Bridging the Paper and Electronic Worlds: The Paper User Interface

Walter Johnson[1], Herbert Jellinek[1], Leigh Klotz[2], Jr. Ramana Rao[1], Stuart Card[1]

[1]Xerox Palo Alto Research Center
[2]Fuji Xerox Corporation
3333 Coyote Hill Road
Palo Alto, CA

ABSTRACT

Since its invention millenia ago, paper has served as one of our primary communications media. Its inherent physical properties make it easy to use, transport, and store, and cheap to manufacture. Despite these advantages, paper remains a second class citizen in the electronic world. In this paper, we present a new technology for bridging the paper and the electronic worlds. In the new technology, the user interface moves beyond the workstation and onto paper itself. We describe paper user interface technology and its, implementation in a particular system called XAX.

INTRODUCTION

Since its invention millenia ago, paper has served as one of our primary communications media. Its inherent physical properties make it easy to use, transport, and store, and cheap to manufacture. But paper also has its shortcomings. It is a passive medium, whose contents are not easily manipulated by hand nor processed by machine. It cannot easily take advantage of new methods for manipulating, communicating, filing, and processing information afforded by advances in desktop computer software. In this paper we present a method for bringing together the paper and electronic worlds through a new hybrid *paper user interface* technology. In the new technology, the user interface moves beyond the workstation and onto paper itself. This allows user interfaces to be produced that so blend into the user's world that they seem to disappear.

The properties of paper are so well suited for its uses as a medium for information that predictions of its early demise seem exaggerated. Its ease of use makes it universally accessible. Its low cost allows it to be used in quantity. Its light weight makes it portable. It can be inscribed with writing or drawings by the simplest hand tools, yet it is also easily inscribed with color photographs in mass quantities by machine and serves well as a primary medium of modern commerce and culture. But the properties of paper are also limiting. Inscriptions on it are fixed and cannot be moved around. It is passive and does not, for instance, automatically rearrange the words in a paragraph when a new one is inserted. Its contents cannot be automatically processed, for example to detect misspelling or to sum columns of numbers. Only slowly can its contents be transmitted over distances.

Electronic media have almost the opposite profile of strengths and weaknesses. Figure 1 shows media divided into three categories: paper, electronic image, and electronic data structure. Many technological inventions can be seen as implementations of transformations between these categories: copying takes paper into more paper. Electronic printing renders data structures into images and prints images onto paper. Scanning takes paper into electronic images. Communications technologies transform these categories over space: fax takes paper into electronic images into electronic images at a distance into paper. Information filing and retrieval transforms these categories over time.

In the 1970's and early 1980's effort was focused on running "downhill" in the diagram, from data structures through to paper (and also on transformation within a category, such as text-editing). Since the late 1980s and surely through the 1990s more effort is being focused on going "uphill" from paper to electronic image to data structure. As this succeeds, we begin to have the possibility of viable processing loops that go through paper, yet have access to the information processes made possible through electronics. The user can potentially interact with such a system either electronically, through a workstation user interface, or, as we shall suggest, via paper, through a paper user interface. The paper user interface we describe in this paper is one of the first attempts to develop user interfaces in this new medium.

PAPER AND THE ELECTRONIC INFRASTRUCTURE

Although an imaging revolution has been predicted for years, scanning systems that can handle volumes of documents are costly and have been limited to use in large routinized environments such as forms processing. Image scanning systems for personal or small workgroup settings have generally been used solely for the purpose of scanning the occasional piece of artwork for desktop publishing.

