

A Multiple Device Approach for Supporting Whiteboard-based Interactions

Jun Rekimoto

Sony Computer Science Laboratory Inc.

3-14-13, Higashi-gotanda, Shinagawa-ku, Tokyo 141 Japan

Phone: +81 3 5448 4380

Fax: +81 3 5448 4273

E-Mail: rekimoto@csl.sony.co.jp

<http://www.csl.sony.co.jp/person/rekimoto.html>

ABSTRACT

In this paper, we propose a multiple-device approach for supporting informal meetings using a digital whiteboard. Traditional digital whiteboard systems often suffer from a limited capability to enter text and the handling of existing data. The large display surface of the whiteboard also makes traditional GUI design ineffective. Our proposed approach provides a hand-held computer for each participant which serves as a tool palette and data entry palette for the whiteboard. Just as an oil painter effectively uses a palette in his/her hand, this hand-held device offers an easy way to create a new text/stroke object, to select existing data from a network, to select pen attributes, and to control the whiteboard application. This paper also reports our experience with the digital whiteboard systems using a proposed multi-device architecture.

KEYWORDS: Multi-computer user interfaces, Pick-and-Drop, digital whiteboard, Ubiquitous Computing, CSCW

INTRODUCTION

During an informal group meeting or at a small lecture, a whiteboard often serves as the central communication medium. Participants gather in front of the whiteboard, and anyone can quickly scribble some text or a diagram to visualize and share his or her thoughts. Similarly, a computerized whiteboard (also called a digital whiteboard), such as the Xerox LiveBoard system [5], is considered to be an important device for computer supported collaboration.

A digital whiteboard is not just a large computer display. In addition to the actual information drawn on it, the physical actions of participants, such as their pointing gestures, are also important.

Recently, many manufactures have been selling affordable LCD projectors and whiteboard-sized tablets. By combining

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.

CHI 98 Los Angeles CA USA

Copyright 1998 0-89791-975-0/98/4...\$5.00

these devices, one might be able to configure an affordable pen-sensitive whiteboard-sized display. However, without a proper understanding of whiteboard activities, such a system would fail become a usable collaborative medium.

Tivoli [13] was one of the first software systems that focused on supporting collaborative activities on a digital whiteboard. During the design and testing of the Tivoli software, several UI research issues relating to digital whiteboards became apparent. We believe that some of these topics can be addressed by enhancing the software design of the system, however others will probably require changing the physical system configuration.

In this paper, we propose a different way to utilize digital whiteboards more effectively by distributing some of the whiteboard functions to a palm-top computer for each participant (Figure 1). We also provide an easy and direct data transfer method between a palm-top and a whiteboard. This palm-top computer works as a tool palette, a remote commander, a text entry box, as well as a temporary data buffer during whiteboard-based collaboration.

DESIGN PROBLEMS OF EXISTING DIGITAL WHITEBOARDS

In our laboratory, we have installed and used a combination of a WACOM Meeting Staff whiteboard (capable of sensing an electronic pen) and a computer projector, as a digital whiteboard. We also have tried several whiteboard emulation systems, as well as off-the-shelf GUI applications on this configuration. During this trial, we immediately noticed a number of limitations that hamper effective collaborative activities when using a digital whiteboard (some of these are also discussed in [5, 13]). These problems can be summarized as:

Text entry is difficult.

During normal whiteboard-based discussions, we frequently write on the board with a marking pen. However, this fundamental activity is not well supported by a digital whiteboard. With the limited precision and tracking rate of current digital whiteboards, writing text using an electronic pointing device is not as successful as on a conventional whiteboard. The resulting pen strokes and characters are often shaky and hard

