ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220877032

Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments

Conference Paper · January 1997

DOI: 10.1145/263407.263505 · Source: DBLP

CITATIONS
505

READS
57

1 author:

Jun Rekimoto
The University of Tokyo

143 PUBLICATIONS **5,988** CITATIONS

SEE PROFILE

Petitioner has added a romanette to page (i).
Otherwise, it leaves the original page numbering.

MICROSOFT CORP.
EXHIBIT 1237

All in-text references underlined in blue are linked to publications on ResearchGate, letting you access and read them immediately.

Available from: Jun Rekimoto
Retrieved on: 19 September 2016

Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments

Jun Rekimoto
Sony Computer Science Laboratory Inc.
3-14-13 Higashigotanda, Shinagawa-ku,
Tokyo 141 Japan
+81-3-5447-4380
rekimoto@csl.sony.co.jp
http://www.csl.sony.co.jp/person/rekimoto.html

ABSTRACT
This paper proposes a new field of user interfaces called multi-computer direct manipulation and presents a pen-based direct manipulation technique that can be used for data transfer between different computers as well as within the same computer. The proposed Pick-and-Drop allows a user to pick up an object on a display and drop it on another display as if he/she were manipulating a physical object. Even though the pen itself does not have storage capabilities, a combination of Pen-ID and the pen manager on the network provides the illusion that the pen can physically pick up and move a computer object. Based on this concept, we have built several experimental applications using palm-sized, desk-top, and wall-sized pen computers. We also considered the importance of physical artifacts in designing user interfaces in a future computing environment.

KEYWORDS: direct manipulation, graphical user interfaces, input devices, stylus interfaces, pen interfaces, drag-and-drop, multi-computer user interfaces, ubiquitous computing, computer augmented environments

INTRODUCTION

In a ubiquitous computing (UbiComp) environment [18], we no longer use a single computer to perform tasks. Instead, many of our daily activities including discussion, documentation, and meetings will be supported by the combination of many (and often different kinds of) computers. Combinations of computers will be quite dynamic and heterogeneous; one may use a personal digital assistant (PDA) as a remote commander for a wall-sized computer in an presentation room, others might want to use two computers on the same desktop for development tasks, or two people in a meeting room might want to exchange information on their PDAs. Other than the UbiComp vision, we often use multiple computers for more practical reasons; PCs, UNIXs, and Macs have their own advantages and disadvantages, and users have to switch between these computers to take full advantage of each (e.g., writing a program on a UNIX while editing a diagram on a Mac).

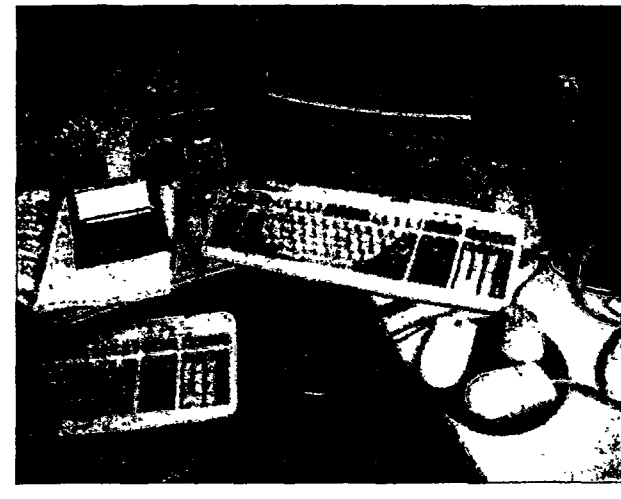

Figure 1: A typical "mouse jungle" in a multi-computer environment

However, using multiple computers without considering the user-interface introduces several problems. The first problem resides in a restriction of today's input devices. Almost all keyboards and pointing devices are tethered to a single computer; we cannot share a mouse between two computers. Therefore, using multiple computers on the same desk top often results in a "mouse (or keyboard) jungle", as shown in Figure 1. It is very confusing to distinguish which input device belongs to which computer.

The other problem is the fact that today's user interface techniques are not designed for multiple-computer environments. Oddly enough, as compared with remote file transmission, it is rather cumbersome to transfer information from one computer to another on the same desk, even though they are connected by a network. A cut-and-paste on a single computer is easy, but the system often forces users to transfer information between

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
UIST 97 Banff, Alberta, Canada
Copyright 1997 ACM 0-89791-881-9/97/10..$3.50

Q1. How many computers do you have on your desktop?

0	1	2	> 3
0 %	7.7 %	38.5 %	53.8 %

Q2. How often do you need to transfer data between computers on the same desktop?

