# USING HANDHELDS AND PCS TOGETHER

BRAD A. MYERS

*Pebbles spreads computing functions and their related user interfaces across all computing and input/output devices available to a particular user or group of users, even when they're communicating wirelessly.*

The age of ubiquitous and nomadic computing is at hand, with computing devices reflecting a spectrum of shapes and sizes being used in offices, homes, classrooms, even in our pockets. Many environments contain embedded computers and data projectors, including offices, meeting rooms, classrooms [1], and homes. One relatively unstudied aspect of these environments is how personal handheld computers interoperate with desktop and built-in computers seamlessly in real time. More and more people carry around programmable computers in the form of personal digital assistants (PDAs), including PalmOS organizers and Pocket PCs (also called Windows CE devices). Mobile phones and even watches also increasingly participate in our computing and information environments. In a project called Pebbles, or PDAs for Entry of Both Bytes and Locations from External Sources, begun in 1997 at Carnegie Mellon University (see www.cs.cmu.edu/~pebbles), we've been researching how computing functions and the related user interface can be spread across all computing and input/output devices available to the user, forming what we call multimachine user interfaces, or MMUIs. In this way, the handhelds augment other computers rather than just functioning as replacements when other computers are not available.[1]

Most other developers and researchers have focused on how handhelds are used to replace PCs when users are mobile. Our focus in the Pebbles project is how handhelds and the PC work together when both are available. MMUI applications, which are used by individuals and by groups, involve heterogeneous devices for both input and output with embedded processors connected and sharing information synchronously while being co-located.

MMUIs reflect a number of interesting properties and design requirements that differ from other styles of user interface. For example, MMUIs can be distinguished from the following other user interface styles:

*Multiple displays with conventional graphical user interfaces.* An example involves multiple monitors connected to a single PC. The displays usually share

[1] We use the terms PDA and handheld computer interchangeably, as most of the ideas here would apply equally to any such device. Our research has focused on PDAs, including those running PalmOS and Windows CE. We say PC when referring to desktop computers that might actually be laptops or wall-mounted displays connected to computer embedded in a room, and are therefore not necessarily only for desks.

TERRY MIURA

similar characteristics and are all controllable through a single set of input devices. In contrast, MMUIs use different input devices and processors for each display.

*A single display used as a front-end to multiple processors.* Examples are the X Windows mechanism for Unix and pcAnywhere from Symantec for displaying another processor's windows on the user's display. The user has the illusion of using only one computer. With MMUIs, each processor is connected to its own display and input devices.

*MMUIs supporting other input/output modalities, such as speech and gesture.* In contrast, MMUIs have multiple means for displaying data, as well as for input, and span multiple computing devices.

*Most groupware work, also called computer supported cooperative work, usually focusing on users interacting remotely from from standard desktop PCs.* MMUIs focus on groups that are co-located and in which all users share some displays and input techniques, and others are private and individualized depending on the location of the individual user.

As part of the Pebbles project, we have developed a variety of applications and performed a number of user studies on MMUIs. For example, in the SlideShow Commander application, a user's laptop displays a PowerPoint presentation, but the user's PDA controls the presentation. The PDA shows 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 through a serial cable or wirelessly using radio or infrared (IR) technology. Another MMUI example is scrolling desktop windows using the PDA in the user's left hand while the mouse is in the user's right hand. Such an interface was shown by a study to be quicker than using conventional scroll bars [7] (see Figure 2).

MMUIs expose a number of new human-computer interface research issues and problems; some are relevant to group work, others to how to facilitate individual work, and still others to how devices communicate.

***Groups.*** When people meet, someone might display a slide presentation or a work product on a PC, helping focus the discussion; others might edit or annotate it as needed. Because we increasingly expect meeting participants to bring handhelds with them, we've been motivated to explore ways to use all their devices together.[2] Important research issues include private displays versus shared displays and interaction techniques for multiple users.

