

PapierCraft: A Gesture-Based Command System for Interactive Paper

CHUNYUAN LIAO, FRANÇOIS GUIMBRETIERE
Department of Computer Science, University of Maryland

KEN HINCKLEY
Microsoft Research

JIM HOLLAN
Department of Cognitive Science, University of California, San Diego

Paper persists as an integral component of *active reading* and other knowledge-worker tasks because it provides ease of use unmatched by digital alternatives. Paper documents are light to carry, easy to annotate, rapid to navigate, flexible to manipulate, and robust to use in varied environments. Interactions with paper documents create rich webs of annotation, cross reference, and spatial organization. Unfortunately, the resulting webs are confined to the physical world of paper and, as they accumulate, become increasingly difficult to store, search, and access. XLibris [Schilit, et al., 1998] and similar systems address these difficulties by simulating paper with tablet PCs. While this approach is promising, it suffers not only from limitations of current tablet computers (e.g., limited screen space) but also from loss of invaluable paper affordances.

In this paper, we describe PapierCraft, a gesture-based command system that allows users to manipulate digital documents using paper printouts as proxies. Using an Anoto [Anoto, 2002] digital pen, users can draw command gestures on paper to tag a paragraph, email a selected area, copy selections to a notepad, or create links to related documents. Upon pen synchronization, PapierCraft executes the commands and presents the results in a digital document viewer. Users can then search the tagged information and navigate the web of annotated digital documents resulting from interactions with the paper proxies. PapierCraft also supports real time interactions across mix-media, for example, letting users copy information from paper to a Tablet PC screen. This paper presents the design and implementation of the PapierCraft system and describes user feedback from initial use.

Categories and Subject Descriptors: H5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Graphical user interfaces*

General Terms: Management, Documentation, Design, Human Factors

Additional Key Words and Phrases: Paper interfaces, ubiquitous computing, marking interfaces, tablet computers, paper-augmented digital documents, gesture-based interfaces, distributed systems

1. INTRODUCTION

In the age of personal computers, PDAs, tablets, smart phones, and a growing variety of ubiquitous computing devices, knowledge workers strongly prefer paper for many tasks [Sellen and Harper, 2001]. Paper is comfortable to read, easy to annotate, light to carry,

This research was supported in part by Microsoft Research for the Microsoft Center for Interaction Design and Visualization at the University of Maryland and the National Science Foundation under Grant #0414699 and #0447703. We thank Anoto for technical support, Logitech for streaming software and HP for hardware donation. Early partial results of this work appeared as a conference Tech Note at UIST 05 [Liao, et al., 2005].

Authors' addresses: Chunyuan Liao and François Guimbretière, Department of Computer Science, University of Maryland MD 20742, U.S.A; Ken Hinckly, Microsoft Research, One Microsoft Way, Redmond, WA 98052, U.S.A; Jim Hollan, Department of Cognitive Science, University of California, San Diego, La Jolla, CA 92093. Permission to make digital/hard copy of part of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date of appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

quick to access, and simple to use. Paper's multiple strengths highlight the weakness of current digital alternatives.

While digital devices will continue to improve, paper offers several distinctive advantages. The key affordances of paper are well adapted to the combination of reading, understanding, critical thinking, annotating, summarizing, elaborating, linking, writing, organizing, and sharing that comprise *active-reading* tasks [O'Hara and Sellen, 1997, Sellen and Harper, 2001]. For example, during active reading users annotate specific text regions with handwritten notes, circle and connect sections of documents, cut and paste information between documents, attach post-it notes, and spatially array documents by placing them side by side. These annotations and physical arrangements comprise a web of information. While the creation of this web is facilitated by the affordances of paper, the information involved is confined to the physical world of paper, making access to computer affordances (e.g., search, navigation, and remote sharing) a problem.

Three approaches have been proposed to address this problem. One is to augment paper with video projection and other digital facilities. For example, the Digital Desk [Wellner, 1993] projects interactive feedback directly onto paper and the A-book system [Mackay, et al., 2002] uses an overlaying PDA to interleave paper and computer-based interactions. Despite recent advances in projection and vision systems [Lee, et al., 2004] and the processing power of computers, these approaches limit the portability of the pen and paper interfaces.

An alternative approach attempts to transfer tasks now accomplished with paper into the digital world. The Xlibris [Schilit, et al., 1998] system, described as an "active reading machine", is one example. More recently, other systems, such as Microsoft OneNote [Microsoft, 2003], have adopted this approach to support notetaking and the ability to assemble information collages on tablet computers. Moving into the digital world facilitates capturing user interactions and makes possible "linking by inking" [Price, et al., 1998]. It also confronts limitations of current computers. For example, one can either look at small parts of many documents at once or flip between multiple windows. While multi-display configurations and large high resolution displays, such as the Stanford mural [Guimbretière, et al., 2001], aim to address this problem, they face portability and cost issues.

A third approach explored by the Paper-Augmented Digital Document (PADD) system [Guimbretiere, 2003] is to put the digital and paper world on a more equal footing so that changes made on one media can be easily transferred to the other. While using PADD, paper printouts and computers are simply two different ways to interact with

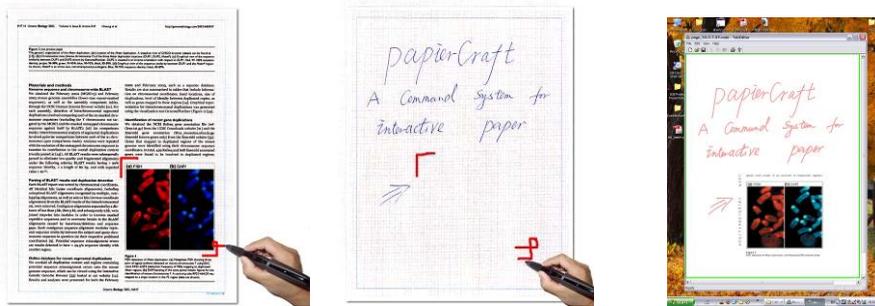


Figure 1. Copy/Paste Interaction in PapierCraft. A picture is copied from a printout (left), then pasted to a note (center). The result is shown on the PapierCraft viewer after pen synchronization (right). Marks are highlighted for clarity. Page taken from Open Access document <http://genomebiology.com/2003/4/8/R47> © 2003, Cheung et al. Used with permission.

documents during their life cycle. If paper affordances are needed, a document is printed on specially patterned paper. Using a digital pen, one can interact with paper as usual but the strokes are automatically recorded by the pen. Upon pen synchronization, the corresponding document is retrieved and data captured on paper are merged with the original digital pages. The resulting augmented document can then be edited, shared, archived, and printed out again to participate in further cycles between paper and digital versions.

The PADD system views paper printouts as *proxies* of digital documents, but it only captures ink strokes made on them. PapierCraft extends PADD by providing a gesture-based command system to enable users to manipulate digital content directly on paper. Using an Anoto digital pen, users can draw pen gestures to tag document content with keywords, email interesting parts to colleagues, copy and paste information from one paper document to another paper (or electronic) document (see Figure 1), create links between content found in two different paper documents, or juxtapose (“stitch”) two paper documents together by drawing a pen stroke across them. Upon pen synchronization with a host computer, PapierCraft retrieves the digital files corresponding to the paper documents using the PADD infrastructure, interprets and executes all gestures on the associated files, and presents the results in a digital document browser. All information gathered in the paper world becomes digitally accessible, allowing users to easily search, navigate, and share the resulting web of information with others. PapierCraft combines the advantages of paper interaction with useful digital facilities of computer-based annotation systems like Xlibris and OneNote.

In this paper, we present the design and implementation of the PapierCraft system. The core of the system is a new pen-based marking interface designed specifically for passive media such as paper. In its basic form, the PapierCraft interface does not require

any *active* feedback beyond inking of paper. We describe the design of pen gestures and error correction mechanisms needed to meet the challenges of limited feedback in paper-based active reading applications, detail command execution, post-processing facilities and real time interactions, report initial user experience, and discuss future possible extension to the system.

2. RELATED WORK

Three main areas of work have influenced the development of PapierCraft: previous approaches to bridging the paper-computer gap, recent research on marking-based interfaces, and new mechanisms for supporting distributed interaction across devices.

2.1 Bridging the Paper-Computer Gap

Earlier systems that attempted to bridge the gap between paper and digital worlds can be classified into five broad categories based on the functional roles played by computer and paper.

One category, exemplified by Xlibris [Schilit, et al., 1998], Dynamite [Wilcox, et al., 1997], OneNote [Microsoft, 2003], and Screen Crayons [Olsen, et al., 2004], attempts to remove the gap by eliminating paper and restricting interaction exclusively to digital media. These systems mainly employ tablet computers for annotating and copying information from one digital document to another. While this allows ready access to digital affordances (e.g., search, recording of interactions, and “linking by inking” [Price, et al., 1998]), some paper affordances, like flexible display size, are either missing or provided in forms that are much less effective than their paper-based equivalents. In addition, the limited screen real estate of tablet displays makes navigation difficult and viewing multiple documents simultaneously is clumsy.