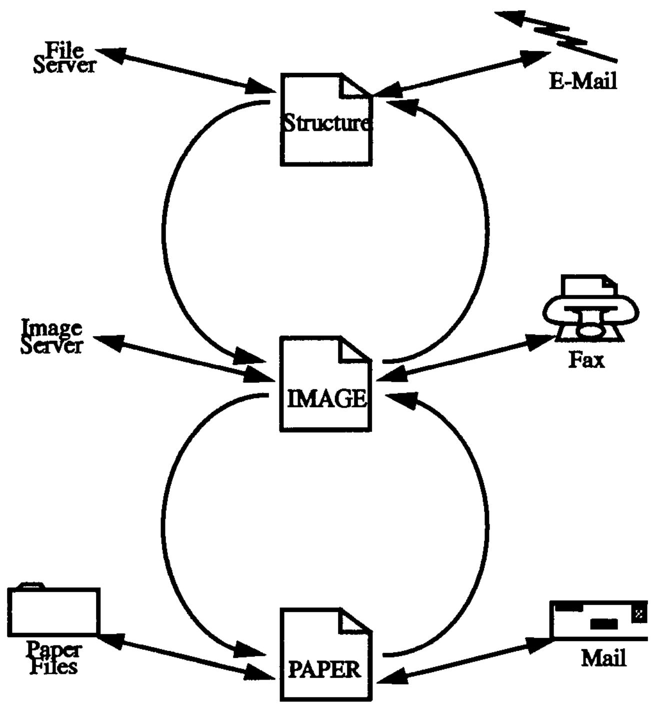

Figure 1: Bridges between Paper and Electronic Worlds

More than the cost and simple lack of desire for imaging systems is responsible for this lack of acceptance. Two other factors have limited the success of this revolution. The first is ease of use: scanners have been available for many years, yet are far less prevalent than fax machines. To use a fax machine, one must know little more than how to make a phone call. To use a scanner, one must know the scanner hardware, the computer's operating system, and a specific scanning software package. The second is functional: even when stored on the computer, image files are notoriously useless to the average user. Most environments provide little support for images, making it difficult to print, fax, display, or integrate them into other documents.

Recently, several technologies have matured to the state where solutions to many of these problems are at hand. Light-lens photocopying is gradually being superseded by digital reprographics, leading to cost reductions in high quality digital document handling systems. Users are beginning to understand and expect advanced image capabilities in reprographic systems. The rapid proliferation of computer fax modems offers a widespread capability to capture image files on personal computers.

This confluence of events in the change of the computational infrastructure removes barriers to the introduction of paper into the electronic document life cycle. The problem that remains is the essential "lifelessness" of paper documents, and the resulting scanned images. Regardless of the structure underlying an electronic document, once it is printed it is just marks on paper. Thus, paper is a dead end, a by-product of the cycle. Much energy has been spent on character and layout recognition as a means to put electronic life back into paper documents. While difficult at best, the vagaries of document and character recognition are exacerbated by the distortions imposed during the processes of printing, distributing and scanning documents. Characters and fonts that can be recognized in an clean 400 spot-per-inch image become unrecognizable in a skewed and stretched 200x100 spot-per-inch fax-based image.

Recent work at Xerox PARC, led by Dan Bloomberg and David Hecht, has seeded the development of an end-run around these problems in document recognition. The basic technology is a method for embedding digital data in documents as diagonal marks on paper, and reading back this data from images of the documents. These marks, which Bloomberg has labeled simply "glyphs," are robust in the face of the range of distortions introduced in printing, copying, faxing, and scanning. Furthermore, they are scalable, and thus useful for encoding information in documents scanned or printed at very low and very high resolutions. Finally, their simple, regular pattern fails to catch the eye, so they are not visually distracting when embedded in a paper document.

The impact of the embedding of digital data on paper goes beyond that of simple character and layout recognition, for it makes possible the encoding of information not only about the marks that actually exist *on* the paper, but also the *structure* that is behind those marks. Accurately determining the content and structure of a document from its image has thereby enabled us to complete the "document loop," going from structure, to image, to paper, to image, and back to structure

PAPER USER INTERFACE

Robust digital data, together with other image processing techniques, enabled us to create a system in which ordinary paper becomes a computational interface. We call this concept a *paper user interface*. Using the paper itself as the computer interface, as opposed to a screen-based interface that accesses page images, provides substantial benefits to users.

First, people are accustomed to using paper. Not only is it the predominant medium in their daily work lives, but it is intertwined with all aspects of life. This historical reliance on paper means that the infrastructure for its use is common and available: pencils, pens, erasers, typewriters, fax machines, copiers, envelopes, file cabinets surround us in the most mundane of environments.

