



A viewer for PostScript documents

Citation

Adam Ginsburg, Joe Marks, and Stuart M. Shieber. A viewer for PostScript documents. In Proceedings of UIST '96, 1996.

Published Version

<http://doi.acm.org/10.1145/237091.237095>

Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:2265290>

Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

A Viewer for PostScript Documents

Adam Ginsburg
Aiken Computation Lab
Harvard University
Cambridge, MA 02138
+1-617-495-2344
ginsburg@eecs.harvard.edu

Joe Marks
MERL
201 Broadway
Cambridge, MA 02139
+1-617-621-7534
marks@merl.com

Stuart Shieber
Aiken Computation Lab
Harvard University
Cambridge, MA 02138
+1-617-495-2344
shieber@eecs.harvard.edu

ABSTRACT

We describe a PostScript viewer that provides navigation and annotation functionality similar to that of paper documents using simple unified user-interface techniques.

KEYWORDS: Document viewing, PostScript, annotation.

INTRODUCTION

Viewing a document on a computer screen is a poor substitute for reading a paper copy of the same document. This is due in large part to the physical limitations of computers and display monitors, but uninspired user-interface design has also been a contributing factor. Our goal in developing *DeckView* has been to design a PostScript viewer that affords some of the same kinds of functionality provided by paper documents. For instance, paper documents provide sequential paging; contextual location of a page; scanning (by riffling); direct access (using thumb indexes, bookmarks, and fingers); and annotation.¹ Our approach, as manifested in the *DeckView* document viewer, has been to provide this functionality not by mimicking the devices that provide for it in paper documents (e.g., simulating the appearance of bookmarks or thumb indexes), but rather by designing simple unified user-interface techniques tailored specifically to on-screen documents.

NAVIGATION FEATURES

We provide the variety of navigational functions supported by paper documents — sequential paging, scanning, permanent and temporary marked direct access, contextual location — through a single novel device. Antialiased thumbnail images of the pages are placed in a “tray” (see Figure 1). By overlapping the thumbnails like a deck of cards, all of the document’s pages can be included in the tray, regardless of document length. Depending on the number of pages, more or less of each page will be visible, except for the current page, which is always fully visible. The position of the fully visible thumbnail locates the current page within the document.

¹There are other ways in which on-line document viewing might be enhanced: the “Virtual Paper” project at DEC SRC [2] is motivated by the same general goals as ours, but its developers have concentrated on different aspects of the document-viewing task.

To scan through the pages of the electronic document, the user drags the pointer through the deck of thumbnails. As the pointer moves through the deck, the thumbnail below the pointer is shown in its entirety and the page number is shown above the thumbnail tray. When the pointer crosses the top or bottom border of the fully visible thumbnail, the next thumbnail is uncovered in place, and the pointer is warped to a set distance from the appropriate edge of the new thumbnail. This pointer-warping strategy ensures that the traversal of a given number of pages requires equidistant mouse movement regardless of the thumbnail sizes, or their degree of overlap.² Once the mouse is released, the page is selected and shown at full size in the page-display window.

The reader can maintain direct access to pages by dragging and dropping pages into the trays adjacent to the main central tray, forming “bookmarks”. The left tray holds permanent bookmarks, the right temporary ones. These bookmark thumbnails remain horizontally aligned with the corresponding thumbnails in the central tray, even when the thumbnails are scanned. Furthermore, thumbnails in the bookmark trays can be scanned in the same way as those in the central tray.

ANNOTATION FEATURES

Readers of paper documents avail themselves of the ability to annotate documents in a variety of ways, adding marks, labels, and longer annotations. On-line document annotation can be an important element of computer-supported collaborative work [3]. Again, our strategy has been to use a single device for all types of annotation and linking. Annotations are added to a document by dragging the annotation button (to the left of the “Previous Page” button) to a specific location on the page in the page-display window. The dialog box that appears (shown in Figure 1) contains fields for an annotation label, annotation text, and hyperlink destination. Any or all of these fields may be specified, allowing for a variety of different annotation types. The destination of a hyperlink is another annotation: it is placed by dragging the hyperlink icon from an open annotation dialog box to the desired location on the target page, or by dragging an existing annotation and dropping it on the hyperlink icon. (Navigation to find the target page can be performed while the annotation dialog box remains open.)

Annotation icons are overlaid on top of the PostScript text and

²Note that this aspect of the interaction permits an arbitrary number of pages, regardless of the height of the tray. In contrast, scroll bars require a minimum of one pixel per item, and are difficult to use without much more resolution.

