Courier: A Collaborative Phone-Based File Exchange System

Amy Karlson, Greg Smith, Brian Meyers, George Robertson, Mary Czerwinski

Contact: karlson@microsoft.com

Technical Report
MSR-TR-2008-05
January 2008

Microsoft Research
One Microsoft Way, Redmond, WA 98052, USA
http://research.microsoft.com

Courier: A Collaborative Phone-Based File Exchange System

Amy Karlson*, Greg Smith, Brian Meyers, George Robertson, Mary Czerwinski

Microsoft Research, One Microsoft Way, Redmond, WA, USA 98052
{karlson, gregsmi, brianme, ggr, marycz}@microsoft.com

Abstract. While people often carry mobile phones for communication purposes, they are generally underutilized as productivity tools, especially in the workplace. In this paper we present Courier, a system that leverages the storage capacity and communication capabilities of the mobile phone to facilitate the viewing and exchange of PC-based documents when users are away from their desks. We describe a formative evaluation of Courier's use in a collaborative meeting setting and report on the results. While participants were enthusiastic about Courier overall, our study also highlighted usability issues that will require further refinement, such as users' concerns with privacy, reliability and speed.

Keywords: Mobile computing, file exchange, collaboration.

1 Introduction

Mobile phones are unquestionably the fastest-spreading and most widely-adopted personal computing technology in history. Not only have cellular subscription rates jumped from 1 out of 5 people to 1 out of every 2 people worldwide over the last five years [17], but more phones were sold in 2007 alone than the total number of PCs and laptops in use today [21]. Moreover, with their small size and the convenience they offer for continuous personal and professional connectivity, cell phones are widely considered an essential accompaniment to life beyond the home [5]. Yet despite the mobile phone's rapid adoption for voice and text communication, they have remained relatively underutilized for data-oriented productivity tasks [9]. Thus in our research, we are exploring how mobile phones' ever-increasing accessibility, storage and computing capabilities can be leveraged to promote efficiency in the workplace, particularly when workers are away from their desks.

We began our investigation by focusing on workplace "meetings" because they are a fixture of modern business practice that still pose significant challenges for generating, disseminating and archiving ideas and information among attendees who are typically not collocated with their primary computing platform. Consider, for example, the following scenario:

* This work was conducted while the first author was visiting Microsoft as a research intern.

> *Once a month, the ten members of a research team gather to discuss emerging trends in their field. In turn, participants discuss the most noteworthy papers, talks, websites, and demonstrations they have encountered during the past month. Most attendees bring a laptop to present their own findings as well as to take notes about the resources others have discovered. The presentation format varies by individual, ranging from formal slide decks to on-the-fly web searches. The disruption caused by speakers swapping both seats and laptops to gain access to the projector, the non-uniformity of the presentation styles, and the difficulty the audience often has reading the relatively small text of documents and web pages that are projected on the large meeting room display, invariably lends a quality of tedium to the event. Despite this, the team's members have resisted standardizing the meeting format because of the actual and perceived overhead it would add to their monthly preparation, which group members admit to giving a low priority by comparison to their other obligations.*

While the narrative of the scenario is certainly unique, the inherent inefficiencies and frustrations are not. One reason is that it is difficult to share data directly between computers today except by prior arrangement, which often is prohibitively constraining for meeting settings. For example, the researchers in the above scenario might have each posted their findings on a shared wiki before the meeting, but the consistent forethought and cooperation this approach requires of the attendees makes it unlikely to succeed in the long term. Furthermore, the solution fails for a variety of other meeting styles, such as spontaneous gatherings, hallway encounters, or those involving visitors from outside the trusted corporate network.

The rigidity of today's data sharing methods has spawned a variety of workarounds, such as directly connecting laptops to meeting displays, emailing files to oneself and others, posting data on web pages, and note-taking to archive data that is viewed but otherwise inaccessible. While these practices are the norm, they involve unnecessary expenditures of time and effort before, during and after the meeting. Instead, consider a variant of the previous scenario:

> *In the weeks leading up to the monthly group meeting, each researcher records any relevant resources she encounters using whatever personalized system she has developed for organizing and archiving ideas, references and prior work. Otherwise, she takes no further action to prepare for the meeting, but ensures she takes her phone with her as she leaves for the meeting room. To present her findings, the researcher uses her phone to browse the documents and URLs that she has accessed from her desktop PC over the last month, and selects those she wishes to share with her colleagues on the meeting display. If another meeting attendee is interested in an item displayed, she can use her own phone to request a copy of the shared item. Finally, when each attendee returns to her office, any meeting artifacts she requested are automatically filed to her desktop PC for future reference.*

Note that this alternative approach has the potential to improve upon on current meeting practices in several ways. First, by enabling remote access to an attendee's desktop documents, she does not need to move or email her documents prior to the meeting for them to be accessible, which can reduce preparation time and allow for spontaneity and flexibility during the presentation. Second, since data transfer between a user's PC and the shared meeting display is mediated by the phone through wireless communication, attendees can remain seated for the duration of the meeting, which avoids the disruption of attendees swapping seats to gain access to the shared projector and associated input devices. Finally, by supporting real-time, personal archival of the original documents and URLs shared during the meeting, attendees can focus on participating in the meeting itself, rather than on straining to read and transcribe the small text of documents and URLs, since any items requested will be available for detailed review at their desktop work spaces after the meeting.