Second, people know *how* to use paper. This knowledge of "paper's syntax" exists even when the semantic knowledge is lacking: for example, in filling out a form, one can easily determine which boxes to check, and which to write in, on an income tax form, even when one has no idea of what numbers actually go in the box. Thus the learning curve for paper-based computation is minimized.

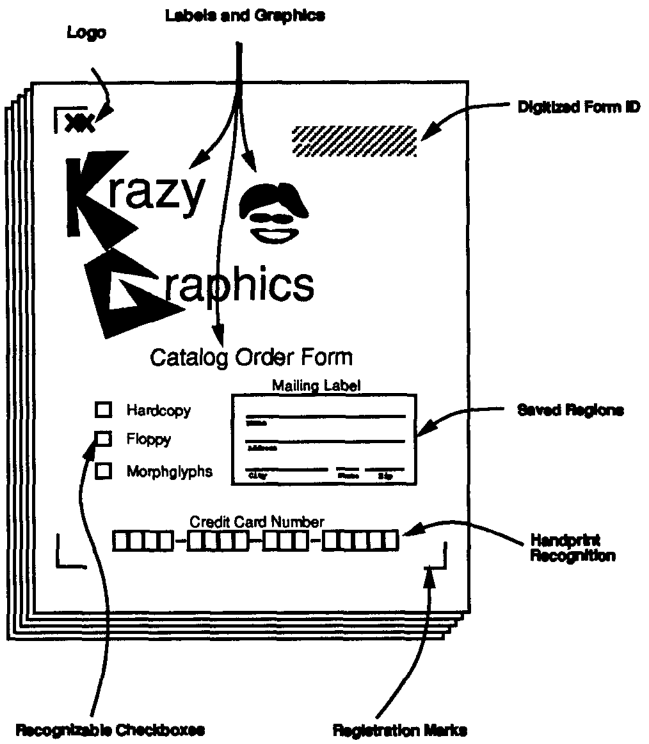

Figure 2: A Paper User Interface

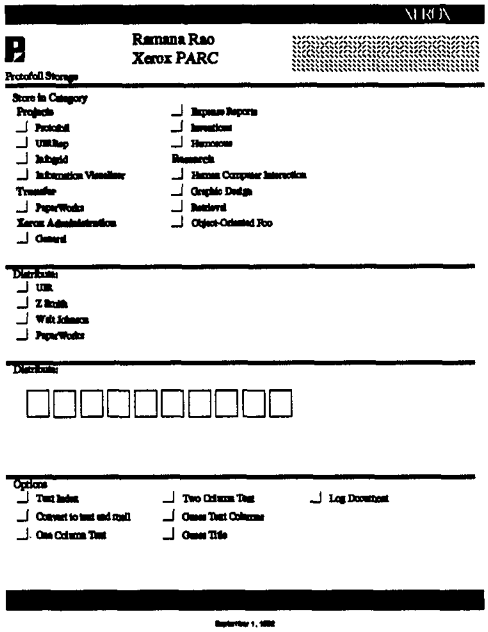

Figure 3: A Protofoil Storage Cover Sheet.

We developed the notion of a *form* as a kind of paper user interface to application programs. A form is a paper interface to a computer application, just as a dialog box is a screen interface to a computer application. Forms combine the simple paper syntax and infrastructure support of paper with the functionality of computer applications.

Figure 2 shows a typical form: it is a piece of paper with machine-readable marks, human-readable marks, and human-markable areas. The logo, form ID, and registration marks are machine-readable marks. These marks are included in every form, and serve to identify it to the machine. The checkboxes, handprint regions, and saved image regions are interface areas for users to give information to the application. Finally, arbitrary text and graphics included on the form allows the application to give information to the user.

Cover Sheets: A Paper User Interface

We explored the use of *cover sheet* forms as a paper user interface to a document services system. A cover sheet is placed in front of a document in a scanner or fax machine connected to a document services system. A cover sheet is a form that allows the user to specify the actions to be taken on the document which follows it.

