

PapierCraft: A Command System for Interactive Paper

Chunyuan Liao, François Guimbretière
Human-Computer Interaction Lab
Department of Computer Science,
University of Maryland,
College Park, MD 20742, U.S.A
{liaomay, francois} @cs.umd.edu

Ken Hinckley
Microsoft Research
One Microsoft Way
Redmond, WA 98052-6399, USA
kenh@microsoft.com



Figure 1 A copy/paste interaction in PapierCraft: an image is copied from a printout (left), then pasted to a note (center). The result is shown on our PapierCraft viewer after pen synchronization (right). Marks were highlighted for clarity. Page from Open Access document <http://genomebiology.com/2003/4/8/R47> © 2003 Cheung et al. Used with permission.

ABSTRACT

Knowledge workers use paper extensively for document reviewing and note-taking due to its versatility and simplicity of use. As users annotate printed documents and gather notes, they create a rich web of annotations and cross references. Unfortunately, as paper is a static media, this web often gets trapped in the physical world. While several digital solutions such as XLibris [15] and Digital Desk [18] have been proposed, they suffer from a small display size or onerous hardware requirements.

To address these limitations, we propose PapierCraft, a gesture-based interface that allows users to manipulate digital documents directly using their printouts as proxies. Using a digital pen, users can annotate a printout or draw command gestures to indicate operations such as copying a document area, pasting an area previously copied, or creating a link. Upon pen synchronization, our infrastructure executes these commands and presents the result in a customized viewer. In this paper we describe the design and implementation of the PapierCraft command system, and report on early user feedback.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST'05, October 23–27, 2005, Seattle, Washington, USA.
Copyright ACM 1-59593-023-X/05/0010...\$5.00.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

KEYWORDS: Paper interfaces, pens, distributed systems.

INTRODUCTION

In this computer age, paper is still used extensively by knowledge worker during the life cycle of a digital document [16]. Paper documents are easy to annotate, they provide an inexpensive way to display large quantities of information by spreading out documents, and they are socially well-accepted in meetings. Paper interactions are also extremely flexible. Users might augment printed documents with standard proofreading marks, their own notation system or add margin annotations that summarize a given section or offer alternate wording. Users also often complement direct annotations on printed documents with annotations made on a separate pad or notebook. In that case, the pad or notebook acts as a structured repository of information gathered by its owner. Pads or notebooks often contain explicit references to other documents (like “See also figure 7 on page 23”) and users literally cut and paste information into them (such as post-it notes). Users can even create elaborate fold-outs by “stitching” multiple documents together side by side.

These notations and physical arrangements represent an implicit web of links between multiple documents manipulated by the users during their work. Unfortunately, while paper makes it easy to create such a web, it also makes it difficult to transfer it to the digital world where the links

could be easily navigated and searched.

Several solutions to this problem have been proposed. One approach, explored by systems such as Xlibris [15] and Microsoft OneNote [11], simulates paper affordances using Tablet-like computers. While moving into the digital world makes it easy to capture all user interactions, and to “link by inking” [12], it also faces the limitation of small displays when compared to the surface area of a physical desk. Another approach is to augment paper with a nearby computer, as proposed by the Digital Desk [18] and A-book systems [10]. These methods offer a unique blend of digital and paper affordances, but require hardware that is much more expensive and much less portable than plain paper.

A third alternative is proposed by the Paper Augmented Digital Documents (PADD) [4] system in which documents can be manipulated either on paper or on a computer screen. Within the PADD system, paper and computers are seen as two collaborating tools to manipulate a digital document. Users can select the best medium to interact with a document as they see fit. This approach was made possible by the recent availability of digital pens such as the Logitech iO2. Using the Anoto technology [2], such pens are able to capture the location of strokes made on special paper as well as the page’s unique ID.

We contribute PapierCraft, a paper-based command system for PADD documents. Using PapierCraft, printouts are considered as *proxies* that capture annotations as well as commands to be executed on the corresponding digital version of each document. For instance, users can copy and paste information from one paper document to another (see Figure 1), create links between two different paper documents, or “stitch” two paper documents together by drawing a pen stroke across them. Upon synchronization of the digital pen with the infrastructure, PapierCraft executes all commands on the corresponding digital documents and presents the results in a digital document browser. All information gathered in the paper world becomes accessible to users, allowing them to easily explore the implicit links they created on paper. Thus, PapierCraft based systems combine the advantages of paper with the advantages of digital systems like Xlibris and OneNote. Compared to other systems which use paper as an input medium (such as Xax [8], PaperPDA [5], or Rasa [3]), PapierCraft has a greater level of flexibility as it offers an extendable command system and can manipulate a wide range of user content.

