Opportunistic Annexing for Handheld Devices: Opportunities and Challenges

Jeffrey S. Pierce, Heather Mahaney
Georgia Institute of Technology
{jpierce, mahaneyh, abowd}@cc.gatech.edu

1 Introduction

People are increasingly using personal digital assistants (PDAs) and cellphones in their daily lives. The primary advantage of these devices is that they are mobile. They allow people to perform tasks requiring computation and communication without tying them to specific locations. Despite the popularity of these devices, they do have drawbacks. Most significantly, they impose a trade-off between luggability and capability. To get smaller and lighter devices, users must accept reduced processing power, storage space, communication capabilities, interaction quality, and battery life. Our goal is to change the balance in this trade-off by improving interaction with mobile devices without increasing their size or weight (Figure 1).

Mobile devices suffer from poor interaction quality because they have limited input and output (I/O) resources, in particular small screens and slow input methods. These limited I/O resources reduce users' willingness to complete tasks with mobile devices, and they limit performance on many tasks users are actually willing to perform. Despite these drawbacks, increasing the I/O resources of mobile devices by decreasing their luggability (i.e. increasing their size or weight) is not a viable option. PDAs and cellphones must remain small enough that users can easily slip them into a pocket.

We propose to allow users to *opportunistically annex* input and output resources, such as desktop monitors, televisions, speakers, keyboards, mice, and microphones, that they encounter in the environment. Opportunistic annexing is the process of temporarily attaching one or more of these resources to a computational device to enhance its capabilities, creating what Olsen et al termed opportunistic assemblies [Olsen01]. In this paper we discuss the potential benefits of opportunistic annexing, describe some of the challenges to designing the user experience, and discuss potential software and security architectures.

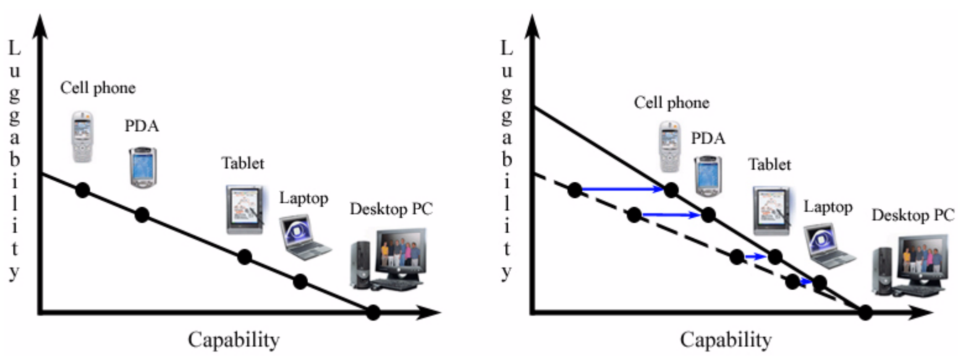

Figure 1: The current state of the luggability vs. capability trade-off (left) and our goal (right).

2 Background

Previous research efforts have explored a variety of approaches to improving the capabilities of mobile devices without decreasing luggability. Grid computing research holds the potential to increase the processing power available to mobile devices (e.g. [Gonzalez]) by allowing them to draw on the processing resources of fixed devices. Research on networked storage (e.g. [Burge]) could allow mobile devices to increase their virtual storage capacity by utilizing information repositories stored in a fixed location. Mesh or ad hoc networking (e.g. [Corson]) could improve the communication capabilities of mobile devices by allowing them to draw on the communication resources of nearby devices. Adaptive power algorithms [Flinn99, Flinn02] could increase the battery capacity of mobile devices by adjusting performance and quality to reduce power demands.

One research direction on improving interaction with mobile devices is to make more effective use of devices' existing capabilities. Researchers have created a variety of different stroke-based [Mankoff, Perlin98] or soft keyboard-based input methods [MacKenzie99a, Zhai00, Zhai03] that improve on the Graffiti stroke-based or QWERTY soft keyboard input methods most PDA users employ. Despite the performance improvements offered by these methods, none has yet achieved widespread use. Another approach researchers have taken is to add sensors to mobile devices. These sensors allow users to interact with PDAs [Hinckley00, Yee] and cell phones [Hinckley01] using a variety of physical gestures.

Researchers have tried to make more effective use of the limited screen real estate provided by mobile devices. One method is to adapt zoomable user interfaces [Bederson94, Perlin94] to mobile devices. WebThumb [Wobbrock] and the Fisheye Calendar [Bederson03] are both examples of this approach. A similar approach is to create interfaces with both focus and context views (e.g. [Bjork99, Bjork00]). Other methods researchers have explored include using transparency to overlap user interface elements without occlusion [Kamba] and adding visual scent [Pirolli] to help communicate information that is currently off screen [Baudisch].