A second category of systems, such as the Digital Desk [Wellner, 1993], Ariel [Mackay, et al., 1995], Video Mosaic [Mackay and Pagani, 1994] and EnhancedDesk [Koike, et al., 2000] bring digital functionality to paper by tightly coupling paper and computers. The early Digital Desk system was a seminal prototype in which digital feedback was directly projected onto paper documents. This approach, while powerful, since the computer can provide direct feedback during paper interactions, requires non-portable combinations of computers, projectors, and cameras. A-Book [Mackay, et al., 2002] addresses this issue to a degree by using a PDA to provide digital overlay for the paper document area beneath the PDA. This simplifies the system setup, and requires a simpler tracking technology, but the overlap requirement still constrains its use in certain settings, such as users working in the field without a stable desk to support the devices.

Systems in the third category, including Intelligent Paper [Dymetman and Copperman, 1998], PaperLink [Arai, et al., 1997], Paper++ [Norrie and Signer, 2003] and Books-with-Voices [Klemmer, et al., 2003], further loosen the paper-computer coupling by using paper and screens as separate displays. These systems offer a balance between paper affordance (e.g., ease of navigation) and access to digital medial, yet their flexibility is limited. Most interactions are “pre-programmed” into the content as link hot spots. Another interesting example is the IBM CrossPad [IBM, 1998]. While it captures handwriting on any paper sheet, using a portable graphics tablet placed beneath a paper notepad, it presents several difficulties: users need to specifically indicate that a page has been turned; only one page per tablet is “active”, so it is difficult to spread out multiple pages on a table for cross-page interactions.

In contrast to the above systems, members of a fourth category of systems, including Xax [Johnson, et al., 1993], PaperPDA [Heiner, et al., 1999], and the Anoto Pen system, use pen and paper as input devices for form filling. Although this fits with conventional pen-paper usage, the systems offer an asymmetric view of the paper-computer relationship, one biased towards entry of structured information. Input must be in pre-printed areas and only a limited set of actions can result from the input. This approach is too restrictive to support active reading tasks that, by their very nature, take unexpected twists as the user works.

A fifth approach to bridging the paper-computer gap places paper and digital documents on a more equal footing. Paper Augmented Digital Documents (PADD) [Guimbretiere, 2003] allow paper-based annotation of digital documents without any restriction on input structure. To accomplish this, the PADD system relies on the tracking capability of the Anoto system. Based on a special pattern in the paper background (see Figure 2) the Anoto system can not only track pen movements but also identify the specific piece of paper involved. As a result, it is possible to create a one-to-one correspondence between pages of digital documents and paper printouts of them.

Using PADD, users can easily capture annotations made on printouts and associate them with the corresponding digital documents at pen synchronization. But PADD does not support specification of commands to be performed on the digital versions. PapierCraft

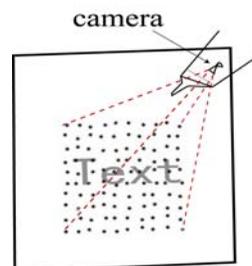


Figure 2. Working mechanism of the Anoto digital pen: The pen uses a built-in camera to recognize a visually unobtrusive dot pattern in the paper background to trace the pen tip's position within the page coordinates and the unique ID of that piece of paper.

addresses this limitation. It uses the PADD infrastructure to establish paper as a proxy for interaction with digital documents and extends the infrastructure to provide a gesture-based command system.

Several paper based command systems have been proposed in the past. The I-book system [Cunha, 2004], a newer version of A-book [Mackay, et al., 2002] using Anoto facilities, supports gestures to select specific document segments and to specify commands by “ticking” buttons previously drawn on paper. The Leap Frog’s Fly pen system [LeapFrog, 2005] combines pre-printed interfaces and user-drawn buttons. PapierCraft’s paper-based command system allows users to execute commands in context without the need for pre-printed buttons or for user-drawn interface elements.

PapierCraft is also related to other tangible interfaces such as NISMap [Cohen and McGee, 2004] and Papier-Mâché [Klemmer, et al., 2001]. NISMap uses the Anoto pen infrastructure, but is focused on group-coordinated annotation of a shared map, while PapierCraft supports document content manipulation for active reading and personal knowledge work. Papier-Mâché is a toolkit aimed at abstracting low-level tangible input (e.g., symbols on paper) and facilitating application development. It does not provide a paper-based command mechanism.

PapierCraft explores an interaction model in which users work with a pen and paper but are still linked to and able to control digital facilities. This approach promises to be especially effective and natural for active reading and early design tasks [Bailey and Konstan, 2003], where pen and paper are still very popular due to their flexibility.

2.2 Marking-Based Interfaces

Pens are natural devices for marking-based interfaces. Pen strokes and gestures can be used to design a visual language for informal sketches and commands. Among the extensive literature on pen-based interaction MATE [Hardock, et al., 1993], a pen-based document proofreading application is closely related to our work. Other systems like SILK [Landay and Myers, 1995], DENIM [Lin, et al., 2000] and DEMAIS [Bailey, et al., 2001] use pen-based interactions to support the early stages of GUI design, web page design, and multimedia applications. More recently, SketchREAD [Alvarado and Davis, 2004] and SimuSketch [Kara Burak and Stahovich, 2004] have demonstrated significant improvements in sketch recognition. All these systems though are designed for interaction with computer displays where active feedback can be provided. In contrast, PapierCraft is designed to support paper and pen interactions without any active feedback

other than the physical pen marks left on paper. Special interface design is required to meet the associated challenges.

2.3 Distributed Interaction

One motivating design model for PapierCraft is to conceptualize each piece of paper as a separate “mobile device” displaying information and capturing users’ strokes. From this perspective, the implementation of “copy and paste across pages” facilities for PapierCraft faces technical challenges similar to those confronted in developing distributed interaction techniques to support “copy and paste between mobile devices” such as *Pick-and-Drop* [Rekimoto, 1997] and *Stitching* [Hinckley, et al., 2004]. PapierCraft demonstrates how similar forms of interaction can be extended to tangible paper interfaces while maintaining the corresponding digital representations.

PapierCraft also enables novel forms of distributed interaction with paper. With the recent availability of digital pens capable of real-time streaming, it is now possible to handle strokes immediately. Thus, users are able to work on both computers and paper simultaneously, with the display surfaces of the different media functioning as united members of a connected work space. In this respect, PapierCraft bears similarities with Rekimoto’s Augmented Surfaces [Rekimoto and Saitoh, 1999]. A fundamental difference though is that the components of the PapierCraft system are loosely-coupled. Pen and paper interaction can continue to function without computers being jointly located or continuously available. Of course, the static nature of paper hinders it from reflecting the digital document changes in real time and thus constrains the continuity of the work space.

3. DESIGN GOALS

As mentioned above, systems have made various tradeoffs in their efforts to bridge the gap between the paper and digital media. Frequently they have sacrificed paper affordances when introducing digital functions. The primary goal of PapierCraft is to bridge this gap while maintaining both digital and paper affordances. While the PADD infrastructure supports simple annotation, PapierCraft provides a command system that enables a richer set of operations. In developing the system we had the following design goals:

1. **Respect current paper practices.** This was the main design goal. The general principle is that users should be able to do whatever they could before with no new constraints but also be able, with minimal extra effort, to access new facilities enabled by a paper-digital linkage. We have focused on several common paper-based

practices: annotating sections of documents, excerpting passages, relating documents, and creating information collages.

2. **Provide flexible, simple, and reliable commands.** The command system should be as flexible as possible to accommodate the informal and unstructured nature of conventional pen-paper interactions, and impose minimal restrictions on the shape or location of command marks. Further, since the marks may be the only immediate feedback provided, the command system needs to be simple and reliable in order to minimize errors.
3. **Ensure commands are human-readable.** Although the gesture commands are intended for computer interpretation, it is equally important that they also be easily understood by people. For example, the execution scope of a paste-command should be clear so that the region will not be overlapped by subsequent notes or other commands. Human-readability is also helpful for information recovery in the event of pen failure (e.g. out-of-battery). This way, we can achieve graceful degradation [Yeh, et al., 2006], in that the digital pen can still be used at least as a normal ink pen working with paper in a conventional way, not stopping users from doing tasks like note-taking and proofreading.
4. **Design an extensible command system.** Although we initially focus on paper-only interactions, the system should be extensible to accommodate and support interactions across a rapidly changing interaction context in which paper, computers, and various ubiquitous wireless devices are involved.

4. DESIGN OF PAPIERCRAFT INTERFACE

The PapierCraft interface was designed for paper-based active reading tasks. Typically this involves writing on notepads and annotating multiple printed documents, often spread out on a desk. To support the portability, robustness and flexibility of this working style, we adopted Anoto digital pen technology [Anoto, 2002] to capture users' strokes on paper. Digital pens using Anoto technology, such as the Logitech io₂ pens [Logitech, 2004], are able to distinguish between different physical pages and provide the local coordinates and timestamps of each stroke made on a page. PapierCraft assumes that document printouts are managed by a system such as PADD to maintain correspondence between a printed page and its original digital version.

