Using Hand-Held Devices and PCs Together

Brad A. Myers,

Robert C. Miller, Jeffrey Nichols, Benjamin Bostwick, Franklin Chen, Carl Evankovich, Herb Stiel, Jack Lin, Rishi Bhatnagar, Brian Yeung, Choon Hong Peck, Dave Kong, Karen Cross, Adrienne Warmack, Sunny Ya-Ting Yang, Chun-Kwok Lee, YuShan Chuang, Marsha Tjandra, Monchu Chen, Kin Pou Lie, Bo-chieh Yang and Zack Rosen

Human Computer Interaction Institute

School of Computer Science

Carnegie Mellon University

Pittsburgh, PA 15213

bam@cs.cmu.edu

http://www.cs.cmu.edu/~pebbles

August 9, 2001

INTRODUCTION

The age of *ubiquitous computing* is at hand with computing devices of different shapes and sizes appearing in offices, homes, classrooms, and in people's pockets. Many environments contain embedded computers and data projectors, including offices, meeting rooms, classrooms [1], and even homes. One little-studied aspect of these environments is how personal, hand-held computers will interoperate with the desktop and built-in computers. More and more people are carrying around programmable computers in the form of Personal Digital Assistants (PDAs) such as a PalmOS organizer or Pocket PC (also called Windows CE). Cell-phones and even watches also are increasingly able to participate in the user's computing and information environment. In the Pebbles project, we are researching the question of how the functions and the user interface can be spread across all of the computing and input / output devices that the user has available to form *multi-machine user interfaces*. In this way, the hand-held devices[1] can be used to *augment* the other

---

[1] This paper uses the terms PDAs and hand-held computers interchangeably, since most of the ideas in this paper would equally apply to any of these devices. Our research has so far focused on using PDAs such as Palm Pilots and Windows CE devices. We will use "PC" to refer to a desktop computer, which might actually be a laptop or a wall-mounted display connected to a computer embedded in a room, and therefore not necessarily only for desks.

computers rather than just being a replacement when other computers are not available. Pebbles stands for: **P**DAs for **E**ntry of **B**oth **B**ytes and **L**ocations from **E**xternal **S**ources.

MULTI-MACHINE USER INTERFACES

Most developers and researchers have focused on how a hand-held computer can be used to *replace* a PC when the user is mobile. Our focus is different: how the hand-held and the PC can work effectively *together* when both are available.

We call these kinds of applications "multi-machine user interfaces (MMUIs)." Important attributes of MMUIs are that they use heterogeneous devices for both input and output, that many of the devices have their own embedded processors, that the devices are all connected and share information synchronously, and that the devices are all co-located and are used by individuals or groups.

MMUIs have a number of interesting properties and design requirements that differ from other styles of user interfaces that have been extensively studied. For example, MMUIs can be compared to:

- Using multiple displays with a conventional graphical user interface (GUI), for example, when multiple monitors are connected to a single PC. In this situation, the displays usually have very similar characteristics and are all controlled by a single set of input devices. In contrast, MMUIs will use different input devices and processors for each display.
- Using a single display as a front-end to multiple processors, for example using the X Windows mechanism or pcAnywhere from Symantec to display another processor's windows on the user's display. Here, the user is given the illusion of using only one computer. With MMUIs, each processor has its own display and input devices.
- "Multi-modal" user interfaces, which support other input modalities such as speech and gestures. MMUIs have multiple means for display as well as for input, and span multiple computing devices.
- Most "groupware" work, also called "Computer Supported Cooperative Work (CSCW)," since MMUIs focus on groups that are co-located and where all of the users share some displays and have other devices that are individualized.

We have developed a wide variety of applications and performed a number of user studies to investigate MMUIs. For example, in the Slide Show Commander application, the user's laptop displays a PowerPoint presentation as usual, but the user's PDA controls the presentation, and the user can see on the PDA a thumbnail of the current slide, the notes for the slide, the titles of all the slides, and a timer (see Figure 1).