A second research direction on improving interaction with mobile devices is to adapt content to be better suited to them. Research on adapting content (e.g. web pages, images, video) to mobile devices has concentrated on both static and dynamic methods. Standards like the Wireless Application Protocol allow designers to create content specifically for mobile devices. Dynamic approaches try to avoid the need to manually redesign content for mobile devices by adapting content on the fly [Buyukkokten, Korva, Metso, Noble]. The drawback to dynamic approaches is that not all content can be easily adapted on the fly. Prior research suggests, for example, that adapting existing World Wide Web (WWW) content can be difficult because of the complexity of web pages [Kim].

Opportunistic annexing complements rather than competes with this prior research. The research on improving input methods, designing for small screens, and adapting content to mobile devices offers approaches to improve interaction when users work with a mobile device by itself, while opportunistic annexing concentrates on improving interaction by drawing on other devices. The research on improving the processing power, storage capacity, communication capabilities, and battery life, when combined with our efforts, should lead to a significant improvement in the overall capabilities of mobile devices without a significant decrease in luggability.

Olsen et al's Join and Capture research [Olsen01] and Intel's research on the Personal Server [Want] are the closest in spirit to opportunistic annexing. Olsen et al proposed taking advantage of interaction resources in the local environment by creating opportunistic assemblies of devices. They proposed a model where users create assemblies by Joining or Capturing resources and discussed an implementation of those primitives using their XWeb architecture [Olsen00]. We aim to extend their work by exploring the potential of, and design space for, annexing devices in more depth.

The goal of Intel's research is for users to carry a small, display-less device, the Personal Server, and access it using I/O resources they encounter. While the Personal Server project also focuses on co-opting local I/O resources, we focus on different goals. Intel's goal is to create an additional device for users to carry that provides all the processing and storage capabilities they require. Our goal is to improve interaction with the mobile devices that users are already carrying. Despite this difference, we anticipate that some of the lessons learned from the Personal Server project will be relevant to our research, just as some of the lessons we learn will be relevant to their work.

Another relevant area of research is the effort to improve the capabilities of mobile devices by utilizing them as part of a heterogeneous collection of devices. Instead of extending the existing capabilities of mobile devices, this research concentrates more on assembling the heterogeneous devices into a larger, aggregate device. Research on single display groupware [Greenberg, Myers98, Rekimoto98] is one example of this type of research. Users typically each employ their own PDA as an input device and coordinate their efforts on a larger, shared display. Another research project in this area is the Pebbles project [Myers01], which is exploring how users can employ PDAs in conjunction with desktop computers [Myers] and consumer electronics devices [Nichols]. Other research has examined how users can employ their mobile devices as access points into smart environments or rooms. The iRoom [Fox, Johanson02], i-Land [Streitz], Cooltown [Kindberg01], EasyLiving [Brummitt], and SCF/CSCDE [Pham00, Pham01] projects are all examples of this type of research. This research focuses primarily on using mobile devices as adjuncts to devices in the environment, while opportunistic annexing will allow devices in the environment to act as adjuncts to mobile devices.

3 Potential Benefits of Annexing

We anticipate that the primary benefit of annexing input resources will be faster input rates. Experts estimate that skilled touch typists can enter an average of 72 words per minute (wpm), while average users enter approximately 50 wpm [Card]. Based on experimentally measured rates for the three most popular input methods for PDAs, Graffiti (21.5 wpm for average users [Fleetwood]), transcription (17.6 wpm for experts [MacKenzie99b]), and soft QWERTY keyboards (20.2 wpm for novices [MacKenzie02]), annexing a keyboard could allow PDA owners to enter data almost four times faster. Estimates for multi-tap (27 wpm) and T9 (41 wpm) [AOL] experts [Silfverberg] suggest that cell phone users could enter data two to three times faster.

To determine the potential benefit of annexing a keyboard, we conducted an initial study of 14 users to gather the number and length of calendar entries, notes, tasks, and contacts on their PDAs. We determined that the average length of their personal information items was 157 characters. If an average Graffiti user enters 21 wpm and an average touch typist enters data at 50 wpm, then an average users could save 52 seconds by annexing a keyboard to enter an average information item.

Annexing a larger display should improve interaction with mobile devices because previous research demonstrates that screen size affects interaction, although the size of the effect when moving from a very small display to a larger one is an open question. Researchers exploring the effects of display size have primarily concentrated on task performance differences between desktop monitors and larger (e.g. projected) displays. For example, studies have shown that larger displays help equalize performance between men and women on navigation tasks [Tan03b] and increase performance on spatial tasks [Tan03a] and on representative office tasks [Czerwinski].

There has been some research on the effects of smaller displays on reading speed and search tasks. According to one study, reading speed increases 25% when document width is increased to full screen (18.7 cm) from a third of the screen [Duchnicky]. Another study suggested that smaller displays force users to reread text more often [Dillon]. Users reading a document on a 20 line display jumped in the text three times more often than users with a 60-line display. Small screens also appear to make it more difficult to search for information. Users completing four question-based search tasks on a 30 line display answered twice as many questions correctly as users with a 15-line display [Jones].

