

The Missing Link: Augmenting Biology Laboratory Notebooks

Wendy E. Mackay¹, Guillaume Pothier¹, Catherine Letondal², Kaare Bøegh³ & Hans Erik Sørensen³

¹INRIA

Domaine de Voluceau
Rocquencourt, B.P. 105
78153 Le Chesnay Cedex, FRANCE
{wendy.mackay,guillaume.pothier}@inria.fr

²Institut Pasteur

Service d'Informatique Scientifique
28 rue du Docteur Roux
75724 Paris CEDEX 15, FRANCE
letondal@pasteur.fr

³Aarhus University

Åbogade 34
DK-8200 N, Aarhus,
DENMARK
{boegh,hes}@daimi.au.dk

ABSTRACT

Using a participatory design process, we created three prototype augmented laboratory notebooks that provide the missing link between paper, physical artifacts and on-line data. The final *a-book* combines a graphics tablet and a PDA. The tablet captures writing on the paper notebook and the PDA acts as an "interaction lens" or window between physical and electronic documents. Our approach is document-centered, with a software architecture based on layers of physical and electronic information.

KEYWORDS: A-book, Augmented laboratory note-books, Augmented reality, interactive paper, information layers, interaction lens, SVG

INTRODUCTION

Research biologists face a complex information processing task, managing physical paper documents, physical research specimens, on-line documents and on-line services. They require paper laboratory notebooks for legal, historical and practical reasons, but they are also active computer users. This forces them to constantly juggle paper and electronic forms of the same information. We are exploring the concept of an "augmented laboratory notebook", to develop a design space that provides the missing link between this physical and on-line information.

We created three prototypes, with active participation by research biologists, archivists and managers at the Institut Pasteur in Paris. Each provides solutions to particular user needs, retaining the best aspects of paper, linking physical and electronic data, and creating on-line access to information that can be preserved for posterity.

The final prototype, the *a-book*, includes a paper notebook and introduces a novel interaction technique called the *Interaction Lens*. This palm-sized physical "window" between the physical and digital realm can be used as a see-through tool or lens. The software is designed to handle persistent data, manage links and facilitate retrieval of a

variety of archived information, including paper and electronic documents as well as other physical objects, such as live animals, research specimens and data images. We use a document-centered approach, with a multi-layer information architecture that crosses the boundary between physical and on-line objects.

This paper begins by presenting the results of our fieldwork with users. We then describe three working prototypes whose functionality were defined by these users and which explore different characteristics of an augmented laboratory notebook. We conclude by describing the document-based data architecture we developed to handle multiple layers of physical and electronic persistent data.

RELATED WORK

For several decades, the trend has been to introduce computer systems that replace paper documents. The goal, of course, is to take advantage of the computer's increased processing power and communication capabilities. Many believe that this makes paper-based systems obsolete. For example, Lysakowski [15], articulates many problems with paper notebooks and praises the many advantages of computers. Unfortunately, he ignores paper's benefits and the potential problems introduced by computers. For example, we can still read Pasteur's century-old paper notebooks, but have lost on-line files from five years ago because they were created on now-discontinued computers.

Researchers have shown that paper has many benefits difficult to replace with standard monitor/keyboard/mouse interfaces. Paper is easier to read [23] and meets subtle needs for people working collaboratively [17]. It is also lightweight, ubiquitous, inexpensive and easy to use. This may help to explain why the failure rate for replacing paper-based systems is so high [13]. Multi-billion dollar projects, e.g. for the U.S. tax service and air traffic control, have been rejected out of hand. As Sellen and Harper [26] point out, the paper-less office is a myth.

One strategy for including the benefits of paper is to create interfaces that directly simulate it. For example, XLibris [27] simulates a paper notebook with a tablet computer that displays on-line or scanned documents, and handles free-form annotations with digital ink. Other researchers use PDAs to help engineers take notes [9] or share them [5].

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'02, October 27-30, 2002, Paris, FRANCE.

Copyright 2002 ACM 1-58113-488-6/02/0010...\$5.00.

Another strategy does not simulate paper, but retains it directly as the user interface medium. The Digital Desk [31] and variants [18] provided early examples of how to manage the complex interplay between paper documents and on-line systems. Since then, we and other researchers have explored augmented books [1], engineering drawings [20], flight strips [19], and maps [22]. Stifleman [28] created a paper notebook with sound annotations, whereas Heiner et al. [11] created a "paper PDA" that enhances personal notebooks and organizers with stickers that are linked to on-line information.

Many questions remain as to how best to integrate physical and electronic material. Our research begins with real-world settings in which attempts to replace paper with have failed and where users have legitimate reasons to use both off-line and on-line versions of documents [16]. Such settings can help us obtain a deeper understanding of the critical features of both paper and electronic documents, while providing a fertile testbed for exploring how to manage information in all its manifestations.

This paper describes our work with biology laboratories and their notebooks. The goal is to create an augmented notebook that meets the legal and ease-of-use requirements of paper, while benefiting from the communication and data analysis features offered by the computer.

FIELD STUDY

The Institut Pasteur in Paris, founded by Louis Pasteur in 1887, is a semi-private, non-profit organization devoted to biological research. We used a participatory design approach in which we observed, interviewed and worked with over twenty research biologists their laboratories in for over two years. We also included archivists and research managers, who have different perspectives and correspondingly-different design priorities. Most of the actual design work was conducted during a series of brainstorming sessions and video prototyping workshops. We asked participants to bring their own notebooks and describe specific examples of uses or problems they had recently faced. We also discussed clips from our interview videos, to identify common themes. We all worked together in small groups to develop real-world use scenarios, which were then discussed by the entire group.

We ran five video prototyping workshops, spread throughout the project, in which small groups of participants created mockups and shot video clips that showed how they would like to interact with an augmented notebook. (See [21] for a description of these techniques.) The programmers on our team (who also participated in the workshops) based the design and specific features of each of the three prototypes on these video scenarios. All of our examples are drawn from real situations faced by biologists in the study and our prototype paper notebook is based on re-written copies (with slight modifications) of a notebook contributed by one of the participants.

We held evaluation forums to present working prototypes and gather feedback about what worked and what did not. Unfortunately, the details of the participatory design are

beyond the scope of this paper, so the next section summarizes the requirements articulated by the three user groups design of the three prototypes.

