Serendipitous sharing through personal information environments

Jeff Pierce

Dr Jeff Pierce is an Assistant Professor in the College of Computing at the Georgia Institute of Technology, where he conducts research on next-generation user interfaces. Pierce leads the Personal Information Environments research group and co-directs the Adaptive Personalized Information Environments lab with Dr Charles Isbell. "We are connected 24/7, but don't yet have reasonable access to data with mobile phones", says Pierce, who is working to change that fact. In *receiver*, he talks about how mobile phones can become an extension of the PC.

http://www.cc.gatech.edu/~jpierce/index.htm
Jeff Pierce's site

Today's information workers have gone beyond the traditional model of a user working in an office with a personal computer. Instead of being tied to a fixed location, workers empowered by mobile devices and pervasive wireless and cellular networks can increasingly access and share digital information when and where they see fit. Workers are also spreading their computing activities beyond a single personal computer: they now interact across a variety of computing devices, including desktop computers, laptops, tablets, personal digital assistants (PDAs), and cellular phones.

Those two trends present an opportunity, but also several challenges. The opportunity is to allow workers to take advantage of their heterogeneous collection of computing devices to access, interact with, and share their information at any time using any of their devices. The challenges are that currently the contents of those devices may differ drastically because workers perform different tasks with different devices at different times; all devices are not physically or network accessible at all times; and current computing devices, because each largely assumes that it is a user's only device, provide little to no support for coordinating activities across them.

The following scenario illustrates these challenges:
Alice is visiting a satellite corporate office and arranges to meet Bob, a colleague, for lunch. They discuss how their respective jobs are going and exchange details of recent projects. During their conversation, Bob expresses an interest in reading a market research report that Alice just recently finished writing. Unfortunately, it is on her desktop computer, which is back in her home office. Rather than immediately sharing the report with Bob, Alice must remember to email it to him when she returns from her trip.

The solution we propose to overcome the challenges and realize the opportunity is to take a worker's collection of heterogeneous personal computing devices and organize them into a personal information environment. Devices in such an environment should be aware of each other, able to contact each other regardless of intervening firewalls and changing network addresses, and able to actively help users manage information and activities across them.

We tackled the problem of helping users serendipitously share files with others as an initial motivating example and step toward our proposed solution. Mobile information workers frequently need to share files with or receive them from others. While workers currently have access to a variety of file-sharing mechanisms (eg email, posting them on a web page, putting them on a USB flash drive, point-to-point "beaming"), existing mechanisms assume either that users have direct access (either physical or network) to the device containing the files or that the user has anticipated the need to share them.

To address those limitations we created SEREFE, an application for SEREndipitous File Exchange between users and devices. SEREFE works on any of a user's devices, but we concentrated our interface design efforts on cellular phones. Cellular phones are small enough and light enough that users are willing to carry them almost anywhere, and they have the advantage of pervasive network access. Cellular phones are thus most likely to be the devices that users are carrying when they want to share or receive a file. Despite these advantages, cellular phones currently provide only limited access to personal information; they are handicapped by their small storage capacities and limited support for accessing data on other devices. SEREFE overcomes those limitations to allow cellular phones to serve as access points to users' personal information no matter where it is stored.

http://www.cc.gatech.edu/projects/PIE/projects/serefe.shtml
SEREFE

The following scenario illustrates SEREFE's user experience:
Alice and Bob meet for lunch during her visit to a satellite corporate office. They discuss how their respective jobs are going and exchange details of recent projects. During their conversation, Bob expresses an interest in reading a market research report that Alice just recently finished writing. Alice pulls out her cellular phone, enters a brief description of the report, selects the correct file from the list of candidate search results, and sends it to Bob. Bob receives an SMS message on his cell phone notifying him that he has an incoming file; he uses his phone to verify that the file is from Alice and requests delivery to his office PC so that he can read it when he returns to work. Alice and Bob then resume their lunch.

To share a file using SEREFE, users must identify the recipient and the file they want to share. To specify the desired recipient, users may either select from a "buddy list" of other SEREFE users (similar to those provided by most instant messenger programs) or enter an email address. The latter option allows users to share files easily with non-SEREFE users. While SEREFE also provides a simple mechanism for sharing files with groups of users (by choosing multiple buddies or specifying multiple email addresses), it arguably still needs a mechanism that allows users to share files easily with a group of physically proximate users (eg to share materials with people attending a presentation).