4 Designing the User Experience

A primary issue for annexing devices is whether the user annexes devices using his handheld or using the devices he wants to annex. The former can be thought of as *pushing* the interface out to other devices from the handheld and the latter as *pulling* the handheld interface out using other devices. Note that users will not annex resources directly, although this may be possible in the future. Instead users will annex resources through the computer to which they are attached; I will refer to this computer as the *annex proxy.* Annexing also presents challenges when designing user interfaces: we must learn how to effectively divide interfaces across annexed devices and design interfaces given uncertainty about the devices users will employ. Users annexing devices are only part of our user population; we must also determine how device owners should specify who can annex their devices. Owners will need to be able to specify which of their devices users can access and what permissions different users have on those devices.

4.1 Annexing Devices

Annexing via pushing provides more privacy because the mobile device does not need to broadcast its presence, and it provides better security by eliminating the risk of unauthorized connections. While researchers have created several methods for establishing communication between two devices using specialized hardware, such as scanning an RFID tag, reading an IR beacon [Kindberg00], bumping the device into the resource [Hinckley03, Tandler] or shaking them together [Holmquist], we believe that there is a further need for light-weight methods, such as entering a numeric code on the mobile device, that does not require specialized hardware.

The primary disadvantage of annexing via pushing is that the user needs direct access to the mobile device; this approach is not possible if the device is currently stowed in the user's backpack. Annexing via pulling allows users to annex resources without interacting with their mobile device. For example, a user talking on a cell phone could use a nearby display to view his calendar without interrupting his phone conversation. The user attaches the resource to his device using the annexing proxy to which the resource is connected. Annexing via pulling requires that mobile devices broadcast their presence, and it forces users to authenticate themselves to their device when annexing a particular resource. Designing this authentication process is challenging because passwords are not a viable option; if users employ a password to log into their mobile device from a borrowed keyboard they leave themselves open to replay attacks. Researchers have explored alternative authentication methods that do not rely on passwords such as hardware tokens [Corner] and selecting a subset of images [Pering].

Regardless of whether users annex devices via pushing or pulling, the process must be sufficiently quick and easy that annexing adds value. If a user could save 5 seconds by typing rather than writing a note on their PDA, annexing the keyboard must take less than 5 seconds to be worth the user's time.

4.2 Designing Under Uncertainty

Designing applications that take advantage of opportunistic annexing will require learning how to design interfaces given uncertainty about which I/O devices users will employ. We need a method of creating interfaces that addresses this uncertainty. Two currently used methods are handcrafting interfaces for a variety of devices in advance and generating interfaces at run-time.

Handcrafted interfaces tend to be easier to use and more aesthetically appealing, but the effort required to create them may restrict the types of devices that users can annex. Automatically generated interfaces (e.g. [Nichols, Olsen00]) provide a semantic description of the desired interface that annexed devices can use at run-time to create an interface suited to their abilities. This approach allows interfaces to adapt to a variety of I/O resources, but it does not make any guarantees about the quality of the interfaces.

Because we expect that, in practice, users will primarily annex desktop displays, keyboards, and mice, we believe that hybrid approach may be optimal. Rather relying solely on one approach or another, we could craft interfaces for common cases and rely on generated interfaces when users annex unanticipated devices. The iCrafter system [Ponnekanti] uses a variant of this approach.

4.3 Dividing Interfaces Across Devices

With the exception of research on computer-supported cooperative work, where researchers have done some exploration into effectively spreading interfaces across multiple devices for multiple users, most existing interfaces presume a single display and one or two input devices. Because opportunistic annexing allows users to employ multiple I/O devices, we must explore how to spread the interface across these devices to utilize them most effectively.

One of the questions we will need to answer is whether we should treat multiple displays as separate spaces or as a single workspace. For some applications the decision may be clear cut, but for others either choice may be viable. Consider a program that allows users to show pictures on an annexed monitor. One way to structure the interface would be to separate the display spaces so that the user selects an image on the handheld and it appears on the monitor. Another way would be to treat the two as a single workspace and make the user drag images between the displays. Previous systems have applied both approaches. For example, the SlideShow Commander application [Myers00], the iRoom's multibrowsing functionality [Johanson01], and the Pick and Drop technique [Rekimoto97] all treat multiple displays as separate workspaces, while the iRoom's PointRight functionality [Johanson00] and the hyper-dragging technique [Rekimoto99] treat multiple displays as a single workspace.

We will also need to learn how users will want to divide an interface to protect their privacy. Given that people are more likely to "eavesdrop" on information on large displays than small ones [Tan03c], users may be reluctant to show certain information on large, public displays. We recognize that an interface designed to protect a user's privacy may not be the most effective in terms of speed or space, but an application that shows a user's social security number on a large public display will probably not be acceptable to users.

4.4 Closing the Connection

A final annexing design issue is determining how long devices remain annexed. In some cases it may make sense to allow users to only annex devices for a single, discrete action. In other situations we might allow users to annex a device for an unlimited amount of time. Between these extremes, we might allow users to "lease" annexed devices for a specified amount of time. The appropriate solution will depend on a variety of factors: the desired use, the user's identity, who owns the annexed device, etc.