Very often	Often	Sometimes	Occasionally	Never
69.4 %	25.0 %	2.8 %	0.0 %	2.8 %

Q3. (under the Q2 situation) How do you transfer data?

By hand	Through shared files	By ftp	By e-mail	Through floppies	Other
62.9 %	62.9 %	57.1 %	34.3 %	20.0 %	22.9%

Q4. How often do you need to transfer data from your computer to another's computer within a short distance?

Very often	Often	Sometimes	Occasionally	Never
28.2 %	23.1 %	35.9 %	5.1 %	5.1 %

Q5. (under the Q4 situation) How do you transfer data?

By hand	Through shared files	By ftp	By e-mail	Through floppies	Other
54.1 %	56.8 %	37.8 %	73.0 %	10.8 %	18.9 %

Table 1: How people transfer information between computers within a proximity distance: A survey conducted on the members of Sony's software laboratories. About 100 people received this survey by e-mail, and 39 of them answered. Note that the answers for Q3 and Q5 are duplicated, so the totals may exceed 100%.

computers in a very different way. A quick survey reveals that people transfer information from display to display quite irregularly (Table 1). Interestingly, quite a few people even prefer to transfer data *by hand* (e.g., read a text string on one display and type it on another computer), especially for short text segments such as an e-mail address or a universal resource locator (URL) for the World Wide Web . We consider these tendencies to be caused by a lack of easy direct data transfer user interfaces (e.g., copy-and-paste or drag-and-drop) between different but nearby computers.

The first problem is partially solved by using more sophisticated input devices such as a stylus. Today's stylus input devices such as WACOM's, provide untethered operation and thus can be shared among many pen sensitive displays. This situation is more natural than that of a mouse, because in the physical world, we do not have to select a specific pencil for each paper. With the second problem, however, we have much room for improvement from the viewpoint of user interfaces.

Although some systems use multi-display configurations [3, 17, 14], direct manipulation techniques for multi-display environments have not been well explored to date. We believe that the concept of multi-display direct manipulation offers many new design challenges to the field of human-computer interfaces.

In this paper, we propose a new pen based interaction technique called "Pick-and-Drop". This technique lets a user exchange information from one display to another in the manner of manipulating a physical object. This technique is a natural extension to the drag-and-drop technique, which is popular in today's many GUI applications. Figure 2 shows the conceptual difference between the traditional data transfer method and Pick-and-Drop.

DESIGNING PICK-AND-DROP

From Drag-and-Drop to Pick-and-Drop

Pick-and-Drop is a direct manipulation technique that is an extrapolation of drag-and-drop, a commonly used interaction technique for moving computer objects (e.g., an icon) by a mouse or other pointing devices. With the traditional drag-and-drop technique, a user first "grabs" an object by pressing a mouse button on it, then "drags" it towards a desired position on the screen with the mouse button depressed, and "drops" it on that location by releasing the button. This technique is highly suitable for a mouse and widely used in today's graphical applications.

However, simply applying the drag-and-drop to pen user interfaces presents a problem. It is rather difficult to drag an object with a pen while keep the pen tip contacted on the display surface. It is often the case that a user accidentally drops an object during the drag operation, especially when dragging over a large display surface.

Development of our proposed Pick-and-Drop method started as useful alternative to drag-and-drop for overcoming this problem. With Pick-and-Drop, the user first *picks up* an computer object by tapping it with the pen tip and then lifts the pen from the screen. After this operation, the pen virtually *holds* the object. Then, the user moves the pen tip towards the designated position on the screen without contacting display surface. When the pen tip comes close enough to the screen, a shadow of the object appears on the screen (Figure 4) as a visual feedback showing that the pen has the data. Then, the user taps the screen with the pen and the object moves from the pen to the screen at the tapped position. This method looks much more natural than that of drag-and-drop. In our real lives, we regularly pick up an object from one place and drop it on another place, rather than sliding it along the surface of something. We would also like to mention that this Pick-and-Drop metaphor might be more familiar to people who normally use chop sticks at meals.