Inspired by this vision, we have developed Courier, a lightweight infrastructure that enables a new way to access, share, and distribute files and URLs across computing systems. While Courier is an end-to-end system, from a user's perspective, Courier serves two main purposes. First, Courier allows people to use their mobile phones to access files and URLs originating from their personal computers and to opportunistically send and display those items to other desktop and laptop computing systems. This functionality lets users take opportunistic advantage of large displays for viewing documents for themselves or sharing documents with others while away from their desks. Second, Courier allows users to request a copy of any item others have shared, again via their mobile phone; this copy can then be viewed while in transit like any other personal document, and also will be automatically uploaded and saved to their personal computer upon returning to their home or office. Because our initial exploration has focused on how these capabilities can be used to improve interaction dynamics and productivity in the meeting environment, Courier also provides secondary functions to support meeting activities, such as by enabling meeting attendees to move and scroll documents that have been shared to a common display.

In the remainder of this paper, we present the Courier architecture and user interface, its formative evaluation, and lessons learned from the evaluation. Overall, user opinion concluded that Courier is a desirable system but that improvements to the system speed, reliability, shared document control, and the user interface are required to transform it into a deployable system.

2 The Courier System

Conceptually, Courier provides the bridge that allows people to access the data that resides on their trusted, personal, often stationary computing resources while they are on the go, as well as allows them bring artifacts they encounter along the way back into their personal computing domain for future viewing and use. Since people often keep their mobile phones handy while in transit, Courier exploits the phone as a means to select items to share with and copy from others. Since our goal at this stage of investigation has been to explore the desirability, usability, viability, and feature set of a novel phone-based remote data access system, we have made some simplifying assumptions in the Courier design to help speed its iterative development and evaluation. Thus before describing the Courier architecture in detail, we will scope the functionality of the current system.

First, people often use more than one PC over the course of a day (e.g., work and home, or desktop and laptop), and it is reasonable to assume they would want to access data from any of those systems while on the go. However, the current version of Courier expects that users have designated only a single laptop or desktop computer as the one whose documents they want to access while mobile. This constraint will certainly be lifted in future versions of the Courier system.

Second, the value of Courier depends on providing users a means to *reliably* access the files and URLs located on their primary computer while they are mobile. With their increasing storage and computing capabilities, mobile phones have evolved into a natural platform for what Want [26] envisioned as the Personal Server – a conveniently-sized device that a) provides reliable access to a user's personal data and b) annexes nearby resources to support a rich computing experience. The benefit of relying only on phone-resident data is that it avoids the challenges imposed by access-controlled computing systems and networks. Taking this approach, Courier automatically caches files and URLs from a user's personal computer to their mobile phone; later, it is the phone-hosted copy of the data, rather than copy on the user's machine, that is shared with others. All transmissions to and from the mobile phone are performed using a short-range wireless radio protocol, so that even in situations where users only have access to the devices in their immediate environment (and not the networks), Courier can support flexible and spontaneous collaboration. As cloud computing [8] matures, we will undoubtedly revisit whether the phone should store the user's data, or simply pointers to the data. In the meantime, storing data on the phone makes it easier for us to perform lab-based studies and small-scale field deployments, so that we may improve our understanding of the usability aspects of Courier that do not depend on the underlying implementation.

Lastly, people will want to share data with others under a variety of circumstances, including formal meetings, casual encounters, social events, and so on. For this version of Courier, however, we have focused on the workplace meeting scenario, where the process of sharing data includes viewing a document as a group, and passing along copies of a document to the meeting participants for their personal use and archival. Since this use scenario expects that meeting participants want to *see* the files on which they are collaborating, and not just transfer files among one another, Courier assumes the availability of a meeting space PC for hosting, displaying, and mediating requests for shared documents. To support the meeting scenario further, Courier offers participants limited means to *interact* with the documents shared to the meeting room PC; participants can perform high-level browsing activities, such scrolling documents and moving windows, but detailed interactions, such as are required to edit a document, are restricted to the person in possession of the meeting PC's mouse and keyboard. Despite Courier's current specialization to the collaborative meeting scenario, the Courier architecture does not preclude peer-to-peer transmission of files in future versions, which would allow for data exchange in hallway or street settings, for example.

2.1 The Courier Architecture

Courier's role is to continually, wirelessly, and silently synchronize the documents and URLs a user creates, edits, and views from her primary PC (e.g., desktop computer at work) to her mobile phone, so that those resources can be readily available and sharable in mobile situations. When the user is away from her desk, collaboration with others is made possible by sharing and downloading files and URLs between the phone and a third-party PC; for the purposes of this paper we will assume this PC is one associated with a meeting room, but of course in practice it could be any Courier-enabled system, including, for example, a colleague's desktop computer. When the user returns to her office, any recently copied (downloaded) files are automatically synchronized back to her own PC. Here we describe in detail the system architecture that supports these processes.

The primary function of the Courier infrastructure is to enable the exchange of files between a Smartphone and (at least) two personal computers – the user's desktop PC and a meeting room PC. Hence, Courier consists of two main pieces of software: CourierMobile for the phone and CourierDesktop for the PCs that communicate with the phone. CourierDesktop is a C# Windows application that offers two independent services: a synchronization service called CourierSync, and a file sharing service called CourierShare. Typically CourierSync will be the service running on a user's personal desktop computer, while CourierShare will be the service running on a meeting room PC, but if desired, both services can run simultaneously on the same machine. CourierSync is designed to maintain a trusted, wireless "pairing" relationship for synchronizing files between a user's Smartphone and her primary personal computer, whereas the CourierShare service is meant to support the "meeting" scenario which allows multiple concurrent users to join a meeting (associated with a shared display) and to *share* or *download* files. Each service is associated with a distinct Bluetooth service GUID so that CourierMobile, which runs on the phones, can distinguish the services from one another.

