

November 11, 1998

---

**SRC** Research  
Report

157

---

## **The Virtual Book**

David Chaiken, Mark Hayter, Jay Kistler, Dave Redell

---

**digital**

**Systems Research Center**

130 Lytton Avenue

Palo Alto, CA 94301

<http://www.research.digital.com/SRC/>

# Systems Research Center

The charter of SRC is to advance both the state of knowledge and the state of the art in computer systems. From our establishment in 1984, we have performed basic and applied research to support Digital's business objectives. Our current work includes exploring distributed personal computing on multiple platforms, networking, programming technology, system modeling and management techniques, and selected applications.

Our strategy is to test the technical and practical value of our ideas by building hardware and software prototypes and using them as daily tools. Interesting systems are too complex to be evaluated solely in the abstract; extended use allows us to investigate their properties in depth. This experience is useful in the short term in refining our designs, and invaluable in the long term in advancing our knowledge. Most of the major advances in information systems have come through this strategy, including personal computing, distributed systems, and the Internet.

We also perform complementary work of a more mathematical flavor. Some of it is in established fields of theoretical computer science, such as the analysis of algorithms, computational geometry, and logics of programming. Other work explores new ground motivated by problems that arise in our systems research.

We have a strong commitment to communicating our results; exposing and testing our ideas in the research and development communities leads to improved understanding. Our research report series supplements publication in professional journals and conferences. We seek users for our prototype systems among those with whom we have common interests, and we encourage collaboration with university researchers.

# **The Virtual Book**

David Chaiken

Mark Hayter

Jay Kistler

Dave Redell

November 11, 1998

**© Digital Equipment Corporation 1998**

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of the Systems Research Center of Digital Equipment Corporation in Palo Alto, California; an acknowledgment of the authors and individual contributors to the work; and all applicable portions of the copyright notice. Copying, reproducing, or republishing for any other purpose shall require a license with payment of fee to the Systems Research Center. All rights reserved.

## **Authors' Abstract**

The Virtual Book Project explored the use of a prototype electronic reading appliance to replace paper as the medium for reading and browsing a wide variety of material. The research hypothesis was that the relevant technologies for high-resolution flat panel displays and the associated storage, processing and communication components were reaching thresholds that enabled them to compete successfully with paper for sustained reading. Our research strategy was to build and deploy a dozen prototype units, gaining hands-on experience with the issues involved in their use.

Our experience with the *Lectrice* prototype shows that reading appliances are indeed crossing the threshold to practicality, but that a number of challenges remain to make these devices fully competitive with some of the more subtle advantages of paper. At the same time, the inherent advantages of electronic media make the long-term prospects for reading appliances very compelling.

This report describes *Lectrice*, summarizes the lessons learned about both software and hardware, and poses some of the questions that need to be addressed for the promise of reading appliances to be fully realized.



# The Virtual Book

David Chaiken, Mark Hayter, Jay Kistler, and Dave Redell

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>2</b>	<b>LECTRICE: THE PROTOTYPE VIRTUAL BOOK .....</b>	<b>2</b>
2.1	OVERVIEW .....	3
2.2	SPECIFICATIONS .....	4
2.2.1	Hardware .....	4
2.2.2	Software .....	5
2.2.3	Packaging .....	6
2.3	USING LECTRICE .....	8
2.3.1	Operating modes.....	8
2.3.2	User Studies .....	10
2.4	USER INPUT DEVICES .....	12
2.4.1	Buttons .....	12
2.4.2	Pen .....	15
2.5	COMMON APPLICATIONS.....	19
2.5.1	Lectk – the Virtual Paper viewer.....	19
2.5.2	Acrobat Reader .....	24
2.5.3	Browser .....	24
<b>3</b>	<b>COMPETING WITH PAPER .....</b>	<b>25</b>
3.1	A BETTER DEFINITION OF THE TARGET.....	25
3.2	LEGIBILITY .....	26
3.2.1	Display Size.....	26
3.2.2	Display Density .....	27
3.2.3	Display Examples.....	28
3.2.4	Other Factors.....	39
3.3	ERGONOMICS .....	39
3.3.1	Physical Characteristics .....	39
3.3.2	User Interface Characteristics.....	40
3.4	MARKUP .....	41
<b>4</b>	<b>SURPASSING PAPER .....</b>	<b>41</b>
4.1	NAVIGATING ELECTRONIC DOCUMENTS .....	42
4.2	TRANSPORTING ELECTRONIC DOCUMENTS .....	43
4.3	PUBLICATION OF ELECTRONIC DOCUMENTS .....	45
4.4	BEYOND PRINT .....	46
<b>5</b>	<b>RELATED WORK .....</b>	<b>48</b>
<b>6</b>	<b>CONCLUSIONS .....</b>	<b>51</b>
6.1	DISPLAYS FOR VIRTUAL BOOKS .....	51
6.2	IMPORTANT FEATURES OF VIRTUAL BOOKS .....	51
6.3	THE GLOBAL LIBRARY: THE FUTURE OF VIRTUAL BOOKS .....	52
<b>7</b>	<b>ACKNOWLEDGEMENTS.....</b>	<b>53</b>
<b>8</b>	<b>REFERENCES.....</b>	<b>54</b>





## 1 Introduction

Computers are machines that you can't write without, and can't read with.<sup>1</sup> Challenged by this observation, the Virtual Book project set out to use emerging technology to build a platform to investigate aspects of on-line reading. The principal goal was to produce a device that would provide as many of the conveniences of paper as possible, supplemented with features that could be provided only by electronic documents. Our hope was to make on-line reading an attractive activity for books, magazines, papers, and other documents that are read today primarily on paper. It was our belief that flat panel displays were passing thresholds in both size and resolution that would allow comfortable reading for extended periods. The displays that passed these thresholds were novel and expensive at the start of the project in 1994, but they are now produced in volume. Once we had built our prototype, we carried out informal studies of its use. These convinced us that reading and browsing on-line is not only possible on the sort of device we had built, but it is often the preferred way of accessing and reading the documents we use in our everyday work.

Reading text on stationary computer screens is generally too unpleasant to be acceptable for documents of even moderate length. Most people print out documents longer than a page or two, rather than struggling to read them on-line. A preceding project, called *Virtual Paper* [Birrell95], started addressing this problem by designing software that would improve on-line reading on a desktop computer. This project used four key approaches. Gray-scale and sub-pixel character positioning were used to enhance the on-screen appearance of text. The speed at which new pages are displayed was increased. The user-interface was intuitive, and conserved screen real estate. Finally, the system provided new and improved capabilities over paper-based reading, such as easy and fast searching and support for document annotation.

These techniques showed the promise of on-line reading, as did Adobe's Acrobat Reader, which was developed contemporaneously. The emergence of the Web has further increased the importance of on-line information access. However, software solutions alone are not sufficient for competing with real paper for most reading tasks because they fail to address important ergonomic issues.

To convert users to reading documents on-line, rather than printing hardcopy, the design of the reading device must also be addressed. The ergonomics of current desktop reading environments are poor largely because the display is immobile. The only degrees of freedom that on-line readers typically have are limited movement in their chairs and turning their necks. Compare this to reading a real book: the reader can sit anywhere, can move easily from one place to another, and can change reading positions arbitrarily. The absence of such flexibility causes users to avoid on-line reading systems despite their potential advantages over the plain paper alternative. Even the most popular portable devices — so-called "laptop" personal computers — are not packaged to allow comfortable reading positions.

The reading appliance built during the Virtual Book project is called *Lectrice*, it features a high-resolution display, button and pen input, audio input and output, and both wired and wireless network options. Lectrice runs software derived from the Virtual Paper system allowing convenient and comfortable reading. A standalone mode allows complete freedom of use, at the expense of limiting the amount of reading material that is available. In networked modes, a virtual book gets access to much

---

<sup>1</sup> Professor Roger Needham, at the time head of department at the University of Cambridge Computer Laboratory, made this observation.

larger document repositories and can serve as a portable web browser. As of 1998, this networked configuration is still an excellent way to read documents on the Web, and it is still in use by researchers at SRC.

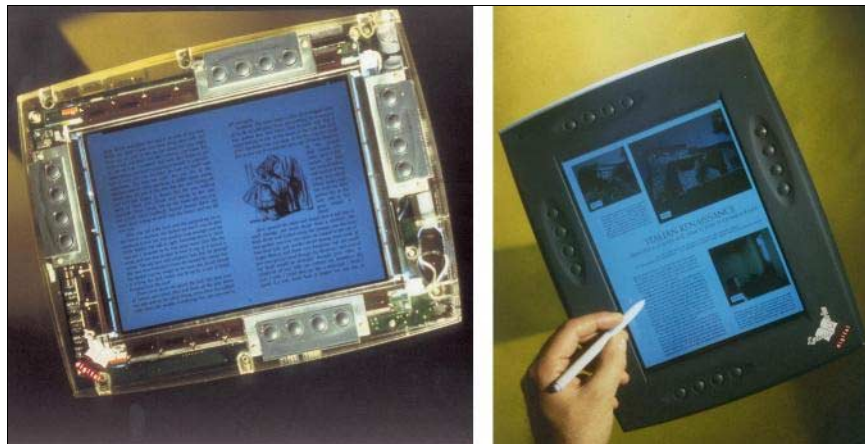
The Virtual Book project started in early 1994. By the end of the year the first demonstrations of the system were given. During 1995 the case and final hardware were designed, the bulk of the software was written and the system was made robust. The majority of the observations of users of the prototype were made during the last quarter of 1995 and first half of 1996.

The rest of this report is organized as follows. Section 2 presents an overview of Lectrice, motivating the key design decisions and highlighting the main characteristics of the software, hardware, and packaging. Section 3 compares Lectrice with paper documents and comments on the trade-offs made in its design. Section 4 describes areas where the electronic format improves upon paper. Finally, Section 5 describes some related work and Section 6 reflects on the future of virtual books.

There are a number of pictures in this document. Those that show the Lectrice case are digitized photographs, those that show applications but no case were captured using a screen dump. A few of the pictures were captured and scaled to exact size to demonstrate quality observations. These will be degraded in photocopies or by low-resolution printers. Where two photographs are shown side-by-side their resolution has been reduced.

## 2 Lectrice: The prototype Virtual Book

The primary deliverable of the Virtual Book project was Lectrice — a tablet computer tuned for on-line reading. A total of fifteen of these units were built, deployed at a number of sites within Digital, presented to select customers, used in focus groups, and shown in public forums.



**Figure 1: Lectrice (photos by Studio Red).**

In producing Lectrice, our intent was to build a research-enabling tool as rapidly as possible, rather than to design a cost-effective product, which would need to include additional features and functions as well as significant engineering optimizations. The central goal of the Lectrice prototype was to enable experiments testing the hypothesis that a well-designed high-resolution reading appliance can compete with paper as a reading medium.

To leverage the previous software work at SRC, the initial thrust of the project was to allow *Lectern* (the Virtual Paper viewer) to be used on portable hardware. A reading appliance being designed today should support the common document formats that have emerged as a result of progress in commercial reading and information access products. These advances have widespread public visibility on the World Wide Web. The most notable new standard is the HyperText Markup Language (HTML), the primary document format used by Web browsers. Next in importance is Adobe Portable Document Format (PDF) and the Adobe Acrobat Reader, which are appropriate for statically formatted documents.

Given our central goal, the most obvious software strategy would have been to program the Lectrice as a single-function device, dedicated to running the Lectern viewer software. However, it was intended that Lectrice would be suitable for uses besides sustained reading — for example, as a Web browser, as a news and mail reader, or as an audio/video conferencing terminal. To ease exploration of these various uses, the base software transforms Lectrice into a mobile, keyboardless X-terminal. This approach allows Lectrice to function as a prototype of a variety of devices. A two-stage process can then be used; first an existing application is run remotely displaying on Lectrice, and then if warranted a native port can be done. The core reading application followed this pattern; initially the standard Lectern viewer was run remotely and later a native port (*Lectk*) was developed. In other cases, the source code was not available for the port so remote operation was the only option. For example, a portable Web browser was created by running Netscape Navigator<sup>2</sup> on a remote server and displaying its window zoomed to full screen on the Lectrice. This combination is a very persuasive demonstration of the benefits of a mobile reading appliance.

## 2.1 Overview

The project's goals, available technology, and physical constraints shaped the design of Lectrice. As far as possible, the design integrated existing technology, saving innovation for areas directly related to the core research on sustained reading. The initial goal was to build a prototype that would be usable for periods of continuous reading. Instead of displaying documents designed for computer screens, the prototype would have to allow readers to view a variety of *existing* documents, most designed for high-resolution printing on paper. Examples include technical and business documents, novels, journals, and newspapers.

The main hypothesis of the project was that LCD size and resolution were crossing thresholds that would allow users to read for long periods. A survey of tablet computers available in mid-1994 showed that none were suitable for this task (they typically had VGA displays of insufficient quality). However, the display manufacturers were able to supply small quantities of the new LCDs. This, along with a brief study of the state of pen and wireless technologies, led us to the conclusion that by using emerging (and therefore expensive) components, it was feasible to build a suitable hardware and software prototype to test the hypothesis.