The PDA communicates to the laptop using a serial cable or wirelessly using radio or infrared. As another example, scrolling desktop windows using the PDA in the user's left hand while the mouse is in the user's right hand (see Figure 2) can be faster than using conventional scroll bars [7].

Figure 1

The Pebbles Slideshow Commander program. (a) A Palm IIIc (color Palm), with the "Scribble" (thumbnail) panel at the front. The other panels for the Palm are "Notes" (b) "Titles" (c) and "Timer" (d). The "Ctrl" tab in (a)-(d) switches to the Shortcutter program (Figure 5) to facilitate demonstrations. Meanwhile, a laptop computer is running PowerPoint and the Palm is in continuous two-way communication with the laptop. (e) The Slide Show Commander application running in Pocket PC on a Compaq iPaq, (f) a detail of the "Titles" view under Windows CE, and (g) the "switcher" pane for PocketPC, which makes it easy to bring a running PC application to the front while PowerPoint is still running in the background.

Figure 2

Typical configuration when using a PDA along with desktop applications. The PDA in its cradle on the left of the keyboard, and the user's hands are on the PDA and the mouse. The PDA is running an application such as a Scroller, which the user is operating with his index finger.

RESEARCH ISSUES IN MULTI-MACHINE USER INTERFACES

Creating multi-machine user interfaces exposes a number of research issues and problems that are different from those with conventional user interfaces. Some of these issues are relevant to group work, others deal with how to facilitate individual work, and finally, there are many issues about how the devices will communicate.

Groups

When multiple people meet to work, slides or a current work product may be displayed on a PC, which serves as a focus of discussion and may be edited by the group. We envision that the meeting participants will have hand-held computers with them, and we are exploring ways to effectively use all the devices together.[2] Important research issues include private displays versus shared displays and interaction techniques for multiple users.

Private Displays versus Shared Displays

The information on the PC display is shared among all the users, whereas the information on the hand-held is private to the individual. The challenge is to show the appropriate information in each place, and to allow fluid transfer of state and information between private and shared displays [3] [11]. In the Slideshow Commander application, for example, the group sees only the slide on the shared display, but the pre-

---

[2] Any remote users not in the same room can use conventional group applications such as Microsoft NetMeeting, which interoperate with the Pebbles applications.

senter's private display shows the notes for each slide, and a timer (see Figure 1). In a study we did of PowerPoint presentations, a number of problems were identified, including that the speaker often desires to walk away from the presentation computer to be closer to the audience (and some people just like to wander around while talking). It can also be awkward to point to and annotate the slides using a mouse. People also often had trouble when trying to go navigate to the previous slide or to a particular slide, or switching from PowerPoint to a demonstration and then back to the PowerPoint slides. The Slideshow Commander program addresses these problems. First, the PDA can be used while walking around. Second, the physical buttons make it easy to move forward and backwards and the titles pane (Figure 1(c) and (f)) makes it easy to jump to particular slides. Third, the user can scribble with the stylus on the slide thumbnails on the PDA (Figure 1(a) and (e)), and these annotations will be shown to the group. Finally, there are two different mechanisms that make it easy to switch to demonstrations and then back to the PowerPoint slide show. The "Ctrl" tab on the PalmOS version switches to the Shortcutter program described below which can invoke and operate programs. Shortcutter can also be used to control room devices such as the video projector or lights (see Figure 5(g-h) below). On PocketPC, the "switcher" pane (Figure 1(g)) displays the programs running on the PC and tapping on a program will switch to it. When the user goes back to Slideshow Commander on the handheld, the PowerPoint presentation is resumed where it left off. In the future, we will be exploring an audience mode for Slideshow Commander, where private scribbles and notes can be saved on each hand-held, and the presenter can call on an individual who can then move the desired information to the public screen.

If each student has a handheld which can communicate wirelessly with the instructor's machine, many interesting applications are possible. Notetaking using Slideshow Commander is one, but also instantaneous test taking. In the last two spring semesters, we have used Jornada handhelds donated by Hewlett Packard in a second-level Chemistry class with about 100 students to enable the instructor to ask multiple choice questions and get a bar graph of all the student's answers [2]. This helps keep the students thinking about the material and also allows the instructor to evaluate the students' level of understanding of the material. The students preferred using handhelds to alternatives such as raising their hands.

