Synchronizing Clipboards of Multiple Computers

Robert C. Miller and Brad A. Myers
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213 USA
+1-412-268-7571
{rcm,bam}@cs.cmu.edu

ABSTRACT

This paper describes a new technique for transferring data between computers, the *synchronized clipboard*. Multiple computers can share a synchronized clipboard for all clipboard operations, so that data copied to the clipboard from one computer, using the standard Copy command, can be pasted directly on another computer using the standard Paste command. Synchronized clipboards are well-suited for a single user moving data among several computers in close proximity. We describe an implementation of synchronized clipboards that works across a wide range of existing systems, including 3Com PalmPilots, Microsoft Windows PCs, Unix workstations, and other Java-capable platforms. Our implementation adds no noticeable overhead to local copy and paste operations.

KEYWORDS: Synchronized clipboard, network clipboard, data transfer, file transfer, drag-and-drop, pick-and-drop, distributed systems, Java, ubiquitous computing, Pebbles

INTRODUCTION

As computing devices proliferate, and more users own and use multiple computers – such as desktop computers, laptops, and personal digital assistants (PDAs) – the problem of moving data among one's own computers becomes increasingly important. This paper describes a new technique for data transfer, *the synchronized clipboard*. A synchronized clipboard ensures that the clipboards of two or more connected computers are always identical. With this technique, data can be moved between different computers using the familiar copy-and-paste model: select an item on one computer, copy it to the clipboard (using the standard Copy command available in most applications), then turn to the other computer and paste the item (using the standard Paste command).

In our model, each user has a personal "clipboard group" consisting of the computers on their desks (and in their pockets), which all share a synchronized clipboard. The clipboard group is currently configured by hand, but personal wireless networks such as BlueTooth [1] may eventually allow automatic group configuration based on physical proximity. Other computers may temporarily join the clipboard group, for instance if visitors want to carry away data on their portable computers. Two users should not share a synchronized clipboard for long, however, because their clipboard operations would interfere. Furthermore, clipboard contents are hidden state, so copied data must be pasted before the user forgets about it. A synchronized clipboard makes sense for a single user transferring data between computers on the same desk, but not between a computer at work and another at home.

Existing "network clipboard" programs (such as the ClipBook Viewer included in Microsoft Windows) allow a computer's clipboard to be shared across the network, but do not provide automatic synchronization. As a result, at some point during a copy-and-paste operation, these programs must ask the user for the name of the other computer. The synchronized clipboard omits this step. Once configured, the synchronized clipboard is *invisible* to the user.

The rest of this paper describes the synchronized clipboard implementation we have built, which has two components. One component synchronizes the clipboards of one or more PalmPilots with a Windows PC. The other component synchronizes the clipboards of Java-capable computers (such as Windows PCs and Unix workstations) across a network.

SYNCHRONIZING PDA AND PC CLIPBOARDS

The 3Com PalmPilot already supports data synchronization with a PC through HotSync, but HotSync is inconvenient for moving a small piece of text, such as an email address or a URL. Forming a HotSync partnership also transfers private information to the PC, which is undesirable for temporary interactions with other users' computers.

We implemented synchronized clipboard support for PalmPilots as a Hackmaster "hack" [2] that intercepts clipboard-related system traps on the PalmPilot and redirects them to the PC clipboard (currently by a serial cable, eventually by wireless infrared). Since the PC clipboard *is* the synchronized clipboard, copy-and-paste operations on the PC do not need to be intercepted or modified in any way. The PalmPilot clipboard only supports copying and pasting text, so the

UIST '99. Asheville, NC
© 1999 ACM 1-58113-075-9/99/11... $5.00

synchronized version of it is similarly limited.

The PalmPilot's clipboard group is configured automatically, simply by connecting or disconnecting it from the PC. Every copy or paste operation on the Pilot triggers an attempt to contact the PC. If the PC is connected and responding, then the PC clipboard is used for the operation. If the PC does not respond within a brief (unnoticeable) timeout, then the Pilot assumes it is disconnected and uses its own clipboard instead. When a copy or paste successfully contacts the PC, a beep informs the user that a remote transfer has occurred.

Our PalmPilot synchronized clipboard, released to the public under the name *Remote Clip Hack*, is one of a family of applications developed by the Pebbles project [3]. Along with the other Pebbles applications, Remote Clip Hack has been downloaded over 12,000 times in recent months and has generated strong positive feedback from users.

SYNCHRONIZING CLIPBOARDS ACROSS A NETWORK

Based on our favorable experience with synchronized clipboards on the PalmPilot, we built *Remote Clip*, a prototype system for synchronizing clipboards across the network. Written in Java 1.1, Remote Clip uses Java's Remote Method Invocation (RMI) to communicate with its peers across the network. In our prototype, a clipboard group is configured manually by running Remote Clip on every machine in the group and specifying the hostnames of the other group members in a setup dialog.

Unlike the PalmPilot system, which uses a client-server architecture centered on the PC's clipboard, the Remote Clip system is peer-to-peer. The synchronized clipboard is "owned" by the machine where the most recent copy operation occurred. When a copy operation occurs on computer A, for example, A notifies the other members of its clipboard group that it is taking ownership of the synchronized clipboard. If a paste operation subsequently occurs on computer B, then B satisfies the paste by retrieving the clipboard contents from A. Note that no clipboard data is transferred until a remote paste actually occurs, so the clipboard owner can change the clipboard contents repeatedly without notifying other group members. Thus, a sequence of local copies and pastes (the common case) proceeds at full speed without any network traffic.

In addition to text, Remote Clip can transfer files and directories, packing them into a ZIP archive that is unpacked automatically by the receiver. On Microsoft Windows, Remote Clip uses native code to copy and paste files directly in the Windows file browser. On other platforms, users can copy and paste files using either a GUI (file-selection dialog boxes) or command-line programs (*rcopy* and *rpaste*).

A number of desirable features are not yet supported by the prototype, including authentication, encryption, and richer data formats (like pictures).

RELATED WORK

Pick-and-drop [4] enables data transfer between pen-based computers by "picking up" an item from one computer with a pen and "dropping" it on another. Hyperdragging [5] extends the reach of drag-and-drop (using a computer's built-in pointing device) across machine boundaries to other computers on an augmented tabletop. Pick-and-drop and hyperdragging require specialized hardware or augmented environments, whereas synchronized clipboards can be implemented on a wide range of current and future hardware and software systems. The only system requirements are a clipboard and a network connection.

Several programs can share clipboard data across the network, such as the ClipBook Viewer included in Windows, but none synchronize the clipboards directly.

CONCLUSION

Synchronized clipboards extend the copy-and-paste model, already familiar to users for data transfer between applications, to allow seamless data transfer between computers.

ACKNOWLEDGMENTS

We are grateful to Edward Keyes (for Hackmaster), Ben Bostwick and Carl Evankovich (for Pebbles), and Eric Tilton, Andrew Faulring, Laura Cassenti, and the anonymous reviewers (for helpful comments).

This research was funded in part by Microsoft and in part in connection with Contract number DAAD17-99-C-0061 with the U.S. Army Research Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

1. BlueTooth Consortium. *BlueTooth.* www.bluetooth.com
2. Daggerware Inc. *Hackmaster.* www.daggerware.com
3. Brad A. Myers, Herb Stiel, and Robert Gargiulo. "Collaboration Using Multiple PDAs Connected to a PC." *CSCW'98*, pp. 285-294. www.cs.cmu.edu/˜pebbles/
4. Rekimoto, J. "Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments." *UIST'97*, pp. 31-39.
5. Rekimoto, J. "Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments." *CHI'99*, pp. 378-385.