CourierMobile is a C# (.NET Compact Framework) Windows Mobile Smartphone application. The CourierMobile user interface (described in detail in section 2.2) offers the user the ability to join meetings, which in practice involves searching for and connecting to PCs within Bluetooth range that are publishing the CourierShare

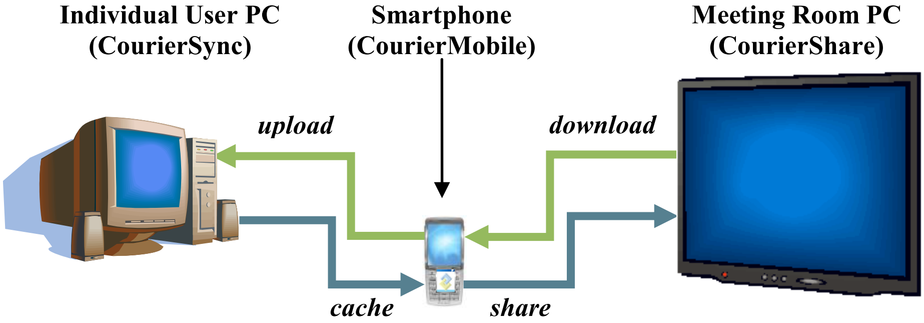


Fig. 1. High-level schematic of the Courier architecture. All communication between the Smartphone and the PCs is conducted via Bluetooth.

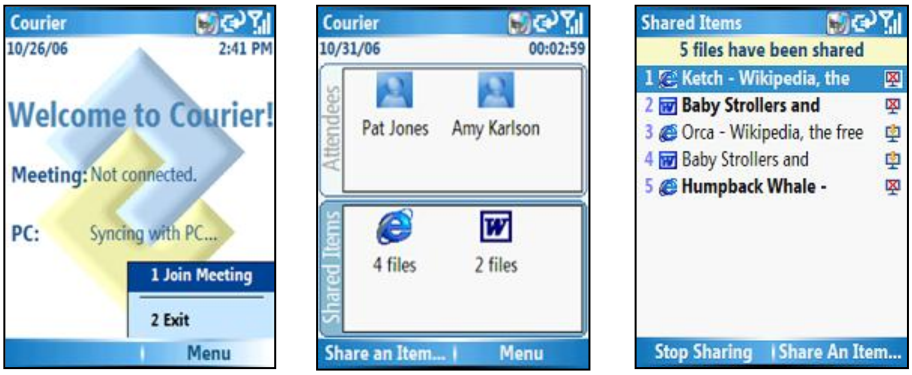
service. Once connected, CourierMobile works in conjunction with CourierShare to offer the user the ability to interactively exchange documents and control document windows.

Unlike CourierShare, which is a user-drive, interactive service, the CourierSync service operates without any user intervention. When CourierMobile is running, it performs a periodic scan to determine whether the user's personal (paired) PC is within Bluetooth range. If so, the phone connects to the paired PC and begins the sync process by requesting files that have changed since the last update. Courier uses the VibeLog [11] which is a Window's logging tool that tracks a user's desktop file and URL access activities. Our assumption is that the files and URLs a user is mostly likely to share are those accessed in the relatively recent past, so the context window is set to a reasonable constant, such as four weeks. Consulting its local VibeLog database, CourierSync determines if any files have changed, and if so, sends each new or updated file to the paired Smartphone, thereby completing the *cache* process. For web links, the streamed file is simply a ".url" file containing the URL to the web page, whereas for non-web file types (such as ".doc" or ".pdf" files) the file contents are transmitted in binary format and thus a full copy is created on the receiving device. CourierMobile then checks if any files downloaded from a meeting room PC or other Courier-enabled device have yet to be sent back to back to the paired PC. If so, these files are streamed back to CourierSync on the desktop and stored in a folder whose name encodes the time and PC name of the meeting from which the files were obtained, thereby completing the *upload* process. This two-way synchronization is repeated periodically as long as the paired connection between the phone and user PC is maintained. When the connection is lost, CourierMobile tries to re-establish it repeatedly on a decaying time schedule with increasingly long pauses between retries.

All Courier connections adhere to a request/response model with CourierMobile as the client and CourierDesktop (e.g., CourierSync or CourierShare) as the server. Each CourierMobile client is serviced by an independent CourierDesktop thread, which for example allows a meeting PC to easily repair a phone connection that gets dropped, as well as isolates each attendee's phone-based meeting connection from faults caused by other attendees' activities.

2.2 Interface Design

This section describes the Courier interface components and the workings of the PC-based CourierShare service in greater detail. On the phone, CourierMobile offers an initial status screen (as shown in Figure 2a) of service activity, and a menu of commands at the bottom right. Selecting the "Join Meeting" command from the menu initiates a Bluetooth search and returns a list of available meeting PCs (Courier devices within Bluetooth range that are publishing the CourierShare service), allowing the user to select a meeting to connect to. When the first CourierMobile user connects to a CourierShare service, CourierShare begins a "meeting" and records its start time. CourierShare subsequently tracks the number of attendees connected to the meeting as well as the files each has shared, and sends updates of this information to each connected device ("meeting attendee") whenever changes occur. While joined to a meeting, the CourierMobile meeting screen (Figure 2b) displays a continually-updated overview of the meeting's participating attendees and shared files.