4.5 Granting Access

Users annexing devices are only part of our user population; we must also determine how device owners should specify who can annex their devices. Owners will need to be able to specify which of their devices users can access and what permissions different users have on those devices. Owners should be able to assign permissions to both individuals and groups of users.

As their collection of devices grows, owners may want a more centralized method of specifying who can annex which devices. For example, the computer support staff for a university might want to be able to set the permissions for all of the devices the university owns simultaneously from a single location, rather than assigning permissions to each device separately.

We must also determine the appropriate levels of granularity for owners to use when setting permissions. For example, the owner of a printer might want to be able to specify different page quotas for different classes of users, while the owner of a display might want to be able to specify different lease lengths and maximum window sizes for different individuals. Real world experience will be needed to determine the types and values of parameters that owners should be able to specify.

4.6 Evaluation

We need to demonstrate that handheld users will derive value from opportunistically annexing I/O devices before we expend the time and effort necessary to create a complete infrastructure. While we can make a priori arguments for the value of some applications, we believe the best way to demonstrate value is to create a variety of applications that simulate opportunistic annexing and formally evaluate them. For tasks that users can perform with a handheld alone, these evaluations will attempt to establish that opportunistic annexing allows users to complete those tasks more quickly. For new tasks that annexing makes possible, these evaluations will attempt to establish that annexing allows users to effectively complete those tasks.

Unfortunately, demonstrating benefits in the laboratory will not be enough, because those benefits do not always transfer to real world use. Users may find applications initially exciting, but stop using them after the novelty wears off. By deploying the applications they created on a large scale and studying long-term use, the ActiveCampus project researchers at UCSD demonstrated that adoption rates for applications can differ significantly [Griswold]. To study whether and how users adopt and use the applications that opportunistic annexing makes possible, we will similarly need to deploy them to a large audience and conduct long-term studies [Consolvo].

5 Architectures for Interfaces

We will need a software architecture that allows mobile devices to display interface elements on annexed displays and to receive user actions from annexed input devices (Figure 2). The primary issues for the architecture are how the mobile device tells the annex proxy what to display and what information the mobile device receives from the annex proxy about the user's actions. Table 1 lists approaches from previous research.

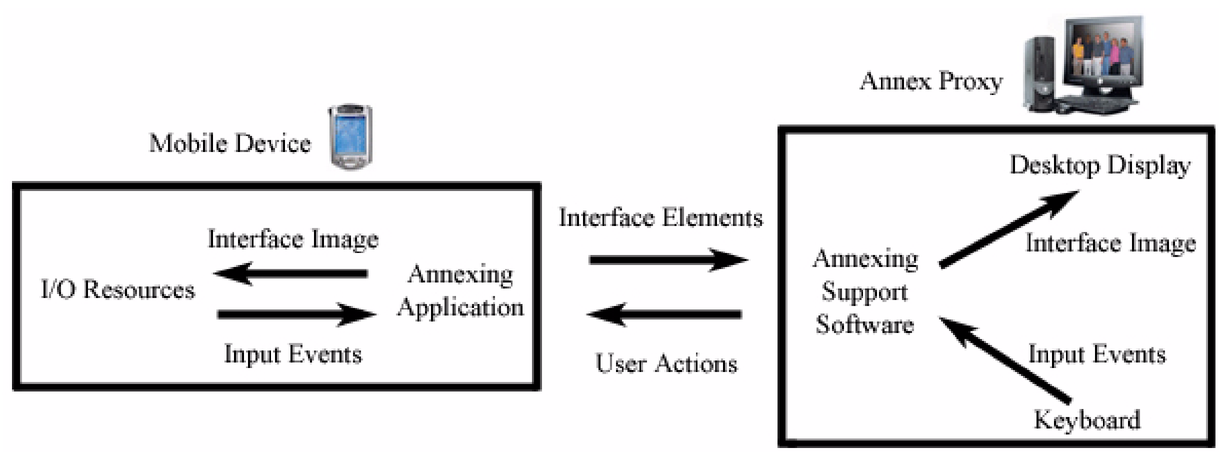

Figure 2: The Annexing Software Architecture

Sending the image to display and receiving a stream of input events minimizes the requirements for annex proxies, allowing mobile devices to annex almost any resource. The biggest drawback of this approach is that it places a heavy burden on the mobile device. A mobile device must be aware of the capabilities of annexed displays, it must be able to render and transmit images of the interface at interactive rates, and it must be able to interpret the incoming stream of raw input events. The processing, storage, bandwidth, and power demands of complex interfaces will quickly exceed the capabilities of today's PDAs and cell phones.

Sending a semantic description of the desired interface reduces the mobile device's burden. Rather than sending images of the interface, the mobile device sends a description of it and relies on the annex proxy to use that description to recreate the desired interface. Returning a higher-level description of the user's actions, instead of raw input events, reduces the communication load even further and avoids the need for the mobile device to understand a wide variety of raw input events. The key to this approach is to determine the appropriate semantic level for describing the interface. A higher-level description reduces the amount of information to communicate, but it places more of a burden on the annex proxy. Specifically, the annex proxy must understand how to create a large number of complex interface elements, instead of a small number of simple elements. A possible solution is to provide both an optimistic, high-level description of the interface and a more conservative description. For example, a mobile device could describe a desired schedule interface as both a calendar and a collection of text boxes.

