18

Designing a Multi-Slate Reading Environment to Support Active Reading Activities

NICHOLAS CHEN, University of Maryland
FRANCOIS GUIMBRETIERE, Cornell University
ABIGAIL SELLEN, Microsoft Research, Cambridge, MA

Despite predictions of the paperless office, most knowledge workers and students still rely heavily on paper in most of their document practices. Research has shown that paper's dominance can be attributed to the fact that it supports a broad range of these users' diverse reading requirements. Our analysis of the literature suggests that a new class of reading device consisting of an interconnected environment of thin and lightweight electronic slates could potentially unify the distinct advantages of e-books, PCs, and tabletop computers to offer an electronic reading solution providing functionality comparable to, or even exceeding, that of paper. This article presents the design and construction of such a system. In it, we explain how data can be mapped to slates, detail interactions for linking the slates, and describe tools that leverage the connectivity between slates. A preliminary study of the system indicates that such a system has the potential of being an electronic alternative to paper.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Interaction styles*; H.5.m [**Information Interfaces and Presentation**]: Miscellaneous

General Terms: Design, Human Factors

Additional Key Words and Phrases: Active reading, electronic books, horizontal display, tablet computer, distributed user interface

ACM Reference Format:
Chen, N., Guimbretiere, F., and Sellen, A. 2012. Designing a multi-slate reading environment to support active reading activites. ACM Trans. Comput.-Hum. Interact. 19, 3, Article 18 (October 2012), 35 pages.
DOI = 10.1145/2362364.2362366 http://doi.acm.org/10.1145/2362364.2362366

1. INTRODUCTION

The reading activities that knowledge workers and students engage in tend to be more complex than those that characterize reading for leisure. Typically, such activities involve a diverse mix of linear reading, skimming, annotating, interleaving reading and writing, and switching between documents that are used simultaneously.[1] Though these activities are well supported by paper, despite considerable advances in

[1]While there is no single agreed upon term to describe these types of reading activities, they have been described as being components of "work-related reading" [Adler et al. 1998], "active reading" [Adler and van Doren 1972], or "responsive reading" [Pugh 1978]. For brevity's sake, however, we will refer to these activities simply as *active reading* in this article.

This work was supported by NSF Grants IIS-0812196, IIS-0936105, and a gift from Microsoft Research. N. Chen is funded by a Google Ph.D. Fellowship.
Authors' addresses: N. Chen and F. Guimbretiere, 301 College Avenue, Information Science, Ithaca, NY 14850; email: nchen@cs.umd.edu, francois@cs.cornell.edu; A. Sellen, 7 J.J. Thomson Avenue, Cambridge CB3 0FB, U.K.; email: asellen@microsoft.com.

DOI 10.1145/2362364.2362366 http://doi.acm.org/10.1145/2362364.2362366

interaction design and computing devices, it is apparent that current electronic reading solutions still fail to provide the wide range of functionality this style of reading demands. For this reason, it is perhaps not surprising that there has been low uptake of electronic reading devices, such as the Amazon Kindle in the professional sector, and that deployments in academic environments have been met with mixed reactions at best [Behler 2009; Young 2009; Thayer et al. 2011].

Researchers who have studied paper use explain that one critical way paper supports active reading is by allowing the reader to work with a large amount of information simultaneously. Tasks like glancing back to re-read, comparing documents, and interleaving reading and writing all benefit from the fact that content is distributed across several sheets of paper. Since sheets of paper are thin and lightweight, they can be picked up, laid out, and rearranged effortlessly. These operations provide organizational and cognitive benefits that aid reading tasks involving several documents [O'Hara et al. 2002]. It follows then that if electronic devices are to more fully support the reading activities of knowledge workers and students, they must provide similar capabilities that will enable readers to spread out, navigate through, and work with multiple documents or pages at the same time.

One strategy is to simply increase the size of the display to enable readers to see more than one document at a time. This is the approach that large tabletop computers like Microsoft's Surface [2010] have adopted. Even if we assume that tabletop systems will eventually become somewhat portable, a major drawback is that increasing the size of the display obviously undermines the mobility of the system. In the context of reading, reducing mobility is undesirable, as it limits when and where reading can be done [Tashman and Edwards 2011a] and also removes an important element of physicality from the reading process [Marshall 2005].

In this article, we explore another approach. Our reading system, *United Slates,* is based on a collection of moderately sized but highly portable slate devices. We draw inspiration from how paper documents derive important functionality by distributing content across several different sheets that are individually mobile. However, our system also takes advantage of the power of the digital world by electronically connecting the slates. Although each slate is individually useful—featuring e-paper displays and the ability to capture strokes from a stylus—the integration of the slates enables unique capabilities unmatched in other reading systems. We present the design and implementation of this new electronic reading system and highlight how it has the potential to more completely support the reading needs of knowledge workers and students than existing electronic solutions.

We begin by surveying the literature detailing the reading requirements of students and knowledge workers. This literature review helped inform the goals of the system and was used to specify its hardware and software requirements. The analysis also highlighted the shortcomings of current solutions in supporting key features of reading, further bolstering the case for taking a different approach. In particular, our analysis showed that commercially available hardware was too heavy and lacked support for writing. For this reason, we collaborated with an industrial partner to develop custom lightweight e-paper-based reading hardware that allowed for pen input.

To test the initial acceptability of the slate form factor and some of the key interaction ideas, we first ran a trial in which a probe was deployed to students in a university course. We found the physical characteristics of the slates, the writing support, and additional display to be beneficial. There was, however, room for improvement in the areas of navigating between different documents, integrating with existing computing devices, and supporting varying numbers of devices.

On the basis of this trial, we then went on to design the interactions that underpin the multi-slate reading environment in more detail. Our interface primarily focuses

Fig. 1. Various usage configurations of the United Slates reading system. The system can be used to support workflows that include the PC, can be used for working with multiple documents, and can adapted for use in diverse environments.

on supporting navigation activities, resolving physically awkward slate interactions, and finding more integrated ways of dealing with documents within the context of the overall active reading workflow. For instance, our system allows for rapid switching between pages across documents, quick side-by-side comparison of documents, remote invocation of operations across devices, reading in a variety of environments, and federation with other electronic devices, such as PCs. Although some of these qualities come from simply having more displays available, features like multi-slate navigation show how new functionality can be gained from the electronic connectivity between devices. Through these interactions, our system lowers the barriers of working in a multi-device environment and takes advantage of such an environment for conveying new capabilities (see Figure 1).

To get a sense of whether our system achieves its goals of supporting reading activities, we conducted an evaluation of our system by having participants compare using a set of slates that incorporated the multi-slate reading tools to isolated devices that did not have that functionality. Based on our results, we believe that our system's ability to accelerate non-sequential navigation across many documents, to assist in working with large collections of documents, and to fluidly integrate with PCs are important features that should be incorporated into reading devices for knowledge workers and students. Our positive results also suggest that a configuration consisting of multiple linked slates plus a PC successfully combines the functionality of PCs, tablet PCs, e-readers, and tabletop computers, each of which would only support a subset of desired reading functionality if used independently. The broader implication here is that multi-device ensembles offer an effective way to amalgamate the functionality of different devices, perhaps more so than creating a new monolithic device. We believe this final insight has general applicability beyond reading technologies.

2. RELATED WORK

2.1. Reading Requirements for Knowledge Workers

Reading is a multi-faceted and complex activity, and as a result, a great deal has been written about its processes and requirements. The literature makes it very clear that there are many different types of reading. Pugh's [1978] study of how university students read identified five distinct styles of reading in which students engage: receptive reading, the process of linearly progressing through the text without interruption; reading to search for a specific piece of information; acquisition of information without a set goal; reading to get an overview about the general structure of the material; and responsive reading, which is the process of actively engaging with the material in the form of note taking, annotation, and cross-referencing. This analysis is mirrored in Adler et al. [1998] who focuses on the work-related purpose of reading and highlights the extent to which work-related reading tasks involve writing and annotation as well as cross-referencing. The reading activities that involve writing, annotation and cross-referencing are frequently referred to as "active reading" [Adler and van Doren 1972]. It is, however, important to note that active reading tasks do not occur in isolation [Thayer et al. 2011]. Rather, these tasks are punctuated with receptive reading and skimming activities. Thus, one cannot support active reading without also supporting these other types of reading.

Another important finding of ethnographic studies is the relationship between reading and writing and the number of documents that are used concurrently. Adler et al. [1998] is informative in this respect, reporting that reading happens more frequently with writing than without, and that reading is largely performed across several documents or displays (e.g., monitor, sheet of paper) concurrently. O'Hara et al. [2002] studyed a wide variety of professionals and students performing writing tasks, also describing the concurrent use of multiple displays. In particular, this study points to the savings gained by being able to quickly glance across displays or pages. O'Hara et al. also found that the spatial layout of these materials in the workspace served an important role in supporting readers' mental organization of the material. Spatial layout of documents has additional significance when we consider reading activities that occur across multiple sessions. Malone [1983] explains that layout structures like files and piles speed up the process of finding and accessing documents and also remind users about documents they need to attend to.

Marshall's work documenting annotation practices in a university setting [Marshall 1997] is also relevant. In particular, Marshall regards the ability to smoothly integrate annotation with reading to be one of the most essential challenges for any reading system. These findings about annotation, spatial layout of information, and interleaving reading and writing are further corroborated by O'Hara and Sellen's [1997] laboratory study comparing reading on paper to reading on a PC. Taken as a whole, these studies strongly motivate system support for seamless annotation and writing capabilities and the ability to work across multiple documents simultaneously.

More recently, Tashman and Edwards [2011a] conducted a diary study and design workshop to determine users' opinions as to how their existing active reading practices could be improved. The major findings revealed a tension in which users wished to have more space to view content for things like multi-document reading but also wanted their reading workspaces to be portable.

2.2. Experiences with Electronic Reading Devices

In addition to studies about how people read from paper, researchers have observed how people use electronic reading technology. From these reports, we can glean more insights about how to support reading by examining what aspects of these systems have been successful and where these systems fall short.

The XLibris reading appliance [Schilit et al. 1998] was an experimental system implemented in a tablet form factor, designed to support active reading. XLibris displayed a single page of a document full screen and allowed users to apply free-form ink annotations on the page. XLibris also featured reading tools like highlighting of key terms to aid skimming, as well as support for hyperlinks and backtracking. As a prototype, it was evaluated with various groups of people, including researchers in a reading group and legal scholars [Marshall et al. 1999; Marshall et al. 2001]. Users in these studies found that XLibris' document presentation, annotation, information extraction, and nonlinear and cross-document navigation features facilitated reading activities. On the other hand, the legal scholars who evaluated the prototype expressed a need for the device to better integrate with the work on PCs, leading researchers to wonder whether a laptop with annotation capabilities would be more suitable.

Marshall and Ruotolo [2002] reported on the deployment of PDAs in various university courses found that the high portability and availability of the devices and text search capabilities were advantageous. However, the small screens on the devices meant that layout-sensitive texts were difficult to use, and random-access navigation on the devices was difficult, particularly in long documents.

Wilson and Landoni's [2003] survey of several different electronic reading devices similarly found that basic text search capabilities on the devices were extremely helpful. However, they also reported that as a whole, electronic devices needed to better support working with multiple documents and provide awareness about one's place within a large document. Pearson et al. [2010] compared three commercial e-book reading devices and noted that ergonomics, navigation, and annotations all played a role in a good e-reader design. Recent deployments of commercial e-books (generally those employing electronic paper displays) into the classroom [Young 2009; Behler 2009] have found serious deficiencies in the navigation systems in these devices for classroom reading. These navigation issues often stemmed from the slow refresh rates on the displays. Another problem these studies identified was that illustrations—color ones in particular—were not reproduced with sufficient fidelity.

Finally, the Kindle DX, a commercial reader, has been tested recently in several universities. Thayer et al. [2011] produced an extensive report of Kindle DX use among Computer Science students. As with previous experiences with e-readers in the classroom, Thayer et al. found that the device they tested lacked support for the diverse navigation needs of student readers. In particular, they noted that the lack of tangibility in the pages of an e-reader removed important physical indicators, or *kinesthetic cues*, that helped in giving users a sense of where they were in the text and how much material remained. Thayer et al. also reported that the absence of free-form ink annotation capabilities severely limited students' abilities to read responsively. A report detailing the deployment of Kindle DX devices at Princeton University in International Affairs and Classics courses mirrored many of the sentiments regarding the difficulty of navigating and the lack of annotation facilities [The Trustees of Princeton University 2010].

One other relevant study of reading devices occurred outside the classroom or workplace. Marshall and Bly's [2005] comparison of magazine reading on paper and on an electronic device found that, compared to paper, electronic versions of magazines lacked support for interactions like glancing back to re-read, looking ahead, and adjusting the reading medium to narrow or broaden focus. This research lends further credence to the idea that reading involves more than visiting pages in a document in a linear sequence.

Table I presents a summary of this literature and, in doing so, also provides guidance as to the range of features identified as important in the support of active reading.