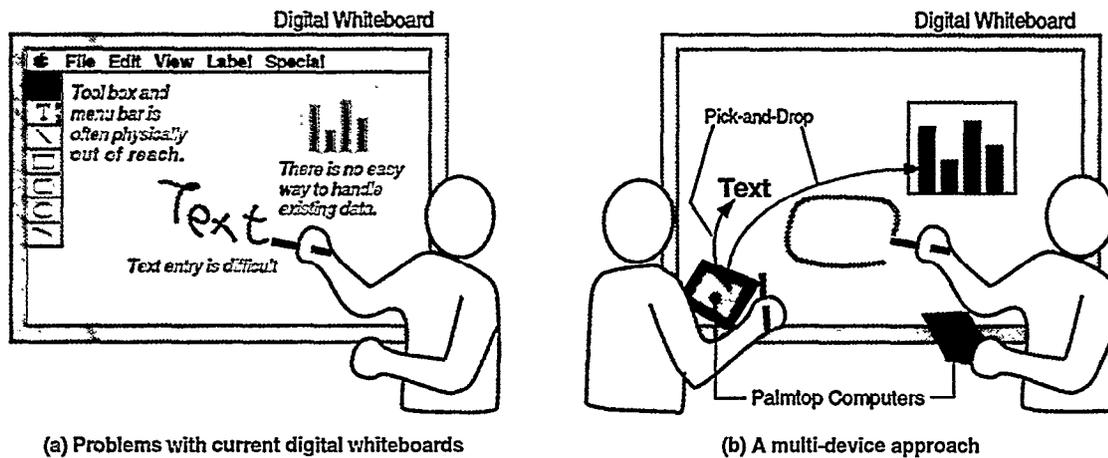


Figure 1: The basic difference between traditional and multi-device approaches to whiteboard user interfaces

to read, not only for a computer (i.e., handwritten recognition software), but also for humans (Figure 2). For this reason, some users hesitate to write characters on a digital whiteboard, preferring to draw very rough diagrams or strokes instead.



Figure 2: Handwritten text vs. printed fonts on a digital whiteboard

Some digital whiteboard systems provide a keyboard located just below the whiteboard (e.g., Xerox LiveBoard), but this layout causes other problems. Only one user can enter text at a time, and doing so requires an unnatural posture. Furthermore, users must also specify the text insertion point on the display using some other method.

One may argue that it is not necessary to provide computer-recognizable characters during an informal meeting; handwritten characters are enough. However, this limitation kills the great potential of computerized whiteboards. If participants can easily enter digits on a spreadsheet, for example, they can take advantage of capabilities offered by computerized whiteboards that are unavailable on physical (non-computer) ones. Recognized characters are far easier to move, copy, erase or search than unrecognized strokes. Recognized texts are also valuable for retrieving information from meeting logs.

Handling of existing data is also problematic.

During whiteboard sessions, participants often need to search for and display existing information from their private online storage, or from public information services such as the

World Wide Web (WWW). However, it is quite irritating and disturbing to the flow of the discussion to wait for one participant to retrieve some desired information while monopolizing the surface of the whiteboard.

Large display size makes current GUI design ineffective.

Using a large screen size can be of itself a problem. Many applications designed for desktop-size displays normally have a menu bar at the top and toolbar(s) at the top or left of the window. As Buxton humorously observed in his talk at CHI '97, this layout causes an out-of-reach problem when used on a wall-sized display. One may have to jump up or walk toward the toolbox or menu bar position, often by displacing other standing participants.

Interactions with the whiteboard become a bottleneck.

Most existing whiteboard applications are designed to be used by one user at a time. As a result, these applications seldom allow simultaneous multiple selections by two or more users. Pop-up and pull-down menus are similarly limited. These single-threaded features prohibit parallel activities among collaborators (e.g., two users entering text segments on different portions of the whiteboard). Given this bottleneck, participants must wait for one user to finish a specific activity. However, this constraint is often violated by mistake. For example, if one participant issues a copy command without waiting for the completion of the other participant's copy-and-paste operation, confusion may result.

A MULTI-DEVICE APPROACH

After considering the present situation, we concluded that a multi-display configuration can potentially solve many of these problems. Existing digital-whiteboard systems have tried to address every problem by enhancing the design of user interfaces only on a whiteboard. However, we feel that if some of the whiteboard functions were given to the participants via hand-held devices that it would offer a simple, yet powerful solution to the problems discussed above. Like an oil painter effectively using a palette in his/her hand to draw a picture on canvas, our proposed approach provides a

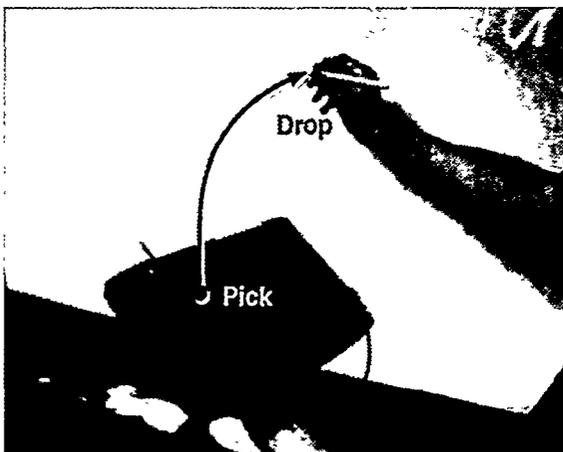


Figure 3: The Pick-and-Drop operation: To transfer data between devices, a user first taps the pen on an object in the first display, then taps again on another display. During this operation, the pen virtually "holds" the data providing an illusion of manipulating digital data as if it were a physical object.

palmtop-sized pen computer (also called a Personal Digital Assistant or PDA) to each participant and allows the user to enter data and manipulate applications on a whiteboard through this PDA. This PDA is used to prepare new text segments, to select existing data, and can be used as a tool palette for controlling the state of the application on the whiteboard.