*Private versus shared displays.* In a meeting, information on a PC display is shared by all users, whereas the information on each handheld is private to the individual group member. The challenge for the design of applications is to show only the appropriate information in each place and allow the fluid transfer of control and information among private and shared displays [3, 11]. For example, in SlideShow Commander, the entire group sees only the slide on the shared display, while the presenter's private display also shows the notes for each slide, along with a timer, as in Figure 1.

We designed many of the features of SlideShow Commander to solve problems uncovered in our 1999 study of conventional PowerPoint presentations. The speaker often wants to walk away from the presentation computer to get closer to the audience; some speakers just like to wander around while talking. It can be awkward to point to and annotate slides using a mouse. We found that users often had trouble navigating to the previous slide or to a particular slide. Another trouble spot was switching from PowerPoint to a demonstration, then back to PowerPoint.

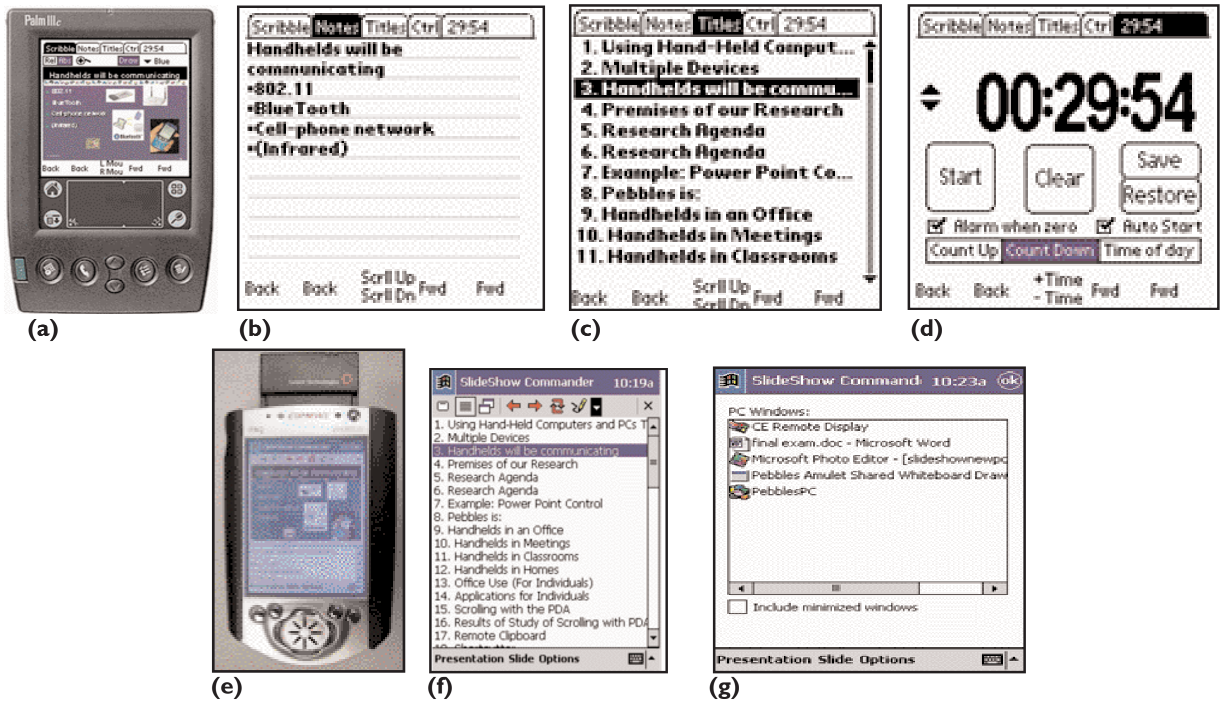
SlideShow Commander addresses these problems. The PDA can be carried in one hand and used while walking around; the PDA's buttons make it easy to move forward and backward among slides, and the Titles pane (Figure 1c and 1f) makes it easy to jump to particular slides. The user can scribble with the stylus on the slide images on the PDA (Figure 1a and 1e); these annotations are shown to the entire group. Two different mechanisms make it easy to switch to a demonstration, then back to a PowerPoint slide show. The Ctrl tab on the PalmOS version switches to the Shortcutter program we've developed to invoke and operate other programs. Shortcutter can also be used to control external devices, including video projectors and room lights. The PocketPC version of SlideShow Commander has a Switcher pane (Figure 1g) that displays the programs running on the PC; tapping on a program name switches to that program. When the user goes back to SlideShow Commander on the handheld, the PowerPoint presentation resumes where it left off.