Mobile Device Transmits	Mobile Device Receives	Example Systems
Image to display	Raw input events (e.g. mouse up, key press)	Microsoft Remote Control [Microsoft], VNC [VNC]
Semantic description (e.g. HTML)	Raw input events	X Windows [X]
Semantic description	Higher-level user actions (e.g. send email message)	Cooltown [Kindberg01], Personal Server [Want], PUC [Nichols], XWeb [Olsen00]
Mobile code	Higher-level user actions	Jini [Waldo], SpeakEasy [Newman02]

Table 1: Previous Architectures for Communicating Interfaces and Actions

A final approach is to employ mobile code to display the interface and return the user's actions. This approach shifts most of the burden from the mobile device to the annex proxy. Every annex proxy must be able to execute the code, and the use of mobile code introduces the risk of malicious code.

6 SECURITY

We will need to provide mechanisms to protect users' devices and data. While current security protocols should be adequate for securing connections between devices, we will need to explore mechanisms to authenticate users, establish trust between users and annexed devices, and guard against unauthorized accesses.

6.1 Authentication and Trust Management

Users who want to annex a device must prove that they have the right to do so. In addition, users must establish a level of trust with the device. This trust goes both ways. Annexable devices need to be able trust users not to use them inappropriately, and users need to be able to trust annexed devices not to capture their data or access their handheld inappropriately. Together these requirements highlight the need for an authentication and trust management process [Blaze].

While passwords, a form of credential, are the most common authentication mechanism, they are one-sided: they verify that the user has the right to access the device, but they say nothing about whether the user can trust the device with his information. A better mechanism might be to use signed certificates as credentials. By issuing certificates to both users (or more accurately their handhelds) and devices, we can allow devices to authenticate users and users to authenticate devices. For example, a university student might have a certificate from the university verifying that she should have student-level access to devices, while a monitor in one of the university's public labs might have a certificate verifying that the monitor is owned by the College. If the user asks to annex the monitor to display some information, the monitor can verify that the user, as a university student, has the right to do so, while the user can verify that the monitor is actually owned by the university.

Streamlining the authentication process, while keeping it reliable, will be a primary concern. While devices can verify credentials themselves, a more common approach will probably be to rely on a separate authentication server, which can introduce delay. Creating dedicated keys for encrypting and decrypting communications with each other, once devices have verified credentials, might help streamline this process. The user's handheld can encrypt future annexing requests to that device with its key; successfully decryption using the corresponding key will authenticate the user.

When annexing via pulling, users will have to perform an additional step: proving to their handheld that they are the ones initiating the annexing request. Unfortunately, certificates will not work and passwords are subject to replay attacks: a borrowed keyboard could store the keystrokes, capturing the user's password and allowing an adversary to reuse it. Researchers have been working on new authentication methods (e.g. [Pering]), but more work is necessary.

The established level of trust will dictate how the user can employ the annexed device. However, users do not have to adhere to the maximum level of trust; they can make their own determination of how much to trust a particular device. For example, a user could have the same permissions when annexing his personal monitors, his organization's monitors, and the monitors in a local coffee shop, but he might trust those devices differently. He might be willing to send his actual data to monitors he owns, while for monitors owned by his organization he might prefer to render a view of his data on his handheld and send the view. For the monitors owned by the coffee shop he might prefer to avoid sending sensitive information at all. When viewing his schedule, for example, the user could display the blocks of time where he is busy on a coffee shop monitor while displaying the event details on his handheld.

We will need to provide some mechanism for users to specify a different level of trust than the one they are granted. We might be able to algorithmically suggest an initial level of trust given information about the sensitivity of the information and the relationship between the user and the device owner. However, this algorithm would probably need to err on the side of caution and provide a simple mechanism for users to override it.

6.2 Protecting Devices

Protecting the data on handheld devices from unauthorized access is a primary concern. Consider a user who annexes the I/O peripherals of a nearby desktop computer, using it as an intermediary, to look at his schedule for the day. The intermediary computer might be able to impersonate the user by pretending that the user has requested his weekly and monthly schedule as well, giving it access to that information.

One possible solution is to make users approve all data requests. While this approach allows users to prevent unauthorized access, approving every request may quickly become too cumbersome. A similar approach would be to allow users to specify the data that annexed devices can access during a particular session, but this approach might also be too cumbersome. Another approach would be to structure user interfaces so that users perform any actions that initiate a data transfer from the handheld, but this approach might prevent the use of more effective interface designs.

A better solution might be to allow users to monitor communications with annexed devices using their handhelds. Monitoring could allow users to detect when unauthorized accesses occur and take the appropriate action, provided we can effectively communicate how the actions (data transferred, applications invoked, etc.) taken by an annexed device differ from the expected and allowed actions.