Another goal was to ensure mobility for Lectrice users. Readers want to be able to shift positions in their immediate environment, for example, while sitting at a desk, relaxing in a living room, or even lying in bed. Readers should also be able to move between locations within a building and ideally would be able to use the device when they are travelling.

---

<sup>2</sup> The project was complete before the Mozilla source release by Netscape.

To be a persuasive information appliance, Lectrice needed to be comfortable to hold. Two of the most important factors are weight and power dissipation. Initial experiments with a weighted clipboard showed that the ideal weight for the device was less than three pounds (a goal considered unachievable with 1994 technology), but that the subjective difference between three and four pounds was small. It was important the device not get too warm since readers would be likely to use it in their laps for long periods. This requirement implies a power dissipation of less than 10 watts<sup>3</sup>. Other factors required careful thought, including the Lectrice's size and shape, placement of buttons, case material, and even the type of exterior paint.

To some extent, the prototype needed to be able to emulate the experience of using paper. For example, it is very common to mark up paper documents, particularly business and technical ones. A general-purpose virtual book should provide an electronic analog, allowing readers to annotate documents.

Extrapolation of the price trends for high resolution LCDs suggested that even if the prototype were successful, products would not be economically viable until 1997 or later. The prototype was therefore designed to be a research platform rather than an immediate candidate for a product. This tradeoff meant that there would be some areas where the prototype would be inferior to a product. For example, small-run plastic cases are almost always heavier and more delicate than production-tooled ones.

## 2.2 Specifications

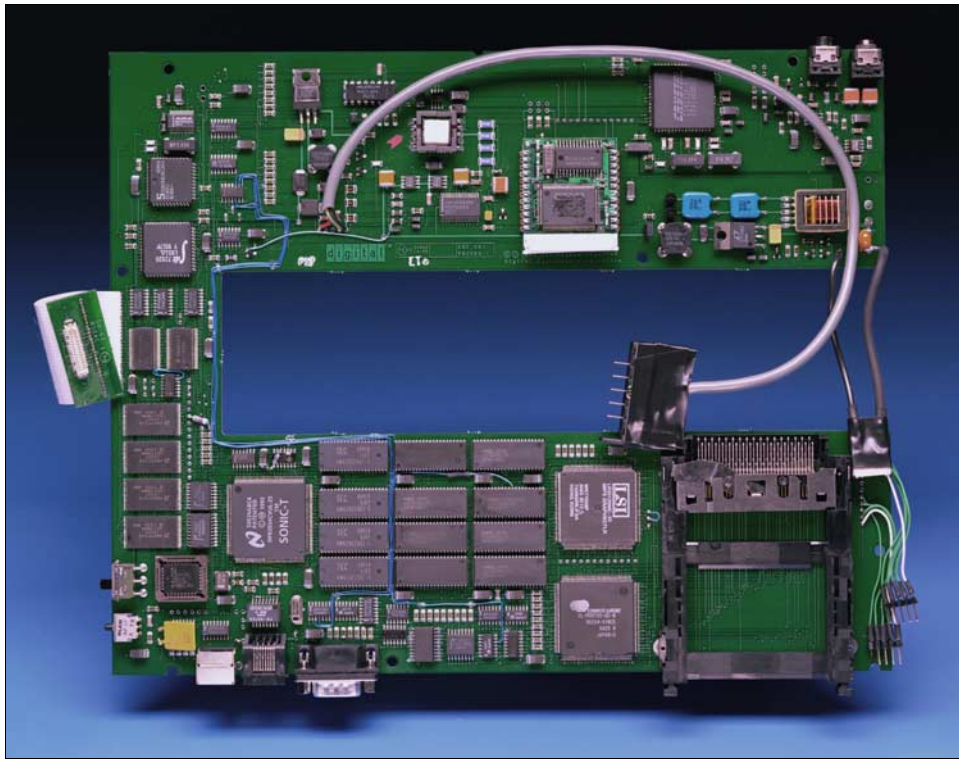
### 2.2.1 Hardware

Lectrice has been described as “a really nice display connected to a really boring computer”. For the purposes of the research goals, the details of the hardware are secondary. In summary, the Lectrice hardware includes:

- LSI Logic LR33120 processor (clocked at 25MHz) — a version of the MIPS R3000 processor with a built-in graphics core. It is designed to run the X server.
- Memory: 16MB of DRAM, 1MB of VRAM, up to 8.5MB of Flash.
- Liquid crystal display (LCD) interface, with support for two 3.5W backlights.
- Wacom pen-input subassembly.
- National Semiconductor's Sonic Ethernet chip to give a 10baseT connection. The spare pairs of wire on the 10baseT connector can be used to supply power, so a single cable can be used to provide network access and power.
- PC-Card (PCMCIA 2.1) connector with 2 slots. The slots were used for a wireless local area network card (the Digital RoamAbout™ product), or for document storage on up to 20-megabyte Flash (non-volatile) memory cards.
- A NiMH battery that supports about 1.5 hours of operation.
- Power supply that generates +5V and +12V from either the battery or a tether.
- Audio I/O using a Crystal Semiconductor CS4231 codec, built-in or external microphone and external stereo headphones.
- PS/2 port for experimenting with other input devices.
- RS-232 serial port for debugging.

---

<sup>3</sup> Consider a cat (which dissipates about 10 watts) sitting in your lap. It is pleasant in the winter but may be too warm in the summer.



**Figure 2: The Lectrice circuit board.**

Lectrices were built with 10.4 inch diagonal TFT LCDs in three different screen resolutions:

- 122 pixels per inch using a Sharp LQ10DX01 XGA (1024 x 768) screen, with two 3.5W backlights.
- 96 pixels per inch using a Sharp LQ10DS01 SVGA (800 x 600) screen, with a single 2.5W backlight.
- 77 pixels per inch using a Sharp LQ10D321 VGA (640 x 480) screen, with a single 2W backlight.

With the XGA display, the total power consumption was approximately 15 watts, almost half of which was consumed by 7 watts of backlight power. Nonetheless, the XGA display was the standard for Lectrice due to the need for high resolution. When the device was designed, this display was in the pre-product stage of development, but it is now a standard part and was manufactured in volume during 1996.

The lower resolution displays were a generation later in technology. The new technology requires only a single backlight, rather than two. This improvement results in part from the larger pixel size, but more from advances in the optical system that spreads the light across the display and in better techniques for forming the pixels.

### 2.2.2 Software

The software for Lectrice provides a comfortable environment for sustained reading. It allowed investigation of both the user interface for reading and the system resources (memory, processing power, I/O bandwidth) needed to support such a system. The project also used Lectrice to investigate alternative forms of input and output (pen, voice and button input for control and simple audio output).

The base-level software ported to Lectrice or written during the project includes everything from the operating system to the user interface. The base system consists of the DEC ELX™ real-time kernel and

associated device drivers. ELX is a derivative of VxWorks™, a real-time kernel by Wind River Systems. The display functionality is provided by an X Windows System (X11R6) server [SG1992] and the FVWM window manager, both of which were extended to allow switching between portrait and landscape orientations. An AudioFile server [AF1993] allows audio input and output. (AudioFile is the audio equivalent of the X Windows server.) Pen control software converts pen events into mouse events and provides input for a handwriting recognition engine. A Graffiti [Palm1994] single stroke handwriting recognizer (similar to the one on the PalmPilot™) allows text and gesture input. A button manager senses the user pressing and releasing Lectrice's buttons (see below) and converts the actions into commands for visible applications. Finally, the Tcl/Tk [Oust1991] runtime provides a toolkit for building applications for the prototype.

Lectk, a derivative of the Virtual Paper viewer, is the flagship application built specifically for Lectrice. Other applications — most notably the Netscape Navigator Web browser — run on a remote server and use the X Server to display on Lectrice's screen. Later sections of this report describe these applications in detail.

### 2.2.3 Packaging

One of the keys to making the Lectrice easy and pleasant to use was the case design. A design team at Studio Red [[www.studioed.com](http://www.studioed.com)] helped evaluate different case shapes and layouts, and eventually produced the packaging for Lectrice. One constraint was the short-run process that used rubber molds, which are suitable for manufacturing prototypes but are not cost-effective for products.

Accepting the constraints of the prototyping process, we identified a number of requirements for making a virtual book easy to use. First and foremost, the unit needs to be as thin and light as possible. To accommodate a variety of documents, both portrait and landscape orientations need to be comfortable to use. The unit needs to be inviting to pick up and to hold.

To satisfy the requests of various researchers, buttons had to be mounted on each side of the display and on the back. One compromise that was made was to require a screwdriver to open the battery compartment, since plastic catches are hard to manufacture in the prototype process.

Foam models helped us investigate various shapes for Lectrice. These models had to be weighted realistically to allow accurate evaluation, since the way they were held depended on the weight. To allow portrait and landscape operation, the center of mass needs to be near the center of the unit. If Lectrice were used only in a single orientation, the ideal center of mass would be slightly below the centerline. The final case has curved edges and is 13.5" x 10.5" x 1.75" (34cm x 27cm x 4.5cm) measuring across the widest points. The thickness is the sum of the thickness of the rear wall, plus the battery, the battery box wall, the LCD, and the front plastic.

The width and length were set by the LCD and its bezel and clearance for the buttons. The margins were made relatively wide to keep the user's thumbs off the screen. In retrospect, readers held Lectrice differently from the way that they held the foam models. Experience with Lectrice suggests that the margins should have been about half as wide. This goal would have been hard to achieve with the XGA display, because it was packaged with electronic drivers extending about an inch beyond the glass. More recent displays require much less clearance, because the drivers are wrapped behind the screen, and permit narrower margins.



Figure 3: Lectrice displaying a magazine page.

Lectrice did not meet its weight goal. The final weight of the XGA-based Lectrice in standalone mode is 5.75 pounds (92oz = 2.6kg). A breakdown of the weight is shown in Table 1.

Item	Weight	
	oz	g
Case	31	875
Liquid Crystal Display (LCD)	26	725
Battery	17	475
Printed Circuit Board (PCB)	13	375
Pen Sensor and Digitizer	4	125
Flash card	1	25
Total	92	2600

**Table 1: Weight of Lectrice components.**

When connected to a wireless local area network (LAN) the 1oz Flash card is replaced by the PCMCIA interface card and antenna, which weigh 6oz (175g). If both LAN and power come from a tether, then the Flash card and battery can be removed, making the total weight slightly over four and a half pounds (2kg).

The case could be reduced in weight by using production rather than prototype techniques. In particular, the plastic is thicker than would be required for production injection molding. In addition, the buttons are all mounted on small metal plates that could be replaced by plastic ribbing. These plates contribute over 8oz (225g) — over 10% of the weight of the tethered device! Even during the design stage, display technology advanced. It is reasonable to expect that the weight of a product's screen would be half of Lectrice's, due to thinner glass and a single backlight.

## 2.3 Using Lectrice

A total of fifteen Lectrices were made. Two were fitted with VGA screens and were used only for experiments in comparing screen resolutions. Two were fitted with SVGA screens. These were acceptable for Web browsing, but were not used for continuous reading. The other units were fitted with XGA screens and have been in use for several years for Web browsing and reading.

### 2.3.1 Operating modes

Lectrice might be used in three classes of environments, corresponding to its three modes of operation. The first is an office; the second a building or campus; and the third a wider area. Lectrice also saw limited use in homes, using both networked and standalone modes.





**Figure 4: View showing the pen in its holder, a Flash card in a slot and the Ethernet/power tether**

#### 2.3.1.1 Office use: Tethered mode

One of the primary areas for information access is in an office. In this setting, Lectrice can get power and an Ethernet connection using a single cable. The standard 10baseT-ethernet connection uses two twisted pairs (one in each direction), but the RJ-45 connector supports four pairs. Lectrice uses the extra two pairs to carry power (12-18V DC) and automatically switches between the battery and wired power. The tether is the blue cable shown in Figure 4. It is positioned to be on the top edge when in landscape mode, and on the bottom left when in the more usual portrait mode. Users preferred the tether to the uncertainty of battery life, but it is hard to say how much this was due to the limited life of Lectrice batteries and the extra weight they add. Using Ethernet gives a high quality network connection and allows access to both server-based Virtual Paper document repositories and the Web.

The advantage that a tethered Lectrice has over using a desktop computer is to allow the user local mobility. Sitting in front of the computer screen forces users to adjust their body position to screen placement. With a Lectrice (even when a thin cable is attached), readers sit in comfortable positions and occasionally shift positions. Conventional computers force users to adapt to them, while Lectrice adapts to the physical needs of the reader.

Anecdotal evidence supports the success of this mode of operation. During a week-long series of meetings, all attendees in a conference room had access to Lectrices operating in tethered mode. A Netscape browser running on each Lectrice allowed the attendees to view on-line versions of the relevant working papers. One attendee started by using a printed version of the material, but by the end of the week had switched to using a virtual book. The browser allowed much more convenient access to the large body of documentation, and allowed the meeting participants to share newly created documents instantly.