Future Pebbles research plans to develop an audience mode for SlideShow Commander in which each audience member is able to add private scribbles and notes to personal copies of the presenter's slides. The presenter will also be able to call on others who might then take control of the presentation and display the desired information on the public screen.

In the classroom, many interesting applications are possible if each student has a handheld that communicates wirelessly with the instructor's machine. For example, we have been studying instantaneous test tak-

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

**Figure 1. The Pebbles SlideShow Commander program: (a) Palm IIIc (color Palm) with Scribble panel showing the current slide in a PowerPoint presentation running on a connected laptop; (b) notes; (c) titles; (d) timer; the Ctrl tab in (a–d) switches to the Shortcutter program to facilitate demonstrations; (e) on a Compaq iPaq; (f) titles view on the iPaq; and (g) switcher view, which makes it easy to bring a running PC application to the front while PowerPoint runs in the background.**



(a)
Palm IIIc
(b)
Scribble Notes Titles Ctrl 29:54
Back Back Scrll Up Scrll Dn Fwd Fwd
(c)
Scribble Notes Titles Ctrl 29:54
Back Back Scrll Up Scrll Dn Fwd Fwd
(d)
Scribble Notes Titles Ctrl 29:54
Start Clear Save Restore
Back Back +Time -Time Fwd Fwd
(e)
(f)
SlideShow Commander 10:19a
Presentation Slide Options
(g)
SlideShow Command 10:23a ok
Presentation Slide Options

(b)

Handhelds will be communicating
•802.11
•BlueTooth
•Cell-phone network
•(Infrared)

(c)

1. Using Hand-Held Comput...
2. Multiple Devices
3. Handhelds will be commu...
4. Premises of our Research
5. Research Agenda
6. Research Agenda
7. Example: Power Point Co...
8. Pebbles is:
9. Handhelds in an Office
10. Handhelds in Meetings
11. Handhelds in Classrooms

(d)

00:29:54

☑ Alarm when zero ☑ Auto Start

Count Up | Count Down | Time of day

(f)

1. Using Hand-Held Computers and PCs T
2. Multiple Devices
3. Handhelds will be communicating
4. Premises of our Research
5. Research Agenda
6. Research Agenda
7. Example: Power Point Control
8. Pebbles is:
9. Handhelds in an Office
10. Handhelds in Meetings
11. Handhelds in Classrooms
12. Handhelds in Homes
13. Office Use (For Individuals)
14. Applications for Individuals
15. Scrolling with the PDA
16. Results of Study of Scrolling with PDA
17. Remote Clipboard

(g)

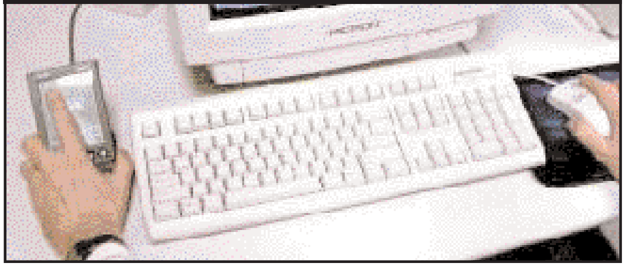
PC Windows:
CE Remote Display
final exam.doc - Microsoft Word
Microsoft Photo Editor - [slideshownewpc
Pebbles Amulet Shared Whiteboard Draw
PebblesPC

□ Include minimized windows

ing. In the last two spring semesters, 2000 and 2001, at Carnegie Mellon University, we used Jornada handhelds donated by Hewlett-Packard Co. in a second-level chemistry class with about 100 undergraduates to enable the instructor to ask multiple-choice questions and generate a bar graph of all the student's answers [2]. This real-time testing and feedback helps keep the students thinking about the material and allows the instructor to evaluate the students' level of understanding during a lecture. A survey of about 50 chemistry students found they preferred using handhelds to other alternatives, including raising their hands.