Summary of User Requests

Research biologists: The primary users of laboratory notebooks are the biologists themselves, who create the content and use it as a resource for future research. It is important to understand that these notebooks are not scratch pads for jotting down ideas or quick reminders: they are official documents and the product of careful reflection. Notebooks are intended as both a personal record and a public document. Research findings are always dated, which is critical for future claims, either of discovery or for patents. The notebooks track procedures, often repeated with minor variations, results and analyses and may be reviewed the following day or years later.

We videotaped a number of biologists in the process of updating their notebooks. Most set aside an hour or two, two or three times a week. They gather relevant documents, such as copies of experimental procedures, hand-written notes taken during an experiment, published research articles, data from the current experiment, and, of course, the laboratory notebook itself. Most use preferred stationery supplies, including highlighter pens, tape or glue, and scissors. They examine their physical data and perhaps annotate it, e.g., using a fluorescent pen to highlight DNA patterns or a silver pen to outline organs on an X-ray image. They use the computer to search gene databases, perform mathematical analyses, or graph data.

The next step is to decide precisely what to record. This involves careful reflection: the notebook must act as an honest record of what happened, but it is also a personal interpretation of the data, made by the individual biologist. They handwrite the date and information about the current experiment: the procedure, the data or the analysis. They also attach and annotate externally-created printed documents or images. The result is a concise, chronologically-organized, permanent record of the biologist's hypotheses, experimental procedures, and research results.

Notebooks are extremely multi-media documents. In addition to writing and drawing, biologists paste in images, such as X-rays and digital microscope photographs. They also add computer printouts, including repeated procedures, raw data and analysis results. Some biologists even paste in physical lab specimens, films or gels from their experiments. To make matters even more complex, notebooks are also used to point to external objects, such as refrigerated or frozen specimens, surgical results stored on slides and even live laboratory animals. Notebooks also provide links to on-line information, including important research articles and web sites.

All the biologists we spoke to are heavy computer users, but most appreciated the simplicity and flexibility of their paper notebooks. They particularly liked the ability to highlight or annotate images or data and create free-form drawings to illustrate a point. However, some tasks are cumbersome, such as creating a table of contents or index

of their experiments. For some, searching is also a problem, especially when the same procedure is used repeatedly, with tiny variations. Several also described situations in which they needed information from another colleague's notebook, but could not find it.

The primary concern of these biologists, with respect to their notebooks, is to support the design and execution of their experiments and the subsequent analysis necessary to publish research articles. They are also aware that the notebooks serve as proof of discovery of ideas and as the basis for patent claims, although few bothered to have their findings witnessed and dated, as required. Only two systematically had their findings officially "witnessed".

Genetic research is highly competitive [3] and research biologists race to publish their findings. We interviewed one biologist who had determined that the Green Monkey in Africa was the source of the AIDS virus, but the first version of her paper was rejected, which allowed another researcher to claim first discovery. Biologists actively seek ways of saving time, through their notebooks or on-line analysis systems. They find paper very efficient: the quantity of hand-writing is small, formatting is easy, with sketches and annotations added at will, and they don't have to fight with an on-line text editor.

Archivists: The Institut has an archives department that is charged with preserving the intellectual property of the Institut Pasteur. Laboratory notebooks are the property of the Institut and are handed over to the archive department when the biologist leaves. Archivists are concerned with both the research and the historical value of notebooks. They operate on a different time-scale, and, with over 300 meters of notebooks, must decide not only what to preserve, but also how and what to throw away. They need methods of indexing and accessing the data they do keep. They worry about storing paper-based data, since ink may spread over pages, glued-on items may become separated and media such as X-rays may decay. They also face a problem when links are lost between notebooks and physical specimens stored in secure facilities or refrigerators. However, they are even more concerned with the growth of on-line data, which quickly becomes inaccessible when stored in obsolete software formats on out-moded computer media. Archivists are also conscious of their role as historians: today's inconsequential finding may become extremely important in twenty years, as other research changes the context in which it is interpreted.

Research managers: The Institut Pasteur obtains a third of its financing from patents and would like to increase this percentage as private and government donations decrease. Although they understand that biologists like to work independently, the managers feel obligated to impose rules on the creation and archiving of laboratory notebooks that ensure that patentable ideas will be pursued. The Institut Pasteur is particularly sensitive about this, after providing a sample of the AIDS virus to an American lab, which proceeded to patent it. After many years of legal battles, it was revealed that the New York researchers had "whited

out or otherwise obscured" their laboratory notebooks in order to lay claim to prior discovery of the virus, and the Institut Pasteur was finally awarded patent rights [8]. But they lost millions of dollars in revenues. The current administration has redesigned laboratory notebooks so "Institut Pasteur" appears diagonally across each page, and a strip at the bottom asks for the signature and dates of the experimenter and a witness. Biologists are issued numbered and dated notebooks, must use ink and may not alter the notebooks. Associated documents are not allowed (although every biologist we interviewed broke this rule). The administration is developing a scheme to scan and preserve pages that biologists believe are patentable, but biologists are resisting, due to the extra time and effort.

TECHNICAL CONSIDERATIONS

We created three research prototypes, which must each must solve three technical problems:

- capturing user information, e.g., hand-writing,
- displaying information in response to commands, and
- managing the link between physical and on-line data.

For each, we can choose among a number of technical solutions, each with advantages and disadvantages. To capture hand-writing, the most precise and easily-programmable solution is a graphics tablet, which permits both writing on paper, with a pen that leaves ink, and writing on a PDA or LCD tablet, with a non-inking pen. We decided against using a camera, as in our earlier work with the Digital Desk [18,20] because we cannot control lighting and it is not portable. Another promising alternative is a pen that captures gestures directly, such as Anoto®, which would address some portability issues.

The second problem is how best to display information. In some use settings, such as when the biologist is already seated next to a computer monitor, the easiest solution is the computer screen itself. Other solutions include portable displays, such as the on-line data window associated with a CrossPad™ or the screen on a PDA or LCD tablet. In the future, technologies such as Xerox's electronic paper (now being produced by Gyricon) and MIT's electronic ink [4] will enable us to display information on paper directly.