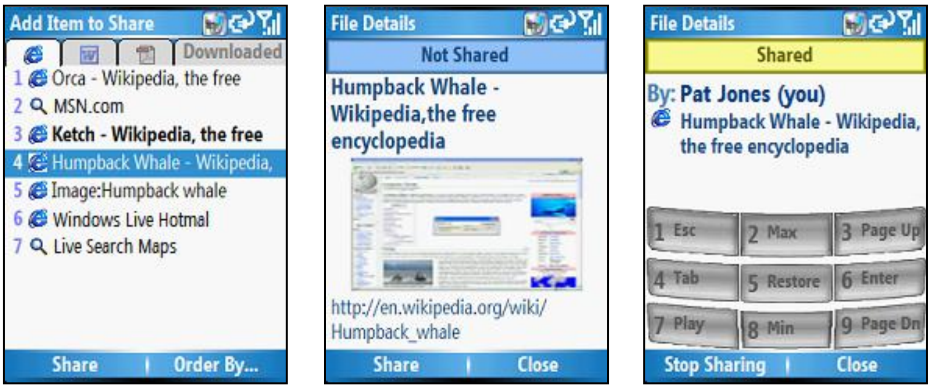
(a) (b) (c)

Fig. 2. (a) The CourierMobile welcome screen. (b) The meeting overview screen. The up/down directional controls toggle between the "Attendees" and "Shared Items". Selecting "Attendees" opens a screen providing contact details about attendees. Selecting "Shared Items" opens the list of shared items (c). Note that for illustrative purposes this screenshot reflects a situation where several files have already been shared by this attendee and others.

The Shared Items screen (Figure 2c) shows a list of all files that are currently available (shared) within the context of the meeting. Items in bold face are those currently shared by the phone's owner, while the remaining items are those shared by other attendees. The ownership attribute of a shared item is important because it dictates the operations that can be performed on the item; Courier allows *all* files to be viewed and manipulated, but only allows files shared by *others* to be downloaded, and only allows files shared by the *owner* of the phone to be unshared (revoked from the public meeting). The icons to the right of each line, together with the left context menu, indicate the operation allowed for each file: "Stop Sharing" (for files the *owner* has shared) or "Download" (for files *others* have shared). When the user chooses to "Download" a file, it is streamed from the meeting PC to the local file system on the phone. When the user chooses to "Stop Sharing" a file, 1) the shared file is removed from the public meeting lists, 2) the associated window on the meeting PC is closed, and 3) the file is deleted from the meeting PC. Note, however, that any copies already downloaded by other attendees are unaffected.

2.3 Interface Interaction

In order to share an item from the phone to the public meeting space, the user selects "Share an Item…" from any available menu (e.g. the meeting overview screen in Figure 2b or the Shared Items screen in Figure 2c). This takes the user to a specialized file browser of phone-resident files, shown in Figure 3a. Motivated by the "recent documents" view of the Windows File Explorer, the file browser displays files in a flat list by name rather than in a folder hierarchy by location. To help users find desired items, the files are grouped into separate tabs by file type, which users select among using the Smartphone's left/right directional navigation hardware. Within a tab, the right menu can be triggered to sort the items by name, cumulative access time, and last access time. These file characteristics are tracked by the VibeLog system,



(a) **(b)** **(c)**

Fig. 3. (a) The file selection screen for phone resident files, grouped into tabs by file type. (b) Details screen for an unshared file. (c) Details screen for a shared file.

which can provide certain document usage information that is not currently offered by the Windows file system. In this way, the browser provides a personalized view of a user's files based on their desktop work history. The default ordering of files is by last access time, under the assumption that the files users are most likely to share are the ones they have recently accessed.

By highlighting an unshared file in the list, the user can choose to share it immediately by pressing the left menu button, or get more information about the file by hitting the 'select' key located in the center of the phone's directional navigation pad. The file details screen (Figure 3b) for an unshared file displays the title of the file, its URL (for a web file) or original file path (for non-web files), along with as a small thumbnail of the file, also provided by the VibeLog system. The blue status area at the top of the details page indicates that the file is not currently shared. When the left "Share" menu is selected, CourierMobile streams the file to the meeting PC, where the file is opened, and made available to other meeting attendees for interaction and download.

Upon sharing an item, the file's status changes to "Shared" and the interface transitions to a screen that allows the user to remotely interact with the shared representation of the document (Figure 3c). Since each shared file corresponds to a window on the meeting PC's shared display, the details screen for a shared item offers the user the ability to send various control commands to the associated window using the keypad (for instance, scrolling the window contents up and down using '3' and '9', or moving it around the display with the joystick). Users can access these commands for items other attendees have shared by selecting an item within the Shared Items list, shown in Figure 2c.

3 User Study

We conducted a formative evaluation of Courier to understand the system's usability in terms of learnability, desirability, perceived effectiveness, and user satisfaction. In particular, we were interested in following issues:

- Can users effectively find and share previously accessed documents and URLs?
- Can users effectively download files that other have shared?
- Can users effectively interact with (e.g., move, scroll, etc) documents that have been shared in a meeting?
- Are users satisfied with the workflow of sharing and downloading files?
- What functions or capabilities do users feel are missing?