Another application that we are exploring is the use of multiple hand-helds along with large displays in a military command post. The big wall displays show maps, visualizations of status or plans, or other information that is shared by all in the room. Each person's hand-held can be used to control the large displays, and to privately get more details about publicly displayed data. In our multi-modal command post, the hand-held devices provide a convenient platform for handwriting and gesturing, since it can be awkward to write directly on the large displays. We are also investigating various ways to use the hand-write directly on the large displays. We are also investigating various ways to use the hand-helds to flexibly enter and carry information, so people can easily and quickly enter their specific knowledge, and easily move data between private and public views.

Interaction Techniques for Multiple Users

Often different people in the meeting will want to take turns controlling the mouse and keyboard to add annotations or input, to try out a system under consideration, or to investigate different options. With standard configurations, they will have to awkwardly swap places with the person at the PC or try to tell the person what to do. The *Remote Commander* application allows each person to use his or her own hand-held device to control the cursor and keyboard of the main PC display. Remote Commander provides three different ways for multiple people to control a single PC. First, each person can take turns controlling the PC's cursor and keyboard input. This allows the hand-helds to control all existing applications, but only one person can be working at a time. The second option is for each person to have their own cursor that appears to float above all applications. This allows everyone to point and scribble on the screen simultaneously, but the cursors cannot control regular PC applications, which can only accept input from a single cursor. Finally, for custom applications that support it, each person can have a separate cursor that controls the application.

Figure 3

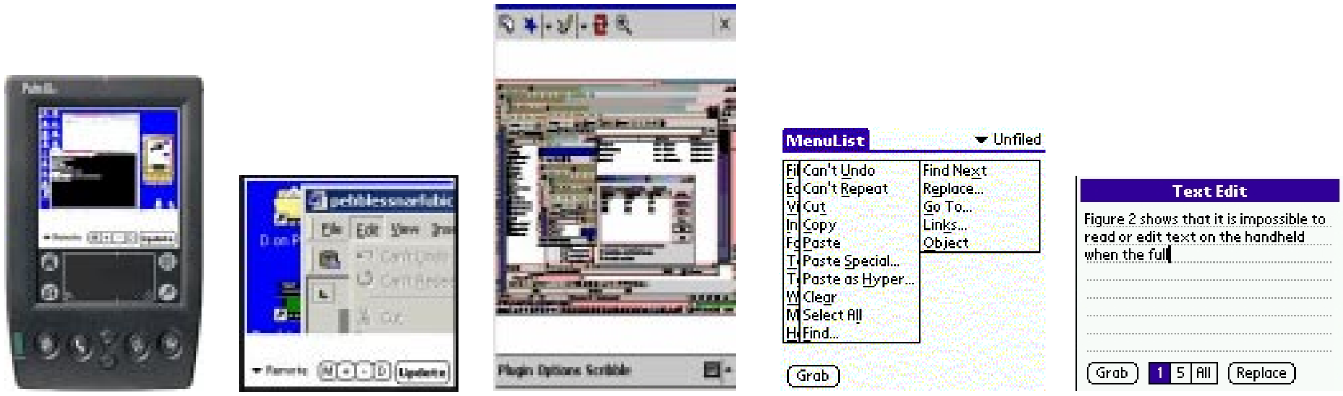
PebblesDraw, a shared drawing program that gives each user a private cursor and selection handles.

We have created a custom drawing program, called PebblesDraw [9], to investigate the interesting issues arising from this shared use (see Figure 3). When all users are sharing the same screen and therefore the same widgets, applications are called "Single Display Groupware" [12]. The user interface issues here are different from those of conventional groupware applications, such as Microsoft NetMeeting, since those assume that each person has a separate display. When all users share a single display, the widgets and interaction techniques must change.