A participant can use the same pen for manipulating his/her PDA and a whiteboard. For easy data transfer between a PDA and a whiteboard computer, we also developed a direct-manipulation method called *Pick-and-Drop* [14]. Pick-and-Drop is an extended concept of the commonly used drag and drop. With this technique, a user picks up an object on one computer display with a stylus, then drops it on a (possibly different) computer display (Figure 3). For example, a user can select or create text on the user's own PDA and pick-and-drop it at any desired location on the whiteboard. From an implementation point of view, data is transferred through the network, but from the user-interface point of view, this technique allows a user to pick up digital data as if it were a physical object (refer [14] for more detailed description).

This multi-device approach naturally solves many whiteboard-based system problems.

Text Entry

To add a text object to the whiteboard, the user first creates the message on his/her own PDA, then picks it up with the stylus and attaches it at the desired position on the whiteboard. Users can also choose their preferred text input method (e.g., handwritten recognition, soft-keyboard, or simplified stroke input methods such as Unistroke [8] or Graffiti [2]).

This feature might also be more beneficial to Japanese (and perhaps to most of multi-byte language) users, because they normally rely on a dictionary-based input conversion system

for text entry (i.e., enter text as an alphabetical-notation, and interactively convert it into Chinese characters using a conversion system). Dictionaries used for input conversion are often highly personalized; people normally register many words to the dictionary to improve conversion accuracy. Keyboard-commands for controlling the input method are also highly dependent on the user's personal preference. Consequently, no single input method can satisfy all users. However, supporting multiple input methods and multiple conversion dictionaries on a single whiteboard display would make situation far more confusing. The multi-device approach solves such a contradiction. Users can simply use their preferred input method on their own PDAs.

This work style also has affinity with brainstorming methodologies such as the KJ-method. With such methods, participants each write their own ideas on small paper cards, and then spread them on a large table for organizing them. Similarly, using a multiple-device approach, people can dynamically create a text segment during discussion and attach it on the whiteboard.

Handling of Existing Data

The multi-device approach also effectively supports handling of existing data. A user can search for data on his/her own PDA, without disturbing other participants' activities. Once the user has found the data, it can be transferred to the whiteboard using a Pick-and-Drop. Similarly, teachers can prepare lecture notes (a list of texts and graphic elements) on their PDAs, and attach them one-by-one during their lectures.

Separation of Personal and Public Workspaces

It is natural to assume that participants might not want to display every piece of information on the whiteboard. For example, during a lecture or a presentation, a presenter might prefer not to display a slide sorter window on a public screen. The multi-device approach is a natural way of separating such personal workspaces from a public display. A presenter could have a slide list on his/her palmtop, select the next slide on-the-fly, then drop it on the whiteboard screen. Similarly, some interactions that are not directly related to the current discussion, such as browsing a help menu, can be better placed on a personal palmtop device. The user might also be able to have a personal work buffer on a PDA to temporarily store text / graphics elements.

Application Control

Instead of placing tool palettes or menu bars on the whiteboard, Participants would have their own tool palettes on a PDA. For example, a user could select pen color or brush width by clicking on tool icons on the PDA, then draw on the surface of the whiteboard with selected pen attributes. Users would not be bothered by the out-of-reach toolboxes on the whiteboard. Having a personal toolbox would also support multiple pen attribute selection naturally; each participant could select pen attributes independently and the state of all pen attributes would be indicated as highlighted tool icons. If only one set of tool palettes is supplied on a whiteboard, it



Figure 4: A multi-device whiteboard system in action (above: digital drawing in an oil-painter's style, below: collaborative diagram making using pen attributes and text entry palettes)

becomes difficult to display each participant's pen state.

AN EXPERIMENTAL MULTI-DEVICE WHITEBOARD SYSTEM

System Configuration

To realize the proposed multi-device approach described in the previous section, we developed a digital whiteboard system by combining a WACOM Meeting Staff whiteboard, an EPSONELP-5000 LCD projector, and a PC for controlling both. The Meeting Staff whiteboard can sense the existence and the position of an untethered electromagnetic pen. It also distinguishes up to three different pens. Pen stroke information is sent to the PC through a serial cable, and the PC draws stroke images on the surfaces of the whiteboard through the projector.¹ Note that users can also use normal marking pens on the surface of the whiteboard. Thus, the user can mix digital data with physical (non-electronic) writing/drawing on the same whiteboard.