Another application we are exploring is the use of multiple handhelds along with large wall displays in a military command post. The displays show maps, visualizations of mission status or plans, and other information shared by everyone in the room. Each person's handheld can control the display and privately get more detail about the publicly displayed data. In our prototype multimodal command post, handheld devices provide a convenient platform for handwriting and gesturing, since it can be awkward to write directly on large displays. We are also investigating ways to use handhelds to flexibly enter and view information, so people can quickly and easily enter their specific knowledge and move data among private and public views.

**Figure 2. Typical configuration when using a PDA with desktop applications. The PDA is in its cradle on the left of the keyboard; the user's hands are on the PDA and the mouse. The PDA is running an application that allows the user to scroll a PC window, similar to the wheel on a Microsoft IntelliMouse.**



*Interaction techniques for multiple users.* Meeting participants often want to take turns controlling the mouse and keyboard to make annotations and provide input, as well as try out systems under consideration. With standard PCs, they either awkwardly swap places with the person at the PC or try to tell the person what to do. The Remote Commander application we developed as part of the Pebbles project allows each person to use a handheld to control the cursor and keyboard

Figure 3. Semantic snarfing onto a handheld: (a) full screenshot on a color Palm; (b) zoomed in; (c) full screenshot on a PocketPC; (d) menubar contents semantically snarfed onto the Palm; and (e) text snarfed onto the Palm.

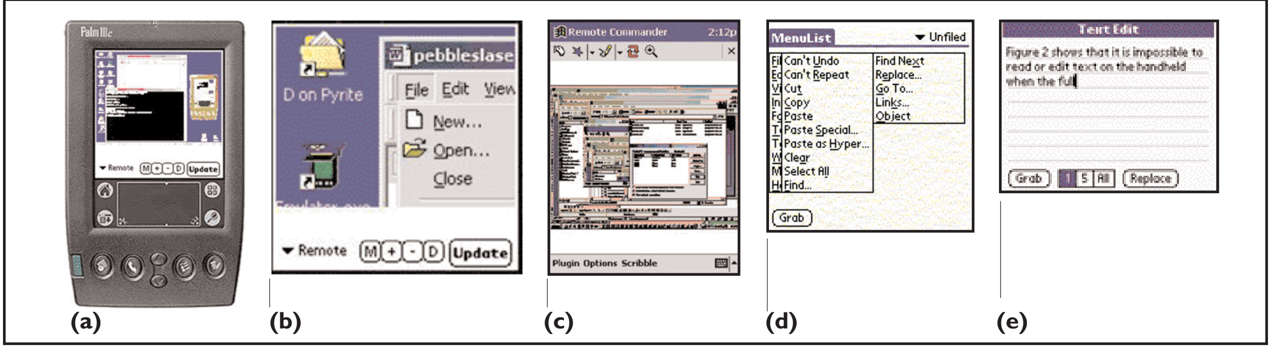


Figure 2 shows that it is impossible to read or edit text on the handheld when the full

of the main PC display.

Remote Commander provides three different ways for handhelds to control a single PC. The first is for each person to take turns controlling the PC's cursor and keyboard input, allowing handhelds to control all existing applications, though only one person can work at a time. The second is for each person to have his or her own cursor; it appears to float above all the applications, allowing everyone to point and scribble on the screen simultaneously, though they cannot control regular PC applications that accept input from only a single cursor. The third way is that each person has a separate cursor but only for custom applications.

We also created a custom drawing program called PebblesDraw [9] to investigate the interesting issues arising from shared use. When all users share the same screen and therefore the same widgets, applications are called single-display groupware [12]. The user interface issues here are different from those in conventional groupware applications, such as Microsoft NetMeeting, which assume each person has a separate display. When all users share a single display, user interface controls and interaction techniques have to change. For example, the conventional way drawing palettes show the current drawing mode or color is by highlighting a button on the palette. This scheme no longer works when multiple users, each with a different mode and color, simultaneously use the same application on the same screen.