### 2.3.1.2 Campus use: Wireless mode

A wireless LAN card can be installed in one of the PCMCIA slots on the Lectrice. Experiments during the Virtual Book project used the Digital RoamAbout™ product (this is based on the WaveLAN™ system). This configuration gives up to 2Mbps per second connection on to a campus LAN. With typical use, Lectrice batteries last between 1 and 1.5 hours. This length of time is sufficient for experiments, but not satisfactory for everyday use. The network bandwidth proved adequate for accessing server-based material.

The wireless mode was also used in a home experiment, including reading in bed. The Lectrice used a wireless connection to access documents on a personal computer located on a different floor of the house. While this configuration was too complex for “real” users, advances in network management should address most of the problems. Lectrice makes an agreeable bed companion: it provides its own light and turns pages almost silently.

### 2.3.1.3 Remote use: Standalone mode

There are many environments where a network connection is not available. Lectrice supports this situation by using a Flash memory card. The Flash card can be seen in the PCMCIA slot in Figure 4. Material is loaded on to the card while the device is still connected to a network and can be accessed later. Only Lectern documents can be read in this mode, because no browser runs natively on Lectrice; however, Web pages can be (inefficiently) converted to Lectern format.



**Figure 5: Standalone mode in use.**

The standalone mode was common for use at home, as well as on trips. When reviewing technical papers or long drafts users find it is much more comfortable to sit back on a couch than in a desk chair. Indeed, on one occasion a user was seen coiled up on the couch deeply engrossed in a thesis!

## 2.3.2 User Studies

The results generated by the Virtual Book project came from informal observations of people using the Lectrice. The three main forums for the observations were day-to-day use, focus groups and a series of intensive interviews. Subsequent sections of this report describe the subjective results and their impact on the final design of the system. Section 6.2 summarizes the results.

The Lectrice was integrated into the team members’ computing environment. The tethered mode was used for browsing news sources, on-line manuals, technical specifications, and other documents available in the Virtual Paper repository or on the Web. The standalone mode (and a variant where the Lectrice

was powered using a tether, but had no network connection) was used for reading and reviewing thesis drafts, and doing peer-review of conference papers (typically done away from the desk at work or in an easy chair at home). The wireless version was used in meetings on a number of occasions, though the poor battery life led to use of the tethered version during the week-long series of meetings mentioned above.

Four focus groups were conducted. Each of these consisted of a discussion between eight people who described themselves as “mobile information users” but who did not work in the computer industry. A facilitator started the discussion by asking the subjects to describe their information needs. Then specification sheets for fictitious devices were presented to gauge their interest in reading appliances. Towards the end of the session a demonstration of the Lectrice was given and the group’s reaction to the device and changes in attitude were noted. The team members watched the sessions from behind a one-way mirror and reviewed them from a video.

The intensive interviews gave us a better understanding of how users might benefit from having a virtual book to help them in their work. Five general areas were studied: healthcare, complex manufacturing, field service/maintenance, field sales force and knowledge-workers<sup>4</sup>. In each area between five and ten people were interviewed to understand the requirements they would have for a mobile information device and the environment they would be using it in. As in the focus groups, the Lectrice was demonstrated towards the end of the discussion. In each interview one team member acted as interviewer and one as scribe. Following the interviews the team met to combine their observations into a common understanding of the requirements from each of the user communities and their reactions to the Lectrice. To aid processing the interview information we followed a modified version of the Concept Engineering process from CQM [[www.cqm.com](http://www.cqm.com)].

In addition, we conducted several customer visits and the device was demonstrated at a number of trade-shows. These events elicited reactions from potential customers and feedback on potential uses for the device.

---

<sup>4</sup> Knowledge workers are professionals who use large quantities of information to carry out their job, examples include bankers, lawyers and corporate executives.

## 2.4 User Input devices

Two atypical input devices are used during normal operation of the Lectrice. The first of these is a set of case-mounted buttons, and the second a pen. Audio input is also supported by the hardware.

### 2.4.1 Buttons



**Figure 6: Lectrice has 32 buttons.**

An initial hypothesis of the project was that case-mounted buttons would be ideal for control functions. This later proved to contrast with the approach taken by Knight-Ridder, for their electronic newspaper design study one of their tenets was that there should be no buttons [Martin1995]. For the Lectrice there was no *a priori* way to determine the right number of buttons, so the design intentionally erred on the side of too many rather than too few. This approach allowed the buttons to be integrated into the case design. If the case had been built with only a few buttons, then some experiments would involve attaching the extra buttons to the Lectrice. We suspected that this would bias results as users reacted to the physical rather than functional characteristics<sup>5</sup>. Since the unit was to be used in both portrait and landscape modes, the button functions needed to be symmetric under a 90-degree rotation. The case was also designed to be symmetric about the horizontal and vertical axes in both orientations, in order to support left and right handed users equally well. Testing showed that the symmetry was required, but for a different reason; all users found that sometimes the left hand buttons were more convenient and sometimes the right, depending on their current reading position.

<sup>5</sup> This assumption proved correct: later experiments using touch pads and button mice proved inconclusive because having a device added on was cumbersome enough that no conclusion could be drawn about usability.

The buttons on the back of the case did not prove useful. Initially, they were intended for both control functions and for chording (one-handed) keyboards. There were three (related) reasons that the rear buttons failed. Primarily, users did not keep their hands in one place on Lectrice during sustained use, so chording would not be a useful interface. Second, it was hard to locate fingers on the rear buttons, even with extra details sculpted in to the case to help users find the buttons. In addition, the difference in people's hand sizes meant that no single position for the rear buttons would allow comfortable use for everyone. The last two reasons seemed true for models of different case designs, so we believe that we were not just seeing an effect caused by the particular case design used for the Lectrice.

One contentious issue in the design was the “feel” of the buttons. The eventual selection used a rubber cover over a metal dome, which pushed onto a membrane switch with a metal backing-plate. The button design balanced the need for low activation force (page turning should not be a strenuous activity!), with the need for tactile feedback. Initially, 12.5mm diagonal, round 170g activation force domes were used, but these proved too soft. Following experiments with a selection of domes, they were replaced by 12mm cross-shaped domes with 200g activation force.

There are two user interface challenges associated with using buttons. The first is how to integrate them into applications — particularly how to experiment with existing applications without requiring modifications. The second is how to label them in a way that will support multiple applications and both orientations.

Our button-input approach is based on the observation that an event stream is used to pass inputs to the application. For most applications, there are keyboard or mouse shortcuts for common actions<sup>6</sup>. For each application, the button manager keeps a binding between each button and the event sequence that is needed to perform the associated action. The binding mechanism is implemented using the X resource database. When a reader presses a button, the manager process determines what application has the input focus, translates according to the current screen orientation, looks up the mapping for the (button, application) pair, and injects the required event sequence.

Button help is provided by a similar mechanism. When the reader presses a designated help button, the button manager looks up application-dependent “reminders” for labeling each of the buttons. Reminders are small icons displayed on the screen adjacent to the buttons on the case. In an ideal device, the reminders might be shown on a small display in each button. It is a little hard to see, but the right hand top icon is an upward pointing arrow (scrollbar operation) in the Netscape picture in Figure 7 and a right pointing button (advance forward) in the Lectk application shown on the right. If the pen is held above a reminder expanded help information pops up in the center of the screen, as shown in the photo on the right. Touching a reminder with the pen performs the same action as the associated button.

---

<sup>6</sup> While the Lectrice implementation is based on X, this observation is true for virtually all window systems, including Microsoft Windows.



**Figure 7: Reminders line up with the buttons.**

The pictures also show the problem with this style of labeling — and with virtual buttons in general. When the expanded help information is not present, the reminders still use up space on the screen, often obscuring the material being read. Even in the case where the reminders are not covering any text, they almost always distract the eye of the reader. These observations are a strong argument against implementing virtual buttons, which permanently occupy some part of the display. The reminders have proved a reasonable solution. Users quickly became accustomed to the positions of the most common buttons and would work with reminders disabled, only displaying them to remind themselves of the position of less common functions.

On a virtual book product, there should be one physically distinguished button that is always used to request more help. If placed appropriately (probably in a corner), the same physical button could be used in both landscape and portrait mode. By convention, Lectrice software always uses the top-right button (relative to the current screen orientation) for the extended help function.

Several other software conventions for button positioning improved the user interface. The four buttons on the left and the four on the right are used for navigation (forward, backward, next page and previous page). These functions are always assigned symmetrically to the left and right buttons. As discussed above, this symmetry allows for use with the most convenient hand. The four buttons at the bottom of the screen are used for application-specific control functions (in the browser shown in the photo these are reload, add bookmark, stop loading and pop-up URL entry box). The four buttons at the top of the screen are always bound to the same system control functions: switch orientation, zoom window to full screen, pop-up window manager commands, and provide extended help.

### 2.4.2 Pen

Pen input is used in three ways. The main use is as a selection device for driving the user interface. In this mode, the pen generates mouse events for applications. The second use is for control gestures as an alternative to using the buttons, and in this mode it generates key events (sending the key sequence for the “keyboard shortcut” the application uses for that function). The third mode is for entering text without using a keyboard.

A variety of sensors for capturing pen input are available. Ideally, such sensors should be integrated with the display, but integration is not economically feasible when building only a small number of research prototypes. Many of the simplest and least expensive pen technologies involve putting a sensor on the front of the LCD. For the purpose of late integration, these products come mounted on a glass sheet (typically a tenth or an eighth of an inch thick). Unfortunately, even the best front-mounted systems have a light transmissivity of only 80%, and the separator particles on the standard membrane systems produce bad interference patterns on a high-resolution display. Lectrice therefore uses an electromagnetic system from Wacom [<http://www.wacom.com/>].

The Wacom system has an antenna array which mounts behind the display. (This is a very thin PCB with a thinner metal sheet behind it.) The antenna array alternates between transmission and reception. The pen has an LC-tuned circuit that resonates with the transmitted signal, and continues to radiate for long enough to be received. When the pen tip is pressed it deforms a capacitor plate causing the resonance frequency to shift a small measurable amount. Buttons on the barrel of the pen make a large change in the resonant frequency. One advantage of this system is its ability to detect the pen position even when the pen is not in contact with the screen (the pen is detected when its tip is about ½” above the screen). It can also report both the position and the pressure applied when the pen is in contact. A major disadvantage is interference between the pen system and the display drive electronics and mounting points. Again, full integration in a product would allow correction factors to be made. The prototypes used shielding tape and calibration software to remove much of the error.

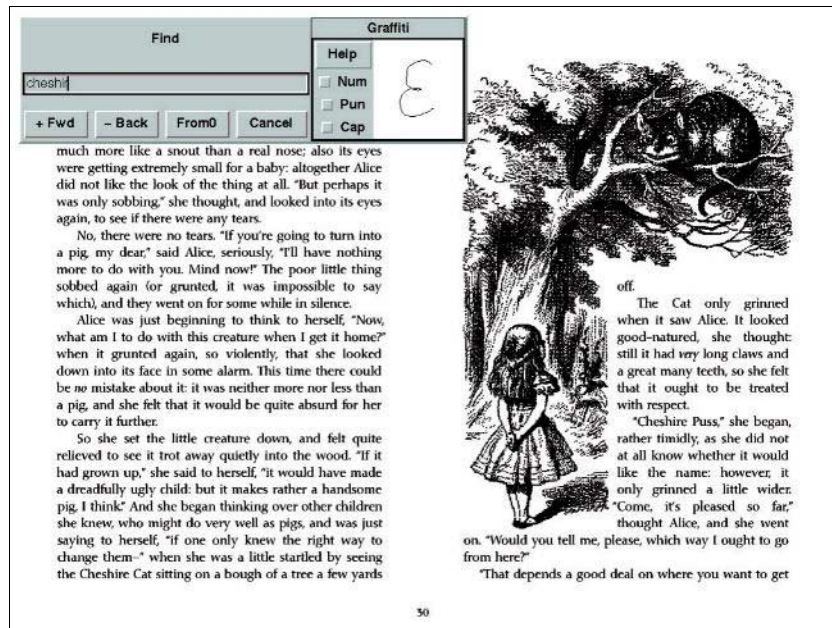
In mouse-replacement mode, the pen position is used to send mouse position events. The pen tip is used as one mouse button, the first barrel button represents another, and the second barrel button or ‘eraser’ end of the pen represents a third.

Research in handwriting recognition was not a goal of this project. However, the project needed to be able to test its use in the interface of a reading appliance. To this end, Lectrice’s software includes a home-grown single stroke recognizer based on the Graffiti™ alphabet. The initial implementation was based on the single stroke recognition engine from Carnegie Mellon University [Rubine1991]. Later, after a study of the literature, a more accurate and efficient recognition engine was implemented based on the work of Li and Yeung [Li1993,Li1995,Li1997]. The recognition rates were not as high as commercial implementations, but they were high enough to prove the utility of the pen for short textual input (search expressions, URLs and simple form filling).

Three mechanisms were available for text input: application integration, input box, and writing on the window. All have their place, but overall the third system proved the most popular.