Inter-Computer Operations

We soon realized that the more interesting part of the Pick-and-Drop operation is in its multi-display capability. That is, with the Pick-and-Drop a user can pick up a computer object from one display and drop it on another (different) display. Pick-and-Drop is a direct manipulation technique that tries to ignore the bound-

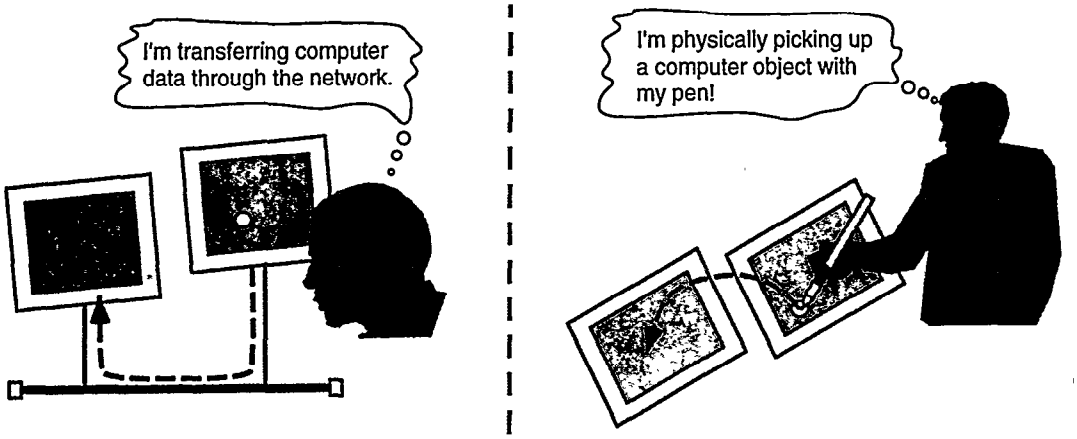

Figure 2: The conceptual difference between remote copy and Pick-and-Drop

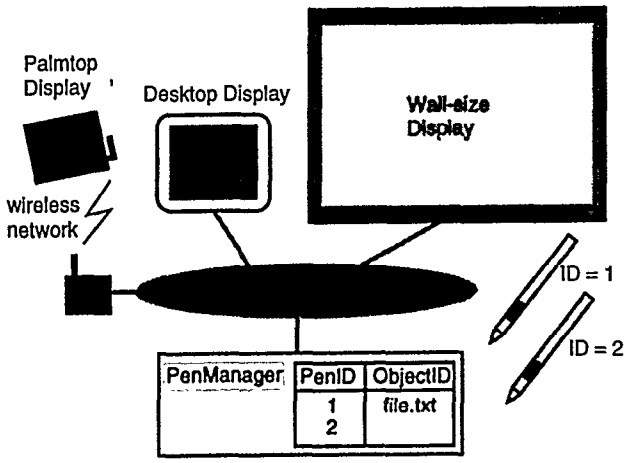

Figure 3: System configuration

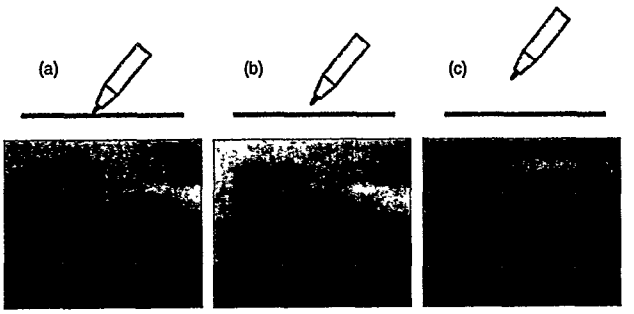

Figure 4: Pen and icons: (a) the pen contacts the display, (b) the pen lifts up, but remains close to the screen, and (c) the pen is away from the screen

ary between computers. We also regard this as one of the first manifestations of a multi-computer direct manipulation technique.

There are a number of opportunities where people need to exchange information from one computer to another. Examples include:

- Copying a file from your PDA to your colleague's PDA.
- When working in front of a wall-sized computer display (such as the LiveBoard [4] or the HoloWall [11]), one may want to pick up a document from one's PDA and attach it to the wall display.
- When using two or more computers simultaneously (e.g., a notebook PC and a desktop PC), one may want to copy a text string from one computer and paste it on the other computer.

Although these operations can also be implemented by using remote copy or shared file systems, we feel that it is more natural to allow a user to manipulate a computer object as if it were a real (physical) object.

Pen-IDs

Storing data on a pen, however, makes the pen device heavy and unwieldy. We developed the multi-computer Pick-and-Drop without making such modifications to the pen by introducing the concept of *Pen IDs*. In our design, each pen is assigned a unique ID. This ID is readable from the computer when a pen is closer enough to its screen. We are currently using a combination of modifier buttons (attached to the pen as a side switch) to represent IDs. We also assume that all computers are connected to the network (either wired or wireless). There is a server called the "pen manager" on the network. (Figure 3).