To answer these questions, we developed a task scenario based on information sharing and document exchange in a workplace setting. At a high level, the scenario consisted of three phases: 1) we asked two participants to work at separate desktop PCs and perform tasks that involved accessing a set of local documents or web URLs; 2) we then asked the participants to join a meeting and perform tasks that required each participant to share and exchange some of the documents and URLs each had accessed at their PC; 3) finally, we asked the participants to return to their PCs and perform tasks that required them to find and open the documents they had received from the other participant during the meeting.

3.1 Tasks

Study tasks were chosen to exercise Courier's core system features, while maintaining reasonable similarity to a real-world use case. Two participants worked in parallel at separate PCs to answer eight questions whose answers could be found either within online Wikipedia (en.wikipedia.org) web pages, or within Consumer Reports articles archived locally as Word and PDF files. Each question included a hyperlink either directly to the associated Wikipedia page, or to a system file directory that contained the required file. For example, the question "At what age do Bald Eagles begin to show their distinctive white head?" included a link to http://en.wikipedia.org/wiki/Bald_eagle. Topics were chosen so that they could be grouped into pairs of documents with similar length and comparable factoids. Paired documents were then split across the two participants' PCs.

After answering the first eight questions at their own PCs, participants then gathered at a "meeting" to answer six questions that required comparing content between two documents – one from each user (e.g., "Which bird can breed at a younger age, the **Emperor Penguin** or the **Bald Eagle**?"). This step required users to share a document they had previously read at their PC, and work together to find the answer. During this collaborative step, participants could use Courier's window control commands to move and scroll windows independently on the large shared meeting display. Although we provided a wireless mouse and keyboard for controlling the meeting room PC, we did not suggest how participants should share those resources for completing their tasks; participants were free to use the mouse or their phones to manage windows, and could close or unshare documents freely.

Before returning to their PCs, participants were instructed to exchange four of their files, which required both *share* and *download* operations. Finally, participants returned to their PCs to answer four questions whose answers were found within the files they had each received from their partner.

3.2 Participants and Procedure

Twenty participants (12 male) were recruited from within a large software corporation. The only criterion for eligibility was that participants were required to have owned a Smartphone for at least 3 months prior to the study; 84% of participants owned their phone for more than 1 year. Most (78%) of the participants used their devices for activities other than phone calls at least four hours per week. The average participant age was 34.5 years. Participants were scheduled in pairs for each of the 10 study sessions. After a verbal welcome, participants read a brief description of the Courier system, after which the administrator gave a slide presentation of the Courier architecture, interface and study procedure. Participants were instructed to think aloud during the “meeting” portion of the study, but were given no explicit instruction about task timing. The introduction activities lasted approximately 30 minutes, after which the study proper was begun. Participants were assigned to an “office” PC with an associated Smartphone, and instructed to begin the study tasks. When all questions on the PC had been answered, participants took their phones to the “meeting”, answered the comparative questions, exchanged files, and returned to their PCs to answer the remaining questions. Participants then performed a second task sequence (PC → meeting → PC), and concluded the study with a subjective questionnaire.

3.3 Measures

Users were asked to rate 25 features of the Courier system, including its usability (e.g., overall satisfaction, learnability and ease of use), workload (e.g., mental demand, time pressure and frustration), and effectiveness for accomplishing the task; participant ratings were recorded on a 7-point Likert scale. Since participants had two opportunities to collaborate in a “meeting” setting, the administrator recorded the total time each “meeting” session lasted. The administrator also recorded instances of technology breakdowns, participants’ comments on the system usability, and problems participants encountered with the user interface.

3.4 Equipment

The PC “office” environment for each user consisted of a 3.6 GHz Pentium 4 HP Workstation with 2GB of RAM, which drove two side by side NEC MultiSync LCD 1880SX monitors set at 1280x1024 resolution. Each workstation ran CourierSync and was configured to pair with an associated Cingular 2125 Windows Mobile Smartphone running the CourierMobile client. The “meeting” environment was hosted by a PC with the same specifications, but which used two side by side projectors set at 1024x768 resolution to create a 2048x768 projected wall display. The "meeting" PC ran CourierShare. All 3 Workstations were equipped with a D-Link DBT-120 USB Bluetooth adapter. During the "meeting", participants sat within 4 meters of both the meeting server and wall display. The "meeting" environment also included a wireless mouse and keyboard.

4 Study Results

4.1 Subjective Ratings

Average subjective ratings (Figure 4) were quite high given that the Courier version studied was a first iteration design. These results are especially interesting in light of the real-time problems users experienced due to limitations of the VibeLog logging system and the design choice to use Bluetooth for data and control communications.