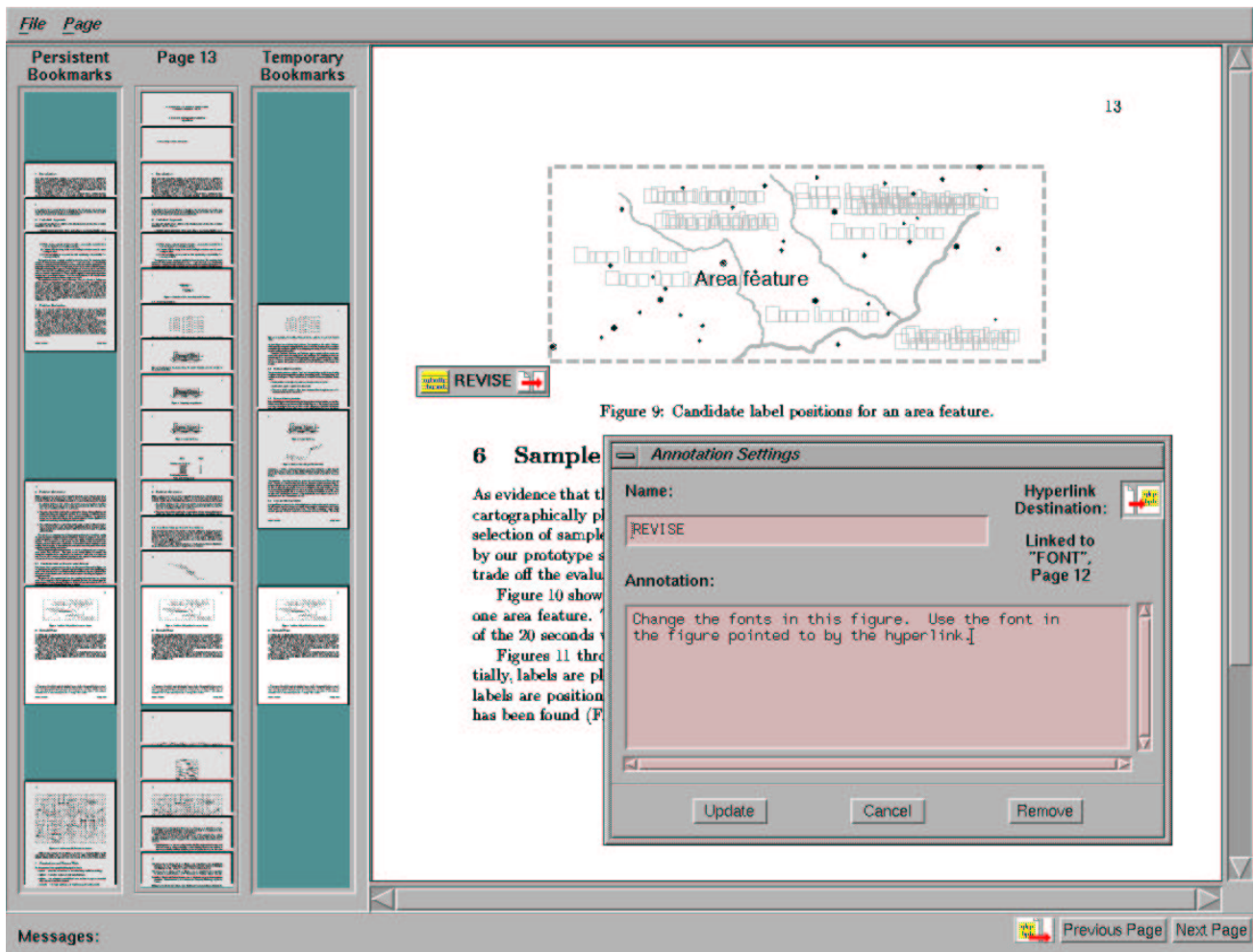


Figure 1: Screen image of user interface.

imagery in the page-display window. In Figure 1, a sample icon is shown to the bottom left of the figure in the page-display window. The reader can access the annotation text (if available) by clicking on the “Post-it” portion at the left end of the icon, and can follow a hyperlink (if instantiated) by clicking on the arrow portion at the right end of the icon. Double clicking on the icon label brings up the dialog box again, allowing further editing.

IMPLEMENTATION DETAILS

DeckView is implemented in C++ and Motif. The GNU Ghostscript interpreter is used to render document images. Documents must be formatted according to the Adobe Document Structuring Conventions (DSC) [1] to enable the navigation and annotation features, though non-DSC PostScript documents are supported with limited functionality.

Permanent-bookmark and annotation data are saved as comments in the PostScript document, so the integrity of the document is not compromised in any way by the addition of bookmarks and annotations, and the marked document can be shared as a single file.

Source code and binaries for *DeckView* are available under a free license at the following URL: “<http://www.eecs.harvard.edu/~shieber/DeckView/>”.

ACKNOWLEDGMENTS

Mary Farbood assisted with system testing. This material is based upon work supported in part by the National Science Foundation under Grant No. IRI-9350192.

REFERENCES

1. Adobe Systems, Inc. Postscript language document structuring conventions specification, version 3.0., September 1992. <ftp://ftp.adobe.com/pub/adobe/Developer-Support/TechNotes/PSfiles/5001.DSC.Spec.ps>.
2. Andrew Birrell and Paul McJones. The Virtual Paper system. <http://www.research.digital.com/SRC/virtualpaper/>.
3. Ellen Francik, Susan Ehrlich Rudman, Donna Cooper, and Stephen Levine. Putting innovation to work: Adoption strategies for multimedia communication systems. *CACM*, 34(12):53–63, December 1991.