For example, palettes, such as those to show the current drawing mode or the current color, normally show the current mode by highlighting one of the items in the palette. This no longer works if there are multiple users each with a *different* mode and color. Therefore, in PebblesDraw, each user's modes are shown in their "home area" at the bottom of the window, as well as in their cursor. The conventional way to identify different users in groupware is by assigning each a different color, but this has been shown to be confusing in a drawing program since the color identifying the user can be mixed up with the color with which the user will draw. Therefore, PebblesDraw assigns each user a different *shape*. The shape is used for the cursor that follows the mouse, as well as for that user's selection handles and text-input cursor. PebblesDraw must also handle the usual Groupware issues, such as how to deal with Undo. PebblesDraw supports both undo-by-user and global undo. If a user selects undo-by-user, then that user's previous command is undone, if possible[3]. This is implemented using a "selective undo" mechanism [6] provided by the underlying Amulet toolkit with which PebblesDraw was implemented.

A common way to interact with presentations in meetings is to use a laser pointer. Many have provided computer-recognition of the laser dot to control the PC's cursor, but this has not been successful due to the inaccuracies and awkwardness of interaction techniques. Therefore, we created a hybrid technique where the laser pointer is used to indicate the area of interest, and the contents there are "snarfed" (copied) to the handheld. Detailed work is performed on the handheld and then any edits are copied back to the PC. We call this technique "semantic snarfing" [8] because often the *meaning* of the contents of the screen, such as the menu items or text, must be copied to the handheld, rather than just the picture. For example, in Figure 4 (a-c) the picture has been snarfed to the handheld, but the menus and text are not visible, so in (d) the menu's full contents are displayed, and in (e) the text is reformatted to be editable on the Palm.

[3] It may not be possible, for example, if a different user has in the interim deleted the object that this user was working on.



(a) (b) (c) (d) (e)

Figure 4

"Semantic Snarfing" onto a handheld. (a) Full screen on a color Palm. (b) Zoomed in. (c) Full screen on a PocketPC. (d) Semantic snarfing of the menu contents onto a Palm. (e) Semantic snarfing of the text onto a Palm.

Individuals

When a person walks into their office or home carrying a PDA, how will it interact with other computers in the environment? We have identified three important research issues in multi-machine user interfaces for individuals: using multiple computers at the same time to control an application, sharing information among the computers, and how handhelds can be used as "personal universal controllers."

MMUIs for Controlling an Application

In the "old days," computers had a variety of input switches and knobs. Today, computers are standardized with just a keyboard and a mouse for input, and connecting custom input devices is difficult and expensive. Although today's computers have high-resolution screens and window managers, users can still need extra space to display information. For people who have already purchased a PDA, we want to exploit it to provide the benefits of having an extra input and output device. For example, Figure 2 shows the PDA being used at the same time as the mouse as an extra input and output device for the non-dominant hand. Some of the newest PDAs such as the Palm V and the Compaq iPaqs have rechargeable batteries that are recharged when the device is in its cradle. Therefore, the user is *supposed* to leave the device connected to the PC whenever the user is next to the PC, so the device might as well be doing something useful.

The PDA can be used to extend the desktop in various ways. It can serve as a customizable input device, with on-screen buttons, sliders, menus and other controls displayed on the screen. These can be made big enough to operate with a finger, even with the non-dominant hand. The interfaces can then be carried around, and used with different PCs. The PDA can also be used as an output and control device to provide secondary views. This is useful when the entire PC screen is engaged and unavailable. For example, the WinAmp MP3 player for PCs can display on the full PC screen an animation based on the music, and a PDA can be used to control playback without interrupting the display (see Figure 5e). Another use is to display information that should not be covered by other windows, for example to display a list of tasks and windows on the PDA to support easy switching. We are also studying how to integrate hand-helds into the user's entire information space.