### 2.4.2.1 Application integration

The simplest way to enable the pen for text input is to modify the dialog boxes in an application to include an area for pen input. This was done by providing a standard Graffiti input widget that displays the stroke drawn by the pen, calls the recognition engine to convert the stroke to a character and inserts the character into a specified text input area. Figure 8 shows the modified “Find” dialog in the Virtual Paper viewer. The Graffiti widget has been added to the right end of the original dialog box and is shown just as the user lifts the pen after writing the letter ‘e’ in the Graffiti alphabet. A fraction of a second later the recognized character will be added to the input area on the left (changing it to read ‘cheshire’). The “ink” from the pen stroke continues to be displayed for a short time, or until the pen is brought back to write the next stroke.



**Figure 8: Pen input integrated with Find dialog box.**

This approach is possible only if all the text input dialog boxes in the application can be modified to include the Graffiti widget. This requires access to the source of the application and involves extra work in both doing the initial modifications and in supporting the modified version. An advantage of doing the integration is that the recognizer can be configured to match the expected input. For example, a numeric input field will only enable number recognition, and will therefore have a smoother user interface and potentially a higher recognition rate.

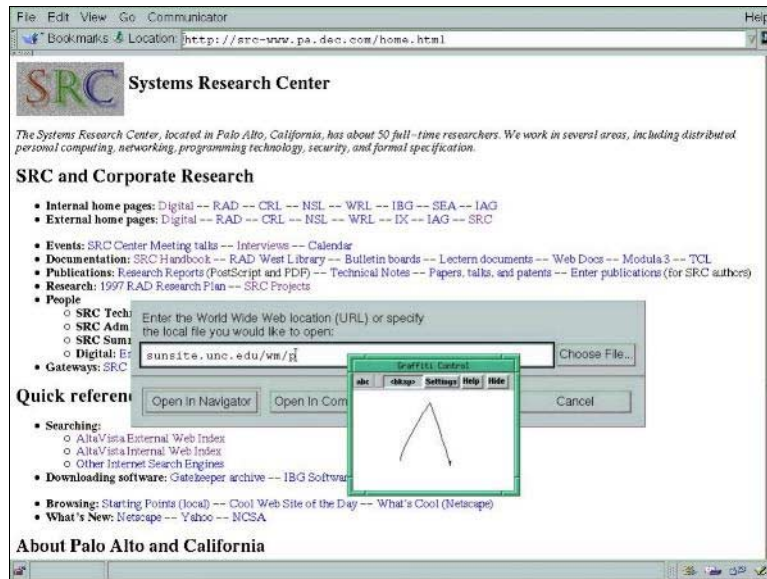
### 2.4.2.2 Input box

The original Graffiti™ implementations and the PalmPilot have a special input area, any pen strokes drawn in this area are sent to the recognizer and the resulting character is passed as input to the current application. Such an input box can be implemented using a pop-up window that does not claim the keyboard focus. Any characters entered into the input box should be sent to the application that currently has the keyboard focus. This implementation allows characters to be sent to all applications that can accept input from the keyboard, without needing any modifications to the application. The appropriate screen placement for the input box depends on the context in which it is used, since the box obscures



some part of the screen. Therefore, the input box must be moveable when it is needed, and hidden when it is not required. If the button on the barrel of the pen is pressed and the pen is tapped on the screen then the input box will appear at (or move to) the point tapped. A “hide” button on the box causes it to disappear.

The input box, which was implemented as a Tcl/Tk application, has the blue border in Figure 9. Netscape Navigator’s URL (uniform resource locator) entry box has the keyboard focus (its cursor flashes to indicate this status). As in the previous example in Figure 8, this screenshot shows the situation just after the user has lifted the pen. The Graffiti letter ‘a’ has been entered and will be sent to the URL box. The row of buttons that can be seen at the top of the input box show the state of the Graffiti recognizer (it is modal – gestures are used to switch between lower case, upper case, numeric and symbol input) and the last character recognized.



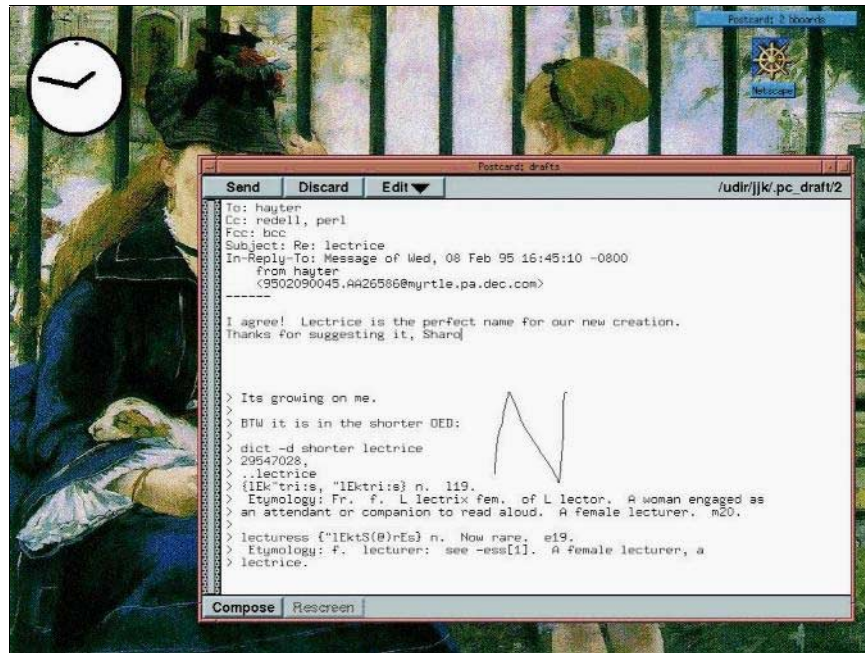
**Figure 9: Graffiti input box above Netscape.**

In addition to providing Graffiti input, the input box can be switched to displaying an on-screen keyboard and characters are entered by tapping them with the pen. Most users found this to be more clumsy than using Graffiti but it was useful occasionally, for example when the input needed to be a key like “Escape” which has no equivalent in the recognized alphabet.

### 2.4.2.3 Writing on the window

The most natural way to enter text is to write directly on the window to which input is directed. This method is similar to using the input box described in the previous section, without having the box obscuring the screen. Since the pen is normally used in place of a mouse to click on buttons or drag items around the screen, there needs to be a way of indicating when a pen stroke should be taken as input for the character recognizer. This was done using the button on the barrel of the pen, when it is pressed the stroke is sent through the recognizer and the resulting character presented as though it was typed on a keyboard. In order to allow the user to see the character drawn, an inking plane shows the stroke and maintains the ink for a short time after the pen has been lifted. This implementation was found to improve usability significantly over one where the pen stroke was tracked but the ink was not shown. While it did not change the recognition error rate, the user was able to see why a character was incorrect.

A more serious problem involves giving feedback to the user. Graffiti is a modal system, the same stroke represents a zero, a lower case letter o, an upper case letter O and an at sign (@) depending on the preceding strokes. We observed several occasions where users were accidentally in numeric mode (or shift or control lock). In this situation, users were unable to understand why letters were not entered correctly. For this reason, many users preferred using the input box for complicated strings of text, like URLs.



**Figure 10: Writing on the window to compose email.**

Figure 10 shows an email message being composed by writing on the window. As with the other examples it shows the state of the screen just after the user has lifted the pen, but before the character is inserted in the composition (in this case the 'n' will complete the name 'Sharon'). If the input box had been used, it would have covered some of the mail message being composed and the user would have had to keep on moving it out of the way. Particularly when editing something like an email message, the ability to position the cursor by tapping with the pen and then just start writing proved highly useable.

### 2.4.3 Audio

The Lectrice hardware was designed with audio input. The original intention was to support experiments with spoken control of the device. Therefore care was taken to ensure noise was not picked up in the cable connecting the built-in microphone to the main board, and the codec was selected to support the 11.25kHz 16-bit sample rate the speech recognition community recommend. We performed a few experiments passing audio samples to a remote speech recognition engine, but it was not used enough to draw conclusions. The lack of progress on audio input was partially due to the fact that relevant speech recognition technology was not sufficiently mature until after the end of the project.

Several potential users reacted favorably to the audio input, mainly wanting it to allow dictation of notes and annotations to documents. During these discussions some people requested a camera to allow still snapshots to be used as annotations. They did not consider this functionality to be extremely important: it was clear potential users would not be prepared to compromise the size, weight or price of the device to get this extra functionality.

## 2.5 Common Applications

Three applications are commonly used with Lectrice. The first, the Lectk Virtual Paper document viewer, is the main application for experimentation. The second is Adobe's Acrobat Reader for PDF documents, the publicly available application that most closely resembles Lectk. The third application is Web browsing, using Netscape Navigator or Communicator, one of the most persuasive uses of this class of device. Lectk can be used in any of Lectrice's operating modes, including standalone. Acrobat and Netscape have not been ported to run natively on the Lectrice, so they run on a server and remote their display. This configuration limits them to the networked modes: wireless and tethered. A more complete software suite would include these two applications in native mode on the virtual book, along with support for web downloading and caching to enable off-line browsing.

### 2.5.1 Lectk – the Virtual Paper viewer

Previous research had shown that standard computer interfaces are poorly suited for long reading sessions. For this reason, sequential reading of sizable on-line documents is uncommon, even though sequential reading is the usual mode for printed material.

The Virtual Paper project had developed the Lectern viewer, designed for on-line reading on a desktop screen. This technology was a natural starting-point for the Lectrice system. A repository of Virtual Paper documents already existed, and it was easy to add documents both from electronic formats (PostScript and later PDF) and by scanning from paper. Lectern is written in Modula-3, which requires a large runtime system. Rather than running the Modula-3 runtime, Lectrice uses the smaller Tcl/Tk scripting system with C extensions to give the required performance. The Lectk application remains faithful to the original user interface, while including extensions for the mobile platform.

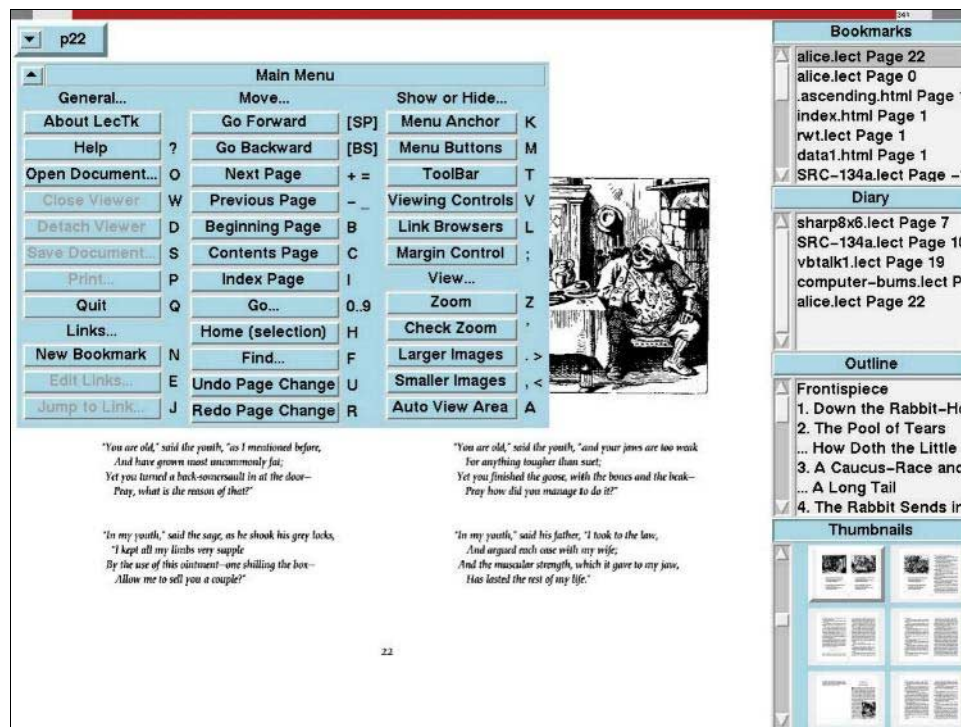


Figure 11: Lectk with the main UI elements showing.

Figure 11 shows the Lectk application. This picture shows an uncommon situation: most of the user interface elements are visible. In typical use, only the material being read would be showing. On the left at the top is the menu anchor, which displays the current page number and a triangular tag to pop up the main menu. This menu is just below, and has buttons for all the viewer functions, together with the keyboard shortcuts used on the desktop version. On the right side of the screen, the “links bar” shows bookmarks (mainly automatically generated) in recent documents, a diary of favorite bookmarks, the table of contents for the current document, and thumbnail views of the pages in the current document. Each of these controls can be revealed or dismissed with a pen stroke or button press. Even with all these controls displayed, the majority of the screen is given to the material being read. Lectrice and Lectk are designed to operate with these controls hidden almost all of the time.

One of the goals of the Virtual Paper project was to support both computer-generated documents and scanned ones. With all scanned documents and many other electronic formats, it is not possible to reformat the document to match the screen; scaling is the only option. Newer electronic representations — HTML, for example — allow dynamic reformatting, but a large body of technical and business documentation is still formatted for paper. In order to allow fast page turning on slow microprocessors (such as Lectrice’s 25MHz R3000), Virtual Paper documents are stored in pre-rendered (and anti-aliased) format using a simple compression. Using current microprocessors and a format like PDF, on-the-fly rendering, scaling and anti-aliasing is possible, but the page layout is still fixed.