We will need to protect annexed devices as well. Owners of devices will doubtlessly be reluctant to allow annexing if there is risk involved. Software architectures that transfer and run code on annexed devices will probably be the most risky; owners might choose to prevent all but the most trusted users from employing mobile code.

7 Conclusions

We believe that opportunistic annexing can address the limited I/O capabilities of handheld devices. Although the idea is promising, much work remains. We are currently engaged in the first steps: creating new annexing techniques, developing a software architecture, and experimenting with different points in the application design space. We hope that other members of the research community will join us in this effort and in addressing the other challenges that we laid forth in this paper.

8 References

[AOL] AOL Mobile's T9. For more information visit http://www.t9.com.

[Baudisch] Baudisch, P., and Rosenholtz, R. (2003) Halo: A Technique for Visualizing Off-Screen Locations. Proceedings of CHI 2003, 481-488.

[Blaze] Blaze, M., Feigenbaum, J., Ioannidis, J., Keromytis, D.A. (1999) The Role of Trust Management in Distributed Systems Security. In Secure Internet Programming: Security Issues for Mobile and Distributed Objects, Jan Vitek and Christian Jensen editors. Springer-Verlag, New York.

[Bederson94] Bederson, B. B., and Hollan, J. D. (1994). Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. Proceedings of UIST 1994, 17-26.

[Bederson03] Bederson, B., Clamage, A., Czerwinski, M., and Robertson, G. (2003) A Fisheye Calendar Interface for PDAs: Providing Overviews for Small Displays. CHI 2003 Video Program.

[Bjork99] Bjork, S., et al. (1999) WEST: A Web Browser for Small Terminals. Proceedings of UIST 1999, 187-196.

[Bjork00] Bjork, S., Redstrom, J., Ljungstrand, P., and Holmquist, L. E. (2000) POWERVIEW: Using information links and information views to navigate and visualize information on small displays. Proceedings of Handheld and Ubiquitous Computing 2000, 46-62.

[Brummitt] Brummitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. (2000) EasyLiving: Technologies for Intelligent Environments. Proceedings of Handheld and Ubiquitous Computing 2000, 12-29.

[Burge] Burge III, L. L., Baajun, S., and Garuba, M. (2001) A Ubiquitous Stable Storage for Mobile Computing Devices. 2001 Symposium on Applied Computing, 401-404.

[Buyukkokten] Buyukkokten, O., Garcia-Molina, H., Paepcke, A., and Winograd, T. (2000) Power Browser: Efficient Web Browsing for PDAs. Proceedings of CHI 2000, 430-437.

[Card] Card, S. K., Moran, T. P., and Newell, A. (1983). The Psychology of Human-Computer Interaction. Lawrence