Table I. A summary of the Reading Requirements for Knowledge Work as Detailed in Previous Research

	In the Literature
Hardware/Physical Requirements	
Mobility: thin, lightweight, graspable	Marshall and Ruotolo 2002; Marshall 2005; Wilson and Landoni 2003
Readability	Wilson and Landoni 2003; Young 2009
Support for writing	O'Hara and Sellen 1997; Adler et al. 1998; O'Hara et al. 2002; Sellen and Harper 2003; Thayer et al. 2011
Page-Level Requirements	
Linear reading	Adler and van Doren 1972; Pugh 1978
Graphics: page layout/illustration	O'Hara and Sellen 1997; Marshall and Ruotolo 2002; Young 2009; Behler 2009;
Superimposed ink annotations	Adler and van Doren 1972; Marshall 1997; Thayer 2011
Text search	Wilson and Landoni 2003; Young 2009
Glancing back to re-read	Marshall et al. 1999; Marshall et al. 2001
Physical/kinesthetic cues	O'Hara and Sellen 1997; Thayer et al. 2011
Document-Level Requirements	
Non-sequential navigation	O'Hara and Sellen 1997; Behler 2009; Tashman and Edwards 2011a
Cognitive map (awareness) of content	O'Hara and Sellen 1997; Wilson and Landoni 2003; Thayer et al. 2011
Skimming to get overview	Pugh 1978; Adler et al. 1997
Discovery of topical knowledge	Pugh 1978; Adler et al. 1997
Switch between navigation styles	Thayer et al. 2011
Workspace-Level Requirements	
Sorting and triage of documents	Adler et al. 1997; Marshall and Shipman 1997
Spatial layout	O'Hara and Sellen 1997; O'Hara et al. 2002
Extracting information	Adler et al. 1997; Morris et al. 2007
Reading from multiple documents	O'Hara and Sellen 1997; O'Hara et al. 2002; Wilson and Landoni 2003; Tashman and Edwards 2011a
Integrating with PC workflows	Marshall et al. 2001; O'Hara et al. 2002; Morris et al. 2007; Tashman and Edwards 2011a; Thayer et al. 2011
Multi-session Reading Requirements	
Reading in different venues	Sellen and Harper 2003; Thayer et al. 2011; Tashman ands Edward 2011a
Filing and archiving	Malone 1983; Sellen and Harper 2003
Restoring reading workspace, resuming reading activites	Malone 1983; Wilson and Landoni 2003; Tashman and Edwards 2011a;

These range from some core features of the devices to features important at the page, document, and workspace level. In addition, there are some more general requirements we distill having to do with the support of document tasks across space and time.

2.3. Distributing Material onto Tangible Surfaces

Our system adopts the strategy of distributing content onto several independent slate devices in light of the importance of supporting free-form annotation alongside multiple simultaneous document use. The technique of distributing content onto separate tangible surfaces has recurred in the literature. Early work like VideoMosaic [Mackay and Pagani 1994] and DigitalDesk [Wellner 1993] were not designed specifically for reading activities but demonstrated how spatial layout capabilities afforded by independent physical surfaces could be leveraged to provide a richer experience, while at the same time being augmented by computers. PaperWindows [Holman et al. 2005] explores interaction techniques with flexible paper-like displays. PaperWindows makes the assumption that electronic displays will eventually have the same physical properties of paper. In contrast, our system takes a more restrained view of the future and, as such, is designed to be sensitive to the limitations we expect slate-type devices to impose in the near future. The dual-display e-book [Chen et al. 2008] and Codex

[Hinckley et al. 2009] both provide two screens that can be separated into two independent displays. Codex, in particular, explores the interactions that are possible between the two screens. Neither the dual-display e-book nor Codex examines the logistical and interactional issues that arise when the number of devices that are available is increased or reduced. PapierCraft [Liao et al. 2008] enables the use of multiple surfaces by allowing users to move between documents on tablet PCs and on sheets of digital paper. Information can be shared between surfaces using an inking-based command system. Our system shares similar goals with PapierCraft but explores the setting in which all surfaces are electronic and interactive.

Most similar to our present work is that of Morris et al. [2007] which evaluates a system composed of three independent tablet PCs that were not functionally linked. This study was the first instance in which more than two independent tablets were used together to complete a reading task. Users indicated that the spatial layout and annotation capabilities of the tablets were helpful and valuable. However, users reported issues from their inability to move information between tablets. Morris et al.'s experience highlights a problem with the default multi-slate configuration, which is that separate slate devices are computationally isolated. The isolation is problematic because it limits the devices' capability to work together in a collaborative fashion. Our solution aims to address this by leveraging network connectivity between devices to implement helpful interactions.

A development that promises improved connectivity between reading devices is CloudBooks [Pearson and Buchanan 2011], which describes a cloud-based infrastructure for linking together multiple iPads. The infrastructure lets devices exchange messages through the cloud as well as provides synchronization functionality. While the system we present does not currently rely on the cloud for its functionality, it would certainly benefit from this type of cloud-based infrastructure to provide increased availability and scalability.

2.4. Interactions for Connecting Slates

While the literature does not provide specific examples of how to construct an interface to unify slate devices for a reading application, there are several techniques that we are able to adapt for the purposes of connecting our reading devices together. The most frequently encountered example of using multiple screens is that of multi-monitor computing. However, multi-slate systems are slightly different from traditional multi-monitor systems in that the relative positions of displays in a multi-monitor setup tend to be static and can share a unified coordinate space. In contrast, in our approach, slates are treated as separate devices and separate spaces, and the focus is instead on bridging these spaces.

One important requirement for bridging these spaces is to simplify the movement of information across the spaces. A great deal of work has been done in streamlining data transfer between devices. The examples most relevant to our work are SyncTap [Rekimoto 2004], Pick-and-Drop [Rekimoto 1997], Synchronized clipboard [Miller and Myers 1999], and Stitching [Hinckley et al. 2004]. The one aspect all of these techniques have in common is that they offer users a fast and direct way of specifying the endpoints of a transfer operation.

Another strategy to help unify disparate devices is to allow one device to control another remotely. Remote control spans the spectrum of functionality provided by Virtual Network Computing [Richardson et al. 1998], which provides a full proxy to a different computer system to systems like Pebbles [Myers 2001] and PointRight [Johanson et al. 2002], where portions of the input and output stream are redirected to control other devices.

Table II. Comparison of Existing Reading Solutions' Support of Reading for Knowledge Work

KEY ○ low support ◒ moderate support ● good support	PC	TabletPC	Tabletop	Pad Appliance	E-book Appliance	Paper
Hardware/Physical Requirements						
Mobility: thin, lightweight, graspable	○	◒	○	●	●	(●)
Readability	◒	◒	○	◒	●	(●)
Support for writing	○	●	◒	○	○	(●)
Page-Level Requirements						
Receptive (linear) reading	●	●	●	●	●	(●)
Illustrations	●	●	●	●	○	(●)
Superimposed annotations	○	●	◒	○	○	(●)
Text search	●	●	●	●	●	○
Glancing back to re-read (see Notes)	◒	○	●	○	○	(●)
Kinesthetic cues	○	○	○	○	○	(●)
Document-Level Requirements						
Non-sequential navigation	●	●	●	●	○	(●)
Cognitive map of content	◒	◒	◒	○	○	(●)
Skimming to get overview	◒	◒	◒	◒	○	(●)
Discovery of topical knowledge	◒	◒	◒	◒	○	(●)
Switch between reading styles (see Notes)	◒	◒	○	○	○	(●)
Workspace-Level Requirements						
Sorting documents	◒	◒	●	○	○	(●)
Spatial layout	○	○	●	○	○	(●)
Extracting information	(●)	●	●	◒	○	○
Reading from multiple documents	◒	◒	●	○	○	(●)
Integrating with PC workflows	(●)	◒	◒	○	○	○
Multi-session Reading Requirements						
Reading in different venues	(●)	●	○	●	●	○
Filing and archiving	(●)	●	●	○	○	◒
Recreating reading workspace	◒	◒	◒	◒	◒	○

Notes: "Glancing back to re-read" is a function of how much content can be viewed at once; hence, better scores for devices with more screen area. "Switch between reading styles" is a function of the ability to reconfigure the interface for different reading styles and requirements. Parentheses denote what are generally regarded as the "target" for interaction experience.

Finally, with improvements in sensing technology, physical connection of devices as a means of specifying relationships between devices has also been explored. Examples include Siftables [Merrill et al. 2007] and ConnecTables [Tandler et al. 2001]. Although we believe that these interactions are valuable, slate devices have appreciable mass and activities that result in having to repeatedly connect or disconnect devices can be cumbersome. As such, we recognize that physical connection can be useful but may not always be appropriate.

2.5. Comparison of E-Reading Technologies

We conclude this section by presenting a summary of how various technologies currently support the reading activities we have found to be central to active reading. Table II shows an analysis of the major technological alternatives. When populating the table, we attempted to factor in the latest developments for each platform.

With its current prevalence in reading, paper is often considered the reference system when it comes to supporting active reading, and we indicate it as such in our table. However, there exist cases, such as working with large collections of documents, where paper suffers from limitations stemming from the cost of distributing, transporting, and archiving paper documents [Sellen and Harper 2003].

The traditional personal computer, which includes both desktop and laptops in our analysis, is a versatile tool well suited for composition and editing tasks that often go hand in hand with reading. PCs have robust multitasking capabilities along with comparatively large screens, making them better suited for cross-referencing, re-reading, and sorting tasks. The lack of pen input was addressed in part by the tablet PC system, but the PC's focus on being a very general, all-in-one tool results in software interfaces that are more complex and hardware that tends to be bulkier and more difficult to handle.

Although we are not aware of a reading system specifically developed for a tabletop configuration, multi-touch computing surfaces, like the Microsoft Surface, are probably good candidates to support reading for knowledge work: tabletops, by virtue of allowing direct manipulation of objects spread across a large surface, are suitable for spatial layout and working with several documents at once. Certain tabletop configurations are also able to receive high-resolution pen input. At the same time, even if we take into account future developments like flexible displays, tabletops will still constrain where and how reading is performed. For one, tabletops make it difficult to support the physicality of reading [Marshall 2005] because documents are confined to the surface. Additionally, these devices would necessarily require environments that provide a correspondingly large and flat working area to accommodate the single large screen.

Pad appliances such as the iPad depart from the standard laptop/tablet PC interface and subscribe to a full-screen application model. The majority of appliances these days forego a pen digitizer—relying instead on touch—and support writing only to a limited degree.[2] Nevertheless, these devices are proving to be good reading platforms for single documents. Moreover, the multi-touch capability found in almost all of these devices offers many possibilities for enhancing navigation, such as those demonstrated in LiquidText [Tashman and Edwards 2011b].

Dedicated electronic reading devices, which include products like the Amazon Kindle, offer highly readable screens, extremely low weight, and long battery lives, making them well suited for receptive reading. However, the slow refreshing black-and-white displays (from which many of their beneficial physical properties derive) limit the navigation capabilities as well as the content that these devices can reproduce. With the exception of devices from iRex (now defunct), these devices also offer little support for free-form writing and superimposed annotation.[3]

3. DESIGN FOR A MULTI-SLATE READING SYSTEM

Examining Table II, we propose that a system able to combine the physical properties of a dedicated e-book reader with the pen input of a tablet PC and the multi-document functionality of a tabletop system would offer the right combination of functionality to match the capabilities of paper. Combining these devices together into some kind of "Franken-device" would have been impossible given inherent conflicts in the nature of the devices. For instance, the size of the tabletop is at considerable odds with the lightweight graspable e-reader. Instead, we took a cue from how paper documents

[2]Aftermarket capacitive styli are available for these pad appliances. However, they simulate inking by simulating finger touch, which severely limits writing precision and can have problems when the user's hand inadvertently comes in contact with the screen.

[3]There exist devices in this category that include a stylus. For example, the Sony Reader Touch Editions (PRS-600, PRS-700). These devices generally employ capacitive or resistive touch screens with the same problems as the touch screens on pad appliances.

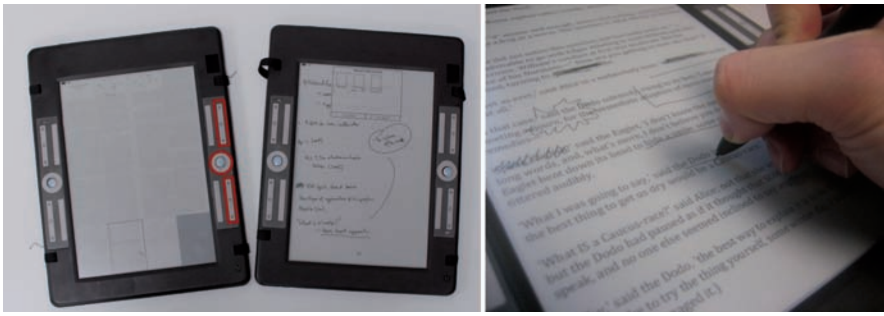

Fig. 2. Left: Reading device UIs for the technology probe. One bank of controls is highlighted in red. The device on the left shows a space-filling thumbnail (SFT) overview of the document. The device on the right displays a notebook page that has been populated with ink notes. Right: Free-form ink being applied to a page.

distribute contents across many sheets of paper that are individually thin and lightweight but powerful when used in aggregate. Using that strategy, we created a multi-slate solution in which each individual slate is useful for reading, but where each can be used as part of a larger ensemble.