Comparing the use of Lectk on a desktop to its use on Lectrice indicates that the on-line reading experience is improved if neither a keyboard nor a pen need be used. Lectrice’s buttons are sufficient for activating common functions, such as moving forward and backward through the material. The pen is reserved for less common activities.

During normal reading, a single button is used to advance through the material. Since typical page sizes are relatively large, readers do not usually keep their hand hovering over the button, preferring to relax into a natural pose. To go backwards, the reader presses a different button. This page-turning process is analogous to the same activity on a paper book. Depending on how they sit, people use either their left or right hand for controlling Lectrice. This reader behavior indicates that button functions should be identical on the left and right sides of any virtual book.

One problem with buttons, compared to page flipping in a standard book, is that the action for moving forward and backwards is so similar. Even when deeply engrossed in a real book, a reader is unlikely to make a mistake and flip backwards rather than forwards. On Lectrice, the page-flipping buttons are positioned for single-handed operation, and readers occasionally hit the wrong button. The solution to this problem is to use visual cues in the user interface. When going forward to the next page, Lectk draws the page from the top right corner down. The underlying algorithm first paints a rectangle in the top right corner then progressively fills in L-shaped sections. For going back to the previous page, the paint is done from the top left. These animations mimic a reader turning book pages by pulling from the top corner.

At first, the page-turning animations just seem cute, but many readers found them helpful after extended use. As has been emphasized above, people prefer to concentrate on their reading material. When readers are following a detailed argument, it can be very jarring to try to connect the words from the bottom of one page with those on the start of the previous page. The simple paint cue reduces this effect, because readers quickly see when they press the wrong button. After a short period using Lectrice, checking the cue is entirely unconscious. An additional advantage with this paint algorithm, as opposed to the simpler one of doing repaints from right-to-left for forward and left-to-right for backwards, is that the bottom lines are left intact until the end of the page flip. This feature allows an experienced user to start the page

turn before finishing the current page (as is often done when reading from paper), permitting a smooth reading flow.

Navigation of the document needs to be fast to prevent the reader rejecting online reading because it seems slower than finding pages in a book. (We did not do any formal tests, but formed the impression that users have higher standards for computer based systems than paper ones.) When flipping through pages, or jumping between sections of a document, the device must have the fastest page turn rate it can. Since the page turning animations take more time than simple screen updates, they should be disabled while navigating, but enabled during sustained reading.

After more experience with the Lectrice, a second problem became evident. There is no feedback to indicate the position within a document. When using paper documents, thickness gives readers an indication of the size of their document and how far through it they are. Again a visual cue seems ideal, but pixels are precious: the 122 dot per inch, 10.4 inch screen is barely sufficient to display full size letter documents (with margins cropped).

The first attempt at providing positional feedback added a thin-as-possible bar at the top of the screen. As can be seen from Figure 12, the bar consists of a central red region with indicators on each end. The size of the indicators is set based on the number of pages in the complete document (one pixel width per page, which works for documents up to a few hundred pages). At the beginning of the document, the right side indicator is dark gray, indicating all the pages are ahead. As pages are turned, the thickness is transferred to the left, indicating that pages are moving behind the current point. This visual cue represents the thickness of book pages on each side. The small size of the bar made it hard to use this information, and it never seemed intuitive.

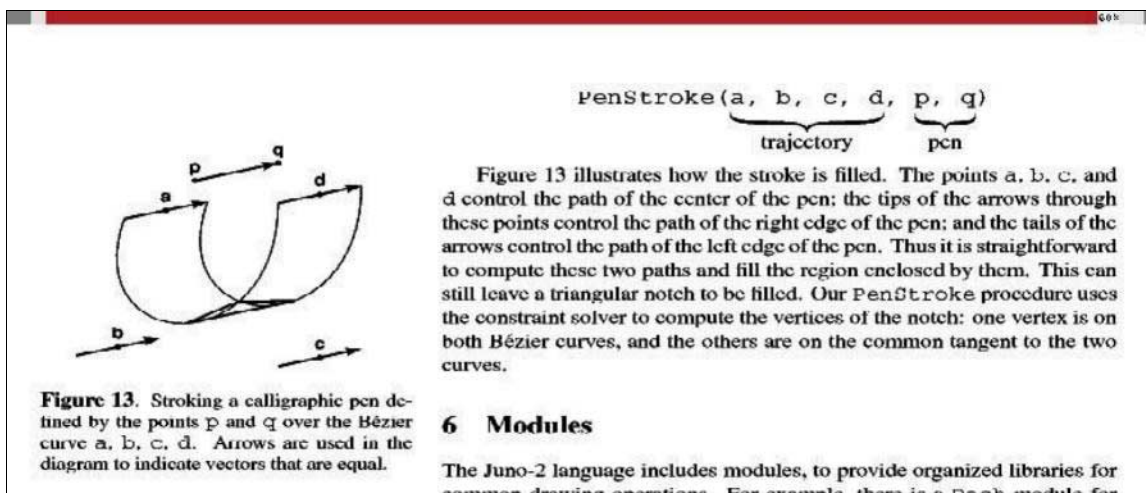


Figure 12: Top part of Lectk window, showing the position bar.

The second method of indicating position was more successful in giving the user feedback, but failed because it used up too many pixels. In this model, the width of the displayed page is reduced to less than the screen width, and a margin placed on the screen. At the start of the document, the page is painted on the left side of the screen and the margin is entirely on the right. Advancing through the document moves the positioning of the page, removing margin thickness from the right and placing it on the left. Again, think of the thickness of pages on the right and left when advancing through a book.

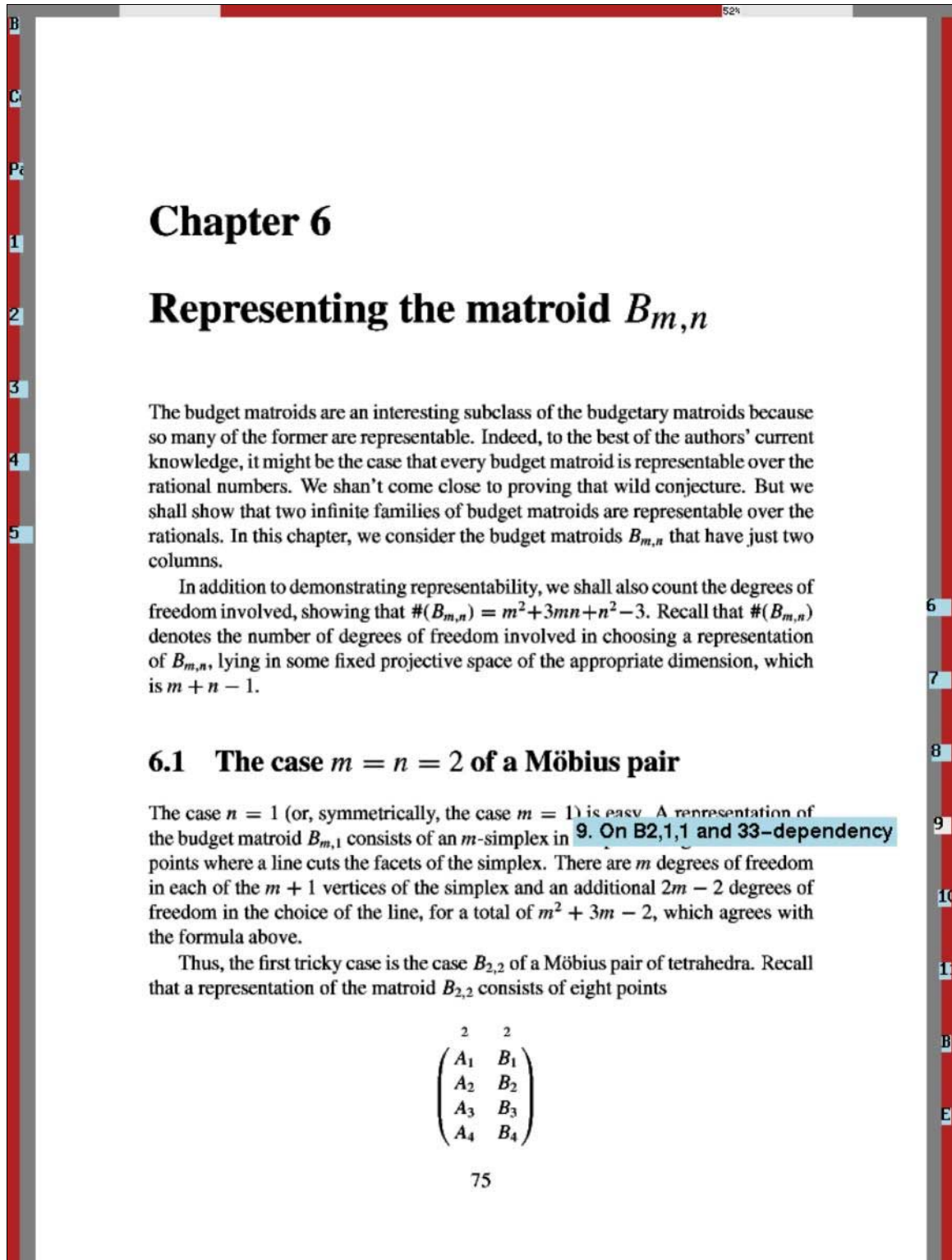


Figure 13: Lectk with position margins.

The position margins can also display chapter marks, if they are present in a document. These chapter marks are similar to the thumb indexes that are typical on a large paper dictionary. Chapter marks on the right are ahead in the document, those on the left are behind. The marks extend from the edge of the

screen so that they just touch the paint area when the marked page is displayed. The goal of the marks is twofold. First, they enable the reader to estimate the distance to the next chapter (feedback from users indicates that this feature is important). Second, the marks can be used like thumb indexes: tapping them with the pen causes Lectk to jump to the associated chapter. “Floating” the cursor in a mark will cause the chapter title to pop-up.

In Figure 13 the page being displayed is positioned at the start of Chapter 6 of the document. The page marker with the number ‘6’ in it indicates this by just touching the edge of the white page display area. Moving forward in the document would cause this marker to jump from the right margin to the left one (since the page starting Chapter 6 is earlier in the document), moving backwards will result in the gray border being visible between this marker and the page. Previous chapters are on the left and subsequent chapters are on the right. The pen is hovering over the “9” mark and has caused the title of Chapter 9 to pop-up.

The future of this kind of mechanism is uncertain. If virtual books have the ability to scale images dynamically, the page can be drawn so the indicator margin never covers any words. On the other hand, the image is reduced in size compared to the printed version, so users are likely to choose the larger image over the visual cue. A potential solution is to use a larger screen, so both the image at normal scale and the margin can be displayed.

One other navigational feature is built into the Lectk application: a slow HTML rendering engine. This was designed for experimentation to understand how Web pages could fit into the Lectk paginated model, since they are designed for a scrolling screen. In practice it is used mainly in standalone mode, displaying an index page that is placed on the Flash card by a download utility. The document titles on this simple HTML index are links that can be tapped with the pen. The HTML feature was also used when connected to the network to display the index pages in the Virtual Paper document repository.



**Figure 14: Standalone Lectrice showing HTML index page.**

### 2.5.2 *Acrobat Reader*

Adobe Portable Document Format (PDF) is replacing the less portable PostScript as the standard way to encode paginated documents for distribution. Running the Acrobat Reader on a Unix machine and using the X protocol to display its windows remotely on the Lectrice allowed experiments with this application. In spirit, Acrobat has many of the same properties as the Lectern viewer. However, the normal mode of its user interface has considerably more visible elements, wasting many pixels. The full screen mode removes all the clutter and acts much like Lectk. The button manager maps buttons to the same function in Acrobat that they have in Lectk.

One experiment with this configuration tried to determine if the Acrobat scrollbar provides an adequate method for indicating progress through a document. In practice, the scrollbar proved little better than any of the other methods, though it was acceptable to frequent computer users.

Unlike Lectk, Acrobat renders pages on-the-fly, which works well on systems with a fast processor. As with Lectk, anti-aliasing improves the effective screen resolution (See Section 3.2.) Unfortunately, the Acrobat anti-aliasing algorithm tends to make the characters lighter. With certain fonts, this effect causes text to be harder to read. Experience reading these fonts suggests that making “light” typefaces darker improves the legibility on a backlit LCD.

### 2.5.3 *Browser*

The third major application used with Lectrice was the Netscape Navigator/Communicator Web browser. The application runs on a remote Unix machine and uses X to display its windows on the Lectrice. This configuration allowed the browser to be used on Lectrice without needing to port it. The document model used by the Web is different from printed material in several ways. HTML documents are formatted at display time to match the display in use (although a number of Web sites confound this functionality by hard-coding sizes in the page layout). Web browsers (and hence designers) use a scrollbar-based model rather than a paginated one. In addition, the use of multiple colors is common in the Web and relatively uncommon in traditional documents (other than magazines).