In PebblesDraw, each user's modes are shown in his or her "home area" at the bottom of the shared window, as well as inside his or her cursor. The conventional way of identifying different users in groupware is by assigning each one a different color, but this technique has been shown to be confusing in drawing programs, as the color identifying a particular user can get mixed up with the color the user uses for drawing. Therefore, PebblesDraw assigns each user a different shape, which is then used as the user's mouse cursor, as well as for his or her selection handles and text-input caret. Moreover, PebblesDraw supports both undo-by-user and global-undo functions. For example, when selecting undo-by-user, a user's previous command is undone, if possible.[3] Undo-by-user is implemented through a selective undo mechanism [6] provided by the underlying Amulet toolkit we developed (www.cs.cmu.edu/~amulet) and with which PebblesDraw was implemented.

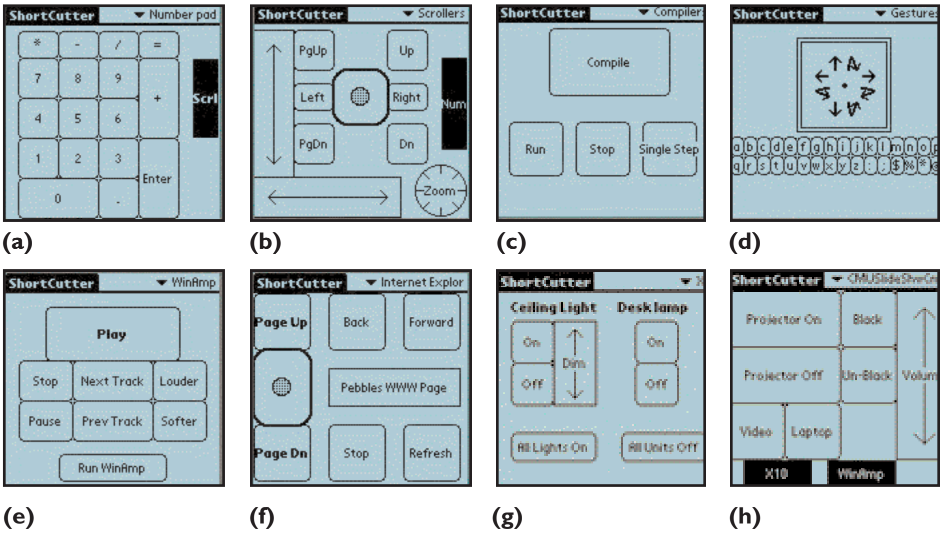
Laser pointers are a popular way to interact with presentations in meetings. Many systems provide computer-recognition of the laser dot to control a PC's cursor, but users have largely rejected this technique due to the inaccuracy and awkwardness of laser-pointer interaction techniques. Therefore, we created a hybrid technique whereby the laser pointer is used only to indicate the area of interest, and the contents there are "snarfed" (copied) to the handheld. Detailed work is performed on the handheld; the final version is then copied back to the PC; we call this technique semantic snarfing [8] because the meaning of a screen's contents, rather than just a screenshot, has to be copied to the handheld. For example, Figure 3a–c shows pictures snarfed to the handheld, though the menus and text are not visible; Figure 3d shows the menu's full contents, and Figure 3e shows the text reformatted to make it editable on the Palm.

***Individuals.*** When a person walks into an office or home carrying a PDA, how might it interact with other computers in the environment? We have identified three important research issues in MMUIs for individual users: using multiple computers simultaneously to control an application; sharing information among the computers; and using handhelds as personal universal controllers.

*MMUIs for controlling applications.* In the early days

[3] It may not be possible if, for example, another user has deleted the object this user had been working on.

**Figure 4. Panels created with Shortcutter: (a) numeric keypad; (b) collection of scrollers and a knob; (c) buttons for controlling items in a set of compilers; (d) gesture pad and two rows of small buttons; (e) controller for the WinAmp PC program for playing music files; (f) panel for browsing via Internet Explorer; (g) panel that interfaces with X-10 to control room lights; and (h) panel that controls a video projector.**



of computing, the machines included 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, some tasks can still benefit from extra space for displaying information. Since so many people have PDAs, applications should be able to exploit them as a customizable and programmable input/output peripheral. For example, Figure 2 shows a PDA being used simultaneously with a mouse as an extra input/output device for the nondominant hand. Since many PDAs, including the Palm V and the Compaq iPaq, have rechargeable batteries (recharged when the device is in its cradle), users are supposed to leave the device connected to the PC whenever it is next to the PC, so it is readily able to communicate.