To provide rich interaction beyond annotation, we have designed and implemented a structured gesture command system customized for a pen and paper environment. Similar to tablet PC gesture systems such as Scriboli [Hinckley, et al., 2005], PapierCraft needs

to distinguish between annotations and commands, designate the scope of commands, select command types, and support command modification and subsequent execution.

4.1 Distinguishing Annotations and Command Gestures

Many solutions have been proposed to distinguish ink intended as content (*annotations*) from ink intended to be interpreted by the system (*commands*). Some systems use an implicit approach [Zelevnik and Miller, 2006] in which the computer automatically distinguishes the two. Other systems employ an explicit approach [Hinckley, et al., 2005, Li, et al., 2005] in which users indicate the current stroke type (e.g., by pressing a button). A mixed approach, in which the computer and the user collaborate to resolve ambiguous input, is also possible [Saund and Lank, 2003].

Given the limited feedback possible with a digital pen and paper, we concluded that both implicit and mixed approach would be problematic because they require immediate feedback for correction or disambiguation. Instead, PapierCraft takes an “explicit” approach, using a “gesture” button present on the pen or in the environment (e.g. a foot pedal). This approach requires only weak synchronization. A stroke is considered a gesture stroke as soon as the gesture button is pressed for some duration during stroke drawing.

With the gesture button down, a gesture can be drawn to specify a PapierCraft *command*. It consists of three basic components: a *command scope*, an *intermediate delimiter*, and a *command type*. Several commands, possibly on different paper sheets, can work in coalition to fulfill a task such as copying and pasting a section or hyperlinking two document sections.

4.2 Specifying Command Scope

The absence of real-time feedback influenced the design of scope selection. Consider the simple example of a copy-and-paste operation. When this operation is performed on a computer (e.g., using a system such as OneNote), one first selects the object to be copied, often using a marquee selection with instant feedback of the area selected. Once the selection is complete, the paste location can be specified. As soon as the paste command is issued, visual feedback is immediately provided to show the result of the operation. Of course, on paper neither type of feedback can be provided. To address this issue, we require users to draw the intended scope of all commands. In the case of the copy operation for example, the scope identifies the region to be copied.

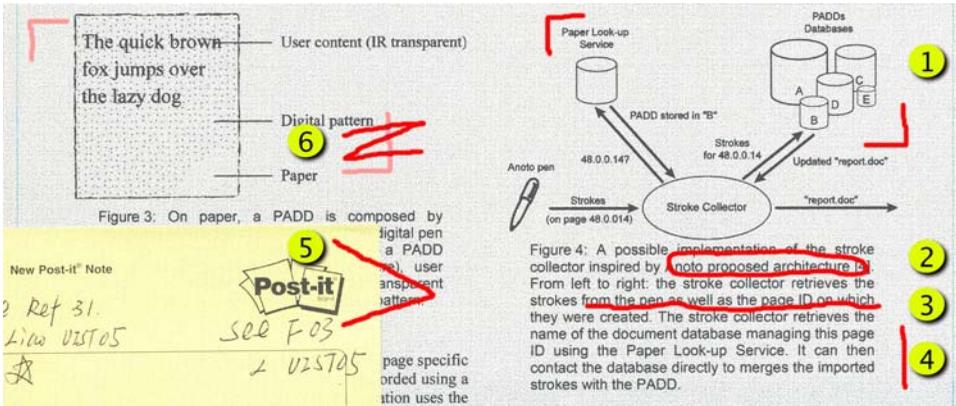


Figure 3. PapierCraft scopes and eraser command: (1) crop marks, (2) lasso, (3) underline, (4) margin bar, (5) stitching mark across two paper sheets, and (6) a "Z" eraser gesture deleting an unwanted crop mark. Its orientation could be arbitrary. Gestures are highlighted in red for clarity.

Inspired by typical marks found on manuscripts (Figure 3), PapierCraft offers five types of scope selectors. One can select content by "cropping" a graphics region, underlining a passage of text, creating a margin bar to select several lines, or simply lassoing an arbitrary document area. We also offer a special "stitching" mark, a "V" reminiscent of the mark carpenters draw on two pieces of wood to specify their alignment. Much like Stitching [Hinckley, et al., 2004], which exploits cross-screen strokes to initialize a connection between displays. PapierCraft uses a sticking mark to indicate that the two adjacent paper sheets should be stitched together in the digital view. It can also be used to "pin" a small Post-It note on a document (Figure 3 item 5).

As pointed out above, during a copy-and-paste operation, a scope gesture is needed to indicate the size and position of the information to be pasted. Before synchronization, the drawn scope serves as a placeholder reminding the user that some information will appear at that location upon synchronization. At synchronization time, the scope is used to scale the content so that it fits into the scope (Figure 1, middle).

Operators such as copy-and-paste only require a simple scope. Other operators, such as the "copy with keyword" command that assigns a keyword to a specific area of a document, use a scope involving multiple parameters to select the target area as well as an assigned keyword (see Table 1). Note that currently, PapierCraft views scope as a region of space. Some systems (e.g., ScanScribe [Saund, et al., 2003]) extract information from images and other document elements to provide object-level selection. We expect to explore this in the future.

After the command scope strokes are made, a delimiter is required to distinguish them from subsequent command-type strokes. We have adopted the Pigtail approach proposed for Scriboli [Hinckley, et al., 2005]. Like Scriboli, our system considers all strokes

between the first gesture stroke and a gesture stroke containing a pigtail as part of the scope selection, and all gesture strokes after the pigtail as the command selection (see Figure 1 for an example). An advantage of the pigtail notation is that it is familiar to proofreaders and, as shown in Hinckley et al. [2005], yields performance comparable to presenting a “handle” (menu box) at the end of any stroke to specify part of the scope.

4.3 Specifying Command Types

Existing paper-based interfaces usually employ a dedicated region on paper for command selection, such as printed buttons in Anoto [Anoto, 2002] and Fly Pen [LeapFrog, 2005] and Guided Gestures in Paper PDA [Avrahami, et al., 2001]. Although easy to learn, this approach typically separates the selected command type and the associated scopes. And, the involved content on paper usually remains unchanged and gives no command feedback. As a result, users may lose clues about what actions have been applied to the document segment. Stroke-based command selection approaches (e.g., marking menus as in [Hardock, et al., 1993] and Sensiva [Sensiva, 2005]) address this issue. Command marks can be drawn right beside the scopes and thus do not require additional feedback.

Given our choice of the pigtail as a scope-command separator, it was natural to pick marking menus for specifying command types. However, single-level marking menus typically offer only 8 different commands. For a richer command set, one option was to use hierarchical marking menus [Hinckley, et al., 2005, Zhao and Balakrishnan, 2004] but there are a number of reasons to expect this to be problematic on paper. First, without any immediate feedback it would be difficult for users to discover and learn the different marks without extra references. Second, it might be even more difficult to remember the meaning of a mark drawn on paper after some time had passed. Finally, a reader cannot discern the temporal order of the marks, violating our design principle of human-readability.

We decided to use a mixed approach for command selection. The commands we expect to be most frequently used, including copy, paste, hyperlink source, and hyperlink target, are accessed directly from the eight cardinal directions of the marking menu. The full command set can also be accessed by simply writing down an unambiguous prefix of the command name immediately after the pigtail. For example, one can directly write the word “Paste” to take precedence over the mark direction which might otherwise be taken as “Copy”.

Writing command names allows for a larger command set. Since our system will recognize unambiguous prefixes, command prefixes can naturally assume the status of

shortcuts. This approach makes it easy for people to read the commands they issued on paper. While having to write a full command name has a cost, we expect that it will be accepted by users as it fits naturally into the context of proofreading or annotating a document during active reading. We also considered the possibility of character/word recognition errors, but given the small vocabulary in the present application we did not expect this to be a significant practical problem. A similar technique known as “mnemonic flicks” has been proposed for a TabletPC-based marking interface [Zelevnik and Miller, 2006]. PapierCraft demonstrates how this technique is particularly well-suited to working on paper.

4.4 An Example: Issuing a Copy/Paste Command in PapierCraft

To further explain the PapierCraft command interface, we describe a simple copy/paste interaction in PapierCraft (Figure 1). To copy an image, the user first indicates the area of the document to be copied. To do so, she presses the gesture button and draws cropping marks around the area of interest. Keeping the gesture button pressed down, she draws a pig tail followed by marking to the right (East). Note that the last stroke of the cropping mark, the pigtail, and the command mark can be issued as one continuous mark, making the interaction fluid. While the current prototype does not support multiple-source-copy, such functionality could be implemented using the existing PapierCraft infrastructure.

To paste, the user follows a similar pattern. First, she indicates the area where the paste will take place by drawing a crop mark and a pigtail. This mark serves as a placeholder such that the user will avoid placing notes or other content there. Then she selects the paste command by marking to the left (West). For multiple-strokes commands, the system is flexible in use of the gesture button. The user may decide to hold it down during the full operation or simply click it each time she is drawing a “command” stroke. During this procedure the user may, as described more fully below, abort any command or remove undesired strokes.