The activity of browsing has some differences from reading. In particular, the pen comes into its own as a navigation device, and the buttons become less important. Using the pen to tap on links is far more natural than clicking on them with a mouse. The links can be anywhere on the page, so it is hard to find a convenient way to select them using the buttons. The user therefore is likely to be holding the pen all of the time (in contrast to the reading case). The buttons become clumsy when holding the pen, but a gesture-based interface becomes compelling. A left-to-right stroke of the pen causes the page to scroll forwards and a right-to-left causes a backward movement. When reading a long document with little or no hypertext, most users eventually put down the pen and return to using the buttons.

Web browsers tend to present a very cluttered interface, with many menus, buttons and controls. Figure 15 shows that these components are normally turned off for Lectrice use and the Web page is viewed filling the whole screen. As with the help system, the user interface gadgets are useful to novice users. With experience, gesture shortcuts (and the “right click” pop-up menu which can be accessed with the pen) become the natural tools, because readers prefer using as many pixels as possible to display useful information.



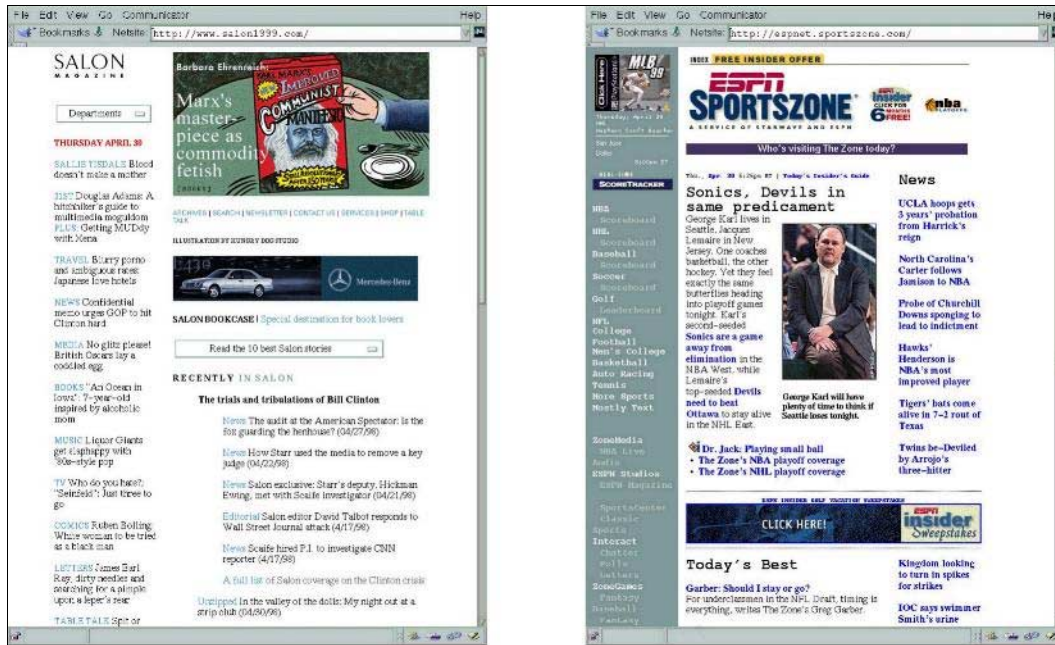


Figure 15: Two screenshots of Netscape on Lectrice.

The pen, normally in the write-on-window mode, proved acceptable for entering URLs and filling out the forms encountered on the Web.

### 3 Competing with Paper

The key question explored by the Virtual Book project was how well electronic media can compete with paper documents. When we began the project in mid-1994, we accepted that digital information appliances could not yet do as well as paper along many dimensions, but believed that an electronic system could sometimes be superior. The task, then, was two-fold: come as close to paper as possible in the dimensions of legibility, ergonomics, robustness, and similar attributes; then, surpass it in other attributes such as searchability, flexibility, and interactivity. This section reports the findings on the first half of the challenge; Section 4 discusses the second half of the challenge.

#### 3.1 A Better Definition of the Target

Paper is such a versatile medium that a virtual book cannot expect to compete across the full range of paper's applications. Instead, the project focused on an important subset: documents ranging in size from roughly paperback book format (*i.e.* about 5" x 7") up to the full 8.5" x 11" page (U.S. letter size, similar to size A4) that is standard for journals, business correspondence, and many books and magazines in the United States. Media and formats such as blueprints, traditional newspapers, studio-quality photographs, and PostIt™ notes were outside the scope of comparison.

Within the chosen size range, a critical assumption was that the source material could not necessarily be re-formatted. This assumption accommodates both scanned material and documents provided in “compiled” formats such as PostScript and PDF, where layout decisions have been fixed and are not practically changeable. Section 4.4 relaxes this assumption to include content available in a markup language such as HTML. Documents that can be formatted dynamically make the task of an electronic device much easier. It is reasonable to believe that eventually, most interesting content will be available in such a format. For the near- and medium-term future, however, fixed-format documents will be quite common. General-purpose electronic reading devices will therefore need to accommodate them.

### **3.2 Legibility**

The most obvious point of comparison between paper and any alternative is legibility. How “readable” are text and figures on the screen? Unfortunately, the answer to this question is complicated and highly subjective. It depends on very qualitative factors such as font design, ambient lighting conditions, and the health of the reader's eyes, in addition to more quantitative factors such as resolution and contrast.

While acknowledging the subjective nature of legibility, the Virtual Book project attempted to gauge the importance of measurable features like dot density and screen size. A key objective was to try to discover acceptability thresholds or ranges; such information could usefully serve display makers as well as product design and planning groups.

The analysis in this section focuses on two factors of the display that affect legibility: physical size and dot density. These are the physical properties behind the parameters that are important to the user: how many symbols per page, how many dots in each symbol and how clear each symbol is. In order to make fair comparisons across combinations of parameters, this discussion will use a “canonical” page and show examples of its appearance in different cases. This page is representative of the target content: formatted for printing on 8.5" x 11" paper, not easily re-formattable, and with typical margins, font style, and font size.

#### **3.2.1 Display Size**

The controlling factor for the size of a virtual book display is the source material size. An 8.5" x 11" page must be scaled to fit on the screen. This section describes the factors that suggested the 10.4" display would work for Lectrice.

Most people's experience with computer displays is that bigger is always better. This observation is true for virtual book displays as well, but only up to a certain point. Beyond a certain size, which varies from person to person, the increased bulk and weight of the virtual book outbalances the increased legibility. Conversely, below a certain size, document pages must be reduced to the point where characters are uncomfortably small. The feasible range of sizes seemed to be from about that of a paperback book up to about 25% bigger than an 8.5" x 11" page. Computer displays are typically specified by their diagonal measurement, assuming a 3 x 4 aspect ratio (when viewed in portrait orientation). The feasible range thus corresponds to displays with diagonals from about 7" to 17". The optimal experimental methodology would have been to buy displays at many sizes from 7" to 17", to package them up in appropriately sized Lectrice cases, and to compare their legibility directly. Unfortunately, this approach would have been very expensive. Even if cost had not been an issue, it would have been very difficult to control for other important display characteristics such as dot density and contrast ratio.

An alternative is to consider the effects of scaling on the text of the page. One metric for measuring this effect is the *x-height* (or body height) of the text in the resulting image. The x-height is the height in printer's points of a lower case "x". Typography texts recommend that x-height be a minimum of about 5 points, requiring fonts to be approximately 9 points or larger. On paper, characters smaller than this size are legible, but are not good for sustained reading.

Diagonal	x-height	Example
7.0	3.5	Sample text
7.5	3.8	Sample text
8.5	4.3	Sample text
9.5	4.8	Sample text
10.4	5.2	Sample text
12.1	6.1	Sample text
13.3	6.7	Sample text
14.1	7.1	Sample text
15.0	7.6	Sample text
16.0	8.1	Sample text
17.0	8.6	Sample text

**Table 2: x-height (points) for various screen diagonals (inches).**

The chart above is generated by measuring the x-height of the characters in a printed copy of the canonical page and computing the new x-height that would result from scaling it (regardless of resolution of the display) to fit on a given display<sup>7</sup>. (For the sizes of the examples to be correct the surround box of this figure should be 3.5" x 2.5".)

Rendered on paper, the samples in the chart have higher dot density and better contrast than on any current electronic display device. Hence, this chart can be used to confirm the lower bound for the size needed to display a full page of text with acceptable quality. This methodology is subjective, and there is no single value that would be correct for all readers. Nonetheless, the samples, the literature and our own experience suggest that an x-height of at least 5 points is necessary for sustained reading on an LCD. This observation indicates that a display with diagonal at least 10 inches is required for the class of content represented by the canonical page.

### 3.2.2 Display Density

Once the physical size of a character has been set the next factor that determines its legibility is the number of dots used to form its shape. After the arguments of the previous section have been used to select a display size, and therefore character size, the number of dots per character is directly related to the pixel density of the display. The examples given in the next section will demonstrate that characters must have sufficient dots to be formed for legibility.

<sup>7</sup> The formula for computing x-height from display diagonal is  $x\text{-height} = 72 * \text{display-diagonal} * .8 * .95 / 108$ :

72 is (roughly) the number of points per inch;

.8 converts diagonal to height (assuming 3 x 4 aspect ratio);

.95 allows 5% for margin;

108 is the empirically derived ratio of x-height to bounding-box for a printed copy of the canonical page.

The case of the Lectrice was designed for a 10.4" diagonal display. Available displays provided three possible dot densities, measured in dots per inch (dpi): a 640 x 480 VGA display gives 77dpi; an 800 x 600 SVGA display gives 96dpi; and the 1024 x 768 XGA display gives 122dpi. Lectrices with all three resolutions were built to enable comparison. The XGA display was one generation earlier technology than the lower resolution displays, and therefore had lower contrast, less brightness, worse color, and more reflections than the other displays. Nonetheless, it was still the best for sustained reading.

These pixel resolutions seem very low compared to the 300 to 600dpi found on typical printers. However, all three displays permit gray-scale rendering, as opposed to the black and white dots on printers. Gray-scale allows anti-aliasing to be used to increase effective density.

Anti-aliasing works by relying on blurring of color between pixels. This can be demonstrated with a simple experiment. Make a copy of Page 32 and hang it at eye-level. Stand back about 10 to 15 meters, and look at the page. Notice that the pixels blend together. Slowly walk towards the page, and watch the pixels resolve into discrete squares. The eye can only resolve objects if the angle subtended between their edges is above a threshold (that varies from person to person). From a long distance the black and gray squares subtend an angle below this minimum and the eye sees an averaged color; up close the squares can be uniquely resolved.

It is worth commenting that anti-aliasing is somewhat more of a challenge on LCDs compared to CRTs. The LCD pixels have very sharp edges, whereas the brightness from a pixel on a CRT falls off with a Gaussian distribution at the pixel edge. Thus the CRT has some blending of the pixels as a result of its physical properties, in addition to that caused by the angle subtended at the eye. With an LCD all the blending must be done by the eye, so the anti-aliasing becomes more effective as the angle subtended at the eye becomes small enough for it to be impossible to resolve a single pixel width. Thus in an appliance for reading it is preferable to have the pixel spacing small enough that pixel wide lines cannot be resolved. This desire conflicts with the expectations set by current laptop computers which (for historical reasons) use system fonts that require the user be able to resolve single pixel wide lines (consider an "m" in a menu).

### 3.2.3 *Display Examples*

This section illustrates the display tradeoffs. While the examples can only really be seen on an actual device, the printed versions shown are a close (but not an exact) match. The canonical page is used in each case to allow easy comparisons. The text selected is a page out of a computer science Ph.D. dissertation. It is a good example of material formatted by the author for printing on 8.5" x 11" paper, but subsequently made available on-line. The next three pages show the full text as rendered on a Lectrice with each of the screen resolutions (VGA, SVGA and XGA). They should print at the exact size they appear on the screen with the diagonal of the bounding box being 10.4".

When looking at the examples, remember the two user parameters: amount of information per page and number of dots per symbol. The first set of examples (rendered using Lectk) keep the amount on each page constant and vary the number of dots per character (resolution). The second set of examples (rendered using Netscape) considers the tradeoff between keeping the amount of information constant, and keeping the number of dots per character constant.

The example page captured from a 122dpi screen (XGA) is satisfactory for sustained reading:

caching, user processes are no longer clients of the file service per se, but clients of a local “file-cache service” instead. The file-cache server is a client—and in fact the only local client—of the file service proper. This indirection is hidden from user processes by machinery below the operating system interface. The file-cache server, or *cache manager*, may be implemented as a module inside the operating system, or as a separate user-level process, or as some combination of the two. Figure 1.1 illustrates the distinction between generic caching and non-caching organizations.

### 1.3 Disconnected Clients

The Achilles’ heel of distributed computing is service availability. In non-distributed systems all services are provided locally, so whenever a user’s process is running it can obtain any of the services supported by the host machine. If the operating system crashes or the hardware fails then this is immediately apparent, and the machine can be rebooted or repaired as necessary. Service availability is determined solely by the reliability of the local hardware and system software.

In contrast, a distributed system client obtains many services remotely, exchanging messages over a network with various server machines. Service availability is contingent upon more than just the local machine being up: the server machine, the server process, and all intervening network components must be operational as well. A client which is functioning normally, but which cannot obtain a service due to remote failure is said to be *disconnected* with respect to that service. A disconnected client will not in general know the cause of disconnection—the only sure evidence is a message sent to the server process which is not acknowledged in a reasonable period of time.