Each user is also provided with a palm-sized computer called

¹To relate the two coordinate systems (the whiteboard's and the projector's), a calibration tool has been developed. At the beginning of the session, a user points to four known points on the surface of the whiteboard with his/her pen. This information is then used to correctly convert position information on the whiteboard into the correct position on the projector.

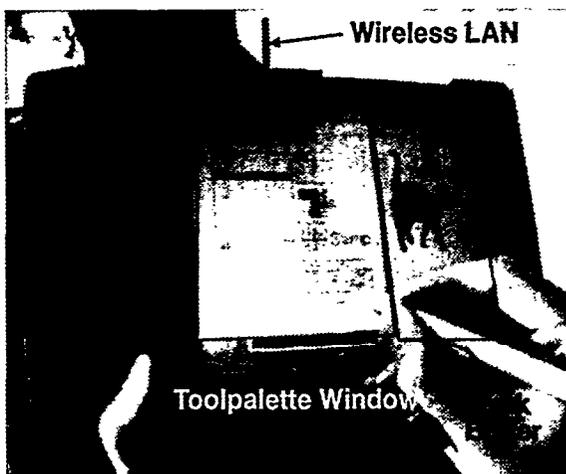


Figure 5: Close-up of M-Pad, a palmtop computer for controlling the digital whiteboard: The window at the left serves as a tool palette, and the window at the right acts as a temporary work buffer. The user can easily transfer information between these windows and the whiteboard by using Pick-and-Drop operations.

the *M-Pad*, which is based on the Mitsubishi AMiTY-SP pen computer, as his/her own tool palette. Since the M-Pad also supports the same pen technology as the whiteboard, the user can manipulate the palm-top computer and the whiteboard with the same pen. The palmtop and whiteboard computers are connected by a spread spectrum wireless network. Users can directly interchange digital objects by using the Pick-and-Drop operation with any participating computer.

M-Draw: A Multi-Device Drawing Tool

Based on the above hardware configuration, we also have developed an experimental multi-device whiteboard tool written in Java called the *M-Draw* (a Multi-device Drawing tool) (Figure 4). The whiteboard part of the M-Draw is simple; there is just a canvas window. With a pen, a user can draw free-form diagrams or handwritten texts on the canvas window, like other ordinary digital/physical whiteboards.

Adding to this basic functionality, users can also use the M-Pad palmtop computer to control the M-Draw application (Figure 5). During operation, the user normally holds this device with his/her non-dominant hand, and use a dominant hand for manipulating a pen.

The window on an M-Pad is split in two. The main window (the left window of Figure 5) is a multi-page tool panel. A user can flip to several tool-palette pages by selecting page tabs on top of the window. Figure 6 shows samples of these tool panels. Using these panels, the user can

- Select pen color, width, and other attributes (dashed-line, etc.) from the pen palette,
- select a prepared diagram element on the picture palette,
- draw a free hand stroke and pick it up on the stroke palette, and
- prepare text segments on the text entry palette.

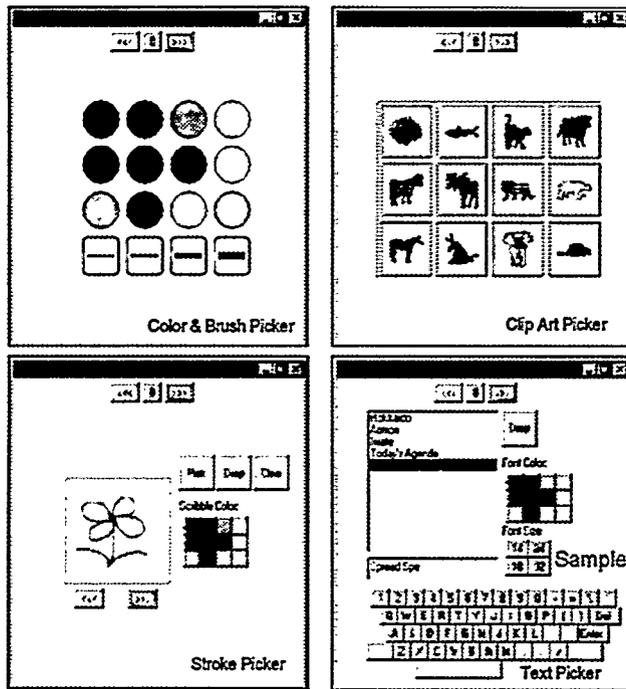


Figure 6: Tool palette examples

Once a user has selected or created a data item on the palmtop computer, he/she can pick it up with the pen and drop it at a desired location on the whiteboard. The user can also change pen attributes by tapping on the pen attribute palette on the M-Pad. This work style is a natural adoption of oil-painting using a (physical) palette.