The third problem is how best to manage the link between physical and electronic data; the solution depends on the specific requirements of the application. In our case, we need to know which paper page is being written upon and the precise location of the writing with respect to the page. A separate problem involves tracking physical objects, either associated directly with the notebook (printouts and images) or independent (research animals and specimens).

Numerous strategies exist for identifying specific paper pages, ranging from the ubiquitous barcode, to glyphs [10] and cybercode tags [24]. Another approach is optical character recognition, with devices such as the PocketReader®, a pen-like device that reads type-written text and displays the interpreted characters in a small window. This avoids the need for non-human-readable codes but works only with typed characters, e.g. preprinted

page numbers, not hand-writing. The main difficulty with these approaches is that the user must remember to verify the current page before writing. Another solution is to permanently attach a bar-code reader or device like the Glyph-o-scope [10], so that when the page is turned, the code for the new page is registered automatically. Researchers have also been exploring strategies for tracking tagged objects from software programs, including Xerox's active badges [12] and electronic tags [30]. Many use RFID technology [1], commonly used in tracking animals, because it is both reliable and inexpensive.

The next section describes each prototype, its goals and functions, the hardware and software configuration, and the key user issues addressed. We conclude with a discussion of the problems resolved and the issues that remain.

Tablet Prototype

The Tablet prototype (fig. 1) was designed to allow biologists to continue to write on paper laboratory notebooks in the normal way, while simultaneously creating an indexed, searchable on-line version. The choice of hardware was influenced by the primary use setting we observed: the biologist's office, usually a desk in a corner of the lab, next to a computer and monitor.

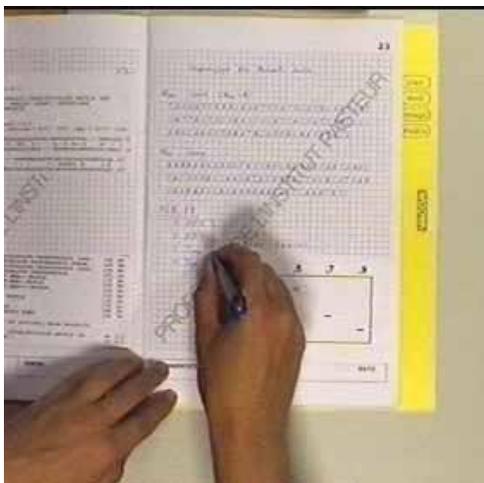


Fig. 1: Tablet prototype: Handwriting in ink on the paper notebook is also captured by the graphics tablet underneath. Command menu appears to the right.

The prototype hardware includes an oversize A4 WACOM graphics tablet, an inking pen (for writing on paper), and a PC running LINUX. A PocketReader detects the current page number. We created a prototype paper notebook with the same pre-printed page format as an official Pasteur notebook. By fixing this to the graphics tablet, we can automatically register the location of hand-written text with respect to the borders of each page. (The tablet can detect writing through the thickness of the notebook.) We took advantage of the fact that the detection surface is slightly larger than the normal A4 sized notebook pages, and added a paper command menu (the yellow band on the right). The software was written in C++ and the individual on-line versions of each notebook page were stored as XML (Extensible Markup Language) files.

Using the paper notebook: The biologist opens the notebook to the desired page and uses the PocketReader to register the page number. He uses the inking pen provided with the tablet and writes or sketches normally on the page. Each gesture is recorded, time-stamped and positioned on the current page. The gestures are then reconstructed to re-create an on-line image of the writing.

The biologist can also make special annotations and issue commands via the menu at the right. We use Rubine's algorithm [25] to detect underlines and boxes. Underlining a text string tells the system to recognize the characters as text. For example, to create a URL, he hand-prints the web address in the notebook and underlines it. The interpreted text appears on the screen image of the notebook and can be corrected if necessary. He can then tap on either the paper or the on-line URL to establish a network connection (fig. 2). The system automatically updates a printable index that includes all underlined, i.e. recognized, words with time-stamps and page numbers.

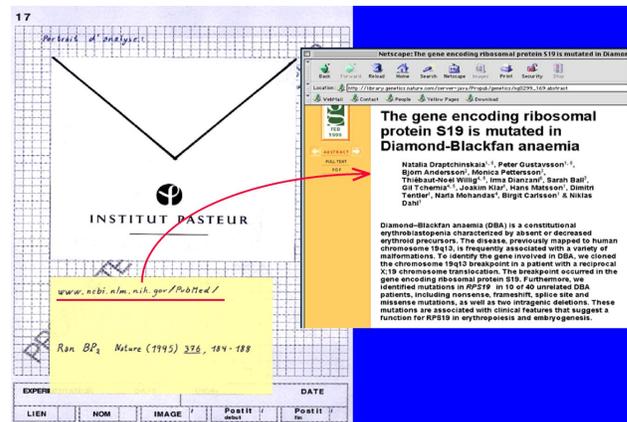


Fig. 2: Following a URL from the notebook page.

The biologist can also create searchable names. For example, in order to re-use a particular procedure, he taps "nom" (name) and then underlines the procedure name, e.g. PCR (Polymerase Chain Reaction). He can also draw a box around the associated procedure and capture its image. He can later refer to the same procedure from a different page, by writing "PCR", tapping "lien" (link) and underlining the text. If he then taps on "PCR", from either the paper or the electronic notebook, the image of the procedure on the original page appears on the computer monitor. Boxed information, which can include sketches or other difficult-to-recognize gestures, are saved as images and also appear in the automatically-generated table of contents.

Tracking on-line/off-line documents: The biologist can paste in externally-created documents, usually computer printouts or images. The problem is how to tell what was actually printed, so we can track it in the on-line version of the notebook. For the former, we take advantage of the fact that most biologists at the Institut use Letondal's PISE [14]: a web interface for over 150 molecular biology command-line driven programs. PISE allows biologists to easily combine analysis programs and create customized procedures. We modified PISE so that each analysis result

has a unique identifier, which is printed on the result page. We use this identifier to retrieve the original on-line analysis result. Images are more difficult: digital images are sometimes coded and tracked, but analog images must be treated as tagged physical objects.