When a user taps an object (typically an icon) on the screen with the pen, the pen manager binds its object ID to the pen ID. This binding represents a situation in which the pen *virtually* holds the object (even though the pen itself does not contain any storage). When the user moves the same pen towards the other display, the pen manager supplies the type of the bound object to the display. Then the shadow of the data appears on the display below the current pen position. At this moment, the pen does not touch the screen. Finally, when the user touches the display with the pen, the pen manager asks the first computer to transfer the data to second computer.

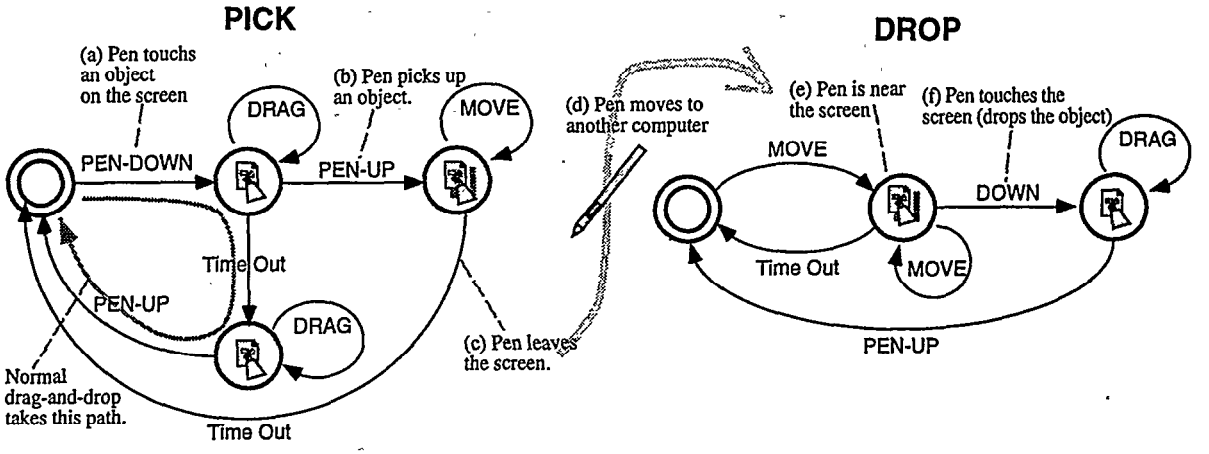

Figure 5: The state transition diagrams of Pick-and-Drop

Figure 6: Information exchange between PDAs

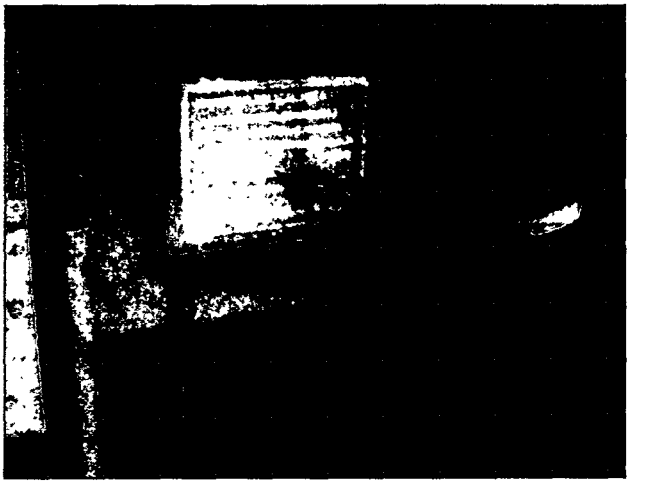

Figure 7: Picking up information from a kiosk terminal

Since each pen has its own ID, simultaneous Pick-and-Drop operations by more than one pen can overlap. This feature would be useful in a collaborative setting.

Note that Pick-and-Drop can also coexist with the normal drag-and-drop by using a time-out. The system distinguishes between these two operation by measuring the period of time between pen-down and pen-up. When a user touches an object with the pen and drags it without lifting the pen tip, it initiates a drag-and-drop instead of a Pick-and-Drop.

The state transition of Pick-and-Drop is shown in Figure 5.

Object Shadows

When a pen holding data approaches a screen, a shadowed object appears on the screen to indicate that the pen has the data (Figure 4). This visual feedback is useful to know what kind of data the pen is holding without having to drop it.

A pen's proximity to the screen can be sensed by combining the motion event and a time-out. When a user moves a pen close to the screen, the screen begins reading motion events from the pen. If motion events occur continuously, the system regards the pen as being near the screen. When a pen leaves the screen, motion events seize and the system can detect it again by setting a time-out. This technique is used for both the Pick and the Drop operations (Figure 5).