3.1. Hardware Design Choices

Based on the literature review, we knew that at the most basic level, the slates we needed for our system would have to be extremely mobile and easy to handle. The devices also needed to have pen-based annotation capabilities. However, we also needed to be able to extend the devices with custom software to support other aspects of reading. Reading devices that matched these requirements were not available when we started the project in 2009 and still are not available. As a result, we were compelled to create a set of prototype reading devices with an industrial partner (Figure 2, left). Our devices have a 9.7" E Ink electronic paper display (EPD) placed over a Wacom pen digitizer. With a resolution of 150 dpi, the EPD offered readability nearly matching paper and considerably lower power consumption compared to standard LCD displays. Furthermore, the use of an EPD helped reduce the thickness and weight of the device by virtue of a thinner screen and smaller battery requirements (at 8.5mm thick and 500g, each slate compares favorably to devices like the Amazon Kindle DX and Apple iPad). One trade-off that use of an EPD entailed was that highly interactive visualizations of the document that are normally possible with LCD screens would not be possible due to the 100ms to 500ms (depending on the size and bit-depth of the image that needed to be drawn) required to refresh the display. Also, the lack of an active backlighting system made placing an overlay to sense touch input impractical, as it would reduce the contrast and display fidelity. Our discussion section details the implications and trade-offs of these hardware component choices.

The computing and communications capabilities come from a Gumstix Overo Air computer-on-module featuring a Texas Instruments OMAP3 application processor, plus Wi-Fi and Bluetooth radios. The choice of a smartphone-class processor was for reducing power consumption. The dual radios enable the devices to communicate amongst themselves and to connect to the Internet via Wi-Fi access points.

Our slates additionally provided a rich set of input and sensing electronics beyond the pen digitizer. Controls on the device include two large round buttons plus four additional tactile buttons that can also capacitively sense finger position and gestures on their surfaces (Figure 2). We placed controls on both sides of the device to support a variety of grips and reading positions. Inside the device, pairs of magnets and

Hall-effect sensors were embedded along the long edges to detect when two devices are placed against each other.

3.2. Technology Probe

With the exception of Morris et al.'s [2007] work, our knowledge of the interactional needs for multi-slate reading derives from what we know about people working with paper. Since slates and paper differ significantly in terms of physical characteristics like weight and thickness, not to mention their interactional capabilities, users most likely would not use slates in the same way they use paper. To better understand how slates would be used for active reading, we conducted a 2.5-month-long trial with a technology probe with 13 undergraduate English students in the Spring 2010 semester. In this trial, we first assessed students' reading activities with a single slate. Starting with a single slate gave us the opportunity to test for usability issues and bugs prior to deploying the more complex dual-slate configuration. Then, after 1.5 months, we provided a second slate and demonstrated additional features. During the course of the study, the students were asked to use one or both devices to carry out the majority of their course readings. We conducted biweekly interviews with students to gather feedback. When possible, we modified the software on the slates to reflect the feedback.

3.2.1. Software Configuration. Each slate displayed one full page of a document at a time, without margins, so that pages of letter-sized documents could be viewed without scrolling. We preloaded the devices with course readings after consulting with the instructor. Students did not have the capability of placing their own documents on the devices.

To turn pages, we relied on the hardware buttons on the slates. For navigation over longer distances, we provided a space-filling thumbnail (SFT) [Cockburn et al. 2006] overview of a document—a natural choice given the relatively slow refresh rate of the EPD. On each page, users could apply free-form ink on the pages with a stylus without entering any special inking modes (Figure 2, right). When more than one device was in use, ink marks made on one slate were transmitted electronically to the other slate so that the annotations on both slates remained consistent. The system also allowed students to create "notebook" documents, which were special documents with 50 blank pages. Users could ink on these pages, and they could also copy and paste regions of pages into the notebook. The clipped regions linked back to the location from which they were extracted so that the user could view the context of the clipping.

3.2.2. Lessons Learned During the Deployment. Students were particularly enthusiastic about the portability aspects of the device, opting to carry the device in lieu of the course readings on paper. Students also mentioned that the form factor and weight of the device were often preferable to a book because it was easier to operate with a single hand and did not flop around. The free-form ink annotation also received a large number of positive comments. Students reported writing more on the electronic devices because they felt they were not "defacing" the original document due to the reversibility of electronic ink marks. The difference reported was particularly pronounced for books, which many students said they wanted to keep pristine. Furthermore, students mentioned that the rigidity of the device actually made annotating easier than on paper. In fact, the simple presence of a pen was a plus in itself; like readers in O'Hara and Sellen's [1997] study, many students reported that reading with a pen was useful for helpful for maintaining one's place in the text. Based on this feedback, we were reasonably confident that the hardware design of the devices—the support for annotation, in particular—were in line with our goals.

Perhaps the biggest shortcoming of the system we deployed was the fact that the devices did not integrate well with the existing electronic devices students employed.

Students desired a way to put their own documents on the device and also to be able to move data off in support of their writing activities. Additionally, many of their regular daily activities revolved around the PC. Users often reiterated their desire to use the slates in conjunction with their computers. This result corroborates a large number of studies that emphasize the integral role PCs play in many reading tasks [Marshall et al. 2001; Morris et al. 2007; Tashman and Edwards 2011a].

In addition, while users reported that receptive reading was satisfactory on the devices, many users ended up working on the paper versions of documents when they needed to complete writing assignments and review for exams. This was particularly pronounced for large books. Students' experiences working with one large text that was several hundred pages long proved particularly illustrative of the challenges they faced. For this particular book, we split up each chapter of the book into a different document in order to simplify access to each chapter. So while SFT could be used to navigate within the chapter, navigating across chapters required a lengthier process of changing documents. Student comments indicated that navigating with SFT was bearable but that for their writing and review activities, they frequently needed to consult disparate information found in the other chapters, and it was simply easier to perform that type of navigation with the paper book. These comments highlight the fact that our system needed to not only support rapid non-sequential navigation within a single document, but also needed a strategy for providing equally quick access across different documents. The wide range in navigation practices we observed is similar to those that Thayer et al. [2011] found in their deployment of electronic reading devices.

Finally, even though we only provided users with two slates in the probe, we were able to observe interesting differences between how the students made use of the additional slate. The number of slates a user employed was heavily dependent on task and individual preferences. For example, users preferred only a single slate when they were on the go, but mentioned that multiple slates were useful for things like clipping out content into the electronic notebook. These comments confirm the fact that practical issues like size and weight will limit how many slates a user will be able to use at any given time.

3.3. Impact on Our Final Design

The technology probe directed our attention to critical parts of our system that we needed to refine in order to produce a viable reading platform. They include the following key points.

—Slates are not just "hard paper." The physical size and weight of a slate device plus the fact that each device multiplexes several pages on a single physical object means that slates are used differently from paper. As a result, our design should seek to achieve the same goals of paper-based interactions, not replicate them on the new platform;
—Slates should be tightly interconnected with each other and other tools like laptops. We believed we had addressed Morris et al.'s recommendations by implementing a distributed clipboard function (which users used and liked), but this was not enough. A successful system must take extensive advantage of the available screen real estate to simplify interactions with each document. It is also important that the multi-slate systems be easy to use during the text creation phase when users are mostly typing text on the computer;
—The "ideal" number of slates changes over time. The number of slates a user employed was heavily dependent on task, environmental constraints, and individual preferences. As such, a reading system should not force users into using a fixed number of screens. Rather, it is preferable for the system to offer a way to reconfigure

itself and remap content as the number of devices change. This ability is particularly important in light of the related research describing the importance of allowing users to read in a variety of venues.

4. DESIGN OF THE UNITED SLATES SYSTEM

We now present an overview of our system design, drawing together insights from the existing literature and from the technology probe of our dual-slate prototype. Our treatment of the system can roughly be divided into four parts. The first two parts are concerned with establishing the basic concepts behind the multi-slate environment and consist of how to distribute information to slate devices and how to move this information around efficiently. These functions are mostly targeted at reading tasks at the page, document, and partially the workspace level (as outlined in Table I. Then, we turn our attention to additional functionality that enables slates to work with PCs in the workspace, and finally, we discuss how our document and workspace model supports reading activities that span multiple reading sessions, venues, and devices.

4.1. Mapping Information to Slates

The mapping of document information onto the slates is a matter of striking a balance between the many layout options afforded by independent sheets of paper against the simplicity of managing more structured collections of documents. We experience similar trade-offs in the paper world when we choose between loose-leaf paper and bound volumes. Fortunately, the dynamic display of electronic devices means that on any given slate, users can operate at several different structural levels when working with their documents. Here, we describe how these different levels relate to one another.

As with most commercial reading appliances, all of a user's documents are aggregated into a single library. However, having a single collection for all of a user's documents can present a problem when the number of documents is large: as with stacks of paper, increasing the number of items in the stack can make managing the collection or switching between documents tedious. To address this problem, our mapping of information treats each slate as representing an *active stack* of documents drawn from the library, along with several inactive stacks that can be swapped in at will. This model resembles how knowledge workers employ a small set of "hot documents" that are actively in use and an archive containing the remainder of the documents [Sellen and Harper 2003]. The inactive documents continue to be available through the library common to all slates. The active stack and global library metaphor is also compatible with how the slate environment interfaces with the outside world. The shared library can be implemented as a special Dropbox[4] folder on a PC. To get new documents into the library, users drag and drop documents into that folder, which makes it available to the entire reading environment.

The document at the top of the active stack is the active document. Navigation within the active document changes the active page, which is what the user sees on the screen in the default, single-page view of the content. Navigation between documents changes the active document in the stack (and changes the active page, by extension).

One major difference between our mapping and paper documents is that we do not allow pages of a document to be split apart from one another, as can be done with paper. At first glance, the policy may appear to limit the expressivity of our system compared to loose-leaf paper. But, in the digital world, moving a whole document is as easy as moving a single page of that document. The practical effects of this policy are seen in some of the operations we present later in this section that allow pages to be moved between slates. When a page is moved from one slate to another, the document to which

[4] `http://www.dropbox.com.`

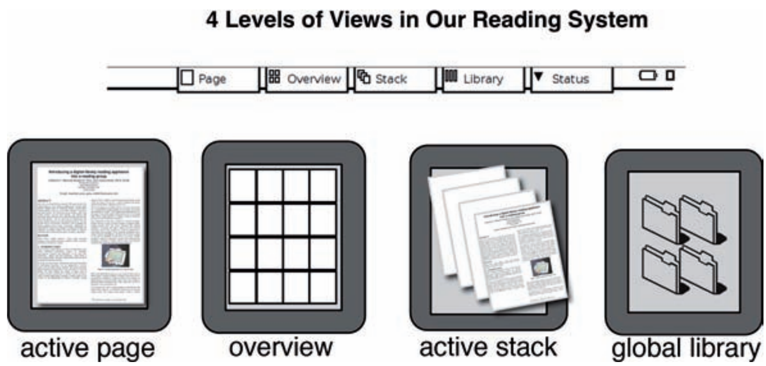

Fig. 3. The system provides four levels at which users can interact with documents. Our multi-slate interactions enable combinations of these views to support reading activities. A row of view selection buttons (top) is placed at the top of the screen and is used to switch between levels (bottom).

the page belongs is placed at the top of the active stack of the destination slate. In this way, we preserve the consistency of the model, retain rich layout capabilities, and avoid the annoyance of having pages of different documents becoming intermingled.

4.2. UI for Viewing Documents at Different Levels

The user interface (UI) of the slates contains a row of view selection buttons at the top of the screen. These buttons allow users to switch between one of four standard views (Figure 3) corresponding to the different information mapping levels previously outlined. To take full advantage of the available screen real estate, each view is always shown full screen. This approach simplifies the interface by avoiding inefficient tiling and complicated panning or scrolling maneuvers that would be necessary if multiple documents or views shared the same screen. The first view is a single-page view of the document; the second is a space-filling thumbnail (SFT) [Cockburn et al. 2006] overview of the active document; the third shows the active stack of the slate; and the final is an interface into the shared library across all of the devices.

The key principle behind our reading system is that users will often use more than one slate. Using multiple devices allows information to be multiplexed in space, which allows more material to be viewed at once. Spatial layout also helps the slates to conform to users' mental models of the content, which is important for organizing and keeping track of information during reading activities [O'Hara et al. 2002]. The cognitive benefits are potentially reinforced by the fact that the information is on physically separate devices, allowing users to take advantage of physical partitioning between devices [Grudin 2001] to better isolate distinct groupings of information.

Having multiple slates in the environment naturally implies that multiple views will be visible to the user at any given time. Consequently, most of our interactions are designed to leverage the opportunities that are possible when multiple views are used in support of each other.

4.3. Moving Information Around

While the initial distribution of information across the workspace is important, reading activities additionally require this information to be manipulated and moved around. For instance, on paper, the process of moving information around allows users to access, organize, and compare information. The key realization is that there are many ways information is moved around in the environment. For paper interactions, similar tasks may involve completely different physical manipulations. For instance, while annotating a document placed far away, a user might reach and annotate in-place, temporarily bring the document closer and then put it back, or more permanently reconfigure the workspace. Depending on their needs, users decide on the fly which movements to employ. Every method of moving information imposes some type of cost/benefit trade-off, making certain movement strategies better suited for a particular task than others. Factors that can influence the appropriateness of a particular transfer technique include the speed of performing the transfer, the physical effort required to specify the relationship, and the duration of the interaction (across which the setup costs are amortized).