We investigated a number of ways to scroll using a PDA. Figure 5b shows some buttons that auto-repeat to scroll up and down a line or a page, or left and right. On the left and bottom of Figure 5-b are sliders, where dragging a finger or stylus drags the text the same amount. The center of Figure 5-b contains a virtual rate-controlled joystick, where pressing in the center and moving out scrolls the text in that direction at a rate proportional to the distance. A user study demonstrated that these could match or beat scrolling using the mouse with conventional scroll bars, as well as other scrolling mechanisms such as the "scroll wheel" built into mice and other two-handed scrolling techniques [7]. Users were fastest with the scroll buttons and slowest with the rate-controlled joystick, although it is clear that all the scrollers can be improved with more iterative refinement. As part of the same study, we measured how long it takes a person to move from the keyboard to acquire input devices, and found that the penalty for moving both hands (the left hand to the PDA and the right hand to the mouse) is only about 15% slower than moving just the right hand to the mouse and leaving the left hand on the keyboard. The average times were about 728 msec to move to just the mouse and 701 msec to move back to the keyboard from the mouse, compared to 838 msec to move both hands to the PDA and the mouse, and about 791 msec to move back to the keyboard. Thus, adding the use of a PDA to the conventional mouse and keyboard does not provide a significant penalty.

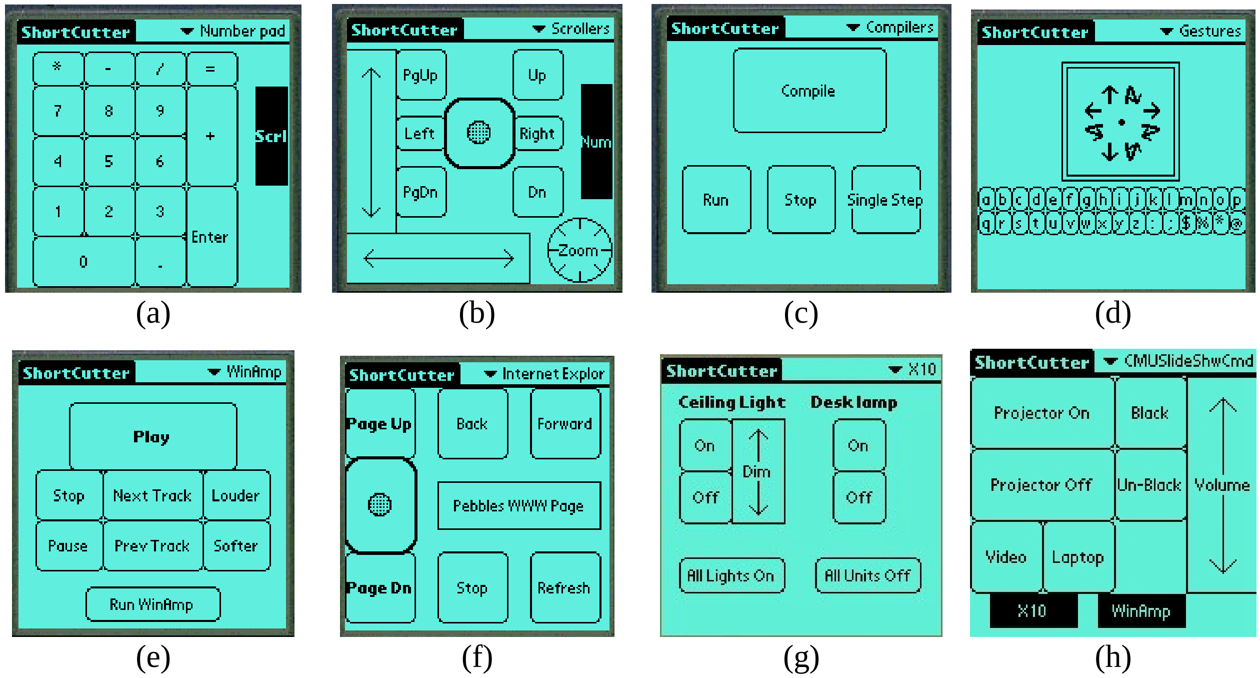


Figure 5

Panels created with Shortcutter: (a) a numeric keypad, (b) a collection of scrollers and a knob, (c) buttons for controlling any of a set of compilers, (d) a gesture pad and two rows of small buttons, (e) a controller for the WinAmp PC program which plays music files, (f) a panel for browsing in Internet Explorer, (g) a panel that interfaces with X-10 to control room lights, and (g) a panel that controls a video projector.