EXAMPLE APPLICATIONS

Since Pick-and-Drop is a natural extension to drag-and-drop, which is a commonly used direct manipulation technique, we should be able to apply this technique to various situations in many user interface systems. We have developed several prototype systems to explore the potential of Pick-and-Drop. The following are some experimental applications that we have identified.

Information Exchange between PDAs and Kiosk Terminals

The simplest usage of Pick-and-Drop is to support the exchange of information between two co-workers. When two people need to transfer a file or a short text segment

Figure 8: The canvas and palette metaphor: drawing on a wall screen with a palm-sized palette

between computers, they can simply pick it up from one's PDA display and drop it on the other's display (Figure 6). Note that these two PDAs are communicating via wireless networks.

It is also possible to pick up information from a kiosk terminal in an public space or an office. In our laboratory, we use a kind of "push media" terminal that periodically retrieves selected information from external and internal news sources on the World Wide Web. The terminals are installed at public spaces in the laboratory such as the coffee corner, and continuously display information [12]. We added a Pick-and-Drop capability to this system so that people can pick up URL information from the terminal and drop it to his/her PDA (Figure 7).

Drawing on the Wall Display with the Tablet

Another possibility is to use a hand-held tablet as support for large (whiteboard-sized) display interfaces. We have developed a simple paint editor using a palm-sized computer as a control palette. The user can select a color and brush type for the pen by tapping the control panel on the palm-sized tablet. This metaphor is similar to physical painting using a canvas and palette (Figure 8). This metaphor is advantageous for drawing on a large display, because the user does not have to click on a tool-palette, which might be out of reach.

This example can also be seen as a variation of Pick-and-Drop. The user picks up pen attributes and drops (draws) on the canvas by using the same pen.

Anonymous Displays

The concept of multi-display operations is also helpful for considering interaction between desk-top computers. For example, when a user is editing a document on a desktop computer, he/she can also use several small tablets on the desk that act as "temporal work buffers". The user can freely Pick-and-Drop diagrams or text elements between the desk-top display and the tablets. We refer to this work style as "Anonymous Displays", because users no longer regard such a tablet as a distinct computer. Instead, the user can easily introduce an additional tablet to the desk space according to their work load. Pick-and-Drop supports intuitive data transfer without bothering with each computer's symbolic name.

As compared with the virtual paste buffers used in traditional GUI systems, employing physical tablets provides a more natural and spatial interface for users. The user can freely arrange tablets on a physical desk-top according to his/her work style. Since all information on the tablets are visible, the user can correctly handle more than two work buffers. Even though the size of the main desk-top display is limited and fixed, the user can add as many work spaces as desired without consuming space on the main display. The concept of Anonymous Displays is to introduce a familiar physical artifact into computer work spaces. Note that we do not have to sacrifice computational power when introducing tangible objects into user-interfaces. For example, the user could perform a "global search" on all of the anonymous tablets. Such a capability is unavailable in a *real* physical environment.

Picking up Paper Icons

Another possible way to extend the concept of multi display user interfaces is to support information exchange between computers and non-computer objects. For example, it would be convenient if we could freely pick up printed icons on a paper document and drop it on the computer screen.

Our prototype system called *PaperIcons* allows Pick-and-Drop between a paper object and a computer display (Figure 9). The user can pick up an object from a printed page and drop it on a display. The page is placed on a pen sensitive tablet and a camera is mounted over the tablet. The camera is used to identify the opened page by reading an ID mark printed on it. The user can freely flip through the booklet to find a desirable icon. The system determines which icon is picked based on the page ID and the picked position on the tablet. Currently, the position of the page on the tablet is assumed and fixed, but it can also be tracked by the video camera by locating markers on the printed page.

Although it is also possible to implement icons book as an electronic display and to provide a Pick-and-Drop operation between booklet; and other computers, the PaperIcons style is quite suitable for selecting "clip art" or "color samples" from a physical book. If the user is accustomed to a frequently used book, he/she can flip through pages very quickly by feeling the thickness of the book.