PDAs can be used to extend the desktop in various ways. The Shortcutter program allows a PDA to serve as a customizable input device, with onscreen buttons, sliders, menus, and other controls displayed on the screen. These controls can be rendered big enough to operate with a finger, even with the nondominant hand. The interfaces might 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—useful especially when the entire PC screen is engaged and unavailable. For example, one mode of the WinAmp MP3 player displays a full-screen animation timed to the music. A PDA can control playback without interrupting the display (see Figure 4e). Another use is displaying information that should not be covered by other windows on the PDA, including lists of tasks and windows, to support easy switching.

*Scrolling (for desktop PDAs).* We've been investigating a number of ways to scroll window contents using a PDA in the user's nondominant hand. Figure 4b shows some buttons that auto-repeat to scroll up and down or left and right. The left and bottom edges of Figure 4b are sliders whereby dragging a finger or stylus causes the text to move a proportional distance in the same direction. The center of Figure 4b is a virtual rate-controlled joystick; pressing in its center and dragging out scrolls the text in that direction at a rate proportional to the distance moved from the center. A user study we conducted [7] demonstrated the PDA scroll buttons could match or beat the scrolling speed of a mouse and conventional scroll bars, as well as other scrolling mechanisms, including the scroll wheel built into mice and other two-handed scrolling techniques. As part of the study, when we measured the time a user takes to move from the keyboard to other input devices, we found that the penalty for moving both hands (the left hand to the PDA, the right hand to the mouse) is only about 15% slower than moving just the right hand to the mouse while leaving the left hand on the keyboard. Thus, adding a PDA to a conventional mouse and keyboard does not represent a significant penalty.

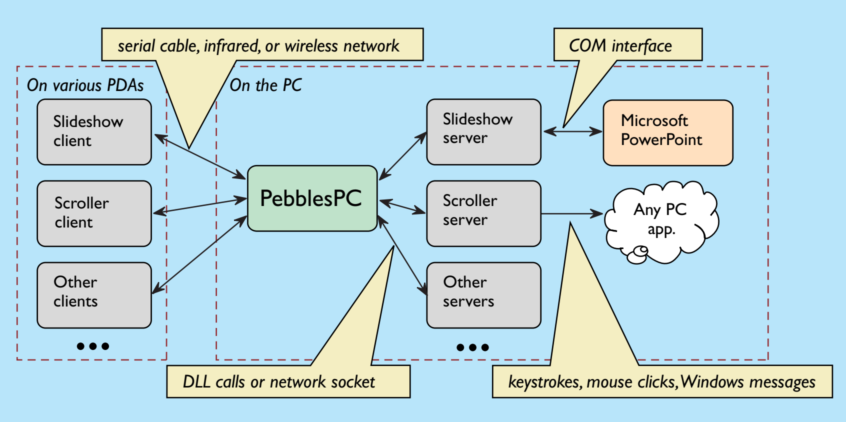
For general application control, Shortcutter allows panels of controls to be drawn on the PDA to control any PC application or external device controlled by the PC. Shortcutter provides customizable interfaces on the PDA, even for applications lacking customization facilities on the PC. Figure 4 shows some panels created in Shortcutter. Shortcutter's widgets include various kinds of buttons, sliders, and knobs, as well as a gesture-recognizer, as in Figure 4d. 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 running macros containing a sequence of other actions.

Another PDA application we've developed, 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 (similar to the Windows menu in some applications). The user taps on an item on the PDA to bring that window to the front of the display on a PC. A similar mechanism allows Shortcutter buttons to be application-dependent. For example, we use many different compilers in our work, but each one uses different shortcut keys as accelerators for various operations, such as compile and run, and none are customizable to allow the keys to be changed. With Shortcutter, we can create a single button on the PDA, shown in Figure 4c, that checks which compiler is in use and sends the appropriate accelerator key.