4.5 Command Execution

The above command example can be carried out without any computer nearby, achieving the same portability as normal pen and paper provides. Users can interact with the digital document wherever they want, with the gestures being cached within the digital pen. Later, when access to the infrastructure is available (perhaps when the user returns to her office), she can simply synchronize the pen (e.g., by plugging it into a cradle or communicating wirelessly) with a host computer. The PapierCraft system will download the stroke data, interpret and execute the series of commands made on paper documents.

The resulting modified digital documents will then be available in a viewer application (CraftViewer, Figure 1 , right). At this point, users can switch to the computer to continue their work. In addition to batch-mode operation, PapierCraft also supports real-time handling of strokes using a Bluetooth connection between pen and computer, enabling immediate command execution. Details of this form of interaction are discussed below.

4.6 Command Modification

From time to time, one may want to change, abort, or cancel a command operation. To address this issue, it is useful to first look at the way users currently deal with errors when using pen and paper. According to Khan [1993], few people use an eraser for correction while taking notes. Instead users simply rewrite or leave errors as they are. This fact reflects the informal and unstructured characteristic of note-taking tasks.

Following this practice and factoring in paper's static nature, PapierCraft adopts a "lazy" approach for command modification: one can implicitly abort the current in-progress operations at any time and reissue a command from scratch. For example, one may abort an on-going copy command by simply ignoring the current scope selection and reselecting a new scope.

One might also want to delete a command after finishing it. PapierCraft supports explicit cancellation with a special "Z" eraser gesture as shown in Figure 3 (item 6). Any gesture stroke is considered void if it is overlapped by an eraser gesture to an extent greater than a specifiable threshold (e.g., being crossed by all the three segments of the "Z"). This eraser gesture is the same as the standard "Scratch Out" gesture in Microsoft Tablet PC handwriting recognition pack [Microsoft, 2005], bearing high user familiarity.

Tasks	Command or Annotations on One Page	
Freeform annotation		
Tagging with reserved keywords		<p>or</p>
Tagging with arbitrary keywords		<p>or</p>
“Google” and “Email” on paper		<p>and</p>
<div style="display: flex; justify-content: space-around;"> Command on Page 1 Command on Page 2 </div>		
Excerption		
Excerption with keyword		
Excerption with command name		
Hyperlink		
Stitching		

Table 1. PapierCraft Operations. The strokes in blue are “Ink” and ones in red “Gesture”. The black circles indicate the starting point of strokes, and the strokes without such indicators can be drawn from either direction. For simplicity, we only use “crop marks” but other scopes are possible (see Figure 3).

5. SUPPORTING ACTIVE READING

A broad spectrum of reading activities are documented in a diary study of work-related reading [Adler, et al., 1998]. Activities range from “lightweight” activities such as skimming to more intensive active forms like summarizing and extracting important information. Of particular relevance to PapierCraft’s design, the study also documents:

Reading occurs more frequently in conjunction with writing than it does in isolation. Thus, it appears that writing (in a variety of forms) is an integral part of work-related reading. Designers need to seriously consider the importance of developing reading devices which also support the marking or writing of documents during the reading process.

Almost half of all the document activities which involved reading involved the concurrent use of multiple “display surfaces” for reading, or reading in conjunction with writing. This points to the need to consider how single display devices can support the range of cross-document activities people carry out. It also emphasizes the importance of considering the benefits of designing devices which are based around the use of multiple, digital displays. [Adler, et al., 1998]

In this section, we describe how the PapierCraft command system was designed to support active reading. This includes managing annotations, linking paper documents, working with a mix of paper and computer media, and handling errors. Our main goal is to demonstrate potential uses of the PapierCraft command system. Other commands could be implemented using the existing infrastructure.

5.1 Managing annotations

The simplest way to support active reading is to capture annotations. All annotations made on any PADD document are stored and automatically merged into the associated digital version of the document. Similar to the Xlibris Reader’s Notebook, all annotated sections can be gathered in a special “Digital Snippets” panel to facilitate review. While our current system only identifies one pen type, it would be easy, since each pen has a unique ID, to create a special category for snippets of information associated with a specific pen. This would support the common practice of using highlighters of different colors to identify different types of information or grouping together annotations made by specific individuals.

5.1.1 Tagging segments of text

Using a different pen for different types of annotations has drawbacks. First, it requires users to carry several pens around and constantly juggle between them. Second, it offers only limited flexibility. Observing that many annotations are often tagged with text (or a symbol) identifying them, we have provided a simple tagging capability. One can select any specific region in a printout or notebook page, and issue a “tag” command using a

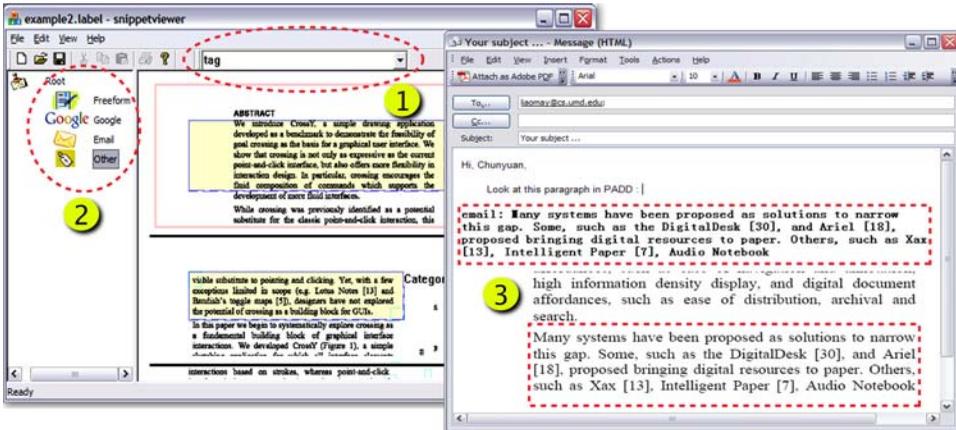


Figure 4. The interface to browse, search and manage snippets: (1) keyword filter, allowing users to type in keywords or to choose one from the list to select specific snippets. (2) a list of the valid categories. (3) an outlook email interface brought by clicking an “email” snippet. The top red rectangle indicates the text extracted from the snippet, and below is an image cropped from the original document (it has been extended to include surrounding content for more context information).

pigtail, in one of the four secondary directions, and optional written keywords (Table 1). In the current implementation, pigtails towards “North West (NW)” and “South West (SW)” are designated to two user-configurable tags, “tag1” and “tag2” by default. Also, to provide additional flexibility, one can write any overriding tag words after these two pigtails.

Upon pen synchronization, one can review, manage, or search the annotated and tagged document segments, *snippets*, in an interface (see Figure 4). The snippets are organized into categories, and one can browse and review them under selected category nodes. In all cases each snippet is automatically linked to its location in the original document to help users rapidly access its context.

The other two secondary directions are used for advanced features. For example, a NE pigtail indicates text to be emailed to a specified address indicated by written words afterward, and a SE pigtail indicates text to be used as parameter to a search engine like Google (see Table 1). If the pen is connected through a real-time link, processing can be executed in real time (see section 5.3); otherwise, commands are executed after synchronization. For example, upon double clicking an email snippet an email program will appear with the snippet embedded in the mail body and the written email address in the recipient field. Note, besides “Google” and “Email”, other commands can be implemented with the same PapierCraft command structure.

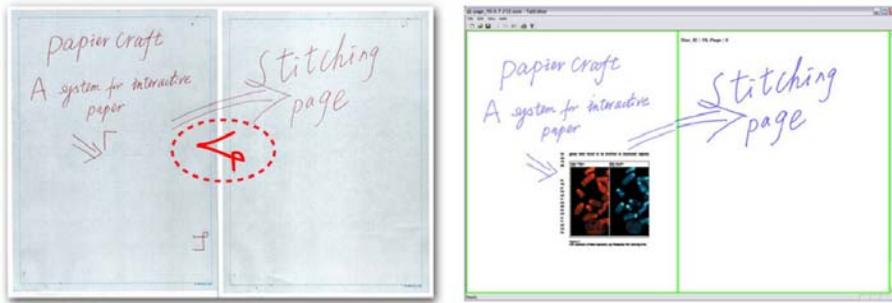


Figure 5. (left) “stitch” in the red circle to extend space. (right) The resulting digital document

5.2 Linking Paper Documents

As described earlier, PapierCraft makes it easy to copy information from a PADD document to one’s notes. It is important to note that the *results* of any earlier commands present in a copied area are also included. This makes it possible to transfer complex collages from one page to another. It also facilitates importing content from legacy documents not printed with the Anoto pattern. For example, an architect could use a piece of translucent vellum with the Anoto pattern to trace a portion of a floor plan from a document. The traced image can then serve as a source for later copy operations.