IMPLEMENTATION DETAILS

Currently, we use MITSUBISHI AMITYs as palmtop pen computers, the WACOM PL300 liquid crystal display as a VGA compatible pen-sensitive desktop screen, and the combination of the WACOM MeetingStaff and

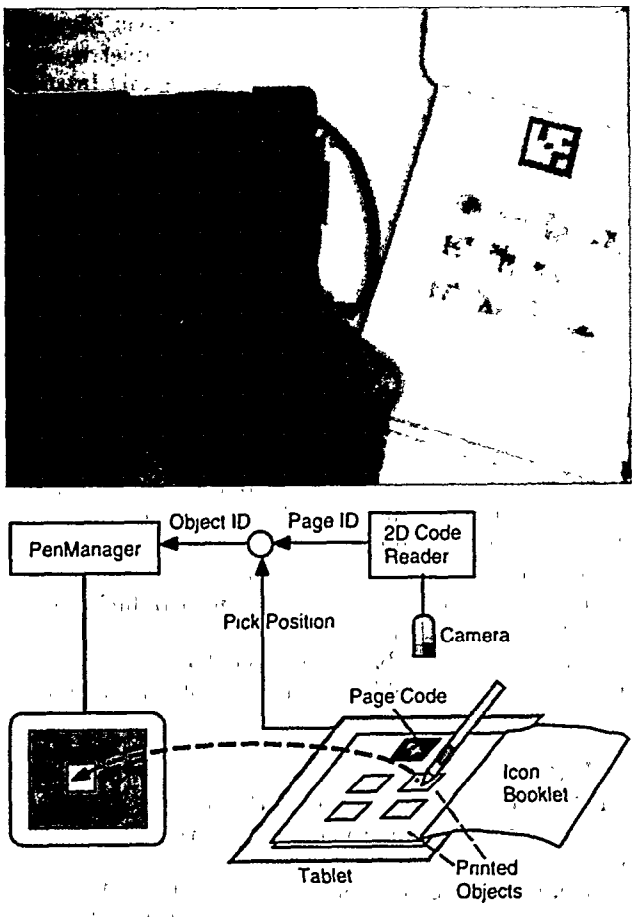

Figure 9: Pick-and-Drop between paper and computer

the projector as a wall-sized display. The same stylus can be used for all these displays because all use the same stylus technology.

Modifier buttons attached to the stylus are used for pen identification. Since WACOM stylus has two modifiers, the system can distinguish up to three pens simultaneously (note that modifier buttons are an alternative). This number is sufficient for testing the Pick-and-Drop concept, but may not be for practical applications. There are several possibilities to extend the number of distinguishable pens. One way is to attach a wireless tag to each pen. Another possibility is to use an infrared beacon.

All the applications described in the APPLICATIONS Section were developed with Java [7]. The pen manager is also a Java application and communicates with applications with TCP/IP connections. When Pick-and-Drop occurs, one (source) application transmits a Java object (e.g., a file icon) to another (destination) application. We use Java's serializable class [8] for implementing object transfers. All instances which are the subclass of class Serializable can be converted to and from a byte sequence. When one computer transfers a Java object, the system first serializes it and sends the resulting byte sequence to the other computer. The receiving computer then de-serializes and recreates the object.

Among the computers described in the APPLICATIONS Section, wall-sized displays (computers) and desk-top displays are directly connected to the Ethernet, while other PDAs use wireless local area networks (LAN). We use Proxim RangeLan2 spread spectrum wireless LAN that employs 2.4 GHz spread spectrum radios and achieves a 1.6M bps data transmission rate.

DISCUSSION

Physical vs. Symbolic

From a functional point of view, a Pick-and-Drop operation is no more than a remote copy command. However, in terms of user interface, we can see several differences between the two.

Pick-and-Drop is physical and visible as opposed to *symbolic*. We observed how people behave when copying information between two different computers and found that they extensively interchange symbolic concepts. In fact, a copy operation could not be completed without verbal support. For example, a typical conversation was: "Mount *Disk C:* of my computer on your computer.", "What is your machine's name?" "*Goethe.*" "Open folder *Document97* on my *Disk C:* and ...". In this example sequence, *"Disk C:"*, *"Goethe"*, and *"Document97"* are symbolic concepts and unnecessary information for simply exchanging files. On the other hand, information exchange using Pick-and-Drop was more direct. They simply moved the icon as if it were a physical object. Although this operation might also be supported verbally, it is more like a conversation for exchanging physical objects (e.g., "Pick up *this* icon", or "Drop it *here*").

The visibility of Pick-and-Drop plays an important role in collaborative settings. Consider, for example, two or more people working together with many computers. When one participant moves data using Pick-and-Drop, this operation is visible and understandable to the others. On the other hand, when a traditional file transfer method is used, the other participants might become confused because its intention could not be effectively communicated.

Shared Files vs. Pick-and-Drop

Many operating systems support "remote file systems". Under such an environment, the user can transfer data from one computer to another by first moving it to a shared file system, and then to the designated computer. As the survey (Table 1) has shown, many people use this technique. If one of the computers can act as a file server, the user can simply mount its file from the other computer and transfer the data by drag-and-drop.