One important characteristic of paper is that moving information around is synonymous with moving the physical sheets of paper. In the case of our slates, moving the physical devices is not only unnecessary, since we can simply move the information over electronic channels, but undesirable, because slates are heavier and thicker than sheets of paper. Instead, what we strive for in our system is to enable users to achieve a similar degree of information movement, but to use electronic functionality to decouple that from physical manipulation of the slates.

With this in mind, our reading environment provides a range of interactions for moving information electronically between the views on our slates. Through these electronic links, we are able to provide a wide range of electronic tools that enhance reading. We describe these interactions and the features they enable in more detail next.

4.3.1. Point-to-Point Connections Between Slates. To provide an unambiguous and fast way of moving information between views, we aimed to give the user a direct method of specifying the slates participating in the operation, while also taking advantage of bimanual processes present in reading [O'Hara and Sellen 1997]. Having the user directly specify the source and destination is well suited for our application, where the slates are in the vicinity of one another. Some of the advantages that direct interaction over a symbolic approach (like picking devices from a list) include reducing confusion about where data is going, side-stepping the complexity of mapping a device to a name, and supporting the use of anonymous displays, which abstract away the idea that each new screen is a separate computer [Rekimoto 1997].

In the resulting technique we developed, the non-dominant hand is first used to designate a target, and then the dominant hand, which offers more precision, selects the item to be transferred. One way to visualize the process is that the user's body forms a *conduit* through which the information is transferred (which we will refer to from now on as the *Conduit technique*). Conduit operations are particularly well suited for activities in which content needs to be transferred to a device with which the user is actively interacting, since the user's non-dominant hand is already close to or holding that device.

One way in which we use the Conduit technique to move information between devices is to allow users to navigate to the page corresponding to a thumbnail on a device other than the one where the thumbnails are displayed. Retargeting the thumbnail navigation makes it possible for users to have devices that are dedicated for thumbnail overviews. To navigate, the user first depresses one of the large circular buttons on the device where he wishes to view the thumbnail in detail (which we will subsequently refer to as the *command button*), which specifies the destination slate of the command. Then, the user selects one of the page thumbnails shown in the overviews (Figure 4). The slate on which the command button is depressed shows the selected page. In this way, the user has the ability to quickly jump to any page to see it in detail, without having to repeatedly switch between page and overview views. With multiple panels of thumbnails, each corresponding to a different document, the user can rapidly access pages across different documents. Another situation where the Conduit technique is

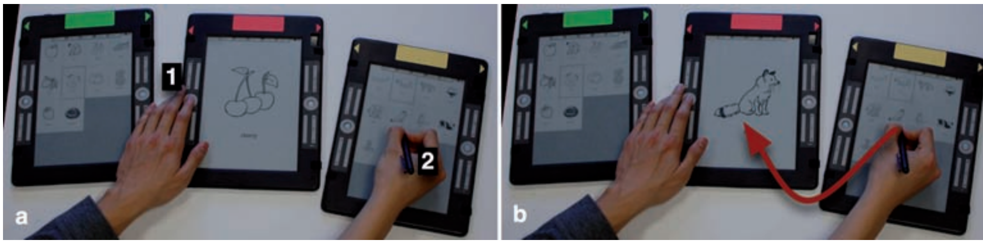

Fig. 4. Distributed thumbnail navigation using the Conduit interaction technique. Left: We navigate to a page on the red slate using a thumbnail on the yellow slate by depressing the command button (1), and then tapping on desired page (2). Right: Result of the operation; note that the user acts as a Conduit through which the data flows.

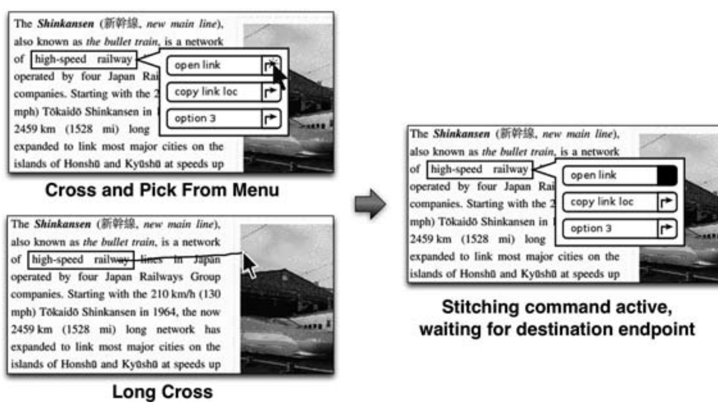

Fig. 5. Performing a Stitching operation using stichable hyperlink elements. Users have a choice of selecting from the full menu of commands (left, top) or using the shortcut of a long cross (left, bottom). While the system waits for a destination endpoint for the command, the indicator square next to the selected command is filled. (This figure is a screenshot from our device simulator and uses desktop cursors for clarity.)

employed in our system is for transferring documents from the active stack of one slate to the active stack of another.

Although Conduit makes quick transfers possible, one limitation is that the transfers are asymmetric toward the slate the non-dominant hand controls. Moreover, given its simplicity, it is also not very well suited for commands in which parameters are needed at both the origin and destination, such as copy and paste. For these cases, our system offers a variant of Stitching [Hinckley et al. 2004]. The basic mechanics of Stitching are the following: first, a stichable element on the source slate is crossed through with the command button depressed, which causes a context menu to pop up. Then, the stitching icon corresponding to a command is picked from the menu. Finally, the user taps on the destination slate, optionally specifying additional parameters for the command, at which point the command is executed. For interactional consistency, the source and destination slates can be the same in a Stitching operation.

To further speed up stitching operations, our system provides an additional shortcut by allowing the user to cross through UI elements with a long cross to automatically select a default command, rather than having to pick from the menu [Dixon et al. 2008] (Figure 5, left). Finally, we want to clarify that Stitching and Conduit can be used interchangeably for all of the interactions we just presented. For instance, even though we demonstrated opening hyperlinks on a different slate using Stitching, Conduit could have been employed instead. Likewise, it is possible to Stitch a thumbnail from one slate to another.

At the page-view level, Stitching hyperlinks to another device offers a way to explore hyperlinks on a page without having to navigate away from that page. By keeping the origin of the hyperlink visible, this feature helps maintain context for the newly opened link, supports comparisons between linked documents, and speeds up backtracking by not requiring extra navigation to return to the source of a link. Figure 5 illustrates the process of stitching a link from one slate to another.

4.3.2. Proxying Interactions through Slates. One interesting feature that can be exploited in a multi-slate system is that since individual slates are functionally identical, one slate can easily double as another. The chief way we apply this functionality is in a remote control feature to provide interaction at a distance.

Since users have trouble stacking up and overlapping slates, they have a tendency to spread them out across a large area on the desktop. Unfortunately, certain interactions such as navigating, writing, or examining a page in detail are difficult to perform when a device is far away. Users can stretch to operate a slate, but that would be uncomfortable for lengthy operations. Users could also pick up the distant slate and move it closer, as we might do with paper. Unfortunately, if a user has a slate in hand already, this requires extra work because it is difficult to hold two slates at once given their size and weight. The remote control feature addresses these problems by allowing the user to use a nearby device as a proxy for a device that is far away. The functionality leverages the fact that users frequently have a device in hand while reading. By using the device in hand as if it were another, we can provide the illusion that the user has virtually picked up a distant device and brought it closer.

In our system, a Conduit interaction is used to activate the remote control feature. As with all Conduit interactions, the command button on the closer device is depressed. Next, one of the view selection buttons at the top of the screen is picked on the distant device, which sends that particular view to the nearby device. When that occurs, the screen on the distant device is frozen and displays a message reminding the user that it is being remotely controlled. One enhancement we added to make the remote control session easier to start was to let users push the command button of the distant device instead of having to tap the view selection button on the distant slate. This enhancement provides users with a larger target to acquire as well as tactile confirmation, which is helpful when the user must reach to access a distant device.

When a remote control session is active, all of the functionality of the system is retained with the exception of being able to invoke an additional remote control session, as we believed it would be difficult for users to keep track of multiple simultaneous remote control sessions or chains of remote control sessions. In order to enable the normal controls to work transparently during the remote control session, we needed to introduce a special input sequence to terminate the remote control session. Thus, the user ends the remote control session by double clicking the command button on the nearby device, at which point, the distant device updates to match the changes made while it was being remotely controlled, and the local device returns to its original state (Figure 6).

4.3.3. Proximity Links between Slates. More permanent connections between slates can be useful for reducing repetitive actions to set up relationships between slates. However, invisible electronic links between slates run the risk of confusing the user if the user is not aware of the connection. For instance, Chen et al. [2008] reported that users were confused when displays on a device changed unexpectedly while the user was working with a different device. These issues are an example of mode confusion. In order to reduce the chance of this type of mode confusion, our system only allows more

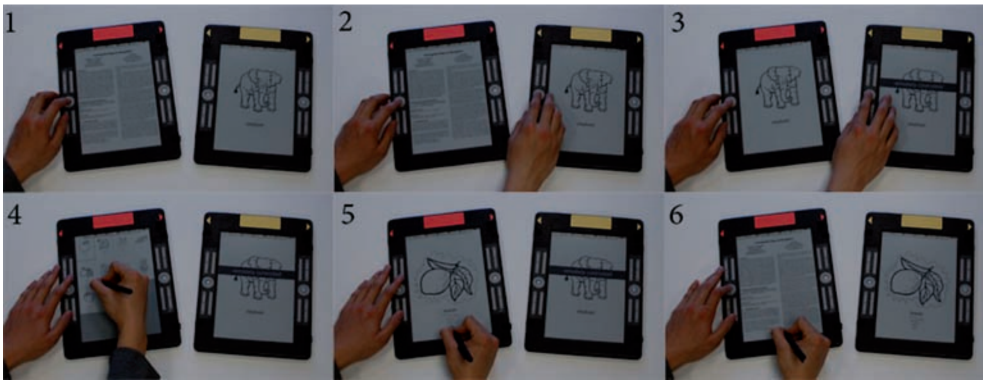

Fig. 6. Example of the slate remote control interaction. (1) Press and hold command button on local device. (2) Click command button on remote device (enhancement). (3) Remote screen is frozen and transferred (note the message informing the user that the session is active). (4, 5) Using local device as a proxy. (6) Double-click command button to end remote control session.

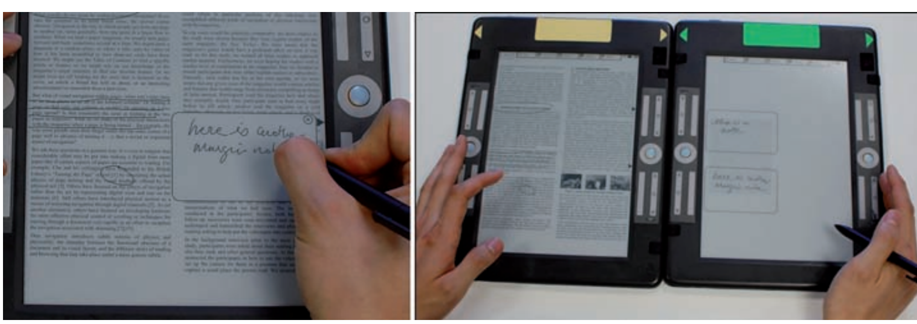

Fig. 7. Left: Single screen margin callout window. Right: Proximity links used to expand the view of a page to include margin notes.

permanent connections to be formed based on slate proximity. This provides strong visual reinforcement that there is a relationship between devices, thereby reducing the chance that slate state changes result in confusion. Although the cost of establishing a proximity-based connection between devices is relatively high, given the physical manipulation required, the resulting implicit connection between devices saves user effort later on and is worth it for connections that are expected to last a long time.

Like the Dual-Display E-Book [Chen et al. 2008], we use proximity links to create a two-page, side-by-side view of a document with synchronized page turns. The two-page view effectively increases the amount of content one sees from a document at a given time. Proximity links are also used in our system to streamline the creation and viewing of margin callouts on the page view of a slate. These callouts provide additional margin space to complement overlaid ink markings [Pearson et al. 2009]. Normally, when using one slate, users can get extra writing space by creating a margin callout with an angle bracket gesture (>) against the edge of the screen. With a single slate, margin callouts need to be called up and viewed one at a time (Figure 7, left). But when the slates are linked, the system expands the margin to the second slate; the right slate displays all of the margin callouts associated with the page shown on the left slate (Figure 7, right). Creating new callouts or editing existing ones also occurs on the right side, eliminating issues of the callout occluding the main text in the single slate case. When slates are initially brought together, the system defaults to the two-page

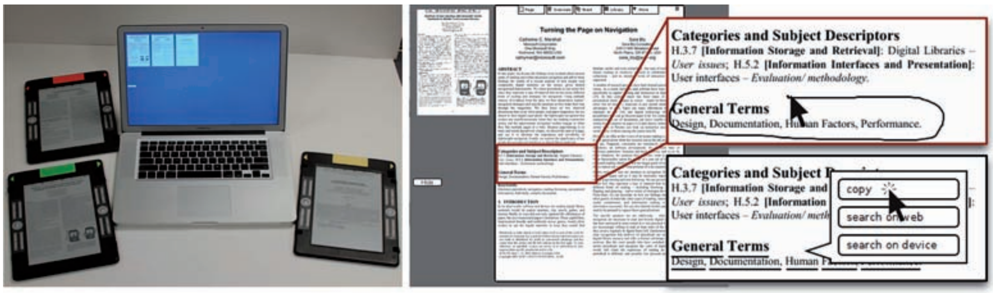

Fig. 8. PC remote control example. Left: PC showing an overview of slate devices. Right: Detailed view of a screen on a PC after selecting a slate to control. Insets: Lassoing text, then selecting the copy command, at which point the text is copied to the distributed clipboard. (Mouse pointers are enlarged for more visibility).

view, but the user can easily switch the behavior to expand the margin using a system setting.