The other window on the M-Pad (the right window of Figure 5) is a temporary work buffer. A user can store several data elements in this window, and paste it to the whiteboard using a Pick-and-Drop operation.

Early User Study

Using this M-Draw system, the author and other colleagues in the laboratory have experimentally tried several collaborative activities including a group meeting (2 or 3 people attended) and a small-size lecture (one presenter used a whiteboard and others audited). Though the functionality of the system was immature, they appreciated the natural separation of the whiteboard and the palmtop. Figure 7 is a typical user operation sequence during a collaborative session. As shown in this diagram, users effectively utilized the personal (palmtop) and the shared (whiteboard) workspaces.

Most of the users felt the benefit of having printed texts on the whiteboard. These are far easier to read than handwritten characters, and also aesthetically pleasing. M-Pad offers an easy and quick way to make "printed" characters of preferred font size and color. We realized, with surprise, that we have had to use handwritten characters on traditional digital whiteboards, only because there were no other (easier and quicker) ways to write a text.

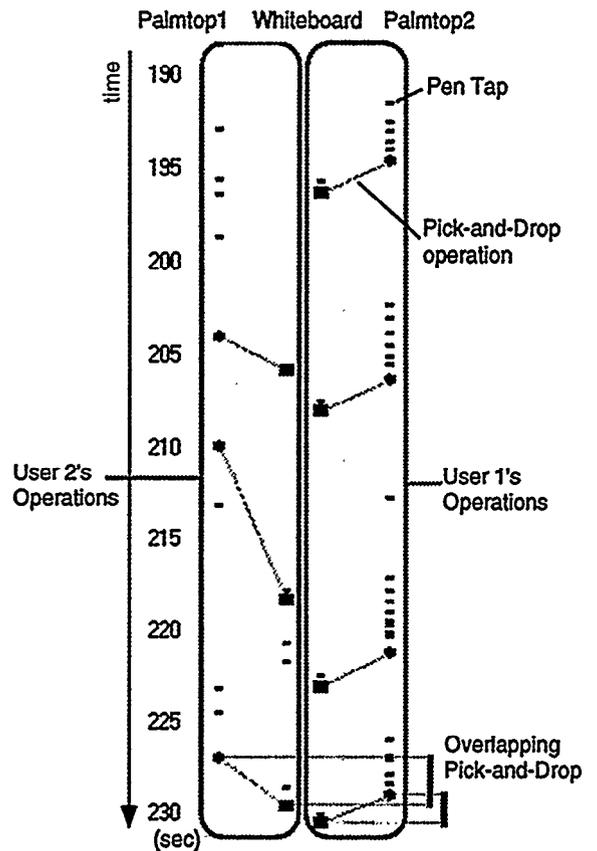


Figure 7: The pattern of pen operations by two participants during a collaborative drawing session

One participant found it effective to use the work buffer on the M-Pad to duplicate data on the whiteboard. He first Picked-and-Dropped it to the work buffer on the M-Pad, then repeatedly used this method to return the information to the whiteboard. To encourage this work style, a copy of the data remains on the M-Pad after the Pick-and-Drop operation. Similarly, after transferring a newly created text from the text entry palette to the whiteboard, the text remains on the item list of the palette; therefore the user can quickly duplicate previously created text items.

It turned out to be quite impressive during the lecture style session to pick up and drop text or diagrams, one-by-one, from a prepared data list on the M-Pad. The lecturer could freely decide which data should be displayed next, according to the flow of his/her talk. The result was a mixture of whiteboard-based lecture style with digital presentation information. Such an effect is not possible using only traditional presentation software.

We did noticed that some of functions should not be placed on a palmtop units. Some design decisions were obvious. For example, if a tool button for flipping a drawing surface is located on a palmtop computer, a participant might surprise other participants by unexpectedly clicking on the flip button. These buttons should be placed on a shared working space.