Although Pick-and-Drop and the shared file solution can be used in conjunction (especially when transferring data to remote computers), there are some issues where Pick-and-Drop looks more natural.

First, as described in the previous section, shared files force the user to deal with certain symbolic concepts

such as a machine's name or a file system's name, even though they can actually transfer data by using drag-and-drop. Since the screen sizes of PDAs are normally limited, thus opening another machines file folder often hides local folders, making operations inconvenient. If the user has to deal with more than two computers, keeping track of "which folder belongs to which machine" becomes a significant problem, which is similar to the "mouse jungle" problem described in the INTRODUCTION Section. In our daily lives, we do not need to have a "remote drawer" mounted on the dresser for moving an physical object from one drawer to another. We simply pick it up and move it.

Secondly, a unit of data transfer is not always a file. We often need to copy a short text segment such as a URL from computer to computer. Although it is possible to transfer such a data element through a temporary file, this operation is more complicated compared to the Pick-and-Drop.

In summary, the shared file approach is a good solution for transferring data between geographically separated computers, but not so intuitive between computers within close proximity.

RELATED WORK

Although there have been a number of researches on improving direct manipulation interfaces, only a few of them dealt with multi-computer environments.

The Spatial Data Management System (SDMS) [3] is a well known multi-modal system that uses hand pointing and voice commands. SDMS is also a multi display system. 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, direct inter-computer manipulation is not considered.

The PARC TAB is a palm sized computer that was developed at Xerox PARC as part of the Ubiquitous Computing project [17]. It is also used in an multi-display environment. For example, the PARC TAB can be used as an telepointer for the LiveBoard system [4]. However, direct manipulation techniques between the PARC TAB and the LiveBoard was not seriously considered.

The DigitalDesk [19] is a computer augmented desk consisting of the combination of a desk, a tablet, a camera, and a projector. The PaperPaint application developed for the DigitalDesk allows select-and-copy operations between paper and a projected image. Video Mosaic [10] also introduces a user interface using physical paper into a video editing system.

The PDA-ITV system [14] tries to use 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 is not considered. For example, it is not possible to grab information from the TV screen and drop it to the PDA.

The PaperLink system [1] is a computer augmented pen with a video camera that is capable of recognizing printed text. Although PaperLink can pick up information from paper, it does not support inter-computer operations. For example, it was not designed to manipulate a computer object and paper information with the same PaperLink pen.

Audio-Notebook system [15] augments paper-based note-taking with voice memos. It allows a user to make links between written notes and voice notes. The system uses printed marks on each page for automatic page detection. The Ultra Magic Key system [16] is another example of a paper-based user interface; it allows a user to manipulate the system through paper. The user mounts a piece of paper (specially printed for this system) on a folder and touches the surface of the paper with their finger. The tip of the finger is tracked by a camera mounted above the tablet. The camera is also used to distinguish the paper type. These configurations have some similarity to the PaperIcon system described in the APPLICATIONS Section, but none of them support interactions *between* paper and the computer.

Finally, the Graspable User Interface [6] proposes a new way to interact with computer objects through physical handles called *bricks*. The user can attach a brick to a computer object on the screen such as a pictorial element in an diagram editor. Pick-and-Drop and Graspable UIs share many concepts in the sense that both try to add *physicalness* to virtual worlds. Unlike Pick-and-Drop, the Graspable UI mainly deals with a single display environment.

CONCLUSION AND FUTURE WORK

We have presented a new interaction technique that allows a user to exchange information in multi-computer environments. By recognizing pen IDs, the system makes it possible to pick up an object from one computer screen and drop it on another screen.

At the moment, the prototype system is immature and there is a lot of room we have for improvement. We would like to expand the number of identifiable pens by introducing radio-frequency (RF) tags. Currently, the system can only exchange Java serializable objects, but it should also be possible to implement Pick-and-Drop with more general file transfer protocols and the cut-and-paste protocols such as the X-Window inter-client communication convention (ICCCM).

In addition to enhancing implementations, there are many ways to extend the idea of multi-display operations. We are planning to build and evaluate an application that supports informal discussions between two or more participants in the same place. Using this system, each participant has their own PDA, while a wall-sized display serves as a common workplace for all participants. Using the Pick-up-Drop, the participants can easily exchange information between their PDA and the

Figure 10: Pick-and-Drop in a ClearBoard setting (ENVISIONMENT)

wall, or between individual PDA's.

Another possible improvement would be to incorporate the Pick-and-Drop with currently available pen interface techniques. For example, several pen gestures like "grouping" can be integrated into Pick-and-Drop ; the user first selects a group of objects by making a group gesture, then picks them up and drops them on another display.