Designing a paper-based command system such as PapierCraft presents a unique set of challenges. While dealing with the limited feedback provided by paper, the system has to be well integrated with current paper-based interactions, provide a reliable and versatile mechanism to execute commands, and offer a simple way for the user to read back commands after they have been issued. Here we present the rationale for our design as well as report on the first implementation of the PapierCraft command system on top of a PADD infrastructure. We also report on preliminary feed-

back from a small sample of users.

A PAPER-BASED COMMAND SYSTEM

Like a digital marking interface, the design of a paper-based marking interface involves three key design decisions: 1) How will the system distinguish between inking strokes and gesture strokes? 2) How will the scope of a given command be specified? 3) Once the scope has been selected, how will user select the command they wish to execute on that scope? The limited feedback provided by paper has a strong influence on each of these design decisions.

Ink and gestures

Many solutions have been proposed to distinguish between strokes intended as content versus strokes intended to be interpreted by the system. Some systems such as SketchREAD [1] propose a fully implicit approach in which the computer automatically distinguishes ink strokes from gesture strokes. Other systems such as Scriboli [7] propose a fully explicit approach, in which users indicate the type of the current stroke by pressing a button. A mixed approach, where the computer and the user collaborate to resolved ambiguous input, is also possible [14]. Given the limited level of feedback provided by paper, a non-explicit or mixed approach would be problematic as these both require immediate feedback. As a result, our system uses a “gesture” button present either on the pen or in the environment (e.g., a foot pedal – as used in our current implementation). Our system only requires a weak synchronization: a stroke is considered a gesture stroke as long as the gesture button is pressed for some duration during that stroke.

Once gestures have been identified, we need to further distinguish between scope selection and command selection. We adopted the pigtail technique proposed by Hinckley et al. in Scriboli [7]. A pigtail is a self-intersecting gesture stroke resembling a script ‘e’; all strokes between the first gesture stroke and a gesture stroke containing the pigtail are considered as part of the scope selection, and all gesture strokes after the pigtail form the command selection. The pigtail notation is familiar to proofreaders and yields similar performance to presenting a “handle” (menu box) at the end of any stroke specifying a scope [7, 14], which of course would be impossible to present on paper.

Specifying the scope of commands

The absence of real-time feedback had the greatest impact on scope selection in our interface. Consider the simple example of a cut 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 the system providing instant feedback of the area selected. Once the selection is complete, one issues the copy command and moves to the paste location. As soon as the paste command is issued, visual feedback is provided immediately to show the result of this operation. Of course, on paper neither type of feedback can be provided. To address this problem, we ask users to draw the intended scope of all their commands. As shown in

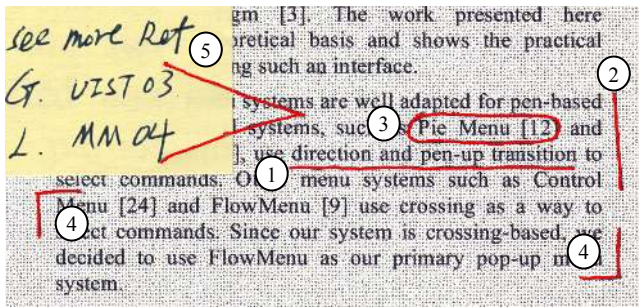


Figure 2 PapierCraft scopes: (1) underline, (2) margin bar, (3) lasso, (4) crop marks and (5) stitching mark (across two documents)

Figure 1, in the case of the copy operation, the scope identifies the region of the paper to be copied; for the paste operation, the scope specifies the desired position and dimensions of the pasted data (the system will automatically scale the copied part to fit in the target scope).

PapierCraft offers five types of scope selectors (Figure 2) inspired by typical marks found on manuscripts, namely underline, margin bar, lasso, crop marks and the stitching mark. They can be used to issue multi-parameters commands (Table 1). For example, the “excerption with keyword” command uses crop marks to select the area and underline to select the assigned keyword within that area. The Stitching command, uses the scope gesture to designates the target documents as well as their relative positions.

Selecting a command

Given the limited feedback available it was natural to consider a marking based system such as marking menu [9] or unistroke gestures as used in Sensiva [17]. We preferred marking menu as it was better adapted to the use of the pigtail delimiter.