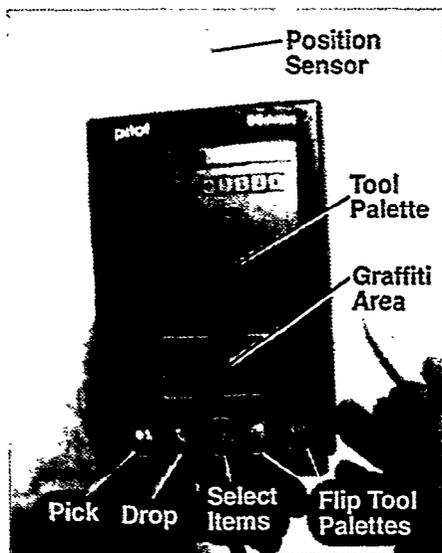


Figure 8: The second generation M-Pad based on PalmPilot

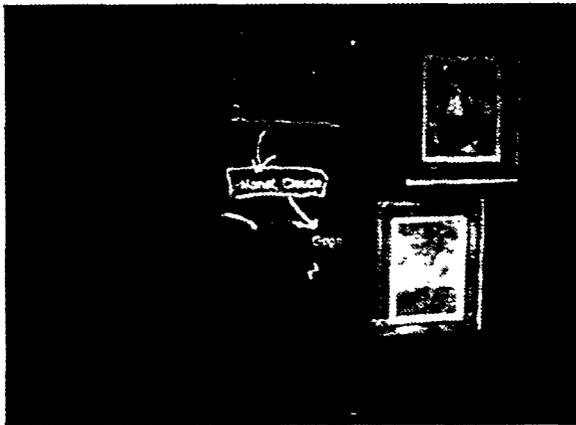


Figure 9: A lecture style session

Others were not clear. If we put too many functions on the palmtops, users tend to keep concentrating on their own palmtop devices, thus degrading mutual awareness among the participants. We feel that there are many UI design issues when distributing functions among multiple devices, however these are still under investigation and require more detailed user studies.

Some of the users complained about the weight of the palmtop device. The device we are using is an off-the-shelf pen computer and it weighs 850g. This weight might not be practical for a long meeting session.

Finally, we also observed that the Pick-and-Drop function within a same whiteboard surface, often worked better than drag-and-drop, particularly when the user had to move data for a long distance (e.g., from corner to corner). Drag-and-drop forces a user to keep the pen-tip in contact with the board during the entire operation, and this feature is not suitable for a large display surface.



Figure 10: Select-and-Paste: Pasting a selected item on the M-Pad with one hand. Note that the other hand is using a stylus.

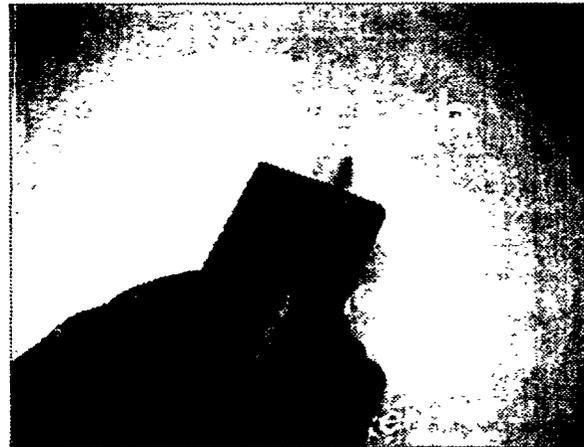


Figure 11: M-Pad is used as a physical Toolglass: A user changes text size and color by clicking-through the M-Pad.

The Second Generation System Design

Based on comments from early users, we are currently developing a second-generation M-Pad device. Since the users' main concern was weight, we decided to use a much lighter hand-held unit. Our choice is the 3Com / U.S. Robotics PalmPilot [4] that weighs only 160g, less than one fifth of the previous device.

Figure 8 shows the second version of our M-Pad device. Unlike the first version, this palmtop device itself has a position-sensing device, so the location of the device can be measured if the device is close enough to the whiteboard surface. It also has several physical buttons to control the operation. These buttons are used for selecting data, flipping tool palette pages, and making pick or drop operations.

The combination of the light weight device and the physical buttons enables a new usage of the M-Pad, one we call the "select-and-paste" technique. Using this new palmtop device, a user can select data and attach it to the whiteboard

by using one (normally a non-dominant) hand, leaving other hand free for normal writing or drawing (Figure 10). The user can also create a new text element by using a soft keyboard, or the simplified stroke recognition system (Graffiti [2]) on the palmtop.

One of the tool palette pages can be used to modify the attributes of data items on a whiteboard. That is, the user places a hand-held device near the data item on the whiteboard, then taps one of the attribute buttons (color, font size, etc.) with the pen (Figure 11). It is a physical variation of the click-through GUI technique (e.g., Toolglass [1]).