It would also be interesting to incorporate Pick-and-Drop with video-conferencing systems such as the ClearBoard [9]. With such a setting, two users could meet over the network through a shared video window. Each user stores his/her own information on the PDA, and they can exchange information between PDAs through the window (Figure 10). Using the Pick-and-Drop metaphor, users can seamlessly integrate their personal work spaces with their shared work spaces.

The concept of Pick-and-Drop is not limited to pen user interfaces. It should be possible to implement interface similar to Pick-and-Drop by using normal displays and a wireless mouse. Since a wireless mouse is not tethered to a single computer, we can suppose that each user owns a wireless mouse as a personalized pointing device. Such a device can be used to transfer information from one computer to another, while providing personal identification. This concept also suggests a future role of PDAs – that of manipulating multiple devices in an networked environment. For example, one can pick up TV program information on a web page (with a PDA), then drop this information on a VCR to request a recording of it.

As a final remark, the common design philosophy behind all these systems is the understanding that we are living in a fusion of physical (real) and virtual (computer) worlds. Each has its own advantages and disadvantages. Pick-and-Drop, for example, adds physicalness to user interfaces, because we feel that traditional data transfer methods are too virtual and hard to learn due to their lack of physical aspects. To the contrary, many augmented reality systems add virtual properties to the physical world [2, 5, 13]. However, these two approaches do not contradict on another. We believe that one of the most important roles of user interface design is to *balance* the virtuality and physicalness of the target area.

ACKNOWLEDGMENTS

We would like to thank Mario Tokoro for supporting this research. We would also like to thank Yuji Ayatsuka, Jeremy Cooperstock and members of the Sony CSL for helpful discussions. Thanks also to the members of the Sony Architecture Laboratories who collaborated on the survey.

REFERENCES

1. Toshifumi Arai, Dietmar Aust, and Scott Hudson. PaperLink: A technique for hyperlinking from real paper to electronic content. In *CHI'97 Proceedings*, 1997. to appear.
2. Michael Bajura, Henry Fuchs, and Ryutarou Ohbuchi. Merging virtual objects with the real world: Seeing ultrasound imagery within the patient. *Computer Graphics*, Vol. 26, No. 2, pp. 203–210, 1992.
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. 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.
5. Steven Feiner, Blair MacIntyre, and Doree Seligmann. Knowledge-based augmented reality. *Communication of the ACM*, Vol. 36, No. 7, pp. 52–62, August 1993.
6. George W. Fitzmaurice, Hiroshi Ishii, and William Buxton. Bricks: Laying the foundations for graspable user interfaces. In *CHI'95 Conference*, pp. 442–449, 1995.
7. Sun Microsystems Inc. The Java language: An overview. http://java.sun.com/doc/Overviews/java/, 1995.
8. Sun Microsystems Inc. Object serialization. http://www.javasoft.com/products/jdk/1.1/docs/, 1997.
9. Hiroshi Ishii and Minoru Kobayashi. ClearBoard: A seamless medium for shared drawing and conversation with eye contact. In *Proceedings of CHI'92*, 1992.
10. W. Mackay and D. Pagani. Video Mosaic: Laying out time in a physical space. In *Proceedings of Multimedia'94*, 1994.

11. Nobuyuki Matsushita and Jun Rekimoto. HoloWall: Designing a Finger, Hand, Body, and Object Sensitive Wall. In *Proceedings of UIST'95, ACM Symposium on User Interface Software and Technology*, October 1997. to appear.

12. Jun Rekimoto. Intellicast home page. http://www.csl.sony.co.jp/person/rekimoto/icast/, 1997.

13. Jun Rekimoto and Katashi Nagao. The world through the computer: Computer augmented interaction with real world environments. In *Proceedings of UIST'95, ACM Symposium on User Interface Software and Technology*, pp. 29–36, November 1995.

14. 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.

15. Lisa J. Stifelman. Augmenting real-world objects: A paper-based audio notebook. In *CHI'96 companion*, pp. 199–120, 1996.

16. Hiroshi Usuda, Motohiro Kobayashi, and Mitsuhiro Miyazaki. The new generation user interface: Ultra Magic Key (UMK). In *SIGGRAPH'97 Visual Proceedings*, 1997. to be published.

17. 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.

18. Mark Weiser. The computer for the twenty-first century. *Scientific American*, pp. 94–104, 1991.

19. Pierre Wellner. Interacting with paper on the DigitalDesk. *Communication of the ACM*, Vol. 36, No. 7, pp. 87–96, August 1993.