By enabling handhelds to act as a PC's keyboard and mouse, Remote Commander has proven especially useful to people with certain types of muscular disabilities. For example, people with muscular dystrophy retain fine motor control and can continue to use a stylus long after they lose arm control and are unable to use conventional keyboards and mice. Using Remote Commander and Shortcutter together, some of them have significantly improved their access to computers.

*Sharing information.* Another important issue is how to fluidly transfer information between devices in MMUIs. Today's PDAs provide synchronization features, including HotSync for the Palm and ActiveSync for Windows CE, that synchronize the device with a PC. However, copying all information from a PDA to a PC is slow and sometimes undesirable, especially when the PC belongs to another user. PDAs can also beam a single record to another PDA using built-in IR, but this mechanism does not work between the PDA and the PC. To provide a quick, convenient way to move small pieces of information to and from the PC, we've adopted the familiar cut-and-paste model while extending it to operate across machines. Whenever users cut or copy text on any of their machines, that text can be pasted into other machines. File names and URLs can also be pasted into PDAs; a command on a PDA causes the PC to open either the application associated with the file or the Web page in a browser. The Remote Clipboard PDA application provides one place to store the data on the PDA, though it can be pasted and copied from any PDA application, including address book, scheduler, and MemoPad.

**Figure 5. The pebbles architecture.**



*Intelligent universal remote control.* We are working on techniques to allow a handheld to serve as a personal universal controller, or PUC, functioning as a remote control for any appliance. The goal is for the system to automatically create easy-to-use control panels from a high-level specification of the appliance's capabilities. The panels can then be customized for each user through interfaces across multiple appliances. For example, the technique for setting the time for an alarm could be the same for all clocks, as well as for VCRs and other appliances, by using the same PDA as a remote controller. Our recent initial user interface studies suggest that users might be twice as fast and make half as many errors using a handheld interface compared to the original manufacturer's interface [10].

## Communication

The main components in the Pebbles architecture 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 (see Figure 5).

We are continuing to experiment with the ways handheld devices might communicate with each other and with PCs. The simplest and cheapest uses the cradles and serial cables supplied with most PDAs. For example, for PalmOS and Windows CE devices, instead of the cradles, serial cables can improve convenience, for, say, using SlideShow Commander for a presentation.

A more attractive option is a wireless connection. Pebbles supports IR communication, since IR is built into most PDAs. Unfortunately, this built-in IR technology is not suitable for most Pebbles applications because it is highly directional and is effective only over a very short range. A better alternative is radio frequency communication. The Wireless Andrew project at Carnegie Mellon University recently created an 802.11 wireless network throughout the campus [5]. Pebbles supports handhelds connected through an 802.11 card, including the Compaq iPaq, as in Figure 1e. The BlueTooth standard for small-device wireless radio communication may eventually be supported by various small devices and seems like an ideal technology for many of our applications [4]. Pebbles also supports other networking options, including the mobile phone network used by the PalmOS Kyocera SmartPhone.

Pebbles client programs run on various PDAs. Most run on both PalmOS and Pocket PC (Windows CE) devices, so multiple PDAs can be connected to the same PC to support the groupware applications discussed here. Server programs run on Windows PCs. The Pebbles architecture supports two kinds of servers: a plugin, which is a dynamic link library loaded into the PebblesPC address space, and a separate process running either on the same PC or on a remote host communicating with PebblesPC through a network socket.

Servers perform their operations in various ways with various levels of application independence. For example, the SlideShow Commander server interacts directly with PowerPoint through PowerPoint's COM interface. At the other extreme, the Scroller server in Pebbles simulates scrolling by inserting keystrokes and Windows messages into the standard Windows event stream, so it can scroll any Windows application that understands the standard messages.

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, IR, network sockets, and Windows message passing. PebblesPC handles the low-level details of each interface.