For general application control, the Shortcutter program allows panels of controls to be drawn on the PDA, and used to control any PC application or external device controlled by the PC. Shortcutter can provide customizable interfaces on the PDA even for applications that do not have a customization facility on the PC. Figure 5 shows some panels we have created. Shortcutter's widgets include various kinds of buttons, sliders and knobs and a gesture-recognizer (Figure 5d). Various actions can be associated with each widget, including sending keyboard keys as if the user typed them, scrolling, sending menu and toolbar commands, invoking applications and opening files, switching to different panels, controlling X10 devices and other appliances, and macros containing a sequence of other actions.

Another PDA application, called Switcher, displays a list on the PDA of the current PC tasks (like the Windows Taskbar) and a list of the windows in each task (like the Windows menu in some applications). The user can tap on an item on the PDA to cause that window to come to the front of the PC. This same mechanism also allows the buttons in Shortcutter to be application-dependent. For example, we use many different compilers in our work, and unfortunately they use different shortcut keys as accelerators for various operations such as Compile or Run (and none are customizable to allow these shortcut keys to be changed). With Shortcutter, we can create a single button on the PDA, shown in Figure 5c, which checks which compiler is currently in the front of the PC, and sends the accelerator key appropriate to that compiler.

The Remote Commander program that allows the handheld to act as the PC's keyboard and mouse has proven useful to certain people with muscular disabilities. With Muscular Dystrophy and other disorders, people retain fine motor control and can use a stylus long after they lose arm control and so cannot use a conventional keyboard and mouse. Using Remote Commander and Shortcutter together, some disabled users have gotten significantly improved access to their computers.

Sharing Information

Another important issue is how to fluidly transfer information between devices in a multi-machine user interface. For example, "HotSync" allows all of the information in a PalmOS device to be synchronized with a PC. Similarly, Microsoft provides "ActiveSync" for Windows CE devices. However, these operations copy *all* of the information from the PDA to the PC, which is not appropriate in many situations, especially when trying to share a specific piece of information with another person. To provide a quicker and more convenient way to selectively move information, we adopt the familiar "Cut and Paste" model, but extend it to operate across machines. Whenever the user copies or cuts text on any of their machines, it can be pasted on another. File names or URLs can also be pasted onto the PDA, and a command on the PDA will cause the PC to open the application associated with the file, or open the web page in the default browser. This allows information on the PC to be copied to the PDA either by value or by reference. If the user copies the information itself and pastes it to the PDA, this corresponds to passing the information "by value." If the user copies the filename or URL of the information, then it is passed "by reference." In this way, the PDA can carry pointers to information that resides elsewhere. The Remote Clipboard PDA application provides one place to store the data on the PDA, but the information can be pasted and copied from any PDA application, including the address book, scheduler, MemoPad, etc.

Intelligent Universal Remote Control

We are working on techniques to allow a handheld to serve as a "Personal Universal Controller" (PUC) that can be a remote control for any appliance. The goal is to *automatically* create easy to use control panels from a high-level specification of the appliance's capabilities. Then the panels can be customized to each user, and use consistent interfaces across multiple appliances. For example, the technique for setting the time for an alarm could be the same for all clocks, and also for VCRs and other appliances. Our initial user interface studies that users might be twice as fast and make one-half the errors using a handheld interface compared to the original manufacturer's interface [10].

COMMUNICATION

The general architecture of Pebbles is shown in Figure 6. The main components are client programs running on (one or more) PDAs, server programs running on the PC, and PebblesPC, a PC program that mediates between clients and servers.

We are experimenting with many different ways for the hand-held devices to communicate with each other and the PCs. The simplest and cheapest is to use the cradles or serial cables that are supplied with most of today's PDAs. For PalmOS and Windows CE devices, you can buy serial cables to use instead of the cradles, which are often more convenient, such as when the user is standing and using Slide Show Commander to give a talk.