One issue we address is how to maintain appropriate contexts of the pasted snippets. We do this in two ways: (1) The system always copies a slightly larger area than was selected so that the original surrounding text and annotations can still be seen in the destination document. (2) The system automatically creates a hyperlink from the pasted region back to the associated section of the source document. This makes it possible, for example, to create thumbnails of several pages by simply copying each to small regions on a single piece of paper. Upon synchronization, one can then use the thumbnails as quick indexes to the associated pages.

5.2.1 Capturing Physical Collages

Arranging pieces of paper in a collage is an effective way of organizing information from multiple sources. PapierCraft allows users to create digital versions of paper collages by



Figure 6. Stitching a post-it note to a map. (left) Draw the stitching mark on paper. (middle) The resulting “anchor” region in the digital version. (right) Expanded “anchor” region for details. Figures are highlighted for clarity.

drawing stitch marks (see Figure 5) across page boundaries. This is similar to the Stitching system that allows stitching across Tablet PC displays [Hinckley, et al., 2004]. After synchronization, the content of the two pages are shown as a single page in the viewer interface. Similarly, one can effectively pin a smaller piece of paper (such as a Post-it note) onto a larger one (Figure 6). The corresponding digital version includes an “anchor region” around the position of the stitching mark vertex. Clicking on an anchor in PapierCraft viewer (Figure 1) allows users to review an attached Post-it note in a separate window.

5.2.2 Explicitly Linking Paper Documents

While our system offers many ways to create implicit links, it is sometime useful to be able to create explicit link between documents. For example one might want to create a link between a reference in a text and the corresponding paper or between a graph and the corresponding data table. PapierCraft supports this practice through the notion of “hot spots”. One can select a region of a document with, for example, a cropping mark and draw a gesture “hyperlink start” followed by another gesture “hyperlink end” on a separate document. The region “hot spot” will be highlighted in the PapierCraft viewer and users can double-click it to open the linked document.

5.3 Working with Paper-Computer Mixed Media

So far we have focused on paper-only operations, but in a typical work environment, paper and digital media coexist and people often use both simultaneously. For example, users may review a paper document and search for a reference or take notes on a PC. While early digital pens only supported synchronization through USB connections, the current generation of pens are able to stream strokes in near real-time through Bluetooth connections. PapierCraft provides support for a streaming interface. This permits users to execute computer commands via gestures on paper. For example, we support a “Google” search on paper. One can select any region in a paper document and issue a pigtail command “google”. The underlying text of the specified region is extracted and sent as a search request to a web browser running on an associated host computer. In this way, users do not have to drop the paper document and switch to a computer to type in keywords, but can immediately review the search results on a screen.

5.4 Error Handling

While users are interacting with paper documents, PapierCraft does not provide any feedback other than ink strokes, so error handling is postponed until pen synchronization. When a user synchronizes her pen, a potentially large number of commands need to be

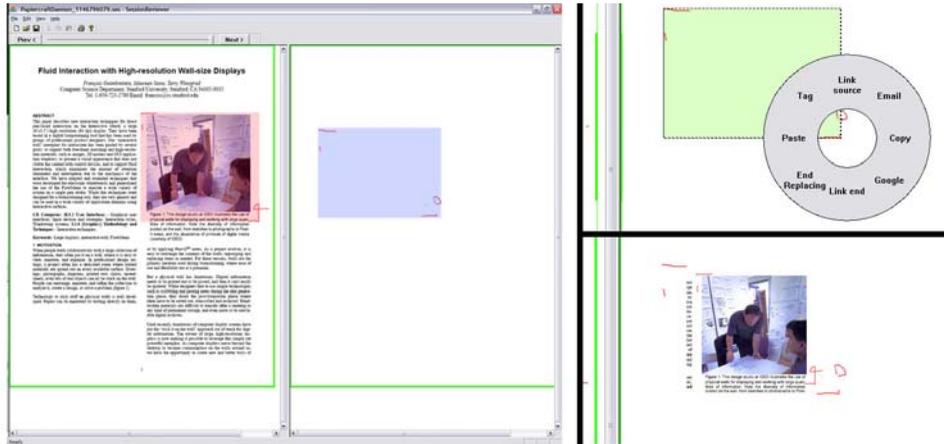


Figure 7. (left) An example of the interface for session review and error correction. A user tried to copy a picture in printout (highlighted in red) to an area in another note sheet (highlighted in blue), but failed due to unrecognized strokes for “paste”. (top right) The user then selects the incorrect strokes and replaces them with a marquee selection in the target area, followed by a pie menu selection for command “paste”. (bottom right) After re-execution, a new version with the desired picture in the note is created.

recognized and executed. Instead of alerting users during synchronization, our system adopts a best effort strategy for command recognition and does not report any errors during this initial phase. As users read their notes, they may discover that errors have been made (for example, that a “Paste” command was not executed). The system provides a simple recovering mechanism to rapidly correct such interpretation errors.

Our approach is based on the observation that while command recognition might be error prone, each stroke is accurately time stamped. Using this information, it is very easy to recover a given stroke’s temporal context including the strokes that immediately precede it. When an error is found, simply clicking on strokes that were not correctly interpreted will automatically launch a Session Reviewer (Figure 7) that allows users to play back the strokes in the context in which they were made. This technique is similar to the Time-Machine Computing approach proposed by Rekimoto [1999] to allow users to restore an earlier context. Once the source of the problem is identified, a user can correct it using the digital interface without returning to paper. For some incorrect gestures, it may be faster to re-issue the commands using standard GUI facilities. Figure 7 illustrates the correction procedure when a “copy” command is mistaken for another gesture. One can select the two strokes and easily replace them with a correct “Copy” command by using PapierCraft gestures or, in traditional way, using a mouse marquee selection and popup menu. In order to retain the correct command sequence, the system will automatically assign the new command with the timestamp of the replaced one. Finally, the user can have the system re-execute the updated command stream for the correct result.

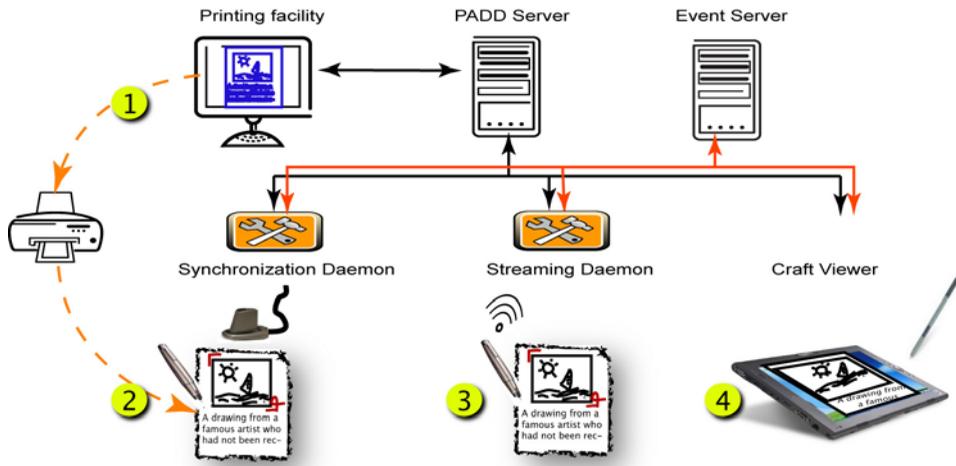


Figure 8. PapierCraft Architecture. (1) A digital document is printed via a PADD printer. The hardcopy become an alternative working media. (2) In “batch” mode, the pen synchronizes with the system via a cradle. (3) In “streaming” mode, the pen interacts with the system in real-time via Bluetooth. (4) A tablet PC can be used directly manipulate the documents through CraftViewer.

6. SYSTEM DESIGN AND IMPLEMENTATION

In this section, we describe details of the design and implementation of PapierCraft. We first present the system architecture and then discuss the command execution procedure.

6.1 System Architecture

The PapierCraft architecture is designed to provide an integrated view of documents and to coordinate and support interaction with them in both digital and paper versions. This leads to the somewhat novel view of a paper page as a simple display device capable of capturing pen strokes. Based on this, it was natural to adopt an architecture similar to Pick-and-Drop [Rekimoto, 1997] and Stitching [Hinckley, et al., 2004]. Similar to these systems, a central server coordinates distributed clients to accomplish users’ operations across devices. For generality, our system supports interactions between paper and paper, between paper and a computer, as well as between two computers.

Figure 8 illustrates the PapierCraft architecture. There are three basic components: *PADD Server*, *Event Server*, and *PapierCraft clients*. The PADD server is the bridge between digital documents and paper documents. It is a repository for digital versions of all printed documents and paper notes and is responsible for maintaining mappings between each digital document page and its paper counterparts. Using the PADD system, the digital version and digital-to-physical mapping information are automatically captured and submitted by the infrastructure when one prints a file through a PADD virtual printer driver. Currently, we support PDF files printed using Adobe Acrobat.