Erlbaum Associates, Hillsdale, New Jersey.
[Consolvo] Consolvo, S., Arnstein, L., and Franza, B. R. (2002) User Study Techniques in the Design and Evaluation of a Ubicomp Environment. Proceedings of Ubicomp 2002, 73-90.
[Corner] Corner, M. and Noble, B. Zero-interaction Authentication. Proceedings of MOBICOM 2002, 1-11.
[Corson] Corson, M. S., Macker, J., and Batsell, S. (1996) Architectural Considerations for Mobile Mesh Networking. Proceedings of IEEE MILCOM 1996.
[Czerwinski] Czerwinski, M., Smith, G., Regan, T., Meyers, B., Robertson, G. and Starkweather, G. (2003) Toward characterizing the productivity benefits of very large displays. To appear at INTERACT 2003.
[Dillon] Dillon, A., Richardson, J., McKnight, C. (1990) The Effects of Display Size and Text Splitting on Reading Lengthy Text from Screen. Behaviour and Information Technology, 9, 215-227.
[Duchnicky] Duchnicky, R.L., Kolers, P.A. (1983) Readability of Text Scrolled on Visual Display Terminals as a Function of Window Size. Human Factors, 25, 683-692.
[Fleetwood] Fleetwood, M. D., Byrne, M. D., Centgraf, P., Dudziak, K. Q., Lin, B., and Mogilev, D. (2002). An Evaluation of Text-Entry in Palm OS - Graffiti and the Virtual Keyboard. Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting, 597-601.
[Flinn99] Flinn, J. and Satyanarayanan, M. (1999) Energy-Aware Adaptation for Mobile Applications. Proceedings of the Symposium on Operating Systems Principles (SOSP), 48-63.
[Flinn02] Flinn, J., Park, S., and Satyanarayanan, M. (2002) Balancing Performance, Energy, and Quality in Pervasive Computing. 22nd International Conference on Distributed Computing Systems, 217-226.
[Fox] Fox, A., Johansen, B., Hanrahan, P., and Winograd, T. (2000) Integrating Information Appliances into an Interactive Workspace. IEEE Computer Graphics and Applications, 20(3), 54-65.
[Gonzalez] Gonzalez-Castano, F. J., Vales-Alonso, J., Livny, M., Costa-Montenegro, E., and Anido-Rifon, L. (2002) Condor Grid Computing from Mobile Handheld Devices. Mobile Computing and Communications Review, 6(2), 18-27.
[Greenberg] Greenberg, S., Boyle, M., and LaBerge, J. (1999) PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal. Personal Technologies, 3(1), 54-64.
[Griswold] Griswold, W. G., Shanahan, P., Brown, S. W., Boyer, R., Ratto, M., Shapiro, R. B., and Truong, T. M. (2003) ActiveCampus - Experiments in Community-Oriented Ubiquitous Computing. Technical Report CS2003-0750, Computer Science and Engineering, UC San Diego, June 2003.
[Hinckley00] Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. (2000) Sensing Techniques for Mobile Interaction. Proceedings of UIST 2000, 91-100. Best Paper Award.
[Hinckley01] Hinckley, K. and Horvitz, E. (2001) Toward More Sensitive Mobile Phones. Proceedings of UIST 2001, 191-192.
[Hinckley03] Hinckley, K. (2003) Synchronous Gestures for Multiple Users and Computers. To appear at UIST 2003.
[Holmquist] Holmquist, L. E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellerson, H. W. (2001) Smart-Its Friends: A Technique for Users to Easily Establish Connections Between Smart Artefacts. Proceedings of Ubicomp 2001, 116-122.
[Johanson01] Johanson, B., Pennekanti, S., Sengupta, C., and Fox, A. (2001) Multibrowsing: Moving Web Content across Multiple Displays. Proceedings of Ubicomp 2001, 346-353.
[Johanson00] Johanson, B., Hutchins, G., Winograd, T., and Stone, M. (2000) PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. Proceedings of UIST 2002.
[Johanson02] Johanson, B., Fox, A., and Winograd, T. (2002) The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. IEEE Pervasive Computing 1(2), 71-78.
[Jones] Jones, M., Marsden, G., Nasir-Mohd, N., Boone, K. (1999) Improving Web Interaction on Small Displays. Computer Networks, 31, 1129-1137.
[Kamba] Kamba, T., Elson, S. A, Harpold, T., Stamper, T., and Sukaviriya, P. Using Small Screen Space More Efficiently. Proceedings of CHI 1996, 383-390.
[Kim] Kim, L., and Albers, M. J. (2001) Web Design Issues When Searching for Information in a Small Screen Display. Proceedings of SIGDOC 2001, 193-200.
[Kindberg00] Kindberg, T. et al. (2000) People, Places, Things: Web Presence for the Real World. Proceedings of WMCSA 2000.
[Kindberg01] Kindberg, T. and Barton, J. (2001) A Web-Based Nomadic Computing System. Computer Networks, 35(4), 443-456.
[Korva] Korva, J., Plomp, J., Maatta, P., and Metso, M. (2001) On-line Service Adaptation for Mobile and Fixed Terminal Devices. Proceedings of the Second International Conference on Mobile Data Management, 252-259.
[MacKenzie99a]MacKenzie, I. S., and Zhang, S. Z. (1999) The Design and Evaluation of a High Performance Soft Keyboard. Proceedings of CHI 1999, 25-31.
[MacKenzie99b]MacKenzie, I. S. and Chang, L. (1999). A Performance Comparison of Two Handwriting Recognizers. Interacting with Computers, 11, 283-297.
[Mackenzie02]MacKenzie, I.S., Soukoreff, R.W. (2002) Text Entry for Mobile Computing: Models and Methods, Theory and Practice. Human Computer Interaction, 17, 147-198.
[Mankoff] Mankoff, J. and Abowd, G. D. (1998) Cirrin: A Word-Level Unistroke Keyboard for Pen Input. Proceedings of UIST 1998, 213-214.
[Metso] Metso, M., Koivisto, A., and Sauvola, J. (2001) Content Model for Mobile Adaptation of Multimedia Information. Journal of VLSI Signal Processing 2, 115-128.
[Microsoft] Microsoft Corporation's Remote Display Control. Available as a Power Toy for Pocket PCs from www.microsoft.com/technet/prodtechnol/ wce/downloads/PPCToys.asp.
[Myers98] Myers, B. A., Stiel, H., and Gargiulo, R. (1998) Collaboration Using Multiple PDAs Connected to a PC. Proceedings of CSCW 1998, 285-294.
[Myers00] Myers, B. A., Miller, R. C., Bostwick, B., and Evankovich, C. (2000) Extending the Windows Desktop Interface With Connected Handheld Computers. 4th USENIX Windows Systems Symposium, 79-88.
[Myers01] Myers, B. A. (2001) Using Handhelds and PCs Together. Communications of the ACM, 44 (11), 34-41.
[Newman02] Newman, M.W., Izadi, S., Edwards, K., Sedivy, J.Z., Smith, T.F. (2002) User Interfaces When and Where They are Needed: An infrastructure for Recombinant Computing. Proceedings of UIST 2002, 171-180.
[Nichols] Nichols, J., Myers, B. A., Higgins, M., Hughes, J., Harris, T. K., Rosenfeld, R., and Pignol, M. (2002) Generating Remote Control Interfaces for Complex Appliances. Proceedings of UIST 2002, 161-170.