4.3.4. Remote Control from PC. We saw during our technology probe that it was important for users to have their reading system well integrated with their main content creation tool, which was generally a laptop or desktop PC. PCs provide a keyboard and a vertical display that faces the user and are generally preferred for composition tasks [Morris et al. 2007]. Other tasks for which a PC is superior include things like accessing data from digital libraries. As such, we expect that many documents will be marshaled together on the PC first before making their way onto the slates. While it is an option to attempt to duplicate PC functionalities on the slates themselves (like a tablet PC), that would necessarily entail adding a great deal of complexity to the interface. Wilson and Landoni [2003] point out that one desirable quality in a reading device is maintaining its simplicity so that operating the device does not begin to interfere with reading activities. Thus, we thought a better approach would be for the slate system to complement, rather than replace, the PC in reading environments. One way we handle the requirements of federating with a PC is through a synchronized distributed clipboard [Miller and Myers 1999], of which the PC is a member. The synchronized clipboard supports information extraction tasks by ensuring that content on a slate can be shared between both slates and PCs.

The concurrent use of PCs and slates presents an additional interactional challenge arising from differences in the devices' input modalities. PCs are primarily operated via keyboard and a pointing device, while our slates are controlled via hardware buttons and a stylus. Tasks that involve repeated switches between PCs and slates are slowed considerably from the user having to re-home on different input devices after every switch. To address this problem, we decided to extend our remote control mechanism so that slates can be controlled from a PC. Upon activation, our PC interface presents a grid of all of the slates in the environment (Figure 8). From this visualization, the user can select a particular slate and then control the slate using the keyboard and mouse. This feature uses the same mechanism that powers the remote control feature described in the previous section. The PC remote control function allows a user to have full control of a slate environment (including text entry and text selection), but to treat each slate as if it were another window on the computer desktop. For example, it is possible to excerpt a piece of text very quickly using this feature (Figure 8, right). A visualization that matches the positioning of the slates in the environment would be possible if we had more sophisticated position localization techniques at our disposal.

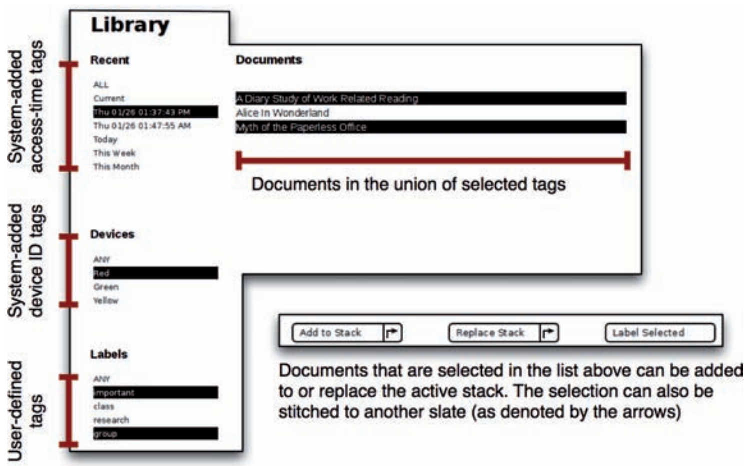

Fig. 9. Stack Manager interface in the Library. Users select a combination of system-added and user-defined tags along the sidebar to filter for documents. The filtered documents (or a subset) can be added to the active stack, replace the active stack, or have tags applied to them. Note that the screenshot only depicts select regions from the full UI to keep text legible.

4.4. Stack Management

The functionality we have presented up to this point has been concerned with reading activities that make use of the small set of documents that are in the active stack of each slate. However, reading activities will almost certainly require documents that are not in the active stack, forcing users to modify, save, or replace the contents of the active stack on a slate. The availability of robust stack management capabilities has far-reaching effects with regard to aiding document organization, providing the ability to suspend and resume reading activities, as well as enhancing the mobility of the system.

4.4.1. Aiding Document Organization. One of the design choices we made in the system is to map several documents to the active stack of each slate, which we detailed in the first part of this section. Recall that the reason for this decision is our belief that users will probably need to work with more documents than there are slates available at any given time. Presumably, like paper stacks, the electronic stacks comprise logical organizations of documents so that users can quickly find and access a desired document. However, with a limited number of slates, a slate may need to be repurposed to view a document that may not necessarily be related to the other documents in the stack of that slate. To illustrate, suppose we have two slates, one with a stack of documents about different types of cats and one about different types of dogs. At some point, we would like to compare two different dogs side by side, meaning we would have to move a dog document onto the stack of cat documents. After several rounds of this, the stacks on each slate lose their organizational significance.

The Stack Manager (Figure 9) component of the document library addresses this problem with an attribute tagging mechanism, like the one found in Presto [Dourish et al. 1999], to group together documents in a more permanent fashion. Tagging of documents is similar to placing them into folders, except that a document can belong to several tag groups. The Stack Manager allows users the ability to quickly specify complex groupings of documents using unions of tag groups.

In one operation, documents in a tag group (or union of tags) can be added to, or replace, the active stack of a slate. The tagging system allows users to quickly and concisely select documents to put on the active stack of a slate, ensuring that users can rapidly get at the documents they need. As for the preceding example, it is simple for the user to "clean" the stack of each slate by reloading the contents with documents that have been appropriately tagged. The reverse process in which a user assigns a tag to the documents in the active stack is useful as well, especially for sorting tasks. Suppose a user is going through a large set of uncategorized documents and wishes to group related documents together. The user can move documents from the slate with a stack of uncategorized documents to other slates that represent the document categories, which initially have empty stacks. After moving the documents to the appropriate slates, the user can tag the documents in each stack in a single operation. This method of sorting allows the user to leverage spatial layout to sort documents, much like what is possible with paper, and avoids repetition by allowing several documents to be tagged together in batch.

4.4.2. Suspending and Resuming Reading Activities. We have shown how the Stack Manager is good for organizing documents, but its functionality also makes it possible for reading activities to be suspended and resumed at a later time, at a different venue, and with different devices. If a user already has slates available in the different places where reading occurs, the ability to restore stacks frees the user from having to transport slates to those venues.

The mechanism that makes this additional aspect of the Stack Manager practical and transparent is the application of automatic, system-added attributes to the documents. As documents are placed onto the active stack of a slate, the Stack Manager automatically tags them with a time-stamp as well as the ID of the device on which it is being placed. Using these automatic tags, it is possible for users to recall the specific documents used on a device at a particular time without needing to explicitly tag the document ahead of time. If a user wishes to restore a stack of documents he was previously working with, he can filter for the documents by time and by slate and quickly resume working with those documents. More powerful, however, is how the user can leverage the functionality to restore a reading session using completely different devices. For instance, if a user is using three slates at work, he can recreate his reading workspace at home using slates he already has at home (Figure 10, right-most panel). The ability to filter by slate makes it easy to map the contents on each work slate to a home slate.

4.4.3. Dealing with Resource Mismatches. The Stack Manager further increases mobility by allowing an environment with many slates to gracefully scale down to one with a smaller number of slates. As a result, users are able to continue working in environments where fewer slates are available (assuming the user does not want to carry a full set of slates), or environments where working with many slates is not practical (e.g., public transit). The method to achieve this is simple; the user can take the union of tags corresponding to documents on different slates and redistribute them onto a smaller number of slates (Figure 10, middle panel). While functionality may suffer from the reduced number of displays, this technique helps reduce the disruption caused by a resource mismatch between reading environments.

4.5. Putting Everything Together

In the preceding sections, we have described many examples of interactions between slates and explain how these interactions can benefit reading activities. In order to give the reader a better idea about how the multi-slate interactions fit within the broad requirements of reading, we place them in context with the different reading requirements we established at the beginning of the article. Figure 11 organizes the features we presented in this section of the paper in a structure matching that of

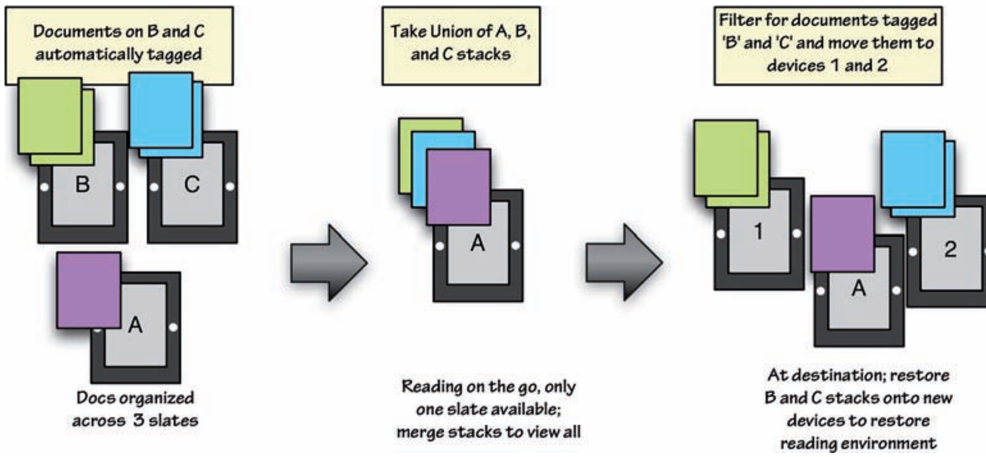

Fig. 10. Using tag operations to manipulate, save, and restore document stacks in order to save and resume reading activities across different environments and times. Although Slate A is always present in all of the venues in this example, there is no restriction that a common slate be present across different reading environments.

Table I. As one can see, the beneficial enhancements of the United Slates system span the full gamut of reading requirements—from the page level to the multi-session reading level. The features made possible through multi-slate use, coupled with the unique functionality of the individual slates (e.g., light weight, readability, and support for writing), translate to more complete support of reading requirements than existing electronic reading technologies.

5. IMPLEMENTATION

Since we already covered the details of our slate hardware in Section 3, we focus now on the software components of the United Slates system, which is composed of the United Slates reading application and underlying system software. The Gumstix computer module we use has a standard ARM-based application processor (TI OMAP3503) for which several standard distributions of Linux are available (our system uses a minimal version of Debian Linux). As a whole, the system operates like any other Linux computer, imposing few restrictions about the programming languages used to implement applications. The main additions made to the operating system were several device drivers that routed data from the input controls to the application and an intermediate layer that presents a virtual Linux frame buffer device to the United Slates application. The driver modifies the graphics data prior to sending it out to the e-paper display controller. Unlike many standard Linux systems, we opted not to use the XWindow system to provide graphics. To reduce overhead, we instead use a lightweight DirectFB graphics layer to draw to the virtual framebuffer. A number of Linux GUI toolkits like GTK+ and Qt can be compiled to output to DirectFB rather than X.