To add a printed document or coded image, the biologist first taps "image" on the menu (fig. 3). If the code is already printed on the document or image, he scans it with the PocketReader. If not, he hand-writes the code on the page and underlines it. He taps two diagonal corners where the image will be positioned on the page. When viewed on-line, a rectangular window appears at the indicated position displaying the printed data analysis result. Note: biologists usually trim the pasted-in page, but we cannot track this directly. Our solution is to provide the correct-sized on-screen window and the original full page. The biologist can then reposition the result within the window, if necessary.

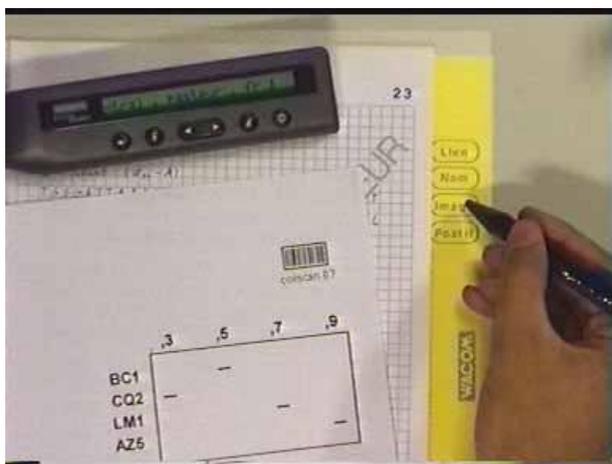


Fig. 3: Adding a printed analysis result, using the PocketReader (top left) and menu (right).

This prototype also handles Post-it® notes, considered to be a layer of temporary information. He taps "post-it" and then places a physical Post-it note on the page. Subsequent writing on the post-it is recorded on a separate, layer of the page. He must tap on "Post-it" again when he is finished.

Using the on-line notebook: The on-line version of the notebook looks very similar to the off-line version, presented as a series of pages with hand-written notes and other data. It also includes an automatically-generated table of contents, using items (URL, procedure names, etc.) selected by the biologist on the paper version. The biologists' primary use of the on-line version is to browse or search the notebook. In addition to flipping through the pages, she can also scan the whole notebook from thumbnail images of each page. She can also select "search" from a popup menu and specify any underlined text. The search can also be restricted by time, e.g., asking for all instances of PCR between May and July this year.

Handwriting recognition: For most biologists, recognizing names and URLs is sufficient. But we interviewed one biologist who wanted to be able to interpret hand-written sequences of DNA (the letters A T C G). We added a "recognize" feature to the on-line version of the notebook.

The biologist highlights the hand-written text to be recognized (Fig. 4) and selects "recognize" from the pop-up menu. The system displays the recognized text in a window and allows her to make any corrections. She can continue to process the sequence by running it through PISE and selecting any of 150 on-line data analysis programs. Of course the final result can be printed and pasted into the notebook, using the procedure outlined earlier.

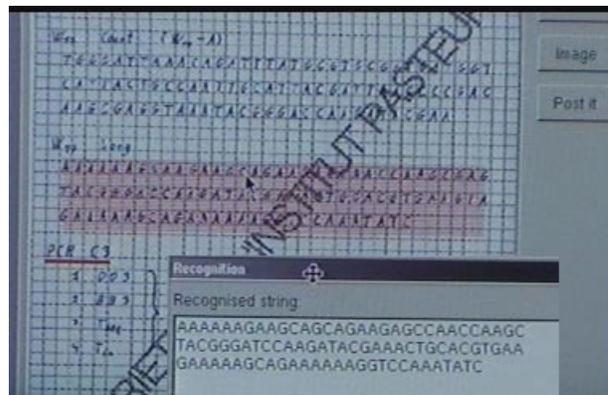


Fig. 4: Recognizing a DNA sequence

Evaluation forum: We presented a live demonstration of notebook and allowed the biologists to try it out. Most felt the hardware configuration (which required that the notebook be updated only when next to the computer) was reasonable. However, several described situations in which they required the ability to make updates to specific pages in other locations (usually another lab or animal room). The main advantages of the system, from the biologist's perspective, had to do with the search facilities, including the ability to automatically retrieve information by time and date (since every gesture is time-stamped) and to track sketches or odd-sized blocks of information, in addition to names. Some of the features proposed by individuals were not viewed as essential by the rest: automated recognition of URLs was deemed only mildly interesting and only the person who proposed it was interested in the Post-it feature.

Not surprisingly, archivists were particularly interested in the automatically-generated table of contents, since biologists rarely create them. Managers were concerned with the cost of the system, but wanted the possibility of time-stamping and tracking potentially-patentable pages. From our perspective, this prototype showed the potential for an augmented laboratory notebook, but page recognition was too cumbersome and many biologists felt we needed to address portability. So we developed the CrossPad prototype, (based on the system from Cross and IBM).

CrossPad Prototype

The Cross-Pad prototype was designed for situations in which biologists carry their notebooks with them, either when they participate in research meetings, travel or work at home. The hardware uses a CrossPad portable graphics tablet, designed by IBM and Cross, which allows the user to write on an ordinary pad of paper. The system is battery operated and can store up to 50 hand-written pages, which must then be uploaded to a PC. The product, now

discontinued, emphasized hand-writing recognition. Users must go through a lengthy training session, write in a restricted manner and accept a high level of errors. Since hand-writing recognition is not a priority in our application, we just captured and time-stamped gestures, freeing the user to write naturally, except when entering names. The lack of interactivity and feedback made it impossible for the user to correct errors as they occurred, making it much more likely that such errors would not be caught.

Like the Tablet prototype, all gestures were time-stamped and it was possible to use the same interaction techniques (underlining, boxing, etc.) to facilitate later on-line search for information. Because of the smaller size, we redesigned the bottom of the standard notebook page (fig. 5) to include the menu items that formerly appeared to the right of the graphics tablet in the Tablet prototype.

EXPERIMENTATEUR		DATE		TEMOIN		DATE	
LIEN	NOM	IMAGE	Post it debut	Post it fin			

Fig. 5: CrossPad prototype menu at the bottom of each paper page, under the temoin (witness) box.

Hand-writing Recognition: We used the IBM recognition software that came with the CrossPad, to recognize text. Unfortunately, this software only runs on the host computer, not the CrossPad itself, so recognition can only be performed after the page is uploaded to the on-line version of the notebook. Our goal was to make the interaction identical to the Tablet prototype. However, the CrossPad is a closed system and we could not program the one-line feedback window, making it impossible to provide feedback about errors as they occurred.