The Event server and PapierCraft clients collaborate to execute the gesture commands distributed on multiple devices. PapierCraft clients are responsible for processing strokes captured either on paper or on a tablet PC in case of mixed media operations, and function on behalf of various display surfaces. Each client can reside on a different host computer. These can include the host computer on which the pen is synchronized, the computer receiving the stroke streams for real-time operation, the computer running the CraftViewer document viewer. The Event Server monitors the PapierCraft client interactions and coordinates them based on pre-defined operation patterns. The architecture is similar to network event monitoring systems like GEM [Mansouri-Samani and Sloman, 1997] and SIENA [Carzaniga, et al., 2001].

6.2 Gesture Command Execution

Gesture commands can be processed in real time or in batch mode. Here we focus on a description of batch mode processing. The procedure for real-time execution is similar except that the strokes are sent without delay. Processing consists of three phases: (1) a *stroke-processing phase* during which pen strokes are uploaded from the pen and interpreted by a host computer, (2) an *execution phase* during which the host computers are synchronized together by the Event Server and the resulting commands executed, and (3) a *display phase* during which client displays are notified of modifications to the documents they are managing. Figure 9 illustrates the phases of execution, and the sections below detail the process.

6.2.1 Stroke processing

Once a pen is plugged in, a PapierCraft client is invoked on the host computer and receives all the strokes captured by the pen (step 1 in Figure 9). After importing the strokes, the client downloads the latest versions of the corresponding digital documents from a central PADD database server, and then creates a Page Proxy for each of the document pages. These proxies act as if they were independent devices interacting with the central Event Server. The client processes strokes in temporal order, labeling each stroke as a command or an annotation stroke. Annotation strokes are simply merged into the digital file. Gesture strokes are passed to a gesture recognition engine that recognizes commands from the stroke stream. Once a command is detected, a corresponding “Local Event” is sent to the event server (step 2, 3 in Figure 9). For example, in the case of copy, the local event includes a client ID, a timestamp, the command type, and the following command-specific parameters: the selected image, text extracted from the digital file, and surrounding annotation strokes, if any, available at that time.

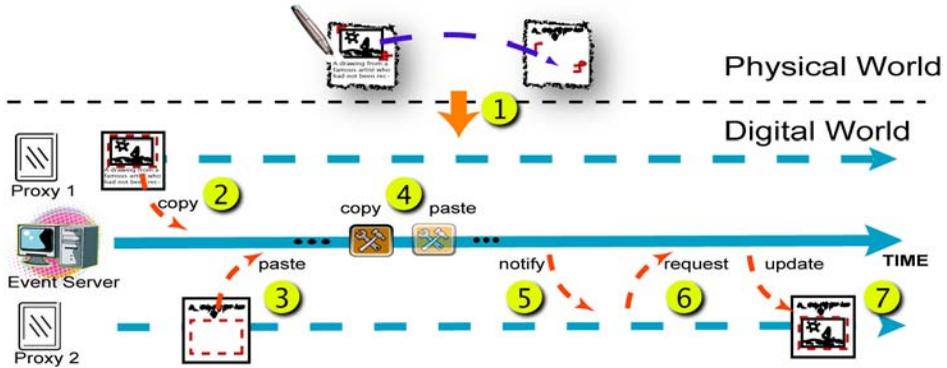


Figure 9. The Time-Space diagram for PapierCraft command execution exemplified with an excerpt operation. (1) pen synchronization starts, (2) proxy 1 sends a local event “copy” to the server, (3) proxy 2 sends event “paste”, (4) the event server checks the received events stream, (5) a distributed “copy”/“paste” operation is detected and a “notify” event is sent back to the “pasting” target, proxy 2. (6) (7) after verification, proxy 2 requests data and updates the digital notes

Parsing gesture strokes is a crucial issue. Gestures before the pigtail crossing are taken as command scopes, and all strokes after the pigtail, if any, as written text commands, of which the number of strokes can vary. We use four rules to determine if a stroke S is the last stroke of the current command: (1) the following stroke is spatially further than a threshold away from S , (2) there is a gap longer than a threshold between their timestamps, (3) the user exists “gesture” mode after S , (4) S is the last stroke drawn on this page. As long as any of the four criteria is satisfied, S is taken as the ending stroke and all additional strokes, if any, are submitted to the handwriting recognizer. These heuristic rules have worked fine in our preliminary tests, but formal evaluation is planned as part of future work.

6.2.2 Event synchronization and command execution

Event synchronization is accomplished by the event server. It maintains a central Event Cache for Local Events coming from various proxies during a session (step 4 in Figure 9). Such a cache is necessary because events may be submitted by the clients out of order of their actual occurrences. Currently, we do not distinguish different pens. Instead, we temporally align all events in the cache and handle them as a single thread. This enables interaction across different pens which might be convenient for a single user. It could be a source of race conditions if a group of users were interacting with a given document. This single thread approach is a policy decision and other policies are possible. For example, each pen could have a private thread, allowing a separate “clipboard” for each pen.

It is important to note that associated events can be interspersed with events of other types. For instance, as in the case of a digital copy and paste sequence, one can first “Copy”, then create a hyperlink, and finally “Paste”. This is equivalent to copy/paste followed by linking. This feature reflects the common “clipboard” semantics familiar to users [Miller and Myers, 1999].

6.2.3 Client notification

Upon execution of a command, the event server sends a “Global Event” notification to the client handling the corresponding page (step 5 in Figure 9). For instance, the client proxy might receive a “Paste data ready” event indicating that some data should be inserted into a region of a given page. If the notification is valid, the client will then send a request to the server and retrieve the pasted data (step 6 in Figure 9). Finally, it updates the opened digital file and uploads it to the PADD server (step 7 in Figure 9). If the client proxy is not responsive, the Global Event will be held until another client opens the digital file, which will receive this notification and update as described above. Currently, we assume, at any given time, there is only one such client opening a digital file, so there is no issue of version control.

After processing all strokes, the daemon opens the latest digital document in the viewer application, so the user can review their previous paper-based work, and continue the task with the digital version.

7. DISCUSSION

We have introduced PapierCraft, a gesture-based command system for interactive paper. PapierCraft’s main contribution is a general software architecture for recognizing and executing pen-based commands on paper documents. It is designed to respect paper practices and to provide a simple, flexible, and extensible command system. With PapierCraft, pen strokes become not only ways of annotating paper documents but mechanisms for issuing commands to interact with associated digital versions and applications. The ability to intersperse commands with free-form annotations is especially well-suited to active reading and related knowledge-work tasks. Here we report on insights gathered during the design process and an initial evaluation.

7.1 Informal user feedback

We have conducted an informal evaluation of PapierCraft with four colleagues not affiliated with this project. We demonstrated how to copy and paste and asked them to perform these operations using different scope selection mechanisms. Some interactions were performed using the pen and pedal configuration described above and some using

two pens (one always in annotation mode and one always in gesture mode; the latter pen used a red cartridge to distinguish it).

Overall participants' reactions were positive. They thought that our gesture set would be easy to remember, especially if proper mnemonic cues were provided (e.g., pointing out that the paste mark looks like a P). The option of writing down the command name was also popular. Participants were uncertain about the trade-off between using one pen (with a gesture button) or two pens (one for annotations and one for gestures). On the one hand, they liked the direct feedback provided by the use of two pens. Because gestures were shown in red, they were easy to identify. On the other hand, they also reported that it was cumbersome to switch pens and, of course, to carry two pens. As expected, the consensus was that one pen that provided both features would be the preferred option.

We also discussed with each participant the minimal feedback provided by the interface. Two main points emerged. First, participants thought the level of feedback required would strongly depend on the reliability of the system. If the system had a high rate of gesture recognition, participants felt that the current level of feedback would be acceptable. Second, two additional forms of feedback were suggested as potentially useful: a gesture mode indicator and confirmation that commands were recognized. These comments are not surprising. Our original design called for a small LED to light the area around the pen tip for gesture mode, and haptic feedback upon gesture recognition. Unfortunately the current Anoto digital pen is not programmable, so we were unable to explore that aspect of the design. This situation is changing as new products such as the Leapfrog [LeapFrog, 2005] pentop computer are introduced. Finally, we queried participants about the lack of content feedback during the paste operation. They thought that simply adding a note inside the paste area would suffice.

We are currently beginning a formal user evaluation of PapierCraft in the context of Active Reading, comparing it to both traditional pen-and-paper and Tablet PC interfaces.

7.2 Use of PapierCraft in ButterflyNet

We have collaborated with researchers developing ButterflyNet [Yeh, et al., 2006], a mobile multimedia capture and access system that integrates paper notes with digital photographs captured during field biology research. In ButterflyNet photos, notes, and other data that contain timestamps are automatically linked in an associated digital notebook. The first field trial of ButterflyNet used a variant of PapierCraft's gesture-based command system to support hotspot association. This enabled users to associate a digital photo with a specific area of a notebook page. To invoke a hotspot association, a

user captured a photo and then drew two brackets in her notebook. The associated section in the digital version of the notebook then contains the photograph resized to fit the bracketed area. We are planning to use hotspot gestures in the future in ButterflyNet to allow users to associate GPS and other sensor data to specific notebook locations in a manner similar to the current association of digital images.