Some of the most positive ratings were attributed to Courier's overall usability; ease of use, satisfaction, and learnability all received average ratings of between 5 and 6. Similarly, participants rated Courier highly with respect to supporting its primary capabilities of sharing (5.7) and downloading (6.1) files. To gauge overall interest in the Courier capability, we asked users to rate how much they would prefer to use Courier over traditional data sharing methods for common workplace scenarios: 74% (N = 14) of the participants stated a preference for using Courier to share documents among a small group of friends or colleagues versus 58% (N = 11) for presenting documents in a meeting. Only 16% (N = 3) of the participants said they would not use Courier in either scenario. Finally, over half of those who chose to enter free-form

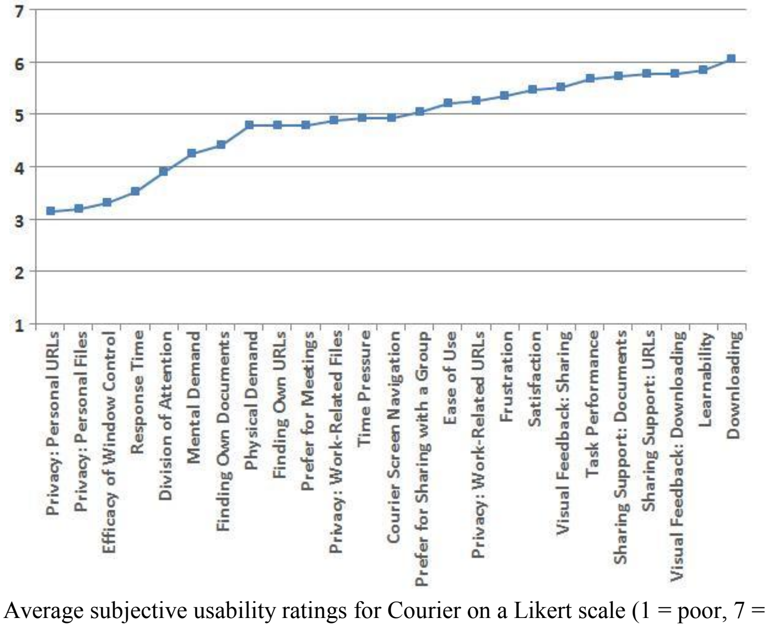


Fig 4. Average subjective usability ratings for Courier on a Likert scale (1 = poor, 7 = good). Workload measures (mental demand, time pressure, frustration) have been inverted so that a high rating always indicates more positive opinion.

comments (9 out of 16) did so to express enthusiasm and immediate interest in Courier ("Please please please make this product real. I would really love to use this in my day to day meetings!", "Overall, I love this system and could use it today. I have a task waiting for me that I could have taken care of in a meeting yesterday.", "Nice product – I already love it", "I love this!").

Despite the generally positive feedback, there were emergent themes that highlighted areas in which Courier would need to be improved before being more broadly deployed. In particular, Courier did not rate highly in the areas of privacy, reliability and response time.

Privacy: Although we deliberately avoided discussing issues of privacy during the study proper, the post-study questionnaire asked participants to record their comfort level for sharing items during the meeting. Users gave the lowest average subjective ratings for their comfort level in sharing personal documents and URLs (3.2). On the other hand, users were considerably more comfortable sharing work-related documents (4.8), and were least concerned about sharing work-related URLs (5.2). One explanation for the uneasiness users felt might be that the automated syncing of personal files between the PC and phone undermines user confidence in terms of controlling and understanding the flow of data to and from their phones. Privacy concerns might be mitigated by increasing the visual feedback about the data flow, providing finer grained and/or explicit control over the documents which are synced to the device, allowing variable access control to the files which are shared, or allowing meeting participants to control meeting membership.

Reliability of Window Control: The capability that Courier offers to allow meeting participants to remotely control (e.g., move and scroll) shared meeting documents was given the third lowest rating (3.3) across all Courier features we evaluated. There was little evidence to suggest that such low opinion was due to a lack of interest in the feature – on the contrary, all participants made considerable use of the remote window movement and scrolling features, even among those who initially expressed skepticism about the feature's usefulness. More likely, users were reacting to the frequent technical failures they encountered when issuing Courier's window control commands. The root of the problem was that the Windows API does not offer a direct mapping between windows and documents, and the heuristics Courier employed for associating documents to open windows had specific failure cases that occurred during the study. Since most participants experienced loss of window control to some degree, it is not surprising that this feature received low ratings.

Bluetooth Response Time: We chose Bluetooth as our wireless communication protocol for the convenient properties it provides with respect to availability (most mobile devices now entering the market are equipped with Bluetooth), short range (which limits interference with simultaneous meetings), and ad hoc organization (users do not need special permissions to access the network). Most current mobile phones employ the Bluetooth 1.0 protocol which supports a 723 Kbps transfer rate. Under these conditions, most files in our study should have taken less than 5 seconds to share and download. However, due to the serial nature of the Bluetooth protocol, we experienced great variability in file transfer rates: from between 150 – 420 Kbps for syncing at the user's PC (CourierSync), and from 25-315 Kbps for sharing and downloading at the meeting (CourierShare), depending on the number of concurrent file transfers that were taking place, and the phone's proximity to the server. In the worst case, the largest files we used (318K, not particularly large by real-world standards) could have taken up to a minute and a half to transfer! Several users commented specifically on the slowness of file transfers, which was conveyed more generally by the low average rating for system response time (3.5). In addition to problems with communication speed, we experienced considerable variability in the persistence of the Bluetooth connection. Approximately a third of the participants were spontaneously disconnected from a meeting during the study, which required them to rejoin and re-share any documents they had been working with.

The low subjective rankings given to Courier's privacy, reliability, and response time all reflect significant usability problems with the current system. Although we have mitigation strategies to address each concern in future versions of Courier, it is encouraging that even despite these issues, the majority of participants reported an interest in using Courier over the methods they currently use to share data with others in meeting settings.