Evaluation forum: Although the biologists liked the concept of the CrossPad prototype, it simply did not work in practice. The advantages of portability were more than offset by the lack of interactivity. We decided to pursue a different solution, building upon the Tablet prototype, which worked well for most situations, but adding a small, portable, interactive device that could be used with or independently of the paper notebook.

A-Book prototype

The third prototype is called the *a-book*, for *augmented notebook*. The idea was inspired by two biologists who needed to record genetic, lineage and other information about specific mice in their experiments. They wanted a portable database tool for recording the characteristics of each mouse in the animal lab and linking them to experimental results recorded in the notebook. This was a different variation of the portability problem: they needed to link the notebook to external objects and carry snapshots of relevant pages with them. The idea was also influenced by a biologist working with the tuberculosis virus, who needed a small portable device to record information during surgery. Access to the operating room is highly restricted and physical objects cannot be carried back and forth, so data needs to be transmitted electronically and linked to the notebook outside. Note that we have refined the concept of portability: few biologists need the full notebook when they

move to these other locations. What they do need is the ability to enter information in a form that does not have to be re-entered when it is time to update the official notebook. This prototype also explicitly addresses the archivists' problem, i.e. to store data over long periods of time and maintain links to physical objects.

The A-book hardware includes an A4 WACOM Intuos graphics tablet with three peripherals: an inking pen (for writing on paper), a non-inking pen (for interacting with the Interaction Lens) and a 4D mouse for tracking the position of the *Interaction Lens* (fig. 6). The latter is a Compaq iPaq 3660 handheld PDA, with a WACOM 4d mouse sensor attached, to enable us to track its position and orientation with respect to the paper page. As with the Tablet prototype, we attach our own printed version of the official Pasteur notebook to the graphics tablet. A small, screenless Single Board Computer fits underneath the tablet and runs the database server and other a-book software. The software is written in Java, except for code written in C to interface to the graphics tablet.

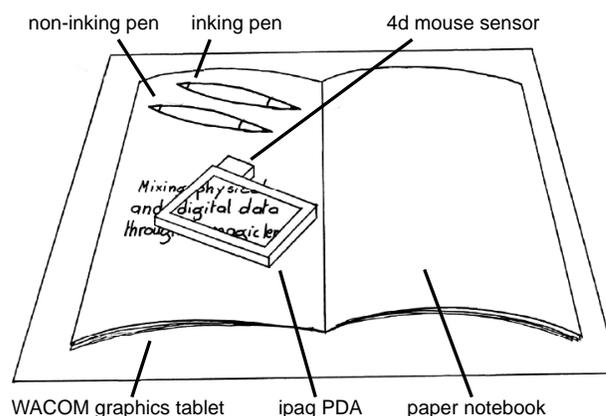


Fig. 6: A-book components

Using the a-book: The basic interaction with paper is similar to the first two prototypes. The system detects the biologist's gestures and uses them to reconstruct individual notebook pages. These pages can be viewed on an ordinary computer monitor if desired. However, the primary use involves the Interaction Lens (fig. 7).

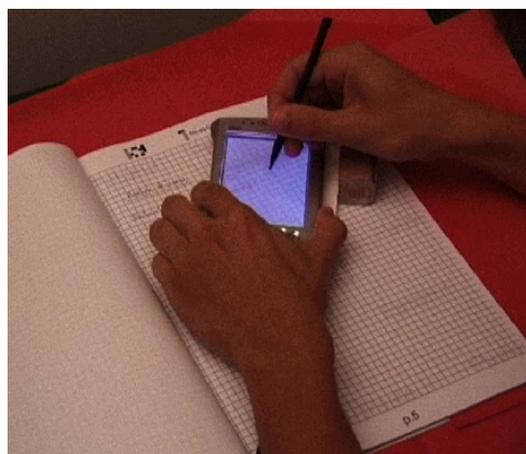


Fig. 7: A-book Prototype: Using Interaction Lens to annotate or create links to external physical objects.

Influenced by Xerox's on-line toolglasses [29] and magic lenses [2] and Fitzmaurice's Chameleon [7], the Interaction Lens acts a physical, movable filter and as an interaction tool. The biologist places it a paper notebook page and immediately sees the underlying page, including any previously recorded hand-written text or data. The viewer's immediate impression is that of a transparent window. The illusion is maintained for any orientation of the PDA.

The biologist uses the non-inking pen to select items from the page through the Interaction Lens. For example, he can create a link to the name of a procedure. If he presses the pen and does not move (with some tolerance) for half a second, a pop-up menu appears. He selects "lien" (link) to define the first anchor of the link, which is displayed as a small icon (fig. 8). He can then specify the target of the link to be an item on another notebook page, using the same technique, or call up another program, called the object-tracker, to associate it with a tagged physical object. Although this is a two-stage process, the biologist need not identify the second anchor point of the link immediately. The system maintains a per-user list of pending actions, which can be consulted when the main Interaction Lens menu appears. This allows biologists flexibility when engaged in complex tagging operations, such as registering and tracking live mice, while allowing them to remember that some actions are still pending.

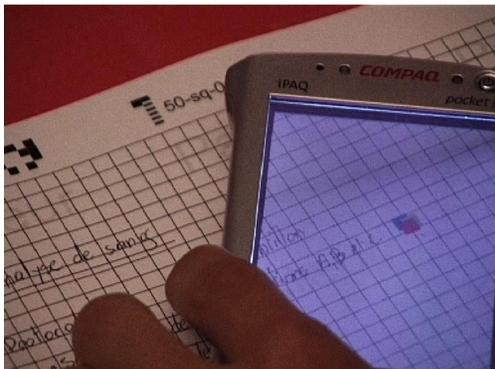


Fig. 8: Interaction Lens: selecting an anchor for a link to an external, physically-tagged object.

The biologist can add items to the table of contents either by drawing a rectangle around the item on the paper page, as in the Tablet prototype, or by placing the Interaction Lens over the desired location on the paper notebook and selecting "toc" (table of contents) from the pop-up menu. A resizable, translucent window appears. He can adjust the size of the window as necessary (fig. 9) and save the entry by clicking the "ok" button or else "cancel".