Although the new version is more practical than the previous PC-based version, especially in terms of the weight, there are some design-tradeoffs between the two. Since the first version is a full-fledged PC, the user was able to retrieve any online data from the network using FTP, a web browser, or any other PC applications. These data can be picked up and attached to the whiteboard. On the other hand, with the new version it is not impossible, but not very practical to operate such complicated GUI applications on such a small-size display.²

RELATED WORK

Our work has been influenced by the vision of Ubiquitous Computing (UbiComp) [17], where many computers, jointly working together, can help people and their real world activities. While UbiComp mainly focuses on "computation in the background", the topic of this paper concentrates more on designing "foreground" interfaces and interaction techniques using multiple computers and devices.

Xerox LiveBoard [5] and its meeting software Tivoli [13] are pioneering research efforts for designing a digital whiteboard and its user interfaces. [13] mentions several differences between traditional desk-top GUIs and desirable UIs for whiteboard systems, which also partly motivated our work. Nakagawa et al. also discusses the user interfaces for the large screen display, and proposed several GUI widgets for such environments [12].

The Spatial Data Management System (SDMS) [3] is a well known multi-modal system that uses hand pointing and voice commands. SDMS is also has a multi-device configuration. Information is displayed on a wall-sized projection display and the operator uses a small touch-sensitive display mounted on the armrest of a chair. Although the user manipulates two different screens to perform a single task, information exchange interfaces between these devices has not be sufficiently studied.

The PARC TAB is a palm sized computer that was developed at Xerox PARC as part of the Ubiquitous Computing project [16]. It is also used in an multi-display environment. For example, the PARC TAB can be used as an tele-pointer for the LiveBoard. However, direct manipulation technique

between the PARC TAB and the LiveBoard was not seriously considered.

The use of the M-Pad work buffer is related to the application on The DigitalDesk [18], a computer augmented desk consisting of the combination of a desk, a camera, and a projector. The PaperPaint application developed for the DigitalDesk allows select-and-copy operations between paper and a projected image.

The PDA-ITV system [15] proposes the usage of a PDA as a commander for interactive TV. Although it uses two different displays for one task, the roles of PDA and TV are static; PDA always acts as a commander for the TV. Inter-computer manipulation was not considered. For example, it is not possible to grab information from the TV screen and drop it to the PDA.

Our hand-held device using the PalmPilot has some similarity to the spatially aware palmtop computer called the Chameleon [6]. Chameleon senses its location in a physical space with an attached 3D sensor, and gives appropriate information related to that position. Our PalmPilot version also shows property information on a display object when the palmtop device is close enough to that object.

The idea of blurring virtual and physical spaces is becoming popular recently. The graspable UI [7] introduced a brick, a small physical object that can be attached to a virtual object on the screen. Tangible UI [11] also extends this concept, and coins a term "phicon" (PHYSICAL ICON). Our innovation to this area is the introduction of multiple computer user interfaces, and the notion of digital information transfer in a physical world. Using pick-and-drop, the user can pick up digital information as if it were a real object, and transfer it across device boundaries.

Finally, this work is also an extension to our previous research called the Pick-and-Drop [14]. Pick-and-Drop allows information transfer across computer boundaries with a direct manipulation interaction technique. Although our system also uses Pick-and-Drop for data transfer, the focus of this paper is on a user interface for supporting whiteboard-based interactions and not just on general data transfer among computers.

CONCLUSION AND FUTURE DIRECTIONS

In this paper, we have described existing problems with the current digital whiteboard systems and have proposed a solution to them. Our proposed multi-display approach has shown to be a natural separation of personal and public work spaces, and encourages parallel activities during whiteboard-based activities. This approach is also a solution for the design of user interfaces using a large-display surface.

There are several ways to extend our proposed method. Immediate possibilities are to extend our system into a distributed environment such as a shared whiteboard, video conferencing, and connection to the WWW. Since the multi-

²The screen resolution of PalmPilot is 160×160 (pixel).