7.3 Paper based interactions

PapierCraft places paper and digital documents on a more equal footing, offering a symmetrical role for paper and digital affordances. It has long been easy to print documents to transfer information from the digital to the tangible paper realm. More recently the PADD system [Guimbretiere, 2003] made it easier to transfer information captured on paper back to the digital realm. PapierCraft expands how paper interactions can contribute to the life cycle of digital documents by allowing users to execute commands on paper.

While our presentation so far has focused on single print-annotations-capture cycle (Note the cycle can also begin from a handwritten paper document, such as the note sheet in Figure 1), the system presented here could be easily extended to support multiple such cycles by simply implementing a print function inside our PapierCraft viewers. As explained by Sellen and Harper [2001] in the *Myth of the Paperless Office*, supporting such a cycle encourages the use of paper as a temporary media only used when paper affordances are needed. While this model has proven to be successful in creating paperless company [Sellen and Harper, 2001], this practice may not be permitted in cases where paper documents are taken as official history records, such as biology laboratory notebooks and legal documents.

Moreover, PapierCraft demonstrates a unifying infrastructure for mixed digital/paper interactions. Because it can be used for both paper-only (batch) and mixed paper/digital interactions (with a real-time pen), it supports a variety of interaction styles proposed in earlier systems such as A-Book [Mackay, et al., 2002] and Caméléon [Mackay, et al., 1998], which rely on a nearby PDA or computer for real time input processing and feedback.

7.4 Supporting Active Reading

Active reading provides many examples of what Schön terms “reflective conversations” with media. [Schön, 1983, Schön and Wiggins, 1992]. Annotations serve numerous conversational and reflective purposes. They can, for example, function as navigational aids for subsequent access, identification of important sections, reminders to oneself (e.g.,

to look up a reference), or comments to a student on an assignment or to collaborating authors on a paper draft. One hallmark of active reading is the variety of thoughts and intentions that are generated during the process. Annotations help harvest those thoughts and intentions. PapierCraft provides novel mechanisms for not only harvesting intentions but helping to satisfy them.

Consider the simple example of discovering a potentially interesting reference while reading a scientific paper. It is common to annotate the reference with a mark or create a note to record the intention to look it up in the future. The alternative is to interrupt reading and look it up when encountered. Both alternatives can be problematic. In the former case, the marked intention resides only in the physical document and the maker of the mark may never return to the paper copy and thus never be reminded of the intention. In the latter case, the on-going reading process is interrupted. With PapierCraft marked references can be automatically looked up and linked into digital versions [Ebrahimi, 2005]. They could also be added to a list of papers we plan to read in the future or the complete texts included in the ebook or laptop we commonly carry while traveling.

Once one starts to consider alternative ways of computationally responding to command-specified intentions, interesting new classes of applications are readily envisioned. PapierCraft's commands for highlighting, tagging, hyperlinking, copying, pasting, and stitching provide a rich base for design of new active-reading applications. While we have only begun to explore them, the potential of a wide range of novel applications spanning paper and computer media (e.g., easily and naturally specifying blog entries) is enticing.

PapierCraft is also applicable in Active Reading in which a group of people read, annotate, and discuss papers in a collaborative setting [Marshall, et al., 1999]. Different from individual reading, group reading usually involves social interactions among participants during collective discussion. PapierCraft's paper-based interface fits well in such a setting. Paper printouts, compared to laptop computers, are less distracting, easier to spread out on a table, and legible at wider view angles to support collective discussion [Sellen and Harper, 2001]. PapierCraft can provide a variety of support for group collaboration. For example, annotations on printouts by different participants can be captured and aggregated. Using streaming pens, group members can draw gestures on their paper copies to select references or specific sections to retrieve and show on a public display to assist group discussion.

8. FUTURE WORK

As our preliminary user study revealed, one of the main challenges facing the paper interface is lack of real-time feedback for actions such as mode switching and gesture recognition. We have begun to explore pen-top feedback mechanisms [Liao, et al., 2006] to enhance the PapierCraft command system via built-in LEDs, vibration motors, sound and speech. We plan to evaluate these mechanisms in both laboratory experiments and in a series of ethnographic studies of use in seminars and other active reading settings. In addition, we are considering various ubiquitous computing applications like 3D modeling as demonstrated in ModelCraft [Song, et al., 2006] in which the Anoto pattern is applied to the surfaces of physical models and command gestures drawn on the surface are used to shape the corresponding digital model.

For PapierCraft to be useful outside of research settings, it is clear from our initial trials that this will require not only wider availability of Anoto-based pen/paper technologies and paper-to-digital document mapping facilities but also careful analysis of how best to use the PapierCraft architecture to support specific real-world tasks. Although most documents are now created in digital form and an increasingly number of legacy documents are being digitalized (e.g., Google Books effort [Google, 2005]) we expect that, just as has been the case in the past, the use of paper will continue to grow and PapierCraft technology will become increasingly valuable for bridging the digital and paper worlds.

9. CONCLUSION

In this paper we have described PapierCraft, a system with a light-weight interface, allowing people to interact with Paper Augmented Digital Documents using an Anoto pen to issue gesture commands directly on the paper proxies of digital documents. It supports an extendable command set that was previously limited to purely digital environments or non-mobile augmented reality environments. PapierCraft introduces digital functions for paper to create interactive paper. This bridges the paper-digital gap and helps overcome difficulties in search, navigation, and sharing of paper documents. It demonstrates the feasibility of adding interactive commands to paper without sacrificing paper's wonderfully felicitous affordances.

The unbundling of the monolithic computer into fragmentary components and its movement off the desktop and into a world of ubiquitous computing devices present enormous challenges for interface design. As with many challenges there is also opportunity. PapierCraft capitalizes on the opportunity of new digital pen technology to

provide a flexible coupling between paper and computer. It enables novel ways of bridging the paper-computer gap with natural gesture-based commands to allow access to the strengths of both paper and digital media as well as flexible movement between them.

ACKNOWLEDGEMENTS

This work is supported by Microsoft Research (as part of the Microsoft Center for Interaction Design and Visualization at the University of Maryland) and NSF under Grant IIS-0414699 and IIS-00447730. We would like to thank Anoto for their support with streaming pattern space, Logitech for streaming software, and HP for hardware donations. We are also grateful to Dave Levin who implemented the first distributed version of the PADD infrastructure on which this system is based, to Ron Stanonik who assisted at UCSD, to Corinna Löckenhoff who supported the production of this document in many ways, and to Patrick Baudisch for valuable advice.

REFERENCES

- ADLER, A., A. GUJAR, L. B. HARRISON, K. O'HARA and A. SELLEN (1998), A diary study of work-related reading: design implications for digital reading devices. *Proceedings of CHI'98*, pp. 241-248.
- ALVARADO, C. and R. DAVIS (2004), SketchREAD: a multi-domain sketch recognition engine. *Proceedings of UIST'04*, pp. 23-32.
- ANOTO (2002), Development Guide for Service Enabled by Anoto Functionality.
- ARAI, T., D. AUST and S. E. HUDSON (1997), PaperLink: a technique for hyperlinking from real paper to electronic content. *Proceedings of CHI'97*, pp. 327 - 334.
- AVRAHAMI, D., S. E. HUDSON, T. P. MORAN and B. D. WILLIAMS (2001), Guided gesture support in the paper PDA. *Proceedings of UIST'01*, pp. 197 - 198.
- BAILEY, P. B. and A. J. KONSTAN (2003), Are informal tools better?: comparing DEMAIS, pencil and paper, and authorware for early multimedia design. *Proceedings of CHI'03*, pp. 313-320.
- BAILEY, P. B., A. J. KONSTAN and V. J. CARLIS (2001), DEMAIS: designing multimedia applications with interactive storyboards. *Proceedings of ACM Multimedia'01*, pp. 241-250.
- CARZANIGA, A., D. S. ROSENBLUM and A. L. WOLF (2001), Design and evaluation of a wide-area event notification service. *ACM Trans. Comput. Syst.*, 2001. **19**(3): pp 332-383.
- COHEN, P. R. and D. R. MCGEE (2004), Tangible multimodal interfaces for safety-critical applications. *Commun. ACM*, 2004. **47**(1): pp 41-46.
- CUNHA, P. C. (2004), The I-Book: a freely interpretive notebook for biologists. *Proceedings of Participatory Design Conference'04*, pp. 159-162.
- DYMETMAN, M. and M. COPPERMAN (1998), Intelligent Paper. *Proceedings of EP'98*, pp. 392 - 406.
- EBRAHIMI, R. (2005), Reading V2.0 : Next Generation Reading Systems and the PENHance Plug-in, unpublished honors thesis, University of California, Sand Diego.
- GOOGLE (2005), Google Books, <http://books.google.com>.
- GUIMBRETIERE, F. (2003), Paper Augmented Digital Documents. *Proceedings of UIST'03*, pp. 51 - 60.
- GUIMBRETIERE, F., M. STONE and T. WINOGRAD (2001), Fluid interaction with high-resolution wall-size displays. *Proceedings of UIST'01*, pp. 21 - 30.
- HARDOCK, G., G. KURTENBACH and W. BUXTON (1993), A marking based interface for collaborative writing. *Proceedings of UIST*, pp. 259-266.
- HEINER, J. M., S. E. HUDSON and K. TANAKA (1999), Linking and messaging from real paper in the Paper PDA. *Proceedings of UIST'99*, pp. 179 - 186.
- HINCKLEY, K., P. BAUDISCH, G. RAMOS and F. GUIMBRETIERE (2005), Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. *Proceedings of CHI'05*, pp. 451-460.