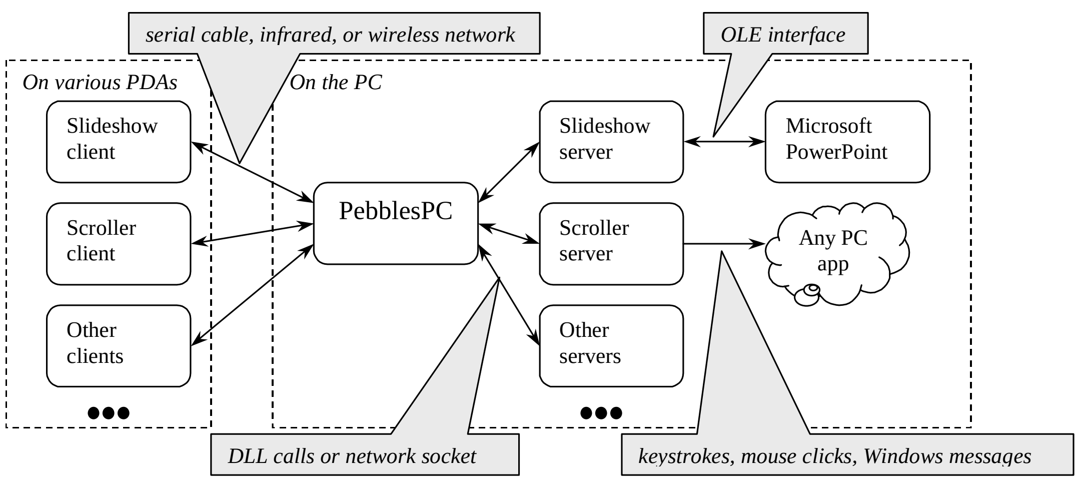


Figure 6

The Pebbles architecture.

An even more attractive option is a wireless connection for the hand-held. Pebbles supports infrared (IR) communication, since IR is built into most PDAs. Unfortunately, IR is not suitable for most of our applications because it is highly directional and quite short range. A better alternative is radio frequency (RF) communication. Carnegie Mellon University has a ubiquitous 802.11 wireless network throughout the campus, in a project called "Wireless Andrew" [5]. Pebbles supports hand-held computers connecting using a 802.11 card such as the Compaq iPaq (Figure 1-e). The BlueTooth standard for small device wireless radio communication [4] may be supported by various small devices and seems like an ideal technology for many of our applications. Pebbles also supports other networking options such as the cell-phone network used by the PalmOS Kyocera SmartPhone.

Pebbles client programs run on various PDAs. Most of our applications run on both PalmOS and Pocket PC (Windows CE) devices. Multiple PDAs can be connected to the same PC to support the groupware applications discussed above. Server programs run on the Windows PC. Our architecture supports two kinds of servers. The first kind uses plugins, which are dynamic link libraries (DLLs) loaded into PebblesPC's address space. The second kind of server is a separate process, running either on the same PC or a remote host, and communicating with PebblesPC through a network socket.

Servers perform their operation in various ways, with various levels of application independence. For example, the Slideshow Commander server interacts directly with PowerPoint through OLE Automation. At the other extreme, the Scroller server simulates scrolling by inserting keystrokes and Windows messages into the standard Windows event stream.

PebblesPC acts as both a naming service and a message router. A server makes itself available to clients by connecting to PebblesPC and registering its name. Clients connect to a server by first connecting to PebblesPC and requesting a server name. If a server by that name is available, then PebblesPC makes a virtual connection between the client and the server, routing messages back and forth. PebblesPC allows clients and servers to connect through heterogeneous I/O interfaces, including serial ports, infrared, network sockets, and Windows message passing. PebblesPC handles the low-level details of each interface.

STATUS AND FUTURE WORK

Most of the applications described here are available for general use from our web site: http://www.pebbles.hcii.cmu.edu. The Slideshow Commander application has been licensed and released commercially by Synergy Solutions (http://www.slideshowcommander.com).