## Future Directions

The SlideShow Commander application was licensed and released commercially by Synergy Solutions (www.slideshowcommander.com). Most of the other applications described here are available for general use from www.pebbles.hcii.cmu.edu.

Many MMUI-related questions remain to be investigated. With the coming wireless technologies, connecting PCs and PDAs will no longer be an occasional event for synchronization of data. Instead, the devices will frequently be in close interactive communication. The Pebbles project team is pursuing the research needed to help guide the design of the interfaces needed for this environment.

## References

1. Abowd, G., Atkeson, C., Brotherton, J., Enqvist, T., and Gully, P. Investigating the capture, integration, and access problem of ubiquitous computing in an educational setting. In *Proceedings of SIGCHI'98: Human Factors in Computing Systems* (Los Angeles, Apr.). ACM Press, New York, 1998, 440–447.
2. Chen, F., Myers, B., and Yaron, D. *Using Handheld Devices for Tests in Classes.* Tech. Rep. CMU-CS-00-152, Carnegie Mellon University School of Computer Science, and Tech. Rep. CMU-HCII-00-101, Human Computer Interaction Institute, July 2000; see 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. *Pers. Techs. 3,* 1 (Mar. 1999), 54–64.
4. Haartsen, J., Naghshineh, M., Inouye, J., Joeressen, O., and Allen, W. Bluetooth: Vision, goals, and architecture. *ACM Mobile Comput. Commun. Rev. 2,* 4 (Oct. 1998), 38–45; see also www.bluetooth.com.
5. Hills, A. Wireless Andrew. *IEEE Spectrum 36,* 6 (June 1999); see teaser.ieee.org/pubs/spectrum/9906/wire.html.
6. Myers, B. and Kosbie, D. Reusable hierarchical command objects. In *Proceedings of CHI'96: Human Factors in Computing Systems* (Vancouver, BC, Canada, Apr.). ACM Press, New York, 1996, 260–267.
7. Myers, B., Lie, K, and Yang, B.-C. Two-handed input using a PDA and a mouse. In *Proceedings of CHI'00: Human Factors in Computing Systems* (The Hague, The Netherlands, Apr.). ACM Press, New York, 41–48.
8. Myers, B., Peck, C., Nichols, J., Kong, D., and Miller, R. Interacting at a distance using semantic snarfing. In *Proceedings of ACM UbiComp'01* (Atlanta, Sept. 30–Oct. 2). ACM Press, New York, 2001.
9. Myers, B., Stiel, H., and Gargiulo, R. Collaboration using multiple PDAs connected to a PC. In *Proceedings of CSCW'98* (Seattle, Nov.). ACM Press, New York, 1998, 285–294; see www.cs.cmu.edu/~pebbles.
10. Nichols, J. Using handhelds as controls for everyday appliances: A paper prototype study. In *Student Posters ACM CHI'01* (Seattle, Apr.). ACM Press, New York, 443–444; see www.cs.cmu.edu/~pebbles/papers/NicholsRemCtrlShortPaper.pdf.
11. Rekimoto, J. A multiple device approach for supporting whiteboard-based interactions. In *Proceedings of SIGCHI'98: Human Factors in Computing Systems* (Los Angeles, Apr.). ACM Press, New York, 344–351.
12. Stewart, J. Single display groupware. In *Adjunct Proceedings of SIGCHI'97: Human Factors in Computer Systems Extended Abstracts* (Atlanta, Mar.). ACM Press, New York, 1997, 71–72.

**Brad A. Myers** (bam@cs.cmu.edu) is a senior research scientist in the Human Computer Interaction Institute in the School of Computer Science of Carnegie Mellon University, Pittsburgh, PA.

Also contributing to this article, as well as to the research and software development it covers, are Robert C. Miller, Jeffrey Nichols, Benjamin Bostwick, Franklin Chen, Carl Evankovich, Herb Stiel, Joonhwan Lee, 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, all of the Human Computer Interaction Institute in the School of Computer Science at Carnegie Mellon University, Pittsburgh, PA.

The research described in this article 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. It was performed in part in connection with contract DAAD17-99-C-0061 with the U.S. Army Research Laboratory. The views and conclusions contained here 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.