Single level marking menus typically only offer 8 different commands which we felt might be too restrictive. For a richer command set, one solution was to use a two-level hierarchy, but it would be difficult for users to discover, learn and remember the different marks. The simple mark hierarchical marking menus [20] would only exacerbate this problem, because on paper, a human reader cannot discern the temporal order of the marks. Instead, we use a mixed approach: the most frequent commands (like *copy* and *paste*) can be accessed directly from the cardinal directions of our marking menu (*East, West* respectively). A richer command set can be accessed by simply writing down an unambiguous prefix of the command name (Table 1, row three). For example, one can directly write the word “Paste” next to the pigtail to take precedence over the mark direction. Writing a command name not only allows for a larger command set, it also makes it easier for people to remember the command they wish to issue. A similar technique known as “mnemonic flicks” has recently been proposed, but not yet published, for a TabletPC-based marking interface [19].

Our current prototype provides a small set of commands. It includes **copy**, **paste**, **hyperlink start**, and **hyperlink end**,

| Operation | Command on Page 1 | Command on Page 2 |
|--------------------------|-------------------|-------------------|
| Excerption | | |
| Excerption with keyword | | |
| Excerption with cmd name | | |
| Hyperlink | | |
| Stitching | | |

Table 1: Operations supported by PapierCraft. Users start each operation on page 1 and finish it on page 2. For the stitching operation the two pages may overlap. For simplicity, we only use “crop marks” but other scopes are possible (see Figure 2).

selected by marking menus towards **East, West, North, and South** respectively. By combining different commands and scope selectors, we support five operations as shown in Table 1. This set could easily be extended by adding more marking menu items or more command words.

Excerpting in PapierCraft

As shown in Figure 1, to excerpt a figure from one printout to a page of notes, the user first draws crop marks around the target figure (while pressing the gesture button), followed by a “Copy” marking menu selection. Then, writing on the page of notes, the user employs the crop marks again to designate the area in which information will appear, and finishes by drawing the “Paste” command. Note that in both cases the lower left crop mark, the pigtail, and the command selection mark can be combined in one stroke. After synchronization, PapierCraft puts the selected figure (as well as any annotations on top of it) in the digital version of the note sheet. To help users recall the context of the pasted data, we paste a slightly larger area than was selected by the user. Our system also creates a hyperlink between the pasted information and the original page so that it is easy for the user to access the original document by simply clicking the figure in her note. Finally, as a convenience, the system also extracts the text, if any, present in the source scope. This text can be used for indexing, searching, or copying to a text editor.

IMPLEMENTATION

As our system was designed to be used without a nearby computer, commands are executed in batch mode upon synchronization of the pen with the PADD infrastructure. First, the synchronization host imports the strokes from the pen using the Anoto SDK 3.0 infrastructure. It then requests the corresponding digital files from the PADD infrastructure. Annotation strokes are simply merged into the files,

while gesture strokes are parsed, recognized, and translated into *local events* (such as “copy this area to the clipboard”). Each local event is then transmitted to a global *Event Server*. As it receives events from different clients, the Event Server aligns them in time, and parses the stream of events for pre-defined patterns. For example, a “Copy” event on page *A* immediately followed by a “Paste” event on page *B* triggers a “Copy/Paste” across the two pages. Once an operation has been recognized, a notification is sent to the client that is handling the affected page. The client will then contact the server and retrieve the pasted data before updating the corresponding digital file and uploading it to the PADD server.

In a sense, PapierCraft conceptualizes each piece of paper as a separate “mobile device” displaying information and capturing users’ strokes. Thus, our implementation of “Copy/Paste across pages” has similar technical requirements as distributed interaction techniques to “copy and paste between mobile devices” such as Pick-and-drop [13] and Stitching [6]. PapierCraft shows how a similar style of interaction can be extended to tangible paper interfaces while maintaining the corresponding digital representations.

EARLY USER FEEDBACK

We conducted a small scale informal evaluation of PapierCraft with four colleagues (not affiliated with this project). Users performed interactions using the digital pen with a foot pedal as a gesture button. For each participant, we provided a hands-on demonstration of how to copy and paste. Next, we asked the participants to perform several copy and paste operations using different scope selection mechanisms. Users also performed commands using two digital pens: one always in annotation mode (with a blue ink cartridge), and one always in gesture mode (with red ink).

