

# Printertainment: Printing With Interactive Cover Sheets

Jason Hong,<sup>1</sup> Morgan N. Price,<sup>2</sup> Bill N. Schilit,<sup>2</sup> Gene Golovchinsky<sup>2</sup>

<sup>1</sup>UC Berkeley Computer Science Division  
387 Soda Hall  
Berkeley, CA 94720-1776 USA  
+1 510 643 7354  
jasonh@cs.berkeley.edu

<sup>2</sup>FX Palo Alto Laboratory  
3400 Hillview Ave., Bldg 4  
Palo Alto, CA 94304  
+1 650 813-7233  
{price, schilit, gene}@pal.xerox.com

## ABSTRACT

We explored a new type of user interface, *interactive cover sheets*: computer forms laid out on the banner pages of print jobs that people can mark on, scan back into a multifunction printer/scanner, and use as input to applications. Cover sheets are commonly strewn around printer rooms; with interactivity, they can let people see what others have to say, add their own comments, or play games, all while waiting for their print jobs. We designed three prototype applications and deployed them briefly in our research lab. We found that interactive cover sheets can be very appealing, that the sheets must be designed so that people can still identify these pages as cover sheets, and that the slow interaction cycle favors asynchronous applications.

## KEYWORDS

Printing, paper user interface (PUI), paper applications, cover sheets, ubiquitous computing

## INTRODUCTION

Printing is universal in today's offices. In common "print rooms," people talk to colleagues or sort print jobs while waiting for their own jobs to complete. Sometimes, however, the experience is reduced to waiting for the machine. Our premise is that we can improve this experience by giving people information to browse, interact with, or socialize over while they wait. We do not, however, want to interfere with the social activities that already surround printers, or to add to printer room equipment clutter. Thus we feel that desktop computers would be inappropriate for these activities.

Instead, our approach is to use a *paper user interface*: paper forms that people can mark on and scan into a digital printer/copier, which takes action based on the marks (c.f. [2]). Paper UIs leverage digital multi-function copiers/printers/scanners that are increasingly common. Paper can be handed around in support of informal communication, and can be carried away. To explore these ideas, we built and briefly deployed a series of three prototypes.

## XAMAGOTCHI: IT LIVES IN THE PRINTER

Inspired by Bandai's Tamagotchi™ virtual reality pet, we designed a "virtual printer pet" that appears on cover sheets (also called banner pages) as shown in Figure 1. People feed and take care of the pet by checking boxes on the cover sheet and then scanning it.

For one day, we simulated Xamagotchi by placing virtual printer pet pages into the cover sheet paper tray of our research lab's central printer/copier. People were surprised and amused, and very positive. Several people found it visually pleasing, and liked how cover pages could be used for something other than identifying the owner of the print job. Several people put them up on their walls, where they remained several months later. But people did not necessarily like the idea of taking care of a virtual printer pet.

## MADLIBS

Our next experiment was a form fill-in party game called MadLibs [3]. People enter nouns, verbs, adjectives, and, using a template, the computer produces an amusing story (see Figure 2). We used Xerox's Document Application Environment (DAE), a toolkit for paper applications, to build a working prototype. The application takes freeform ink input and generates a story with ink intermingled with typeset text.

We placed copies of MadLibs forms next to our workgroup printer/copier. People found MadLibs simple, amusing, and social. On the other hand, MadLibs did not work well as a printer room application because it took too long to output the story. There is a general problem of contention between printing and cover sheet interaction: the printer/copier can scan the sheet while printing, but cannot output the results until the print job completes. Thus we decided to explore an application requiring only asynchronous interaction.

## COVER NOTES

Our third experiment was a lightweight group communication application running on the cover sheets of print jobs, called Cover Notes. Because cover sheets are public, Cover Notes are analogous to a bulletin board. Unlike a bulletin board, Cover Notes let people take material away with them.

The Cover Notes layout includes four boxes on the right margin of each cover sheet (see Figure 3). Handwritten