Figure 4 shows what happens when a person uses a cover sheet to interact with a document services system. In this example, the user uses a cover sheet with document distribution (fax telephone number) options. She marks the intended recipients on the cover sheet, and places it together with a document in the input feeder of a fax machine. The page images are then sent via to the document services system, where it interprets the form image as a request to fax the document following it to a specified location. The document services system then uses its computer fax modem to send the document to its destinations.

Requirements for Paper User Interface

We have identified several conditions required for paper to be useful as a computational medium. First, processing has to be reliable and rapid. Although PC and workstation users have become accustomed to periodic unexplained breakdowns, users of ordinary paper-handling office equipment have developed an intuitive understanding of the failure modes of these machines: if the paper goes in, it will come out or become jammed in a visible, physical way. This statement is true for copiers, printers, and most easy-to-use fax machines. Speed depends on the application: for example, time-dependent applications such as broadcast distribution of faxes must be initiated in a timely fashion. The system must have a mechanism for prioritizing applications if more than one job can be in progress.

Second, the user's time must be separated from the scanner's time. One of the main benefits of paper is that it enables the user to perform the work anywhere that she might ordinarily use paper---at the office, at home, on an airplane, in the car. The physical interaction with the computer---via a fax machine or a scanner---must not be a bottleneck. Since the paper itself directs the processing, the user need not be present while scanning occurs, since her role in the process is

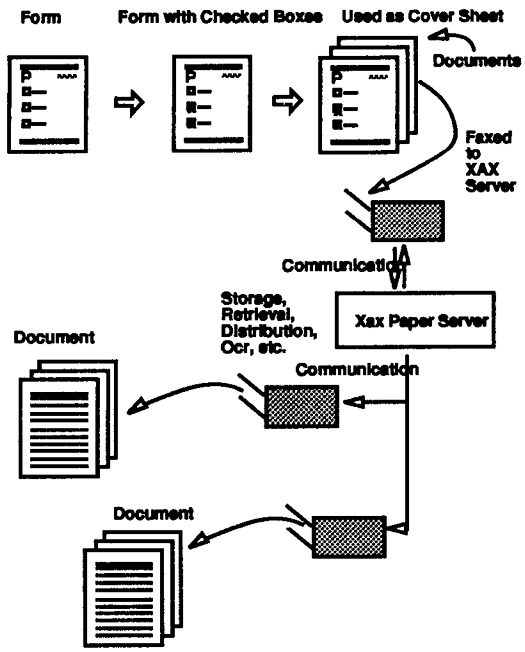

Figure 4: Using Cover Sheets.

over once the paper starts moving. This implies that the system must be capable of batch processing, since the user may want to submit an hour's or day's work at one time.

Finally, since integration with the electronic world was one of our goals, we wanted to provide electronic access to whatever functionality we provided via paper, as well as to the documents stored via paper. An unpublished companion paper [1] describes an electronic filing system that uses paper user interface as well as workstation interface technology.

XAX PAPER SERVER

In order to explore these issues in context, we designed and implemented a prototype "paper server," XAX, which is still in use at PARC. XAX consists of a server, a forms editor, and a client-server interface. XAX provides a framework for supporting applications which use paper interfaces. We describe one such application, Protofoil, and then focus on the components of XAX.

The Protofoil Application

Protofoil is a prototype document management system which provides an electronic filing cabinet that can be accessed by both paper and computer workstation. Protofoil gives users three basic services: document storage, document distribution, and document retrieval. The Protofoil server is built on top of XAX, which provides the paper interface to these document services. (Workstation access comes from a separate client program, not discussed here.)

Users can fax or scan documents into Protofoil, have them indexed by preselected keywords or by the recognized text of the documents, and retrieve the documents based on these features. Indexing, search, and retrieval are all accessable via both paper and computer display [2]. Documents can be distributed by fax or by electronic mail of the recognized text.