5.1. System Architecture

In our current system implementation, hardware devices are fully interlinked using a Bluetooth personal area network (PAN). Bluetooth PANs require a master device to route traffic between devices, but this aspect of the network is abstracted away at the OS level. Applications instead simply see a standard TCP/IP-based network linking the devices. The reason we adopted this architecture was largely due to the fact that it requires no extra hardware, like a wireless access point, which was useful

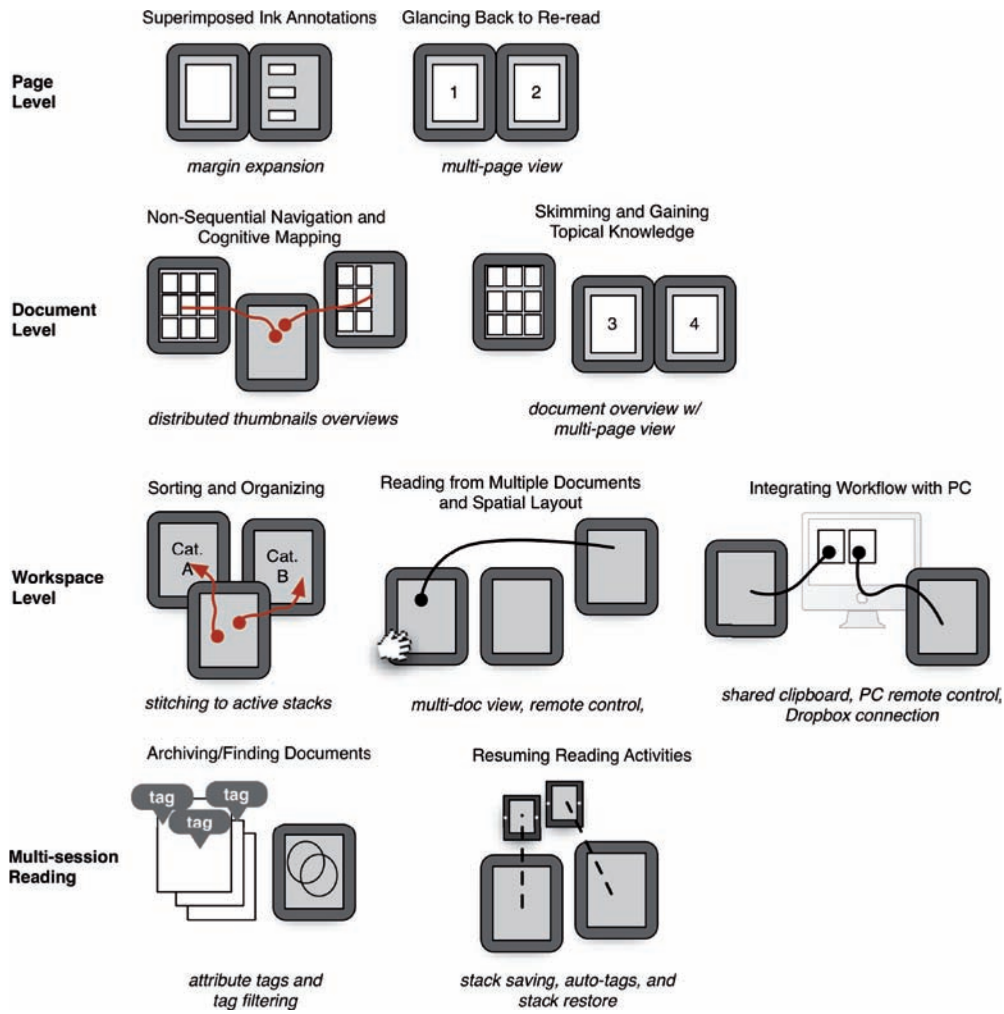

Fig. 11. Multi-slate interactions in the context of the reading requirements presented in Section 2.

when we performed our technology probe and could not reliably guarantee wireless network access. The one limitation this configuration has is that our master device must always be on. An easy and transparent modification that would make the slates fully interchangeable would be to use Wi-Fi networks instead.

In our system, each device maintains a copy of each document being viewed. As annotations are applied, only the annotation data needs to be sent to each device. The device on which the annotation is made is responsible for sending out that data to all over the other devices. We use a non-optimistic serialization scheme [Greenberg and Marwood 1994] (last input in wins) across devices. This scheme works for our application because the scenario is of a single user operating several devices. Also, the modifications users make to the document are simple and can be applied quickly. Strokes generally appear on remote devices within a second.

In order to implement the stack tagging, saving, and restoring features described in our system, a central server storing the state of the various stacks must be available to convey the data to new devices that were not previously present. In the current implementation, we simply placed the server on the Bluetooth master device, because we knew for sure that that particular device would always be present in any configuration. However, in a real-world implementation, the server would be in the cloud and always be accessible. For that reason, our slates' ability to access the Internet is advantageous.

5.2. Application Software

The application software on the slates is written in Python and uses the GTK+ toolkit along with PyGTK Python bindings. Inter-device communication is done through Python Remote Objects (PyRO), which provides facilities for object serialization and remote procedure calls. The tablet federation application on the PC is also written in Python and shares much of the same codebase as the application running on the slates. The application is structured into several view classes, each providing the UI for one level of the system (e.g., page-level, overview, active stack, library). A view manager class routes input events to the view that is active. View classes maintain all variables related to their state and can export the state on demand or temporarily load a new set of state variables. It is through this capability that we implement the remote control features of the system. The system we tested in this article uses a custom document format consisting of page images organized into directories on the filesystem. Strokes are stored in separate files and overlaid on top of the page images. The system also has the ability to view PDF documents, in which case annotations are placed directly into the PDF document. We use the ShortStraw [Wolin et al. 2008] algorithm for many of our gesture recognition requirements.

6. PRELIMINARY EVALUATION STUDY

As the next step in the iterative development of this system, we decided to carry out an evaluation study in which we asked participant users to describe some of their typical work practices involving active reading, and then to try out some of the features of our system assessing them against the real-world tasks they usually carry out. This was not a controlled study aimed at comparing our system against others. Rather, this was an exploratory study in which we aimed to provoke discussion and feedback from potential real users against a grounded understanding of their real work practices.

For this study, we recruited 12 students from Cornell University. We did not recruit for specific majors or class level. Our participants consisted of 11 undergraduate students and 1 graduate student. Participants received $20 compensation for their participation.

The study was set up as a semi-directed interview in which the participants had the opportunity to perform and compare different kinds of interactions on the slates. We started each session with an interview about the reading (e.g., for class, work, or research) which they had undertaken in the previous semester. This was not only to give us an understanding of their practices, but also to get the participants to reflect on the ways in which they really do their work prior to using our system. We asked participants to describe the materials they used, what and why they printed to paper, and the resources they employed when reading and writing, situations where multiple documents were used together, and instances when documents were on different types of media.

6.1. Observing Users Interacting with the System

Following the interviews, we explained that the purpose of the study was to assess the efficacy of a set of features that would make use of situations in which one might have more than one slate from which to read. First, we outlined the basic capabilities of a single slate and let participants familiarize themselves with the controls and inking capabilities of the device. Then, we tested the multi-slate reading support tools we developed in the following order: margin expansion, overview navigation, hyperlink navigation, screen teleport, shared clipboard, PC remote control, and stacks and reading sessions.

The general process we used for testing the tools was to have users perform short tasks that simulated portions of real-world reading activities using each tool. We first had users perform the tasks with the United Slates condition which employed the interactions we have presented. Then we had users use a "naive" condition that represented the experience of using a contemporary single-screen reading device or the user experience of a multi-slate system that did not provide features integrating the slates. The purpose of the naïve condition was to give users a point of comparison against which to discuss the features in the United Slates system. There were two features for which we did not provide a naive condition: copy and paste to the PC using the distributed clipboard, and stack management across slates because there was no plausible method of performing those tasks efficiently without the functionality our system provides. Table III provides specific details about the task that we used to test each feature, the procedure employing the features of the United Slates system, and the procedure for the naive condition.

After testing the multi-slate and single-slate conditions for each feature, we asked users to compare the single-slate method of accomplishing that task and the multi-slate approach. Then we asked participants to describe whether the feature that was just tested would be useful for their personal reading tasks, whether the feature was confusing, the advantages and disadvantages of the feature, and how the feature might be improved to make it better suited for the style of reading they typically performed.

6.2. Results

Users appeared to understand the operation of all of the techniques with minimal intervention and explanation on our part. For nearly all of the features, users had no trouble jumping in and performing the interactions we asked them to perform during the evaluation. Therefore, it would appear that these techniques could readily be used in a system outside of our lab.

Overall, users were positive about all of the multi-slate functionality we tested with them. The positive feedback is perhaps not surprising given the fact that participants in such studies are generally inclined to be positive about prototype systems in which they see the investigator as having a stake. Therefore, the more interesting takeaways are in the specific comments that participants made with respect to each feature, including their suggestions for improvement.

Margin expansion. Participants reported that expanding the space allocated to the margin callouts by physically attaching two devices made creating and viewing annotations easier than the process when using a single screen. Participants mentioned that using this technique meant that callouts no longer occluded the main text, allowing one to read while writing (P4, 7, 9, 10, 11, 12), and that showing all of the notes at once on the second screen made it easy to find pertinent notes (P4, 5, 7, 8, 10, 12). The most requested improvement was to provide facilities to either expand the writing area or to allow writing on the entirety of the second device (P6, 7, 8, 9, 11). Participants added that the proximity link system could also be used for other features, like automatically searching selected text on a second screen (P7) or automatically displaying documents that the user has associated with a particular page (P2, 8, 12).

Distributed thumbnail navigation. Every participant except for P8 mentioned that overview navigation would be very useful for working with multiple documents or locations in tasks such as writing papers, comparing articles, and browsing lecture notes. Most participants (P1, 2, 3, 5, 7, 8, 9, 11, 12) pointed out that they liked the simplicity

Table III. Task and Procedures Used to Test Each Feature with Participants

Feature	Task	United Slates Procedure	Naive Slate Procedure
Margin Expansion	Make margin callout annotations on several pages; return and review the annotations.	Use two slates that are attached using proximity links so that annotations are created and displayed alongside the main text.	Use a single slate to create margin annotations and cycle through annotations one at a time.
Distributed Thumbnail Navigation	Navigate to a different page in Document A (A), navigate to a page in Document B (B), and then navigate back to A.	Use a total of three slates, with two slates showing thumbnail overviews of A and B and one slate for viewing a page in detail. Use the distributed thumbnails to jump to target pages.	Use a single slate to go between A and B and back to A. Users needed to go to the stack view to switch documents and then use the thumbnail overview in each document to navigate to the target page.
Hyperlink Navigation	Visit a sequence of hyperlinks. The sequence includes backtracking navigation where the next hyperlink to follow is on a previously viewed page.	Perform the navigation using three slates using a combination of Conduit and Stitching interactions to open links on other devices in the environment.	Use a single slate to visit the sequence of links. Backtracking was performed using a dedicated "Back" button.
Screen Teleport	With a slate in hand, change the page on a different device to a specific target page. Once at target, make ink markings on that page.	Use screen teleport function to move the display of the distant device to the slate in hand. Perform navigation and inking tasks using slate in hand and send display back to distant device.	Reach and pick up the distant slate, perform navigation and inking task, and then return the distant slate back to its original location.
Distributed Clipboard	Copy and paste text passages from the slate to a word processing application on the PC.	Copy selected text to the distributed clipboard. Paste text on PC using either mouse or keyboard (up to the user).	N/A
Remote Control from PC	Navigate on the two slates alongside the PC so that they are displaying specific target pages.	Use the PC interface to remotely control the slates. Perform navigation without touching the slates.	Navigate by directly operating the two slates.
Stack Manager, Tagging	Place a selection of documents into a newly created stack. Shut off a device; use stack manager to restore that stack.	Use stack management interface and perform tasks using tags.	N/A

and speed of using the thumbnails compared to the alternative we had participants try, which involved going to the current stack, switching documents, and then navigating to the desired page. Improvements that participants suggested included showing annotations and margin notes more prominently in the thumbnails (P2, 4) and making the process of calling up a thumbnail on a different slate more like drag-and-drop (i.e., providing visual feedback showing the process of moving the thumbnail from one device to another) (P6). P11 mentioned that he probably would not want to use a separate device for each document overview, as that could become difficult to manage. Instead, he said he would prefer to "swipe" or otherwise cycle between SFT overviews on one device while employing an additional device to view the page in detail.

Hyperlink navigation. Participants found several reasons why they would want to open a link on a different device. First, opening a link on a second device made sure both the source and target documents are visible, ensuring the user did not get lost. It also reduced the amount of flipping that needed to happen to work with both documents (P1, 2, 6, 9, 10, 11, 12). Cross-device hyperlinking also helped participants focus on reading material by allowing them to move content from a peripheral slate onto the "main" device that they were holding (P6, 8, 9, 10, 11). One problem that some participants encountered when we introduced Stitching after demonstrating Conduit (for thumbnails in the previous task) was confusion about which technique to use when attempting to open a link on a different device. When asked to open a link, some users proceeded to pick up the target device to use Conduit. In these situations, we reminded the participant to use Stitching. Most users, when asked about this behavior, attributed it to having the (incorrect) impression that information could only move from the dominant hand to non-dominant hand. They added that once they became more familiar with the interaction, Stitching made sense as well. Based on this, we believe that a more consistent way to view the Conduit interaction is as a shortcut for special cases of Stitching. That is, every operation is a Stitching operation, but by virtue of the command button, Stitching to the slate the user is holding does not require an endpoint to be specified. A few participants (P11, 12) expressed concerns that spreading documents among too many devices could be confusing. Two participants (P10, 12) suggested that a simple list-based approach for selecting a target, rather than interacting with the target device itself, might be preferable.

Screen teleport. In the non-teleport condition in which users had to manually operate the remote slate, all participants placed the slate they were holding down on the table before picking up the other slate, confirming the difficulty of holding onto more than one slate at any given time. However, the utility of using one slate to control another split user opinion. Most participants found using a nearby device to control a remote device to be preferable than the alternative of reaching over to pick up the remote device (P5, 6, 7, 8, 9, 10, 11, 12). P8 described it as being "more efficient—3 presses of a button did the same as reaching over." However, a contingent of participants (P1, 2, 3, 4) found the feature to be confusing, mentioning that it would be simpler to operate the devices directly, even if it meant physically shifting devices around. These participants said that the prototypes were light enough to move around without being awkward and indicated that the experience of using the screen teleport feature to be slightly worse than directly moving the slates. Like with the hyperlinks, participants mentioned it would be helpful to pick from a list (P10, 12) rather than having to stretch to interact with the device to proxy to. P12 specifically mentioned that picking from a list would be useful for situations in which she moves from reading at her desk to reading on the bed and wants to access a resource that was collocated in the environment but just out of reach.