Overall, participants’ reaction was positive. They believed that our gesture set would be easy to remember, especially if proper mnemonic cues were provided (e.g., users noted that the Paste mark looks like a “P”, as shown in Figure 1, middle). The option of writing down the command name was very popular, and participants commented that it would be especially useful if the system was user configurable. 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, users liked the direct feedback provided by the use of two pens; because the gestures appeared on the paper in red ink, they were easy to identify. On the other hand, users also reported that switching pens all the time was cumbersome, and would require them to carry two pens. The consensus was that as users become familiar with PapierCraft, one pen would be the preferred option.

We also discussed the minimal feedback provided by the interface with each participant. Users noted that the level of feedback required would strongly depend on the reliability of the system. If the system had a very high rate of gesture recognition, some participants felt that the current level of feedback might be acceptable. To improve our prototype,

we are exploring how to provide interactive feedback within the constraints imposed by the pen form factor.

CONCLUSION

PapierCraft lets people interact with Paper Augmented Digital Documents by marking directly on their printouts. It supports an extendable command set (including Copy/Paste, excerpting with keywords, hyperlinking documents, and stitching documents together) which was previously limited to purely digital environments or non-mobile augmented reality environments. PapierCraft thus demonstrates the feasibility of interactive commands for real paper while retaining its original affordances.

ACKNOWLEDGEMENTS

This work was supported in part by a grant from Microsoft Research. We would like to thank HP for providing hardware and Anoto for their technical support. Jim Hollan, David Levin, Corinna Löckenhoff and Patrick Baudisch provided support, help and advice.

REFERENCES

1. Alvarado, C. and R. Davis. SketchREAD: a multi-domain sketch recognition engine. *Proceedings of UIST’04*, pp. 23 - 32.
2. Anoto, <http://www.anoto.com>.
3. Cohen, P.R. and D.R. McGee, Tangible multimodal interfaces for safety-critical applications. *Communications of the ACM*, 2004. **47**(1): p. 41 - 46.
4. Guimbretiere, F. Paper Augmented Digital Documents. *Proceedings of UIST’03*, pp. 51 - 60.
5. Heiner, J.M., S.E. Hudson, and K. Tanaka. Linking and messaging from real paper in the Paper PDA. *Proceedings of UIST’99*, pp. 179 - 186.
6. Hinckley, K., G. Ramos, F. Guimbretiere, P. Baudisch, and M. Smith. Stitching: Pen Gestures that Span Multiple Displays. *Proceedings of AVI’04*, pp. 23 - 31.
7. Hinckley, K., P. Baudisch, G. Ramos, and F. Guimbretiere. Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in Scribboli. *Proceedings of CHI’05*, pp. 451 - 460.
8. Johnson, W., H. Jellinek, J. Leigh Klotz, R. Rao, and S.K. Card. Bridging the paper and electronic worlds: the paper user interface. *Proceedings of CHI’93*, pp. 507 - 512.
9. Kurtenbach, G., The design and Evaluation of Marking Menus, PhD thesis, University of Toronto. 1993
10. Mackay, W.E., G. Pothier, C. Letondal, K. Bøegh, and H.E. Sørensen. The missing link: augmenting biology laboratory notebooks. *Proceedings of UIST’02*, pp. 41 - 50.
11. OneNote, <http://office.microsoft.com/>.
12. Price, M.N., G. Golovchinsky, and B.N. Schilit. Linking by inking: trailblazing in a paper-like hypertext. *Proceedings of Conference on Hypertext and Hypermedia*, pp. 30 - 39.
13. Rekimoto, J. Pick-and-drop: a direct manipulation technique for multiple computer environments. *Proceedings of UIST’97*, pp. 31 - 39.
14. Saund, E. and E. Lank. Stylus input and editing without prior selection of mode. *Proceedings of UIST’03*, pp. 213 - 216.
15. Schilit, B.N., G. Golovchinsky, and M.N. Price. Beyond paper: supporting active reading with free form digital ink annotations. *Proceedings of CHI’98*, pp. 249 - 256.
16. Sellen, A.J. and R.H.R. Harper, *The Myth of the Paperless Office*. 1st ed. 2001: MIT press.
17. Sensiva, <http://www.sensiva.com/>.
18. Wellner, P., Interacting with paper on the DigitalDesk. *Communications of the ACM*, 1993. **36**(7): p. 87 - 96.
19. Zeleznik, R., T. Miller, Fluid Inking: Modelless Integration of Gestures with Inking. Technical report CS-05-10, Brown University, 2005,
20. Zhao, S. and R. Balakrishnan. Simple vs. compound mark hierarchical marking menus. *Proceedings of UIST’04*, pp. 33 - 42.