The paper interface to Protofoil consists of a set of cover sheets. A Protofoil cover sheet is a configurable paper interface that gives the user access to any pre-selected collection of Protofoil services. Users can create new, personal Protofoil cover sheets using the XAX forms editor. The cover sheet shown in Figure 3 allows the bearer to to store a document in a local Protofoil document database, add any of a list of predetermined keywords, and distribute a document to any of a set of pre-determined recipients.

The XAX Server

XAX consists of a WYSIWYG form editor, a form description language, an action description language, and a server. The SPARC-based server processes the forms, performing recognition of form type, execution of actions, storage of documents, and (optionally) dynamic creation of new forms resulting from retrieval requests.

The XAX server (Figure 5) is charged with the recognition and interpretation of incoming documents, which arrive via scanner or fax. Independent scanner and fax driver programs, running anywhere on the Xerox Internet computer network, create sequences of page image files in a special spool directory known to the Xax server. When a new image sequence arrives, the server first splits it into one or more runs of pages, each consisting of a form followed by zero or more pages that it pertains to.

Decoding the glyph information, the server determines the identity of the form, locates the appropriate action description file, and processes the page image, looking for the glyphs, clipped regions, and checkboxes it expects to find there. The server passes that information, along with pointers to the individual page images, to the action procedure for the form, which is then free to use this information in any way required.

XAX Form Editor

The form editor, XAXEDIT, assists the user in designing forms to meet new information-processing needs. Superficially similar to structured graphics editors like MacDraw, XAXEDIT presents the user with a palette of elements, such as checkboxes and text strings, with which to draw a new form. Those elements can be broadly lumped into two classes: those with semantics and those without. Examples of the former include checkboxes and clipped regions, and examples of the other class include text, line art, and bitmaps.

XAX Forms

Forms must always bear certain registration marks and glyphs

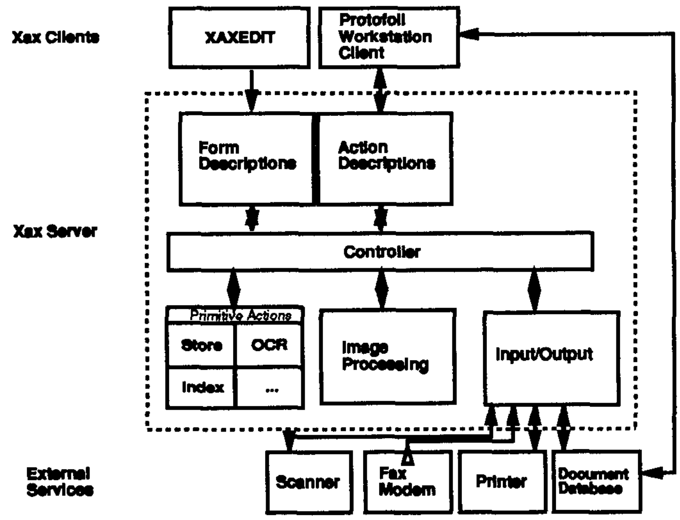

Figure 5: The XAX Server.

in given positions - this is how the system recognizes them as forms. The user is free to add other elements to the form, as determined by the application she is creating. A completed form is thus comprised of three types of information: a form description, which lists all elements of the form in a simple Lisp-based language, and which XAXEDIT can both read and write; Action descriptions, which represent the semantic elements of the form and the actions they perform, and drive the XAX server's image-processing routines; and a description in either Postscript or Xerox Interpress format that encodes how to print the form. (See Figure 3.) A pointer to the files containing this information is encoded in the glyphs on every form.

While forms can be created manually, the system is also required to compute them automatically in response to user queries via another form. For example, upon storing any document in Protofoil, the user needs some way of accessing and interacting with it later. The Protofoil-store action procedure automatically creates a new form that "points at" the document, bearing checkboxes that allow the user to print or delete it. This ability to compute new forms on the fly helps to close the paper--computer loop.

Image processing

In order to recognize forms, several types of image processing operations are performed on every document entering the system. Logo and registration mark detection are performed on every page in order to determine the presence of a form, correct for skew, and enable batch processing of jobs. Glyph recognition enables to server to decode a unique 256 bit ID for each form. Another operation locates each checkbox on a form, and determine its status, checked or unchecked, while others lift and scale hand-printed regions that contain titles, recipient names, and other information a user may handprint on a form. Finally, the system uses the ScanWorx™ system from Xerox Imaging Systems for optical character recognition.