- HINCKLEY, K., G. RAMOS, F. GUIMBRETIERE, P. BAUDISCH and M. SMITH (2004), Stitching: Pen Gestures that Span Multiple Displays. *Proceedings of AVI'04*, pp. 23 - 31.
- IBM (1998), CrossPadd, <http://www.research.ibm.com/electricInk/>.
- JOHNSON, W., H. JELLINEK, J. LEIGH KLOTZ, R. RAO and S. K. CARD (1993), Bridging the paper and electronic worlds: the paper user interface. *Proceedings of CHI'93*, pp. 507 - 512.
- KARA BURAK, L. and F. T. STAHOVICH (2004), Hierarchical parsing and recognition of hand-sketched diagrams. *Proceedings of UIST'04*, pp. 13-22.
- KHAN, F. (1993), A Survey of Note-Taking Practices, Tech Report HPL-93-107, Personal Systems Laboratory, HP Laboratory.
- KLEMMER, S. R., J. GRAHAM, G. J. WOLFF and J. A. LANDAY (2003), Books with voices: paper transcripts as a physical interface to oral histories. *Proceedings of CHI'03*, pp. 89-96.
- KLEMMER, S. R., M. W. NEWMAN, R. FARRELL, M. BILEZIKJIAN and J. A. LANDAY (2001), The designers' outpost: a tangible interface for collaborative web site. *Proceedings of UIST'01*, pp. 1 - 10.
- KOIKE, H., Y. SATO, Y. KOBAYASHI, H. TOBITA and M. KOBAYASHI (2000), Interactive textbook and interactive Venn diagram: natural and intuitive interfaces on augmented desk system. *Proceedings of CHI'00*, pp. 121 - 128.
- LANDAY, A. J. and A. B. MYERS (1995), Interactive sketching for the early stages of user interface design. *Proceedings of CHI'95*, pp. 43-50.
- LEAPFROG (2005), Fly Pen, <http://www.leapfrog.com>.
- LEE, J. C., P. H. DIETZ, D. MAYNES-AMINZADE, R. RASKAR and S. E. HUDSON (2004), Automatic projector calibration with embedded light sensors. *Proceedings of UIST'04*, pp. 123 - 26
- LI, Y., K. HINCKLEY, Z. GUAN and J. A. LANDAY (2005), Experimental analysis of mode switching techniques in pen-based user interfaces. *Proceedings of CHI'05*, pp. 461-470.
- LIAO, C., F. GUIMBRETIERE and K. HINCKLEY (2005), PapierCraft: a command system for interactive paper. *Proceedings of UIST'05*, pp. 241-244.
- LIAO, C., F. GUIMBRETIERE and C. E. LOECKENHOFF (2006), Pentop feedback for paper-based interfaces. *Proceedings of UIST'06*, pp. 211-220.
- LIN, J., W. M. NEWMAN, I. J. HONG and A. J. LANDAY (2000), DENIM: finding a tighter fit between tools and practice for Web site design. *Proceedings of CHI'00*, pp. 510-517.
- LOGITECH (2004), IO Digital Pen, <http://www.logitech.com>.
- MACKAY, W. and D. PAGANI (1994), Video mosaic: laying out time in a physical space. *Proceedings of MM'94*, pp. 165 - 172.
- MACKAY, W. E., A.-L. FAYARD, L. FROBERT and L. MÉDINI (1998), Reinventing the familiar: exploring an augmented reality design space for air traffic control. *Proceedings of CHI'98*, pp. 558 - 565.
- MACKAY, W. E., D. S. PAGANI, L. FABER, B. INWOOD, P. LAUNIAINEN, L. BRENTA and V. POUZOL (1995), Ariel: augmenting paper engineering drawings. *Proceedings of CHI'95*, pp. 421 - 422.
- MACKAY, W. E., G. POTHIER, C. LETONDAL, K. BØEGH and H. E. SØRENSEN (2002), The missing link: augmenting biology laboratory notebooks. *Proceedings of UIST'02*, pp. 41 - 50.
- MANSOURI-SAMANI, M. and M. SLOMAN (1997), GEM A Generalised Event Monitoring Language for Distributed Systems. *IEEE/IOP/BCS Distributed Systems Engineering Journal*, 1997. 4(25).
- MARSHALL, C. C., M. N. PRICE, G. GOLOVCHINSKY and B. N. SCHLIT (1999), Introducing a digital library reading appliance into a reading group. *Proceedings of the fourth ACM conference on Digital libraries*, pp. 77-84.
- MICROSOFT (2003), OneNote, <http://www.microsoft.com/onenote>.
- MICROSOFT (2005), Microsoft Windows XP Tablet PC Edition 2005 Recognizer Pack.
- MILLER, R. C. and B. A. MYERS (1999), Synchronizing clipboards of multiple computers. *Proceedings of UIST'99*, pp. 65-66.
- NORRIE, M. C. and B. SIGNER (2003), Switching Over to Paper: A New Web Channel. *Proceedings of Web Information Systems Engineering'03*, pp. 209-218.
- O'HARA, K. and A. SELLEN (1997), A comparison of reading paper and on-line documents. *Proceedings of CHI*, pp. 335 - 342.
- OLSEN, R. D., JR., T. TAUFER and J. A. FAILS (2004), ScreenCrayons: annotating anything. *Proceedings of UIST'04*, pp. 165-174.
- PRICE, M. N., G. GOLOVCHINSKY and B. N. SCHLIT (1998), Linking by inking: trailblazing in a paper-like hypertext. *Proceedings of Conference on Hypertext and Hypermedia*, pp. 30 - 39.

- REKIMOTO, J. (1997), Pick-and-drop: a direct manipulation technique for multiple computer environments. *Proceedings of UIST'97*, pp. 31 - 39.
- REKIMOTO, J. (1999), Time-machine computing: a time-centric approach for the information environment. *Proceedings of UIST'99*, pp. 45-54.
- REKIMOTO, J. and M. SAITOH (1999), Augmented surfaces: a spatially continuous work space for hybrid computing environments. *Proceedings of CHI'99*, pp. 378 - 385.
- SAUND, E., D. FLEET, D. LARNER and J. MAHONEY (2003), Perceptually-supported image editing of text and graphics. *Proceedings of UIST'03*, pp. 183-192.
- SAUND, E. and E. LANK (2003), Stylus input and editing without prior selection of mode. *Proceedings of UIST'03*, pp. 213 - 216.
- SCHLIT, B. N., G. GOLOVCHINSKY and M. N. PRICE (1998), Beyond paper: supporting active reading with free form digital ink annotations. *Proceedings of CHI'98*, pp. 249 - 256.
- SCHÖN, D. A. (1983), *The Reflective Practitioner: How professionals think in action*. 1983, London: Temple Smith.
- SCHÖN, D. A. and G. WIGGINS (1992), Kinds of seeing and their functions in designing. *Design Studies*, 1992. **13**(2): pp 135-156.
- SELLEN, A. J. and R. H. R. HARPER (2001), *The Myth of the Paperless Office*. 1st ed. 2001: MIT press.
- SENSIVA (2005), [http:// www.sensiva.com/](http://www.sensiva.com/).
- SONG, H., F. GUIMBRETIERE, C. HU and H. LIPSON (2006), ModelCraft: capturing freehand annotations and edits on physical 3D models. *Proceedings of UIST'06*, pp. 13-22.
- WELLNER, P. (1993), Interacting with paper on the DigitalDesk. *Communications of the ACM*, 1993. **36**(7): pp 87 - 96.
- WILCOX, L. D., B. N. SCHLIT and N. SAWHNEY (1997), Dynamite: a dynamically organized ink and audio notebook. *Proceedings of CHI'97*, pp. 186 - 193.
- YEH, R. B., C. LIAO, S. R. KLEMMER, F. GUIMBRETIERE, B. LEE, B. KAKARADOV, J. STAMBERGER and A. PAEPCKE (2006), ButterflyNet: A Mobile Capture and Access System for Field Biology Research. *Proceedings of CHI'06*, pp. 571-580.
- ZELEZNIK, R. and T. MILLER (2006), Fluid Inking : Augmenting the Medium of Free-Form Inking with Gestures. *Proceedings of Graphics Interface '06*, pp. 155-162.
- ZHAO, S. and R. BALAKRISHNAN (2004), Simple vs. compound mark hierarchical marking menus. *Proceedings of UIST'04*, pp. 33-42.