4.2 Time

We recorded the elapsed time of the two "meetings" for each session. Of course, because we encouraged think-aloud feedback, the meeting time included participant comments, communication about the task, recovery from lost Bluetooth connections, and so on. Even so, we wanted to see if a trend could be observed in the amount of time it took for users to perform tasks from one meeting to the next. Two of the 10 pairs of participants did not finish the second meeting due to a 2 hour limit on the total study time. The average time of the first meeting for the remaining 8 groups was 24.8 minutes, versus 15.1 minutes for the second meeting. Paired t-test analysis revealed that on average the second meeting was significantly faster than the first meeting ($p < .001$). Since users answered the same number and types of questions, we attribute this difference to increased comfort with Courier. The high subjective score for system learnability (5.8) supports this conclusion.

5 Related Work

Many research efforts have focused on improving productivity and collaboration in meeting settings. Structured meetings - characterized by agendas, advanced scheduling, and prepared materials - lend themselves well to technology-rich solutions that seamlessly integrate individual computing resources (e.g., workstations, PDAs or laptops) with shared displays (e.g., walls or tabletops). Examples of such meeting environments include CoLab [23], iRoom [13], and i-LAND [24] and related cross-device interaction techniques [4, 14, 19]. While these infrastructures break new ground towards improving productivity in a meeting setting, they come with high material, setup and maintenance costs. By contrast, Courier is decentralized, lightweight, and low-cost.

Other research has centered primarily on collaborative *application interaction*. Single display groupware (SDG) was coined to describe systems that support multiple co-present users of an application on a shared display, exemplified by the multimouse drawing program Kidpad [6]. The use of personal mobile devices to enable multi-user interaction with a shared display was a natural follow-on to this work; since then, PDAs have been explored as general input devices [16], local object palettes [18], and equal partners in applications that split content between private and public devices [10]. More recently, mobile phones have been adapted to serve as general input controls to large public displays [2]. While Courier does not limit interaction to a single application, it restricts multi-user input to serialized, high-level window commands. As an alternative to SDG, Biehl and Bailey [3] have proposed a model for multi-device collaborative settings whereby shared applications are redirected from public displays to private devices when users need to perform editing tasks, and then returned to the public display; this approach avoids issues of concurrent input entirely, but is better suited for personal devices with ample input and output capabilities, such as laptops, than for PDAs or phones.

A mobile device may be just one component of a user's personal arsenal of networked computing resources. Infrastructures such as Segank [22] focus on providing users flexible, remote access to personal information spaces by creating a unified, consistent view of a users' distributed resources. Such systems consider primarily issues of replication, freshness, and network quality, rather than the support of real world data exchange tasks from small mobile devices. However, mobility has certainly spawned interest in flexible, spontaneous sharing and exchange of data among individuals. For laptops, the focus has been on mechanisms for data movement and sharing: WinCuts [25] propagates subregions of a user's laptop screen to a large display for shared viewing, but does not actually facilitate the exchange of source documents. Dynamo [12] supports simultaneous connections to a multi-user desktop environment for synchronous and asynchronous media exchange, but its focus on laptops and USB drives means that data exchanges must be more planned than with Courier, since people do not always carry their laptops, and must anticipate an exchange by explicitly copying files to a USB drive. UbiTable [20] allows multiple users to transfer materials between laptops and an interactive table. Courier shares the spirit of these systems by using a shared display as a proxy for file exchange among users, but instead targets spontaneous, phone-based interaction.

Several recent forays into mobile file exchange have been motivated by peer to peer (P2P) systems. Speakeasy [7] is a framework for opportunistically sharing arbitrary services, resources, and data among laptops in a decentralized manner. For smaller devices like phones and PDAs, the focus has been on accessing and sharing *pointers* to users' remote data, offering various strategies to overcoming access privileges, firewalls, and dynamic IP allocation. Satchel [15] targets PDAs and allows users to browse the file hierarchies of their remote, networked systems, then to send security-enhanced URLs to devices, viewers, printers, and scanners within the vicinity. SEREFE [1] avoids the problems with firewalls and IP allocation by performing remote file query and forwarding via an augmented instant messenger service. Both Satchel and SEREFE require the user's remote devices to be on and accessible, which is not an underlying assumption of Courier. In addition, neither Satchel nor SEREFE address group collaboration dynamics, only data exchange.

6 Conclusion

Courier leverages mobile phone ubiquity to allow people to view, share, and exchange files and URLs from their personal work environment anywhere they have their phones and access to a shared display. Despite technical limitations of our initial implementation, a formative evaluation demonstrated that Courier's strategy of automatically syncing recently-accessed documents between a user's primary PC and their phone, and allowing users to share that information in an ad hoc, opportunistic manner, is an attractive alternative to the methods that are available to users today. We intend to improve Courier's remote document control features and the performance of the Bluetooth transport mechanism before deploying a field study to learn more about Courier's utility and performance when users are working with their own documents. We are particularly interested in developing a better understanding of users' privacy concerns and evaluating whether our data caching algorithm meets user needs and expectations with respect to the documents they have readily available to share when they are away from their laptop or desktop computer.

Acknowledgments. We extend our sincerest thanks to A.J. Brush for her helpful edits and suggestions for improving the presentation of this work.