The table of contents is updated in real time and the biologist can browse it via the Interaction Lens. Entries, which are graphical snapshots of human-written text or sketches, are displayed chronologically, in the order in which they were originally time-stamped, with date and page number. The biologist can also use the Interaction Lens to make on-line annotations of the paper notebook pages. Such annotations make it possible to add informal remarks or ideas to an other-wise formal document.

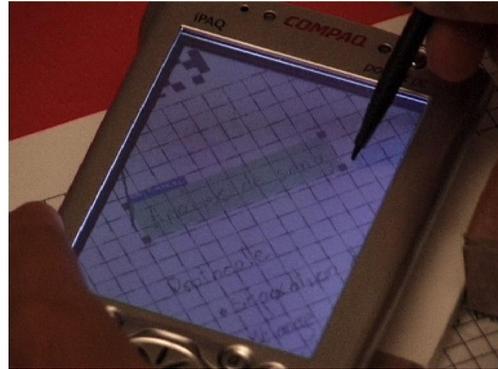


Fig. 9: Interaction Lens: Identifying an item to be added to the table of contents.

The Interaction Lens is also used as a stand-alone PDA, which can be used to browse the notebook or perform other tasks. Fig. 10 shows a browser (left) with a thumbnail of a notebook page, which selects the current page and helps determine whether a particular page has annotations or links. The object tracker (right) maintains a database of externally tagged objects. The Interaction Lens can also display the table of contents and has a print function for printing new notebooks, individual pages, the table of contents and labels.

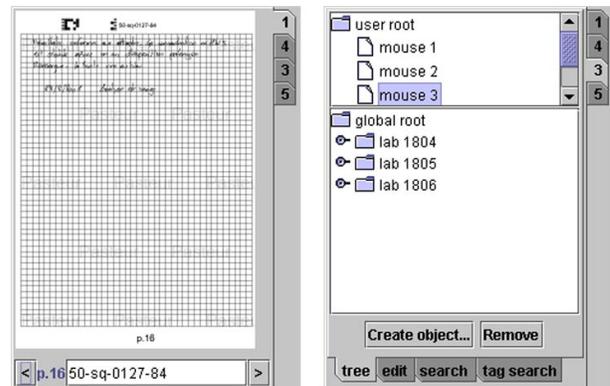


Fig. 10: Interaction Lens, stand-alone mode. Left: Thumbnail notebook page. Right: Object tracker.

Evaluation forum: We presented a live demonstration of the a-book and let the biologists work with it. This prototype was the most successful; biologists appreciated the portability of the Interaction Lens and its use as a simple database. The archive department is interested in pursuing the project and we are now discussing how to implement a version that is robust enough to be used on a daily basis.

SYSTEM ARCHITECTURE

The architecture of the a-book builds upon Fekete & Beaudouin-Lafon's [6] multi-layer model, extended to incorporate layers of physical as well as electronic information. The a-book is distributed between two computers: the interaction lens PDA and the host computer for the rest of the system.

The Layered Model and XML Database

The multi-layer model decomposes a graphical application into a set of layers stacked like transparencies. Interaction is handled by attaching tools to each layer. When an event

occurs, it is passed down to each layer in turn until one of them handles it. For the a-book, we introduce physical layers corresponding to the pre-printed notebook and the physical ink, and layers that link physical and on-line information, such as an on-line annotation that links a handwritten text to a physical specimen.

Fig. 11 shows three layers: the paper, ink and annotation layers. The paper layer correspond to the physical paper, pre-printed with a grid, the Institut Pasteur logo and page and notebook number. It cannot be changed. The ink layer corresponds to the handwriting. Handwriting is captured by the tablet, time-stamped and stored as raw data. Handwriting can only be added to a page and cannot be changed. Later interpretations of this raw data, such as character recognition or changes made the user, are stored in a separate layer. The annotation layer consists of photographs, Post-It® notes, computer printouts, etc. They are also time-stamped and their location on the page is recorded. When the annotation refers to on-line information, a link to this information is also recorded. All the layers are represented with Scalable Vector Graphics (SVG), a vector-based format for graphics.

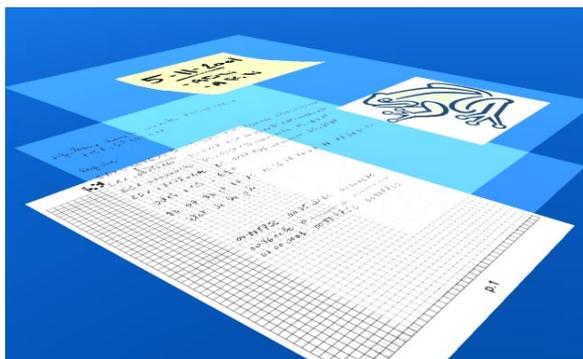


Fig. 11: Schematic representation of the layered information model, with a pre-printed page, handwritten ink, and pasted-in image and post-it note.

A notebook is a set of pages, each with several layers. An XML database on the host computers stores one or more notebooks, as well as information about off-line specimens. This XML database is the heart of the system, providing the persistent representation of the physical world and the links between physical and on-line data.

Since it may be shared by multiple client applications, the database is accessed through a server that implements a simple persistent Document Object Model or DOM. Clients can retrieve a node of the database and its subtree, such as a page and its layers. In order to modify the database, the client locks the subtree, modifies it locally, commits the changes and unlocks the subtree. When committing the changes, a set of updates are sent to the server rather than the updated tree. An update is one of the following operations: add node, remove node, change node name, set attribute, remove attribute.

This design has two advantages: round trips to the server are minimized and the server can keep a log of updates. Storing this log makes it possible to reconstruct the state of

the database at any point in time, which may prove useful for legal claims. The log can be used to reconstruct the most up-to-date version of the database in case of a crash. Finally, clients can listen to parts of the database and be notified when it changes.

The Distributed User Interface

The a-book uses a graphics tablet connected to the host computer to capture handwriting and to track the position and orientation of the PDA implementing the interaction lens. The PDA accesses the XML database to display and update its contents and must know its position to properly display the pages. Since the PDA has limited computational power and memory and the bandwidth of the link to the host computer is limited, task distribution between the PDA and the host must be carefully designed.