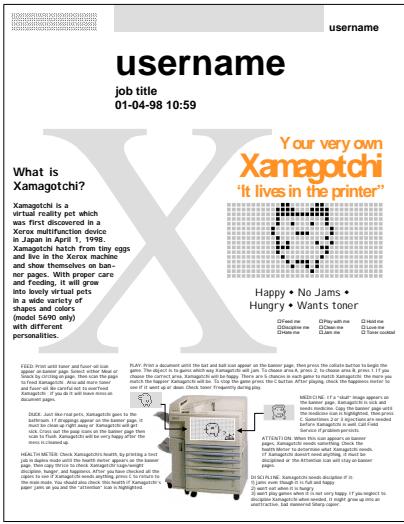


Figure 1: Xamagotchi Printer Pet

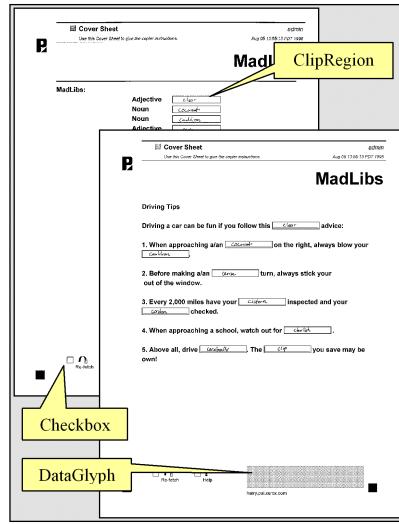


Figure 2: MadLibs

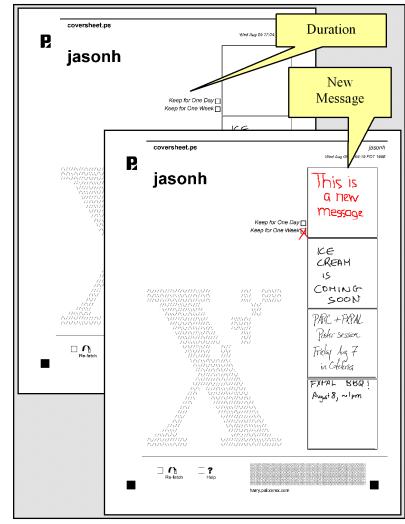


Figure 3: Cover Notes

notes, drawn randomly from a pool of existing notes, are printed in the bottom three boxes; the top box is empty. People can add a new note to the pool by drawing in the empty box and then scanning in the cover sheet. The next time a print job arrives, the new note might be randomly selected and appear on the cover sheet. We also designed a way to remove old items: when entering notes, people can check a box to specify whether the note should persist for a week or a day. We implemented this application using the architecture shown in Figure 4.

We deployed the Cover Notes application for one week on our printer/copier. Because of prototype limitations, people had to install a special “proxy” printer at their desktop, and could not print double-sided or collated. Nevertheless, a substantial proportion of printouts during this period included Cover Notes. Between the jobs which had not yet been picked up, and the cover sheets that people discarded by leaving near the printer (rather than in the recycle bin), a variety of Cover Notes were usually visible. Thus we believe Cover Notes succeeded in publicizing the information it contained. On the other hand, people did not always recognize that Cover Notes were also cover sheets, a problem that did not arise with Xamagotchi. There were no complaints about waiting: the asynchronous strategy seemed effective.

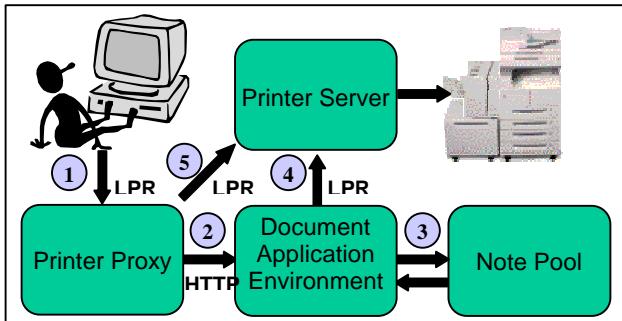


Figure 4: Cover Notes Architecture

## CONCLUSIONS AND FUTURE WORK

Our experiments show that interactive cover sheets can make printing more fun and interesting. To avoid the delays in receiving output from paper applications, interactive cover sheet applications should be asynchronous as much as possible: people should receive interesting cover sheets without any interaction, and should be able to respond to them without being expected to wait for output. To avoid confusion, interactive cover sheets must remain identifiable as such, for example, by being decorations on the existing cover sheets (as in Xamagotchi).

Our challenge is to base these applications on sustainable sources of entertainment and to make them more informative. To meet both challenges, we propose extending Cover Notes to form an office newspaper that combines handwritten notes, announcements from email, and news and entertainment from the Web (Web-based cover sheets were explored in [1]). By sharing this information across printers, this newspaper could help to bring together workgroups that are distributed across floors or buildings.

## ACKNOWLEDGMENTS

Thanks to James Landay for his helpful comments, Kei Tanaka and our other colleagues at FXPAL, and the Xerox DAE team. We thank Joe Sullivan and Jim Baker for supporting this research.

## REFERENCES

1. LaMarca, A. and Goldberg, D. CoverUp: A Paper-based Push System. Xerox PARC unpublished report. September 1997.
2. Johnson, W., et al. Bridging the Paper and Electronic Worlds: The Paper User Interface. In *INTERCHI '93 Conference Proceedings*, Addison-Wesley, 507-512.
3. Price, R., and Stern, L. The Original #1 Mad Libs. Price Stern Sloan Pub. 1974