Shared PC clipboard. All participants thought the shared clipboard feature would be helpful, although P8 believed that working with electronic documents directly on his PC would still be preferable. Not surprisingly, several participants (P1, 2, 7, 9, 10) pointed out that it was annoying to switch between the slate and the PC. There were some differences in how our participants employed the clipboard, however. Instead of using pull-down menus on the PC, P5, and P11 used keyboard shortcuts to perform the paste operation on the PC side. Since it was possible to press the shortcut keys while holding the pen, these two users did not report issues with having to switch devices.

Remote control from PC. All participants, with the exception of P12, found controlling slates remotely through the PC to be the more preferable option for operating the slates when we asked them to control the slates while performing a simulated writing task on the PC. P12 found it easier to operate the devices directly. Many participants cited the fact that they did not have to switch devices as an advantage (P3, 4, 5, 6, 8, 11). P4 and P8 remarked that the remote control allowed the slate alongside the computer to act as an extra, portable monitor. Another advantage participants cited as a plus for the remote control functionality was the fact that they were familiar with using a traditional PC (P1, 2, 3, 9, 10). P7 and P10 mentioned that the PC interface made controlling many devices in rapid succession easier. One unforeseen benefit of the PC remote control feature was that turning pages and switching documents on the PC was much faster, owing to the faster processor and display on the computer. Participants had mixed feelings about the lack of an accurate rendering of slate location in our visualization. A majority believed that the visualization actually felt that it provided a more organized view of the slates in the environment, allowing one to more easily see and access the slates (P1, 2, 4, 5, 6, 9), but several participants mentioned that it was somewhat confusing to identify a target slate because the ordered grid of slates in the PC UI did not match the spatial layout of the slates on the desk.

Stacks and reading sessions. The vast majority of participants thought that tagging documents and organizing them into tag groups to be a good translation of how they currently organize documents. P9 made the comment that the organizational system made going through the documents easier by making it “feel like checking email.” Some users additionally mentioned that they sort documents by time as well, which the time-based tagging automatically supported. Users also mentioned that the stacks would allow them to go beyond what they currently do. A number of participants mentioned that they frequently lose track of documents, so the automatic session tracking feature would be helpful (P3, 6, 7, 9, 12). Finally, participants confirmed that the ability to save reading sessions to be resumed later on other devices would increase portability (i.e., not having to carry devices everywhere) and would allow them to just carry a single “main/master device” (P5, 6, 10, 11, 12). The participants who were not as enthusiastic about the stacks and sessions (P4, 8) mentioned that the use of the tags resulted in too many grouping options. Instead, these users said that they would just prefer to organize documents using traditional folders.

6.3. Evaluation Summary

Participants found the interoperability of the slates with the PC to be the most compelling feature, since the isolation of information on paper documents from tasks on the PC was a problem familiar to all of our users. The other feature of our reading system that participants judged to be directly applicable to their current reading needs is the ability to manage large collections of reading material we provide in our stack manager.

One particularly interesting observation was that while users were split about the utility of screen teleport between slates, they were considerably more positive about the application of the same core functionality for controling the devices from the PC. This is an interesting finding on several levels. For one, the fact that many users did not find picking up a slate and moving it around onerous provides a data point about the threshold of how heavy a reading device can be before it becomes burdensome to handle. Our devices, which weigh in at 500g, appear to be under this threshold. The discrepancy between slate and PC remote control is enlightening as well. We believe there are two main reasons for the discrepancy. First, there is more work involved in switching from using a PC to using slates than in switching from slate to slate. Second, many users found that performing navigation tasks on the slates using the PC interface provided a better experience than working on the slates directly: on the PC, all the slates can be controlled from a single location and using a keyboard and mouse was fast and familiar.

Finally, we noticed that there was a practical limit in the number of slates that could reasonably fit in a user's immediate working area, especially when a computer was present. The main reason for the limit is that users do not overlap slates: the thickness of the slates causes slates to wobble when partially stacked on top of each other. Therefore, P11's suggestion of allowing cycling through multiple panes of thumbnails on one device is a particularly useful suggestion.

7. DISCUSSION

7.1. Results of Our Study

Based on our results, we can give some recommendations about the areas where creators of reading devices targeting work- or academically-related active reading might wish to focus their immediate attention.

Existing electronic reading devices have rudimentary systems for working with multiple documents. Managing, locating, and navigating between documents are all poorly supported. Current devices, designed for the linear reading of long e-books, can get by with the linear lists they employ because movement between documents is infrequent. The current state of affairs will be increasingly intractable for more general-purpose reading systems. The types of documents that users potentially wanted to use on our system (e.g., lecture notes, articles, and paper-writing) indicate that with added functionality, an electronic reading solution would be used for a wide range of electronic documents, not just e-books. When that occurs, the number of documents one loads a device with will likely balloon, as will the importance of being able to rapidly move between these documents. Even the document management interface we present with United Slates can be improved with features like thumbnail previews of items in the active stack to further accelerate navigation between documents in use.

Although our study has found two or more devices to be useful, we think it is important to point out that a single device is just fine for many reading activities. While there has been great interest in dual-display reading devices (often because they emulate the look of a printed book), we believe that for maximum flexibility, it is better to enable single-display devices to work together on an ad hoc basis, as we have done. An extensible environment of independent slates could grow to fill needs that might not be adequately handled by a system fixed at two displays. Also, a multi-slate design would probably have an easier time gaining broad acceptance, since it leaves it up to the end-user to decide whether to make the necessary investment for additional displays.

7.2. Using Multiple Devices to Sidestep Reading Tensions

Tashman and Edwards [2011a] identified two major tensions in active reading technologies. The first of these tensions was that users wanted a large amount of space to work with many documents, but also wanted the reading environment to be portable so that they could set up in arbitrary locations. The second tension was that users liked the experience of reading from a dedicated device, given their unique affordances, but required the use of their PC for many of the support activities connected to that reading. We believe that the multi-device strategy employed by the United Slates system offers an effective solution for resolving both of these active reading tensions.

One of the fundamental motivations underpinning our system is providing more space to work with documents in ways that are both flexible and optimized for cross-document use. At the same time, we believe that our system is also a more portable solution than other approaches. At the most basic level, transporting a set of slates is a far more practical proposition than transporting a large multi-touch surface. However, with the ability to save and restore reading sessions, users may not need to transport any slates at all, as they can capitalize on slates that are already present at the venue where they wish to read. One way to further increase the mobility of our system is for public venues to provide slate devices as infrastructural tools, much like a projector. The reason this vision is compatible with our system is that the stack manager blurs the lines between public and private devices. Work performed on a shared public device is not lost once the user returns the public device. Furthermore, we might use physical proximity links as a way to temporarily associate public devices with a user in a fast and unambiguous manner. The physical nature of this type of authentication may also be useful for dealing with rights management issues.

How United Slates integrates reading tasks with activities that occur on other tools in the reading workspace was another requirement that we expressly designed our interface to support. We recognized that PCs and reading devices have unique niches owing to the significantly different input and output modalities that each support. As such, we believed it was unwise to force users to constrain their reading tasks to the PC or relegate standard computing tasks to a device that offers a smaller screen and lacks keyboard and indirect pointing control. This belief led us to implement features that streamlined the use of both slates and PC simultaneously. In our study, users expressed great enthusiasm about the ability to copy and paste between the slates and the PC rather than a desire to write papers on the tablet. Therefore, we believe that designers should think of slate reading devices not as a separate computing device to be used in isolation, but as a peripheral where certain activities can be offloaded. In some sense, designers should aim to reduce the costs of choosing the “best” device for a particular task. To support this pattern of use, we believe it will be important (and probably not too difficult given advances in cloud computing) to offer even tighter integration with the PC. For example, files and annotations on the PC should be readily available on slates, and changes made on either type of device should immediately propagate.

During our evaluation, an even more interesting phenomenon involving the PC emerged: we discovered that the PC served as a good platform for performing meta-tasks in the multi-slate environment. For example, the PC provided a central place to quickly set the state of multiple slates. Thus, we believe that connecting the multi-slate reading environment with external devices goes beyond making sure that we can move data between slates and PC. Instead, researchers and designers should consider new interactional options that further take advantage of the union of slates and PCs.

Overall, the United Slates system presents a multi-device solution that enables users to work with many documents at once, provides the annotation qualities and form factor found in dedicated reading devices, and allows users to fluidly work across PC and reading slate. This is a particularly interesting result because it indicates that a composition of devices, rather than a single hybrid device, is a good way to combine the desirable functionalities found across different devices. Don Norman once observed that, “When one machine does everything, it in some sense does nothing especially well, although its complexity increases.” [Rheinfrank 1995] Our results indicate that the strategy of using device ensembles [Schilit and Sengupta 2004] is an effective way to maintain the unique qualities of a device while increasing overall functionality.

7.3. Areas for Improvement

Although our system, as presented, attempts to address many of the outstanding problems in supporting reading activities on digital systems, there are still many enhancements that could be made to the system to improve its performance.

7.3.1. Fast, Touch-Capable Displays. In constructing our system, we were forced to make a number of trade-offs about the technologies we used for the system. In the end, we elected to use slower but lighter and more readable e-paper displays. Practical considerations for things like battery life arising from our plans for future deployments with these devices, shaped our decisions as well. A consequence of this choice was that not only was our screen refresh rate limited, we could not employ direct touch sensing, because touch sensing, overlays considerably degraded the screen image. Fortunately, these trade-offs are not fundamental, as a number of viable solutions are around the corner. For instance, the Mirasol[5] display promises a fast, full-color display with energy consumption characteristics of current electronic paper. For situations where writing and weight are not critical, it is also important to point out that many elements of our system could be readily implemented on commercial tablet appliances like an iPad.

It is clear that faster screens and touch control would both be valuable for improving the interactions on reading devices. Faster displays would enable devices to more closely match PCs and tablet appliances in reading activities like skimming, where rapid presentation of content is beneficial. Furthermore, more interactive displays translate to more visualization possibilities for improving users' cognitive awareness of the state of the document, stack, or reading environment.

Touch input would open the door to richer bi-manual operation of the devices, allowing them to approximate more closely what is possible on a tabletop computing environment. Projects like LiquidText [Tashman and Edwards 2011b] have further demonstrated how the combination of highly interactive displays and multi-touch can be used to support active reading. These techniques would be great additions to the system we present in the article. Moreover, touch input could further extend the expressiveness of the techniques we present in this article. For instance, with touch, one can directly specify screen positions at both the source and destination slates of a Conduit operation, allowing the fast mode of transfer Conduit provides to be employed in a wider variety of cross-slate operations.

7.3.2. Slate Localization and Identification. For certain interactions like opening thumbnails and remote control, participants in our evaluation mentioned that indirect interaction that did not require physical reaching, like picking a device from a list, would be faster. This preference was further reflected in users' enthusiasm about controlling many devices indirectly from the PC, which could be done quickly without any reaching required. However, in an environment where slates are similar in appearance (and/or showing similar-looking content), relying on indirect interaction could be confusing, since users would not be able to reliably correspond a choice in a list to a slate in the environment. Although providing additional feedback or identification cues through multi-monitor window notification techniques [Hoffmann et al. 2008] could help alleviate this problem, we earlier mentioned that it would be desirable to recover the position of the devices in the environment.

Having the positions of the devices in the environment would, for one, allow the remote control interface to render the virtual slates in a way that more closely matches the physical layout of the actual slates. However, slate localization has additional benefits. Recovering the relative positions of devices would allow richer indirect interactions, like those seen in Hyperdragging [Rekimoto and Saitoh 1999], which might further address problems of reaching. For instance, this feature could be used to open a link on a different slate merely by making a flicking gesture in the general direction of the destination device.

We omitted this functionality in our current implementation because existing positioning systems require infrastructural support at the room level, which would limit

[5] http://www.mirasoldisplays.com.

where the slates could be used. However, one compromise is to have the functionality only be active in select environments. A better solution, of course, is to have a localization system that does not require any environmental infrastructure. One possibility would be to use acoustic localization technology like that demonstrated in BeepBeep [Peng et al. 2007] to obtain centimeter-level positioning accuracy or to adapt localization techniques developed for sensor networks [Broxton et al. 2006].

7.3.3. Integration with Other Computing Platforms. Although the United Slates system has shown promise integrating beneficial features from a variety of existing computing configurations, it does not reproduce all of their beneficial functionality. Therefore, it might be helpful to consider how multiple slates might coexist with these other types of computing systems. For example, an actual tabletop computer would be a great complement to slates when they are available. Tabletops make working with very large quantities of information tractable. Moving things around, creating, and deleting items is fast and easy on a tabletop computer. Slates, as we have shown in this article, provide a dynamic display in a highly mobile form factor. The two together would support a wide range of reading situations that include working with very large numbers of documents and multi-user reading with a shared display. Another technology that should not be discounted is paper. Paper is both inexpensive and dispensable. Furthermore, there do not appear to be any technologies on the horizon that match the writing experience on paper. Therefore, digital paper technologies like PapierCraft [Liao et al. 2008] would be a good candidate to add to this integrated, multi-device reading environment.