The system consists of three main components implemented as separate applications: the database server runs on the host computer, the notebook application is distributed between the host and the PDA, and the object tracker application runs on the PDA. A fourth application prints new notebooks and add them to the XML database.

The applications communicate through Java Remote Method Invocation (RMI). They share objects by storing them in a Content object in the RMI registry. Any application can access objects by name through the Content object, irrespective of their physical location. Shared objects include the XML database, which is used by all applications, and the input and output devices. Input devices include the main tablet, which is really a dual device since it captures both handwriting and the position and orientation of the PDA, and the PDA's touch screen. The only output device is the PDA screen, although this could change, for example displaying information on the main tablet. The PDA screen is used mostly by the applications running on the PDA, but is also used by the database manager to display error messages.

The notebook application is split into two components, one on the host computer and one on the PDA, because of the limitations of the PDA in terms of speed, memory and communication bandwidth. In particular, the PDA is not powerful enough to render the SVG description of the page in real-time. The rendering of SVG is done on the host computer using Batik, and the part running on the PDA manages a bitmap image of the whole page. It receives updates to this image from the host computer in the form of compressed tiles of raster data. The image is transformed by an affine transformation computed from the current position and orientation of the PDA. This transformation is computed on the host computer, and the PDA application is notified when it changes. The frame rate is still fairly slow, 2-3 frames per second, which could be improved with a better SVG renderer and a more efficient implementation of Java on the PDA. A better approach would implement a dedicated SVG renderer on the PDA.

The PDA part of the notebook application displays two additional layers of information: the transient feedback when interacting with the page, e.g. when the user selects

an area of the screen to create a link or add an entry to the table of contents and the associated GUI, such as the pop-up menu used to specify commands. Since it is the part of the application running on the host computer that executes those commands, the GUI components are created by the host computer and sent to the PDA with RMI. This way, the listener objects that are notified when a command is executed can reside on the host computer. The interface also uses gesture recognition [25], e.g. to underline or circle a written element either when writing on the notebook or through the interaction lens. When using the interaction lens, gestures must be transformed with the inverse of the affine transformation used to display the page. This adds noise that can degrade recognition of small gestures.

The part of the notebook application running on the host is responsible for capturing handwriting on the paper notebook and storing it as SVG data in the XML representation of the page. It also renders the page and sends updates to the PDA component. It also implements the commands invoked by the user from the PDA, such as creating a new link or entry in the table of content.

The interaction lens must be calibrated in order to properly align the image displayed on the PDA with the underlying notebook, giving the illusion of transparency. First, the user moves the PDA so as to see a corner of the current page. Then he selects the calibration command in the popup menu. The display freezes (i.e. the PDA stops tracking its position on the page). The user then moves the PDA to align its image with the underlying page. When the alignment is correct, the user taps the screen, which terminates the calibration and resumes normal operation. The precision of the calibration is not critical, since interaction with the content of the page is done through the PDA display and the parallax error (from the PDA's thickness) is larger than the calibration error.

The notebook application is notified by the tablet manager when the PDA is removed or lifted from the tablet. In this case, it changes the display to a thumbnail of the current page, and presents an interface so the user can flip through the pages of the notebook. As soon as the PDA is put back on the tablet, the interaction lens is reactivated. The user can switch to the object tracker at any time. It runs on the PDA and displays the part of the XML database that stores physical specimens and objects. The user can browse this part of the database, e.g. to create links between handwritten information and physical objects.

SUMMARY AND CONCLUSION

Biologists are required to keep paper-based laboratory notebook that serve as a legal record of research findings. Also active computer users, they work with digital images, analyze data and exchange information over the net. They have developed complex strategies for juggling information in both the physical and virtual world.

Using a participatory design approach, we worked with biologists, archivists and managers to video prototype ideas for an augmented laboratory notebook that integrates physical and electronic documents. We built three working

prototypes that allow biologists to write, annotate and interact with the physical notebook as before, while capturing the user's gestures and identifying the associated documents. Users can thus name or tag information, search, create links among pages and to external information, interpret and process certain types of data and automatically generate an index. Witnessed pages can also be identified and registered for later patent claims.

The a-book prototype introduces a physical interaction lens that lets users highlight, link and annotate information, as well as track physical objects, such as tagged animals or laboratory specimens. We developed a multi-layer architecture that identifies different characteristics of layers of physical data and maps them to corresponding on-line information layers. We also addressed the problem of persistent data, facilitating long-term storage, search and retrieval.

This project is one of a series that examine settings in which attempts to replace paper artifacts have failed [16,17,18,19,20]. We observe and work with users as we try to understand their interaction with both paper artifacts and on-line systems. Then, through a series of brainstorming and prototyping participatory design sessions, we create prototypes that integrate paper and on-line documents, attempting to integrate the different media and benefit from the advantages of both. Each setting poses unique design challenges, but when examined together, they begin to offer a more complete understanding of how to effectively manage the link between physical and on-line documents and questions the assumption that documents of the future will exist solely in electronic form.

Future research

All three users groups at the Institut Pasteur are interested in pursuing the a-book as a working prototype. The immediate problem to solve, which would make the system pay for itself, is to provide a reliable method of tracking notebook pages with patentable results. Of course, we are interested in providing a more generally useful system that addresses the needs of each of the stake-holders.

A real-world system needs to be reliable, inexpensive, and able to gracefully handle breakdowns. We plan to test the Anoto digital pen as a less-expensive and more portable solution than the graphics tablet. We also need a better solution to the page-detection problem. We also need further work with users to identify a prioritized list of functionality to be included in the first version of the system. In theory, since the paper notebooks are the same, the system should "work", even if all the links to the on-line system break. However, we need to understand what happens if some, but not all, of the system fails and how that affects use. The system must be reliable and biologists must be willing to use the interface on a daily basis. We also plan to work with archivists on the problem of archiving persistent physical and electronic data.

Finally, we are interested in using the a-book approach in other settings that require users to juggle physical paper, physical objects and on-line information. Although the

interface design meets specific user requirements at the Institut Pasteur, we believe that the architecture and implementation of the a-book can be developed into a full-fledged toolkit for paper-based augmented reality.