Document Database

XAX provides a generic interface to a document database. The initial design of the system was based upon a remote database, but was judged by users to be too slow. The current design utilizes a local database provided by a system built on top of XAX, Protofoil. Although storage is local to each user's primary workstation, remote access is enabled both via electronic and fax networks.

EXPERIENCE

In our experience with XAX, we found three classes of user: casual users, who used the system infrequently or for a short time, document storage (Protofoil) users, and document services (mainly OCR) users. The top four document storage users stored over 400 documents, totaling over 1300 pages; the top two OCR users scanned 40 documents, totaling over 140 pages.

All users previously had access to scanners, workstations, document storage systems and OCR software. Our interviews revealed that the primary advantage of paper user interfaces over existing systems was ease of task initiation; the users felt that document entry, retrieval, and OCR were simplified by elimination of cumbersome workstation interfaces.

To compensate for the static nature of their paper forms, our users came to modify and add to their form sets frequently. We observed that the patterns of change were different for the document storage and OCR users.

Document storage users tended to have three phases of form use: introduction, specialization, and separation. In the introductory phase, users experimented with one or two forms created for them, with storage options based on categories they supplied. In the specialization phase, users developed new forms, adding and deleting categories from the initial list. In the separation phase, users grouped related sets of categories and moved them onto separate forms.

OCR users underwent different phases of form use. They tended to do little experimentation, but instead used a simple form for mailing the recognized text to themselves. Eventually, they would request new forms allowing access to parameters of the OCR package, such as specification of column layout or language. Unlike document storage users, OCR application users tended to reuse filled-in forms with complex option selections, exploiting the persistent nature of the paper medium.

Complexity of interaction is a significant barrier to the use of any system. In document service systems, where users are handling large volumes of paper, we found use of a paper interface minimized complexity by keeping interactions within the single paper domain.

CONCLUSION

Pen-based computing, as instantiated in notebook computers, leverages only some of the properties of paper. The same holds true for "paper-like" interfaces [3], which are more appropriately called "stylus-based." The use of paper in the modern world persists *because* of its physical properties, not only despite them. Paper is convenient, requiring no batteries. It is light, readable, and durable. It has a tangible, persistent existence that is valued in legal and business transactions. These properties do not diminish the value of alternative media, such as notebook computers, but they do imply that paper has utility that will not simply disappear with the increasing electronization of the world. And as long as a significant amount of paper exists among us, then pragmatically, one of our main goals should be {\em integrating}, and not *removing* paper from our electronic lives.

In this paper we have argued that the physical and historical properties of paper make it suitable for many kinds of work, and that paper itself can leverage these properties and additionally become a computational interface. The XAX Paper Server was created to demonstrate the infrastructure needed to enable computational paper interfaces. The basic technological advance needed to build a scalable PaperUI processing system was the capability of embedding data on paper, thus enabling us to robustly determine the identity and semantics from paper forms, our basic interface mechanism. The feasibility of XAX is further demonstrated by the existence of a PC software product based on it and its successors, Xerox PaperWorks™.

REFERENCES

[1] R. Rao, W. Johnson, L. Klotz, and S. Card. Bridging the paper and electronic worlds: An electronic file cabinet for the information worker's paper documents. 1993. *Unpublished manuscript.*

[2] *D. R. Cutting, J. Pedersen, and P-K. Halvorsen. In* object-oriented architecture for text retrieval. In Conference Proceedings of RIAO '91, Intelligent Text and Image Handling, Barcelona, Spain, *pp. 285-298, April, 1991.*

[3] C. G. Wolf, J. R. Rhyne, and H. A. Ellozy. Paper-like user interfaces. In Designing and Using Human-Computer Interfaces and Knowledge-based Systems: Proceedings of the Third International Conference on Human-Computer Interaction, *pp. 494-501. Elsevier, 1989.*