SEREFE provides two methods of identifying the file to share. First, it keeps a list of recently sent files on the theory that users are likely to share files that they have already shared. Ideally, users simply choose the desired file from that list, allowing them to minimize use of cellular phones' rather cumbersome input mechanisms. When the desired file is not on the list, users enter a small number of terms (potentially including part of its name, its type, or the names of devices that might contain it) and initiate a search for it. SEREFE sends a message to the specified devices (or all of them, if users did not specify particular devices) containing the remaining terms and assembles the results into a list of candidates. Users then either choose the desired file or initiate a new search.

After identifying the recipient and locating the correct file, users instruct SEREFE to send the file. SEREFE sends the recipient notification of an incoming file either directly (if they are currently using SEREFE) or via an SMS message to their cellular phone (if they are not). The recipient can view details about the file (such as the sender's identity and the file name, type, and size) and choose whether or not to accept it. If he accepts it, he can specify both the delivery mechanism (SEREFE currently supports email, IM, and direct FTP between devices) and the email address or device to receive it. We allow recipients to make those choices because they know best how they plan to use the file and the limits on their devices (eg storage capacity) and delivery mechanisms (eg whether a file is too large for their email inbox).

While we assume that users will primarily employ SEREFE when co-located, we did want to support remote sharing as well. For example, if Bob had only realized after lunch that he would like to read Alice's report, SEREFE should allow him to request it from her. We provide this functionality by allowing users to send file requests to other users. When a user receives a file request, SEREFE explicitly asks him to grant or deny it; if the user grants the request, SEREFE initiates the process of sending a file to the requester and displays the file search interface.

We implemented SEREFE by layering on top of an instant messaging (IM) network to create a simple virtual personal network. In addition to assigning users their own IM IDs, we also issue an ID to each of their devices. We then tell each device the IDs of the user's other devices and provide a shared encryption key to allow them to communicate securely with each other. Layering on top of an IM network allows us to avoid firewalls that block incoming connections because devices open connections to the IM server, rather than vice versa. The fixed IM IDs also avoid problems with device IP addresses that change over time because, for example, the devices acquired them at run-time via DHCP. Both benefits help keep devices accessible via the network.

Devices that users turn off will still be inaccessible, even to SEREFE. In those cases we concentrate on removing the burden from users of remembering to share by allowing them to queue messages for later. Users may choose to queue a search to handle it later when a particular device is available or when they have more time. Users may also queue file requests. They can view and complete queued searches and file requests whenever they log into SEREFE.

We conducted an initial usability evaluation of SEREFE that suggests that users find it both usable and useful: they were able to discover quickly how to send and receive files, and they expressed the belief that they definitely would make use of it to share files. We were, however, surprised to discover that users also expressed significant interest in using SEREFE to transfer files between their own devices, a capability that we added essentially as an afterthought. Currently, users can transfer a file between their own devices by choosing themselves as the recipient and specifying how and where they want the file delivered. That mechanism allows, for example, a consultant at a job site who needs a file from his office desktop computer to use SEREFE to FTP it directly to his laptop. As a next step toward developing personal information environments we are exploring how to further simplify transferring and synchronizing files between personal devices.

We are also exploring how to allow users to better utilize the combined input and output resources of their devices, as well as devices they encounter. Users should, for example, be able to borrow the keyboard and monitor of a nearby desktop computer in order to read email stored on their cellular phones, rather than rely just on the phone's resources. We are currently developing DIAMOND, a framework for Dividing Interfaces Across Multiple Opportunistically aNnexed Devices, to explore how multi-device interfaces can improve interaction with users' personal information while addressing their privacy and security concerns.

http://www.cc.gatech.edu/projects/PIE/projects/diamond.shtml
DIAMOND

We have made significant progress in allowing users to access and share information residing on any of their devices using any of their devices. As we move toward increasingly nomadic work structures, this capability will provide information workers with full access to their information no matter where they are or what device they use. This work is, however, just the first step in the path from personal computers to personal information environments. More work remains to be done to realize our eventual goal of allowing users to employ any device (or combination of devices), whether owned or temporarily annexed, to access, interact with, and share any information in their personal information environment.

This article was written exclusively for *receiver*
contact: jpierce@cc.gatech.edu