ACKNOWLEDGMENTS

Special thanks to Denise Ogilvy, Stéphane Kraxner, Laurent Bloch and the many biologists and others who participated in interviews, discussions and workshops. Also, thanks to Nicolas Roussel and Michel Beaudouin-Lafon for their comments on earlier drafts of this paper.

REFERENCES

1. Back, M., Cohen, J., Gold, R., Harrison, S. & Minneman, S. (2001) Listen Reader: an electronically augmented paper-based book. In *Proc. of CHI'01*, Seattle: ACM, p. 23-29.
2. Bier, E., Stone, M., Pier, K., Buxton, W., De Rose, T. (1993) Toolglass and Magic Lenses: the See-Through Interface. In *Proc. of SIGGRAPH'93* Minneapolis: ACM, 1993, p.73-80.
3. Cook-Deegan, R. (1994) *The Gene Wars: Science, Politics and the Human Genome*. Norton: NY.
4. Comiskey, B., Albert, J., Yoshizawa, H., Jacobson, J. (1998) An Electrophoretic Ink for All-Printed Reflective Electronic Displays *Nature* 394 (6690) p. 253-255.
5. Davis, R., Landay, J., Chen, V., Huang, J., Lee, R., Li, F. Lin, J., Morrey, C., Schleimer, B., Price, M. & Schilit, B. (1999) NotePals: Lightweight note sharing by the group, for the group. In *Proc. of CHI'99*, Pittsburgh: ACM, p. 338-345.
6. Fekete, J. & Beaudouin-Lafon, M. (1996) Using the Multi-Layer Model for Building Interactive Graphical Applications. In *Proc. of UIST'96* ACM. p.109-118.
7. Fitzmaurice, G. (1993) Situated Information Spaces and Spatially Aware Palmtop computers. In *CACM*, 36(7) p. 38-49.
8. Crewdson, J. (1995) In Gallo case, truth termed a casualty: Science subverted in AIDS dispute, *The Chicago Tribune*, January 1, 1995.
9. Gwizdka, J., Louie, J. & Fox, M.S. (1996) EEN: a pen-based electronic notebook for unintrusive acquisition of engineering design knowledge. In *Proc. of IEEE 5th Workshop on Enabling Technology: Infrastructure for Collaborative Enterprises*.
10. Hecht (1994) Embedded Data Glyph Technology for Hardcopy Digital Documents. In *Proc. of Symposium on Electronic Imaging, Science & Technology*, San Jose, Vol. 2171, p. 341-352.
11. Heiner, J., Hudson, S. & Tanaka, K. (1999) Linking and Messaging from Real Paper in the Paper PDA. In *Proc. of UIST '98*, p. 179-186.
12. Hopper, A., Harter, A. & Blackie, T. (1993) The Active Badge System. In *Proc. of INTERCHI'93*, Amsterdam: ACM, p. 533-534.
13. Landauer, T. (1996) *The Trouble with Computers*. Cambridge: MIT Press.
14. Letondal C. (2001) A Web interface generator for molecular biology programs in Unix, *Bio-informatics*, 17(1), p 73-82.
15. Lysakowski, Rich. (1997) Comparing Paper and Electronic Laboratory Notebooks. *Scientific Computing and Automation*.
16. Mackay, W. (1998) Augmented Reality: Linking real and virtual worlds. In *Proceedings of AVI'98*, L'Aquila, Italy: ACM. p. 1-8.
17. Mackay, W. (1999) Is Paper Safer? The Role of Paper Flight Strips in Air Traffic Control. *ACM Transactions on Computer-Human Interaction*, 6(4), p. 311-340.
18. Mackay, W. and Pagani, D. (1994). Video Mosaic: Laying out time in a physical space. In *Proc. of Multimedia '94*. San Francisco: ACM, p. 165-172.
19. Mackay, W., Fayard, A., Frobert, L. & Médini, L., (1998) Reinventing the Familiar: Exploring an Augmented Reality Design Space for Air Traffic Control. In *Proc. of CHI '98*, L.A.: ACM, p. 558-565.
20. Mackay, W., Pagani D., Faber L., Inwood B., Launiainen P., Brenta L. & Pouzol V. (1995). Ariel: Augmenting Paper Engineering Drawings. In *CHI '95 Video Program*, San Francisco CA: ACM.
21. Mackay, W. (2002) *Using Video to Support Interaction Design*. DVD Tutorial distributed by ACM/SIGCHI. ISBN 1-58113-516-5.
22. McGee, D., Cohen, P. & Wu, L. (2000) Something from nothing: augmenting a paper-based work practice with multimodal interaction. In *Proc. of DARE'00*, Copenhagen: ACM. p. 71-80.
23. O'Hara, K. & Sellen A. (1997) A Comparison of Reading Paper and On-Line Documents. In *Proc. of CHI '97*, Atlanta: ACM, p. 335-342.
24. Rekimoto, J. & Ayatsuka, Y. (2000) CyberCode: Designing augmented reality environments with visual tags. In *Proc. of DARE'00*, Copenhagen: ACM p.1-8.
25. Rubine, D. (1991) Specifying Gestures by Example. *Proc. of SIGGRAPH '91*, ACM, p. 329-337.
26. Sellen, A. & Harper, R. (2001) *The Myth of the Paperless Office*. Cambridge: MIT Press.
27. Schilit, B., Golovchinsky, G. & Price, M. (1998) Beyond paper: supporting active reading with free-form digital ink annotations. In *Proc. of CHI'98*, Los Angeles: ACM, p. 249-256.
28. Stifelman, L. (1996) Augmenting Real-World Objects: A Paper-Based Audio Notebook. In *CHI '96 Conference Companion*, p. 199-200.
29. Stone, M., Fishkin, K. & Bier, E. (1994) The Movable Filter as a User Interface Tool. In *Proc. of CHI'94*, Boston: ACM, p. 306-312.
30. Want, R., Fishkin, K., Gujar, A. & Harrison, B.. Bridging physical and virtual worlds with electronic tags. In *Proc. of CHI '99*, Pittsburgh: ACM p.370-377.
31. Wellner, P. (1992) The DigitalDesk calculator: Tactile manipulation on a desktop display. In *Proc. of UIST'92*, Hilton Head: ACM p. 27-33.