[Noble] Noble, B. D., Satyanarayanan, M., Narayanan, D., Tilton, J. E., Flinn, J., and Walker, K. R. (1997) Agile Application-Aware Adaptation for Mobility. Proceedings of SOSP, 276-287.
[Olsen00] Olsen, D. R., Jefferies, S., Nielsen, T., Moyes, W., Fredrickson, P. (2000) Cross-modal Interaction using XWeb. Proceedings of UIST 2000, 191-200.
[Olsen01] Olsen, D. R, Nielsen, T., and Parslow, D. (2001) Join and Capture: a Model for Nomadic Interaction. Proceedings of UIST 2001, 131-140.
[Perin] Pering, T., Sundar, M., Light, J., and Want, R. (2003) Photographic Authentication through Untrusted Terminals. IEEE Pervasive Computing, 2(1), 30-36.
[Perlin93] Perlin, K. and Fox, D. (1993) Pad: An Alternative Approach to the Computer Interface. Proceedings of SIGGRAPH 1993, 57-64.
[Perlin98] Perlin, K. (1998) Quikwriting: Continuous Stylus-based Text Entry. Proceedings of UIST 1998, 215-216.
[Pham00] Pham, T-L., Schneider, G., and Goose, S. (2000) Exploiting Location-Based Composite Devices to Support and Facilitate Situated Ubiquitous Computing. Proceedings of Handheld and Ubiquitous Computing 2000, 143-156
[Pham01] Pham, T-L., Schneider, G., Goose, S., and Pizano, A. (2001) Composite Device Computing Environment: A Framework for Situated Interaction Using Small Screen Devices. Personal and Ubiquitous Computing, 5(1), 25-28.
[Pirolli] Pirolli, P. and Card, S. (1999) Information Foraging. Psychological Review, 106(4), 643-675.
[Ponnekanti] Ponnekanti, S.R., Lee, B., Fox, A., Hanarahan, P., Winograd, T. (2001) ICrafter: A service framework for ubiquitous computing environments. Proceedings of Ubicomp 2001, 56-75.
[Rekimoto97] Rekimoto, J. (1997) Pick and Drop: A Direct Manipulation Interface for Multiple Computer Environments. Proceedings of UIST 1997, 31-39.
[Rekimoto98] Rekimoto, J. (1998) A Multiple Device Approach for Supporting Whiteboard-Based Interactions. Proceedings of CHI 1998, 344-351.
[Rekimoto99] Rekimoto, J. and Saitoh, Masanori. (1999) Augmented Surfaces: A spatially continuous workspace for hybrid computing environments. Proceedings of CHI 1999, 378-385.
[Silfverberg] Silfverberg, M., MacKenzie, I. S., & Korhonen, P. (2000). Predicting Text Entry Speeds on Mobile Phones. Proceedings of CHI 2000, 9-16.
[Streitz] Streitz, N. A., et al. (1999) i-LAND: An interactive Landscape for Creativity and Innovation. Proceedings of CHI 1999, 120-127.
[Tan03a] Tan, D. S., Gergle, D., Scupelli, P. G., and Pausch, R. (2003) With Similar Visual Angles, Larger Displays Improve Spatial Performance. Proceedings of CHI 2003, 217-232.
[Tan03b] Tan, D. S., Czerwinski, M., and Robertson, G.G. (2003). Women Go With the (Optical) Flow. Proceedings of CHI 2003.
[Tan03c] Tan, D. S. and Czerwinski, M. (2003) Information Voyeurism: Social Impact of Physically Large Displays on Information Privacy. CHI 2003 Extended Abstracts.
[Tandler] Tandler, P., Prante, T., Muller-Tomfelde, C., Streitz, N., and Steinmetz, R. (2001) ConnecTables: Dynamic Coupling of Displays for the Flexible Creation of Shared Workspaces. Proceedings of UIST 2001, 11-20.
[VNC] VNC, or Virtual Network Computing. More information is available at http://www.uk.research.att.com/vnc/.
[Waldo] Waldo, J. (1999) The Jini Architecture for Network-Centric Computing. Communications of the ACM, 42 (7).
[Want] Want, R., Pering, T., Danneels, G., Kuman, M., Sundar, M., Light, J. (2002) The Personal Server: Changing the way we thing about ubiquitous computing. Proceedings of UbiComp 2002, 194-209.
[Wobbrock] Wobbrock, J. O., Forlizzi, J., Hudson, S. E., and Myers, B. A. (2002) WebThumb: Interaction Techniques for Small-Screen Browsers. Proceedings of UIST 2002, 205-208.
[X] X Window System. More information is available at http://www.x.org.
[Yee] Yee, K-P. Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers. Proceedings of CHI 2003, 1-8.
[Zhai00] Zhai, S., Hunter, M., and Smith, B. A. The Metropolis Keyboard - An Exploration of Quantitative Techniques for Virtual Keyboard Design. Proceedings of UIST 2000, 119-128.
[Zhai03] Zhai, S., and Kristensson, P-O. (2003) Shorthand Writing on Stylus Keyboard. Proceedings of CHI 2003, 97-104.