8. FUTURE WORK

We believe that we have only scratched the surface with regards to the functionality of multi-device reading systems. For one, we believe such a system would mesh well with other techniques that have been developed for pen-based computing platforms, such as tablet PCs. For example, providing active note-taking functionality like that presented in InkSeine [Hinckley et al. 2007] could be beneficial for reading. InkSeine's in-situ ink search capability and interface for creating custom layouts of extracted content and hyperlinks would help users organize concepts they encounter, as well as speed navigation between documents. The annotation browsing, content recommendation, and automatic hyperlink creation capabilities found in XLibris [Schilit et al. 1998; Price et al. 1998] could also be adapted to take advantage of several linked displays.

In light of our observation that there are many other possibilities for slates and PCs to work together as a unified whole, we also believe that it will be important to push further in that area. For one, our current project has largely focused on workflows that occur mostly on the PC but require the occasional use of slate resources. Scenarios involving the reverse direction are equally likely, namely reading-heavy activities where the PC is occasionally used to access resources like digital libraries or Wikipedia. To support these scenarios, applications on the PC should be made slate-aware so that operations on the slates can propagate to the PC with less user intervention. Perhaps, to add a level of generic control, it should be possible to temporarily transfer the contents of any window on the PC to a slate and allow the user to interact with that window through the slate. Also, there is no reason to believe that multi-slate environments should only be confined to reading activities. There are other types of activities for which a thin, pen-based slate system would also be desirable. One candidate is that of design practice, in which pen input is used extensively. Additional electronic functionality can make the multi-slate system suitable for sketching and other graphical design work.

As a next step, it will be important to extend our understanding of the implications of multi-slate reading through real-world deployments where actual reading tasks are performed. We would, for instance, like to know which multi-slate functions are most applicable to reading tasks in the real world. Also of interest to us is how multi-slate reading systems alter existing reading practice. Lastly, we would like to observe how users split their electronic reading requirements when both slates and PCs are available to read from in the environment.

9. CONCLUSION

In this article, we presented the design of a multi-slate system for supporting the complex reading activities of knowledge workers and students. The design of the reading system is informed by an extensive survey of the reading requirements for these users, which indicated that existing reading technologies each supported only a subset of these requirements. Our United Slates reading system combines positive traits of different reading technologies, which include the portability and physicality of e-book appliances, inking capabilities of tablet PCs, and the spatial layout possible with a tabletop computer. The electronic aspects of the system also simplify the process of working with large quantities of documents, reading in different venues, and interfacing with PCs—activities that tend to be difficult when reading with paper.

The ideas that make the multi-slate design viable for reading activities form the key contribution of our work. These ideas concern the model of mapping many documents onto a limited number of slates, a range of interactions that support flow of information between slates, and techniques that support reading activities that occur across different venues and using different slates. From these principles, we develop a range of reading tools to support the reading requirements we identified.

The evaluation of our prototype demonstrates the potential advantages of this system and provides useful feedback to further iterate on its design. Ultimately, we hope to refine this system even more in terms of its interface design, as well as improve on its technical robustness. At that point, the system could be more systematically evaluated in rich, real-world deployments. All of this, we believe, helps to open up a new category of reading appliances—ones that we hope build upon the affordances of paper while exploiting the power and flexibility of digital tools.

ACKNOWLEDGMENTS

The authors would like to thank Ken Hinckley for this suggestions for improving this article, as well as Richard Harper, Richard Banks, and Bill Buxton who provided a number of helpful ideas during the preliminary discussion about this area of investigaion.

REFERENCES

ADLER, A., GUJAR, A., HARRISON, B. L., O'HARA, K., AND SELLEN, A. 1998. A diary study of work-related reading: Design implications for digital reading devices. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'98).* 241–248.

ADLER, M. J. AND VAN DOREN, C. 1972. *How to Read a Book*. Simon and Schuster, New York, NY.

BEHLER, A. 2009. E-readers in action. *Am. Lib.* October.

BROXTON, M., LIFTON, J., AND PARADISO, J. A. 2006. Localization on the pushpin computing sensor network using spectral graph drawing and mesh relaxation. *SIGMOBILE Mob. Comput. Commun. Rev. 10*, 1, 1–12.

CHEN, N., GUIMBRETIERE, F., DIXON, M., LEWIS, C., AND AGRAWALA, M. 2008. Navigation techniques for dual-display e-book readers. In *Proceedings of the 26th Annual ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'08).* 1779–1788.

COCKBURN, A., GUTWIN, C., AND ALEXANDER, J. 2006. Faster document navigation with space-filling thumbnails. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'06).* 1–10.

DIXON, M., GUIMBRETIÈRE, F., AND CHEN, N. 2008. Optimal parameters for efficient crossing-based dialog boxes. In *Proceedings of the 26th Annual ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'08).* 1623–1632.

DOURISH, P., EDWARDS, W. K., LAMARCA, A., AND SALISBURY, M. 1999. Presto: An experimental architecture for fluid interactive document spaces. *ACM Trans. Comput.-Hum. Interact. 6*, 2, 133–161.

GREENBERG, S. AND MARWOOD, D. 1994. Real time groupware as a distributed system. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'94)*. 207–217.

GRUDIN, J. 2001. Partitioning digital worlds: Focal and peripheral awareness in multiple monitor use. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'01)*. 458–465.

HINCKLEY, K., DIXON, M., SARIN, R., GUIMBRETIERE, F., AND BALAKRISHNAN, R. 2009. Codex: A dual screen tablet Computer. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*.

HINCKLEY, K., RAMOS, G., GUIMBRETIERE, F., BAUDISCH, P., AND SMITH, M. 2004. Stitching: Pen gestures that span multiple displays. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI'04)*. 23–31.

HINCKLEY, K., ZHAO, S., SARIN, R., BAUDISCH, P., CUTRELL, E., SHILMAN, M., AND TAN, D. 2007. InkSeine: In Situ search for active note taking. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*. 251–260.

HOFFMANN, R., BAUDISCH, P., AND WELD, D. S. 2008. Evaluating visual cues for window switching on large screens. In *Proceedings of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*. 929–938.

HOLMAN, D., VERTEGAAL, R., ALTOSAAR, M., TROJE, N., AND JOHNS, D. 2005. Paper windows: Interaction techniques for digital paper. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'05)*. 591–599.

JOHANSON, B., HUTCHINS, G., WINOGRAD, T., AND STONE, M. 2002. PointRight: Experience with flexible input redirection in interactive workspaces. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST'02)*. 227.

LIAO, C., GUIMBRETIÈRE, F., HINCKLEY, K., AND HOLLAN, J. 2008. Papiercraft: A gesture-based command system for interactive paper. *ACM Trans. Comput.-Hum. Interact. 14*, 4, 1–27.

MACKAY, W. AND PAGANI, D. 1994. Video mosaic: Laying out time in a physical space. In *Proceedings of the 2nd ACM International Conference on Multimedia*. 165–172.

MALONE, T. W. 1983. How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst. 1*, 1, 99–112.

MARSHALL, C. 1997. Annotation: From paper books to the digital library. In *Proceedings of the 2nd ACM International Conference on Digital Libraries (DL'97)*. 131–140.

MARSHALL, C. 2005. Reading and interactivity in the digital library: Creating an experience that transcends paper. In *Digital Library Development: The View from Kanazawa*, D. Marcum and G. George, Eds. Libraries Unlimited, Westport, CT.

MARSHALL, C. C. AND BLY, S. 2005. Turning the page on navigation. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*. 225–234.

MARSHALL, C. C., PRICE, M. N., GOLOVCHINSKY, G., AND SCHILIT, B. N. 1999. Introducing a digital library reading appliance into a reading group. In *Proceedings of the 4th ACM Conference on Digital Libraries (DL'99)*. 77–84.

MARSHALL, C. C., PRICE, M. N., GOLOVCHINSKY, G., AND SCHILIT, B. N. 2001. Designing e-books for legal research. In *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'01)*. 41–48.

MARSHALL, C. C. AND RUOTOLO, C. 2002. Reading-in-the-small: A study of reading on small form factor devices. In *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'02)*. 56–64.

MARSHALL, C. C. AND SHIPMAN, F. M. 1997. Spatial hypertext and the practice of information triage. In *Proceedings of the 8th ACM Conference on Hypertext (HYPERTEXT'97)*. 124–133.

MERRILL, D., KALANITHI, J., AND MAES, P. 2007. Siftables: Towards sensor network user interfaces. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction (TEI'07)*. 75–78.

MICROSOFT. 2010. Welcome to Microsoft Surface. http://www.microsoft.com/surface/.

MILLER, R.C. AND MYERS, B. A. 1999. Synchronizing clipboards of multiple computers. In *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology (UIST'99)*. 65–66.

MORRIS, M. R., BRUSH, A. J. B., AND MEYERS, B. R. 2007. Reading revisted: Evaluating the usability of digital display surfaces for active reading tasks. In *Proceedings of the 2nd Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. 79–86.

MYERS, B. A. 2001. Using handhelds and PCs together. *Commun. ACM 44*, 11, 34–41.

O'HARA, K. AND SELLEN, A. 1997. A comparison of reading paper and on-line documents. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'97)*. 335–342.

O'HARA, K., TAYLOR, A. S., NEWMAN, W. M., AND SELLEN, A. 2002. Understanding the materiality of writing from multiple sources. *Int. J. Hum.-Comput. Stud. 56*, 3, 269–305.

PEARSON, J. AND BUCHANAN, G. 2011. CloudBooks: An infrastructure for reading on multiple devices. In *Proceedings of the 15th International Conference on Theory and Practice of Digital Libraries: Research and Advanced Technology for Digital Libraries (TPDL'11)*. 488–492.

PEARSON, J., BUCHANAN, G., AND THIMBLEBY, H. 2009. Improving annotations in digital documents. In *Proceedings of the 13th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'09)*. 429–432.

PEARSON, J., BUCHANAN, G., AND THIMBLEBY, H. 2010. HCI design principles for ereaders. In *Proceedings of the 3rd Workshop on Research Advances in Large Digital Book Repositories and Complementary Media*. 15–24.

PENG, C., SHEN, G., ZHANG, Y., LI, Y., AND TAN, K. 2007. BeepBeep: A high accuracy acoustic ranging system using COTS mobile devices. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys'07)*. 1–14.

PRICE, M. N., GOLOVCHINSKY, G., AND SCHILIT, B. N. 1998. Linking by inking. In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*. 30–39.

PUGH, A. K. 1978. *Silent Reading: An Introduction to Its Study and Teaching*. Heinemann Educational, London, U.K.

REKIMOTO, J. 1997. Pick-and-drop: A direct manipulation technique for multiple computer environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST'97)*. 31–39.

REKIMOTO, J. 2004. SyncTap: Synchronous user operation for spontaneous network connection. *Personal Ubiquitous Comput. 8*, 2, 126–134.

REKIMOTO, J. AND SAITOH, M. 1999. Augmented surfaces: A spatially continuous work space for hybrid computing environments. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'99)*. 378–385.

RHEINFRANK, J. 1995. A conversation with Don Norman. *Interactions 2*, 2, 47–55.

RICHARDSON, T., STAFFORD-FRASER, Q., WOOD, K. R., AND HOPPER, A. 1998. Virtual network computing. *IEEE Internet Comput. 2*, 1, 33–38.

SCHILIT, B. N., GOLOVCHINSKY, G., AND PRICE, M. N. 1998. Beyond paper: Supporting active reading with free form digital ink annotations. In *Proceedings of the ACM SIGCHI Conference on Ituman Factors in Computing Systems (CHI'98)*. 249–256.

SCHILIT, B. N. AND SENGUPTA, U. 2004. Device ensembles. *Computer 37*, 12, 56–64.

SELLEN, A. J. AND HARPER, R. H. R. 2003. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA.

TANDLER, P., PRANTE, T., MÜLLER-TOMFELDE, C., STREITZ, N., AND STEINMETZ, R. 2001. Connectables: Dynamic coupling of displays for the flexible creation of shared workspaces. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST'01)*. 11–20.

TASHMAN, C. S. AND EDWARDS, W. K. 2011a. Active reading and its discontents: The situations, problems and ideas of readers. In *Proceedings of the ACM SIGCHI Annual Conference on Human Factors in Computing Systems (CHI'11)*. 2927–2936.

TASHMAN, C. S. AND EDWARDS, W. K. 2011b. LiquidText: A flexible, multitouch environment to support active reading. In *Proceedings of the ACM SIGCHI Annual Conference on Human Factors in Computing Systems (CHI'11)*. 3285–3294.

THAYER, A., LEE, C. P., HWANG, L. H., SALES, H., SEN, P., AND DALAL, N. 2011. The imposition and superimposition of digital reading technology: The academic potential of e-readers. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*. 2917–2926.

THE TRUSTEES OF PRINCETON UNIVERSITY. 2010. Princeton e-reader pilot, final report. October 27.

WELLNER, P. 1993. Interacting with paper on the DigitalDesk. *Commun. ACM 36*, 7, 87–96.

WILSON, R. AND LANDONI, M. 2003. Evaluating the usability of portable electronic books. In *Proceedings of the ACM Symposium on Applied Computing (SAC'03)*. 564–568.

WOLIN, A., EOFF, B., AND HAMMOND, T. 2008. ShortStraw: A simple and effective corner finder for polylines. In *Proceedings of the EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*. 33–40.

YOUNG, J. R. 2009. 6 lessons One campus learned about e-textbooks. *The Chronicle of Higher Education*, June 4, sec. Technology.

Received August 2011; revised February 2012; accepted April 2012