There are many open questions to be investigated in the area of multi-machine user interfaces. We would like to generalize from the specific applications to develop lessons and rules-of-thumb to help guide future user interface designs. With the coming wireless technologies, connecting the PCs and PDAs together will no longer be an occasional event for synchronization. Instead, the devices will frequently be in close, interactive communication. We are pursuing the research needed to help guide the design of interfaces that will run in this environment and span multiple computers.

ACKNOWLEDGMENTS

For help with this paper, we would like to thank the referees, Bernita Myers, and all the users of the Pebbles applications who have provided useful feedback.

This research is supported by grants from DARPA, Microsoft and the Pittsburgh Digital Greenhouse, and equipment grants from Symbol Technologies, Palm, Hewlett-Packard, Lucent, IBM and SMART Technologies, Inc. This research was performed in part in connection with contract number DAAD17-99-C-0061 with the U.S. Army Research Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

REFERENCES

1. Abowd, G.D., *et al.* "Investigating the capture, integration and access problem of ubiquitous computing in an educational setting," in *Proceedings SIGCHI'98: Human Factors in Computing Systems.* 1998. Los Angeles, CA: pp. 440-447.

2. Chen, F., Myers, B., and Yaron, D., *Using Handheld Devices for Tests in Classes.* Carnegie Mellon University, School of Computer Science Technical Report no. CMU-CS-00-152 and Human Computer Interaction Institute Technical Report no. CMU-HCII-00-101, July, 2000. Pittsburgh, PA. http://www.cs.cmu.edu/~pebbles/papers/CMU-CS-00-152.pdf.

3. Greenberg, S., Boyle, M., and Laberg, J., "PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal." *Personal Technologies*, 1999. **3**(1): pp. 54-64. March.

4. Haartsen, J., *et al.*, "Bluetooth: Vision, Goals, and Architecture." *ACM Mobile Computing and Communications Review*, 1998. **2**(4): pp. 38-45. Oct. www.bluetooth.com.

5. Hills, A., "Wireless Andrew." *IEEE Spectrum*, 1999. **36**(6) http://teaser.ieee.org/pubs/spectrum/9906/wire.html. June.

6. Myers, B.A. and Kosbie, D. "Reusable Hierarchical Command Objects," in *Proceedings CHI'96: Human Factors in Computing Systems.* 1996. Vancouver, BC, Canada: pp. 260-267.

7. Myers, B.A., Lie, K.P.L., and Yang, B.-C.J. "Two-Handed Input Using a PDA And a Mouse," in *Proceedings CHI'2000: Human Factors in Computing Systems*. 2000. The Hague, The Netherlands: pp. 41-48.

8. Myers, B.A., *et al.* "Interacting At a Distance Using Semantic Snarfing," in *ACM UbiComp'2001*. 2001. Atlanta, Georgia: pp. To appear.

9. Myers, B.A., Stiel, H., and Gargiulo, R. "Collaboration Using Multiple PDAs Connected to a PC," in *Proceedings CSCW'98: ACM Conference on Computer-Supported Cooperative Work*. 1998. Seattle, WA: pp. 285-294. http://www.cs.cmu.edu/~pebbles.

10. Nichols, J.W. "Using Handhelds as Controls for Everyday Appliances: A Paper Prototype Study," in *ACM CHI'2001 Student Posters*. 2001. Seattle, WA: pp. 443-444. http://www.cs.cmu.edu/~pebbles/papers/NicholsRemCtrlShortPaper.pdf.

11. Rekimoto, J. "A Multiple Device Approach for Supporting Whiteboard-based Interactions," in *Proceedings SIGCHI'98: Human Factors in Computing Systems*. 1998. Los Angeles, CA: pp. 344-351.

12. Stewart, J.E. "Single Display Groupware," in *SIGCHI'97 Adjunct Proceedings: Human Factors in Computer Systems, Extended Abstracts*. 1997. Atlanta, GA: pp. 71-72.