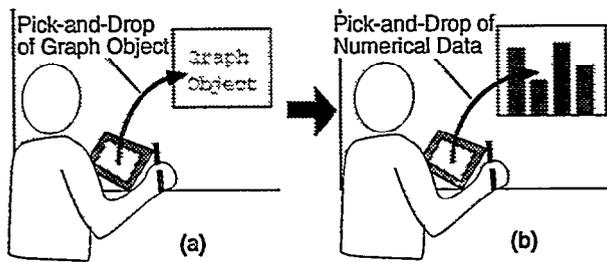


Figure 12: Pick-and-Drop of an active object: The user first attaches a business graph object to the whiteboard (a), then picks-and-drops numerical data on that object (b).

device approach is a fundamental improvement to whiteboard interfaces, it should be possible to combine this technique with other applications such as a meeting support tool or a group decision making tool. In particular, we are interested in combining a multi-device approach with a drawing tool that has interactive beautification capabilities (such as [9]).

Another possible extension is to add a mechanism for handling active objects (such as JavaBeans [10]) to the system. That is, the user can pick up an object from the palmtop device and attach it on the whiteboard. For example, he/she first picks up a business-graph object, drops it on the whiteboard, then he/she can also pick up other numerical data and drop it on the business-graph object (Figure 12). The appearance of the graph object would change according to the dropped data.

ACKNOWLEDGMENTS

We would like to thank Toshiyuki Masui and Jeremy Cooperstock for their helpful discussions and valuable comments on the draft. Nobuyuki Matsushita and Yuji Ayatsuka contributed the photographs in this paper and the video production. We are also in debt to Mario Tokoro and Toshi Doi for their continuing support on our research.

REFERENCES

1. Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony DeRose. Toolglass and Magic Lenses: The see-through interface. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pp. 73-80, August 1993.
2. C. H. Blickenstorfer. Graffiti: Wow! *Pen Computing Magazine*, pp. 30-31, 1995.
3. R. A. Bolt. Put-That-There: voice and gesture at the graphics interface. *ACM SIGGRAPH Comput. Graph.*, Vol. 14, No. 3, pp. 262-270, 1980.
4. 3Com Corporation. PalmPilot home page. <http://www.usr.com/palm/>, 1997.
5. Scott Elrod, Richard Bruce, Rich Gold, David Goldberg, Frank Halasz, William Janssen, David Lee, Kim McCall, Elin Pedersen, Ken Pier, Jhon Tang, and Brent Welch. LiveBoard: A large interactive display supporting group meetings, presentations and remote collaboration. In *CHI'92 Proceedings*, pp. 599-607, 1992.
6. George W. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Communication of the ACM*, Vol. 36, No. 7, pp. 38-49, July 1993.
7. George W. Fitzmaurice, Hiroshi Ishii, and William Buxton. Bricks: laying the foundations for graspable user interfaces. In *CHI'95 Conference*, pp. 442-449, 1995.
8. D. Goldberg and C. Richardson. Touch-typing with a stylus. In *Proc. InterCHI '93*, pp. 80-87. ACM, 1993.
9. Takeo Igarashi, Satoshi Matsuoka, Sachiko Kawachiya, and Hidehiko Tanaka. Interactive beautification: A technique for rapid geometric design. In *Proceedings of UIST'97*, October 1997. to appear.
10. Sun Microsystems Inc. JavaBeans home page. <http://java.sun.com/beans/>, 1997.
11. Hiroshi Ishii and Brygg Ullmer. Tangible Bits: Towards seamless interfaces between people, bits and atoms. In *CHI'96 Proceedings*, pp. 234-241, 1997.
12. M. Nakagawa, T. Oguni, T. Yoshino, K. Horiba, and S. Sawada. Interactive dynamic whiteboard for educational applications. In *International Conference on Virtual Systems and Multimedia (VSMM) '96*, pp. 479-484, September 1996.
13. Elin R. Pedersen, Kim McCall, Thomas P. Moran, and Frank G. Halasz. Tivoli: An electronic whiteboard for informal workgroup meetings. In *Proc. InterCHI '93*, pp. 80-87. ACM, 1993.
14. Jun Rekimoto. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of UIST'97*, October 1997. to appear.
15. Stott Robertson, Cathleen Wharton, Catherine Ashworth, and Marita Franzke. Dual device user interface design: PDAs and interactive television. In *CHI'96 Proceedings*, pp. 79-86, 1996.
16. Roy Want, Bill Schilit, Norman Adams, Rich Gold, Karin Petersen, John Ellis, David Goldberg, and Mark Weiser. The PARCTAB ubiquitous computing experiment. Technical Report CSL-95-1, Xerox Palo Alto Research Center, March 1995.
17. Mark Weiser. The computer for the twenty-first century. *Scientific American*, Vol. 265, No. 3, pp. 94-104, 1991.
18. Pierre Wellner. Interacting with paper on the DigitalDesk. *Communication of the ACM*, Vol. 36, No. 7, pp. 87-96, August 1993.