References

1. Ahn, J. and Pierce, J.S. SEREFE: serendipitous file exchange between users and devices. *Proc. MobileHCI 2005*, pp. 39-46, ACM Press, 2005.
2. Ballagas, R., Rohs, M. and Sheridan, J.G. Sweep and point and shoot: phonecam-based interactions for large public displays. *Extended abstracts of CHI 2005*, pp. 1200-1203, ACM Press, 2005.
3. Biehl, J.T. and Bailey, B.P. Improving interfaces for managing applications in multiple-device environments. *Proc. AVI*, pp. 35-42, ACM Press, 2006.
4. Booth, K.S., Fisher, B.D., Lin, C.J.R. and Argue, R. The "mighty mouse" multi-screen collaboration tool. *Proc. UIST 2002*, pp. 209-212, ACM Press, 2002.
5. Chipchase, J., Persson, P., Piippo, P., Aarras, M. and Yamamoto, T. Mobile essentials: field study and concepting. *Proc. DUX*, ACM Press, 2005.
6. Druin, A., Stewart, J., Profit, D., Bederson, B.B. and Hollan, J.D. KidPad: a design collaboration between children, technologies, and educators. *Proc. CHI 1997*, pp. 463-470, ACM Press, 1997.
7. Edwards, W.K., Newman, M.W., et al. Using speakeasy for ad hoc peer-to-peer collaboration. *Proc. CSCW 2002*, pp. 256-265, ACM Press, 2002.
8. Gilter, G. The Information Factories. *Wired Magazine*, Vol. 14, No. 10, pp. 178-202, 2006.
9. Golvin, C.S., Schadler, T., Fiorentino, R. and Lo, H. *The State of Consumers and Technology*. Forrester Research, Inc., 2007.
10. Greenberg, S., Boyle, M. and Laberge, J. PDAs and shared public displays: making personal information public and public information personal. *Personal and Ubiquitous Computing*, Vol. 3, No. 1-2, pp. 54-64, 1999.

11. Hutchings, D.R., Smith, G., Meyers, B., Czerwinski, M. and Robertson, G. Display space usage and window management operation comparisons between single monitor and multiple monitor users. *Proc. AVI 2005*, pp. 32-39, ACM Press, 2004.
12. Izadi, S., Brignull, H., Rodden, T., Rogers, Y. and Underwood, M. Dynamo: A public interactive surface supporting the cooperative sharing and exchange of media. *UIST 2003*, pp. 159-168, ACM Press, 2003.
13. Johanson, B., Fox, A. and Winograd, T. The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *IEEE Pervasive Computing*, Vol. 1, No. 2, pp. 67-74, 2002.
14. Johanson, B., Hutchins, G., Winograd, T. and Stone, M. PointRight: experience with flexible input redirection in interactive workspaces. *Proc. UIST 2002*, pp. 227-234, ACM Press, 2002.
15. Lamming, M., Eldridge, M., Flynn, M., Jones, C. and Pendlebury, D. Satchel: providing access to any document, any time, anywhere. *ACM Trans. Comput.-Hum. Interact.*, Vol. 7, No. 3, pp. 322-352, 2000.
16. Myers, B.A., Stiel, H. and Gargiulo, R. Collaboration using multiple PDAs connected to a PC. *Proc. CSCW 1998*, pp. 285-294, ACM Press, 1998.
17. Pittet, S., Ekholm, J., Gupta, M., Hart, T.J., Mitsuyama, N. and Nguyen, T.H. *Forecast: Mobile Services, Worldwide, 2002-2011 (3Q07 Update)*. Gartner, Inc., 2007.
18. Rekimoto, J. Pick-and-drop: a direct manipulation technique for multiple computer environments. *Proc. UIST 1997*, pp. 31-39, ACM Press, 1997.
19. Rekimoto, J. and Saitoh, M. Augmented surfaces: a spatially continuous work space for hybrid computing environments. *Proc. CHI 1999*, pp. 378-385, ACM Press, 1999.
20. Shen, C., Everitt, K. and Ryall, K. UbiTable: Impromptu fact-to-face collaboration on horizontal interactive surfaces. *Proc. UbiComp 2003*, pp. 281-288, Springer, 2003.
21. Shiffler, G. *Forecast: PC Installed Base, Worldwide, 2003-2011, September 2007 Update*. Gartner, Inc., 2007.
22. Sobti, S., Garg, N., et al. Segank: A Distributed Mobile Storage System. *Proc. USENIX Conf. on File and Storage Technologies*, pp. 239-252, USENIX Association, 2004.
23. Stefik, M., Foster, G., Bobrow, D.G., Kahn, K., lanning, S. and Suchman, L. Beyond the chalkboard: computer support for collaboration and problem solving in meetings. *Communications of the ACM*, Vol. 30, No. 1, pp. 32-47, 1987.
24. Streitz, N.A., Geißler, J., et al. i-LAND: an interactive landscape for creativity and innovation. *Proc. CHI 1999*, pp. 120-127, ACM Press, 1999.
25. Tan, D.S., Meyers, B. and Czerwinski, M. WinCuts: manipulating arbitrary window regions for more effective use of screen space. *Extended abstracts CHI 2004*, pp. 1525-1528, ACM Press, 2004.
26. Want, R., Pering, T., Danneels, G., Kumar, M., Sundar, M. and Light, J. The personal server: changing the way we think about ubiquitous computing. *Proc. UbiComp 2002*, pp. 194-209, Springer, 2002.