Disconnections are bad because they impede computation. Typically, a process which invokes a service from which it is disconnected will block indefinitely or abort. Either result is likely to frustrate the user. This is particularly true when the service is within the capabilities of the local machine, but remote access has been configured for the purpose of sharing. In such circumstances the autonomy gained by decentralizing computation is lost; the user does indeed have control over local resources, but they are not sufficient to get real work done.

Disconnections are a real-life problem—they are not hypothetical. Every serious user of a distributed system has faced situations in which critical work has been impeded by remote failure. Lamport noted this long ago with his wry definition of a distributed system as “one where I can’t get my work done because of some computer that I’ve never heard of.” The severity of the problem varies from system to system, but, in general, the larger and more heterogeneous the system the more acute it is likely to be.

The example page captured from a 96dpi (SVGA) display is less clear:

caching, user processes are no longer clients of the file service per se, but clients of a local “file-cache service” instead. The file-cache server is a client—and in fact the only local client—of the file service proper. This indirection is hidden from user processes by machinery below the operating system interface. The file-cache server, or *cache manager*, may be implemented as a module inside the operating system, or as a separate user-level process, or as some combination of the two. Figure 1.1 illustrates the distinction between generic caching and non-caching organizations.

### 1.3 Disconnected Clients

The Achilles’ heel of distributed computing is service availability. In non-distributed systems all services are provided locally, so whenever a user’s process is running it can obtain any of the services supported by the host machine. If the operating system crashes or the hardware fails then this is immediately apparent, and the machine can be rebooted or repaired as necessary. Service availability is determined solely by the reliability of the local hardware and system software.

In contrast, a distributed system client obtains many services remotely, exchanging messages over a network with various server machines. Service availability is contingent upon more than just the local machine being up: the server machine, the server process, and all intervening network components must be operational as well. A client which is functioning normally, but which cannot obtain a service due to remote failure is said to be *disconnected* with respect to that service. A disconnected client will not in general know the cause of disconnection—the only sure evidence is a message sent to the server process which is not acknowledged in a reasonable period of time.

Disconnections are bad because they impede computation. Typically, a process which invokes a service from which it is disconnected will block indefinitely or abort. Either result is likely to frustrate the user. This is particularly true when the service is within the capabilities of the local machine, but remote access has been configured for the purpose of sharing. In such circumstances the autonomy gained by decentralizing computation is lost; the user does indeed have control over local resources, but they are not sufficient to get real work done.

Disconnections are a real-life problem—they are not hypothetical. Every serious user of a distributed system has faced situations in which critical work has been impeded by remote failure. Lamport noted this long ago with his wry definition of a distributed system as “one where I can’t get my work done because of some computer that I’ve never heard of.” The severity of the problem varies from system to system, but, in general, the larger and more heterogeneous the system the more acute it is likely to be.

The example page captured from a 77dpi (VGA) screen is too hard to read for long periods:

caching, user processes are no longer clients of the file service per se, but clients of a local “file-cache service” instead. The file-cache server is a client—and in fact the only local client—of the file service proper. This indirection is hidden from user processes by machinery below the operating system interface. The file-cache server, or *cache manager*, may be implemented as a module inside the operating system, or as a separate user-level process, or as some combination of the two. Figure 1.1 illustrates the distinction between generic caching and non-caching organizations.

### 1.3 Disconnected Clients

The Achilles’ heel of distributed computing is service availability. In non-distributed systems all services are provided locally, so whenever a user’s process is running it can obtain any of the services supported by the host machine. If the operating system crashes or the hardware fails then this is immediately apparent, and the machine can be rebooted or repaired as necessary. Service availability is determined solely by the reliability of the local hardware and system software.

In contrast, a distributed system client obtains many services remotely, exchanging messages over a network with various server machines. Service availability is contingent upon more than just the local machine being up: the server machine, the server process, and all intervening network components must be operational as well. A client which is functioning normally, but which cannot obtain a service due to remote failure is said to be *disconnected* with respect to that service. A disconnected client will not in general know the cause of disconnection—the only sure evidence is a message sent to the server process which is not acknowledged in a reasonable period of time.

Disconnections are bad because they impede computation. Typically, a process which invokes a service from which it is disconnected will block indefinitely or abort. Either result is likely to frustrate the user. This is particularly true when the service is within the capabilities of the local machine, but remote access has been configured for the purpose of sharing. In such circumstances the autonomy gained by decentralizing computation is lost; the user does indeed have control over local resources, but they are not sufficient to get real work done.

Disconnections are a real-life problem—they are not hypothetical. Every serious user of a distributed system has faced situations in which critical work has been impeded by remote failure. Lamport noted this long ago with his wry definition of a distributed system as “one where I can’t get my work done because of some computer that I’ve never heard of.” The severity of the problem varies from system to system, but, in general, the larger and more heterogeneous the system the more acute it is likely to be.

The previous three examples show the appearance of the page when rendered to be the same size on each of the Lectrice displays. Since the characters are the same size, the differences in legibility arise from the change in the number of pixels per character as can be seen from the magnified words. These words also show how two gray levels are used to anti-alias the character edges. First at the full 122dpi:



Then at 96dpi:



And at the lowest 77dpi:





It is instructive to consider the case where reformatting the material is possible. In this case the number of pixels per character (i.e. the screen font size) can be adjusted to fit the same amount of material on each of the screen sizes, or the number of pixels per character can be kept constant and the amount of material reduced. To generate screenshots for all these cases the Netscape browser was used to render an HTML version of the example page. Netscape did not use the same font as Lectk, and it was monochrome rather than anti-aliased. Nonetheless, since the same information was displayed, these magnified words can be compared with the previous page. At 122dpi:



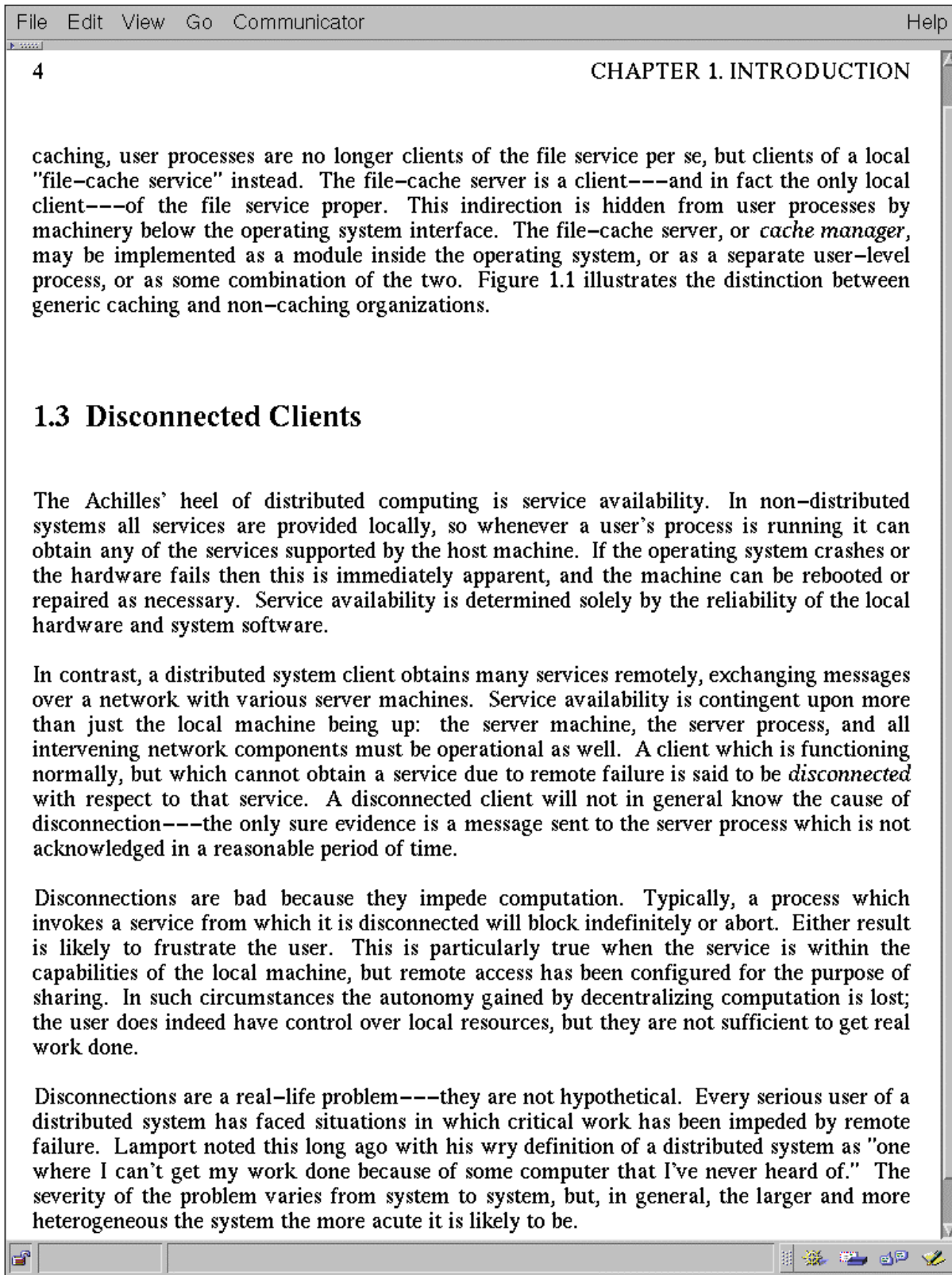
And at 96dpi, using fewer dots per character to keep the same information content:



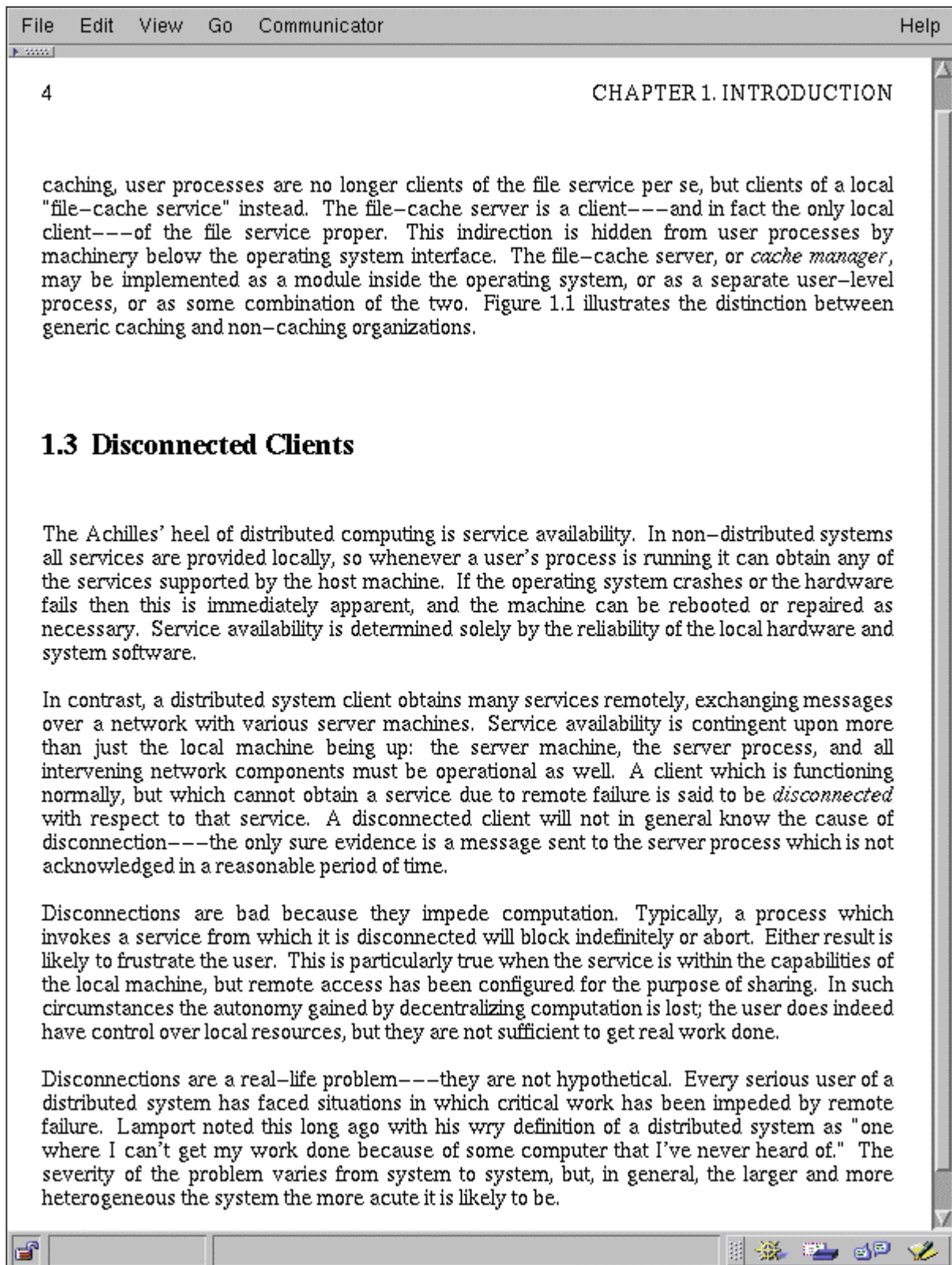
And at 77dpi, using even fewer dots per character:



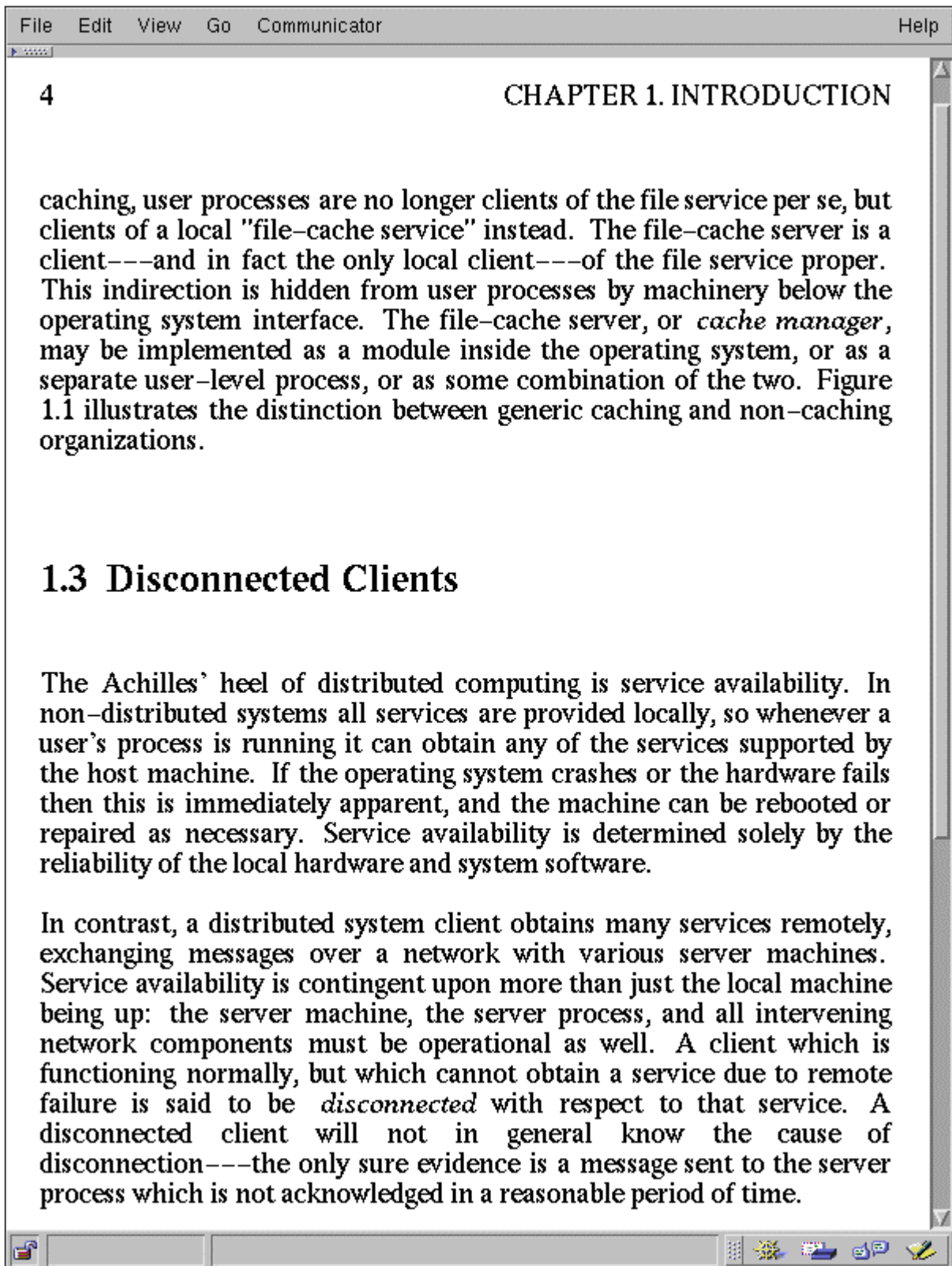
This is the full page, displayed at 122dpi:



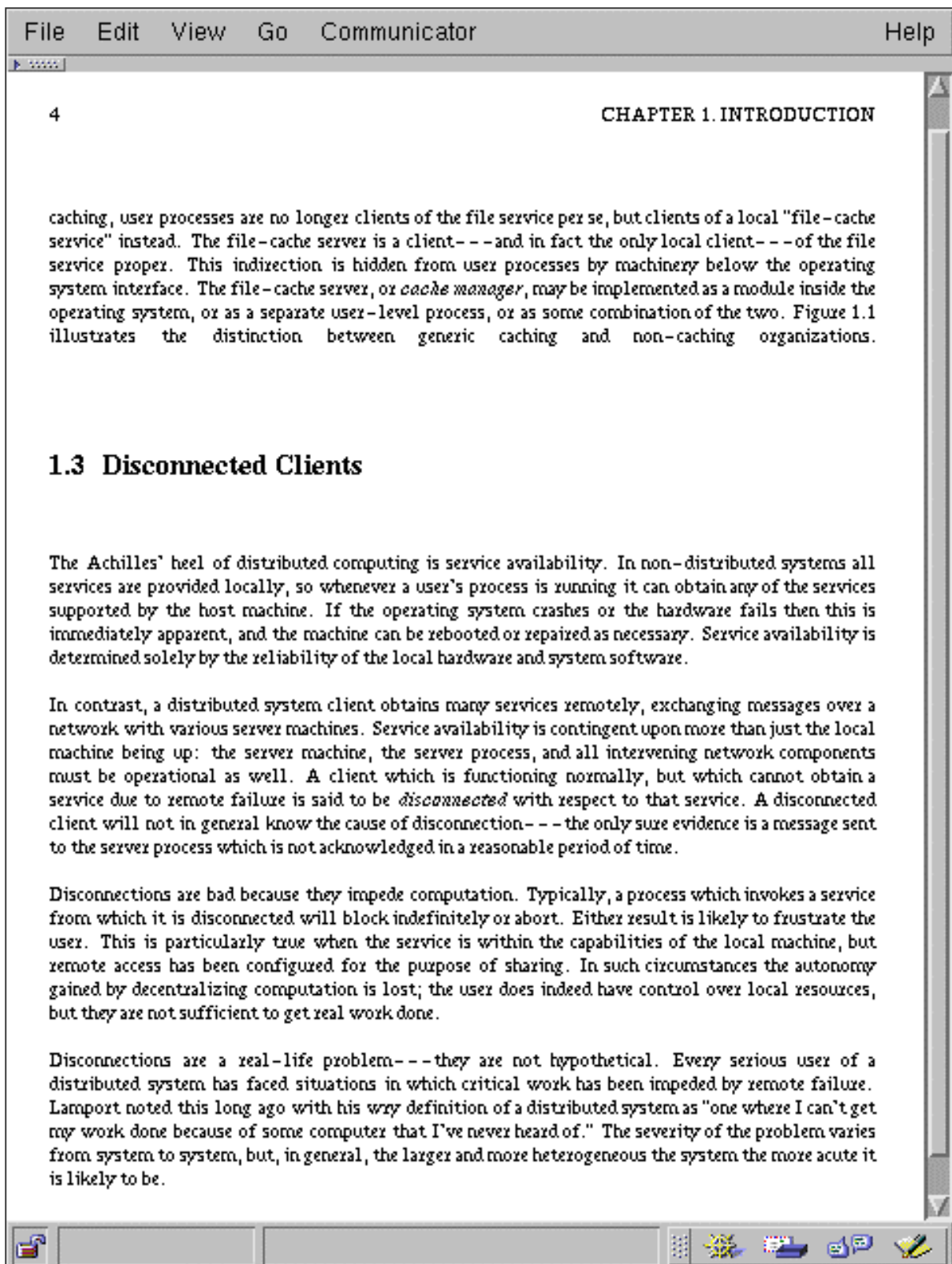
The full page at 96dpi demonstrates that the lack of anti-aliasing makes the type lighter than the LECTK version:



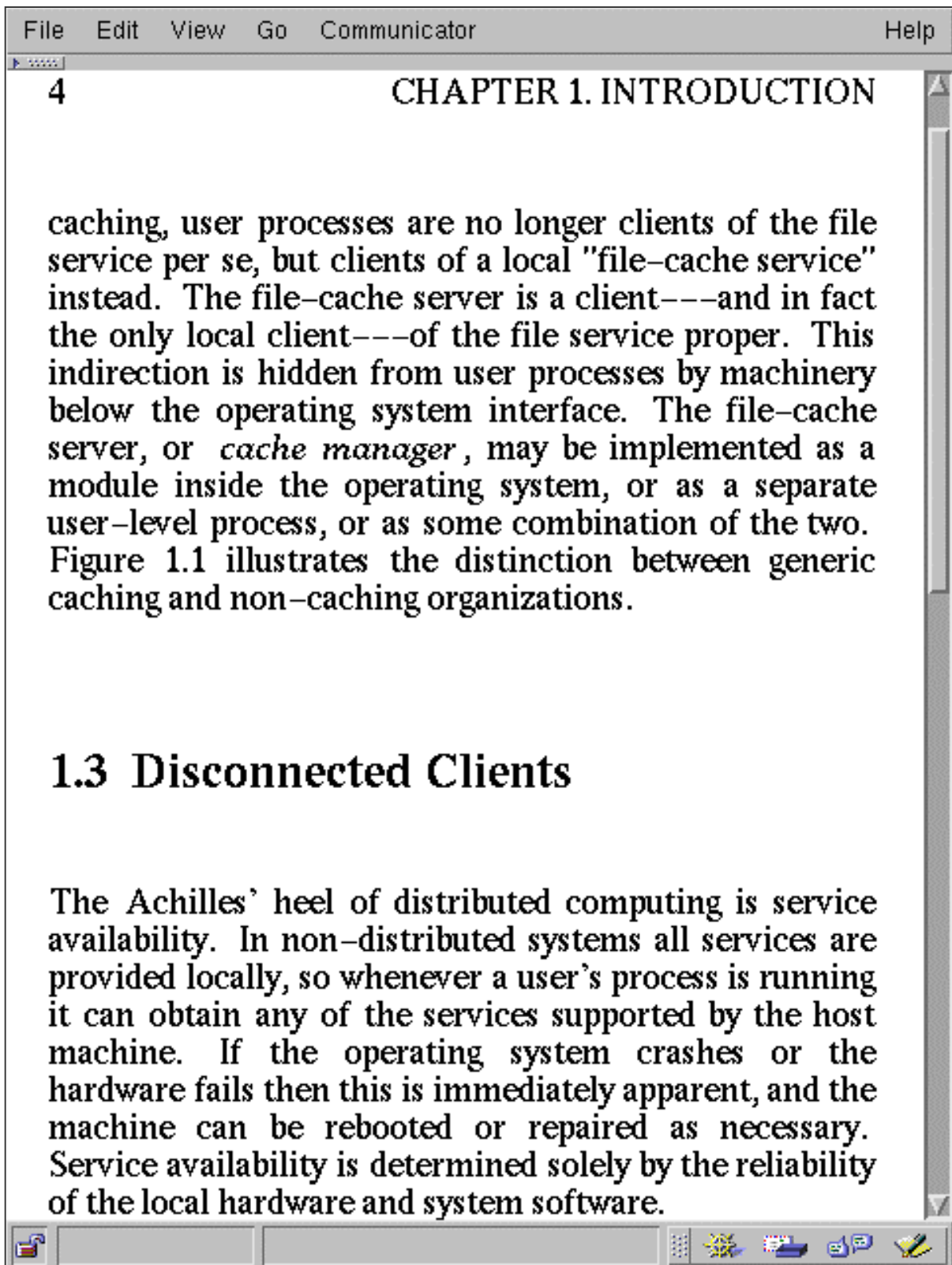
When displaying the text at 96dpi and maintaining the same number of pixels per character, the characters are easier to read, but a couple of paragraphs are lost:



On the 77dpi VGA screen reducing the font size to get the full page makes the text unpleasant to read:



On the other hand, maintaining the number of pixels per character results in very little material being visible. This amount of text may be acceptable for a novel, but not for detailed study:



### 3.2.4 *Other Factors*

Other factors such as brightness and contrast affect legibility. Increased brightness and sharper contrast make for a better display. However, it was hard to design experiments to measure these factors because none of the displays used with Lectrice allows much variation in contrast, and the backlight could be set to only two brightness levels. Newer generations of backlit displays have improved contrast and brightness compared to the displays used in Lectrice, so they would certainly be suitable for a reading appliance. It would be good to establish the minimum acceptable contrast and brightness because these would allow assessment of the usability new display technologies. For example, the reflective displays that are now becoming available seem compelling for virtual books, since they consume about a tenth the power of traditional TFT displays because they have no backlight. Unfortunately, they also have much lower contrast.

One significant difference between Lectrice's XGA screen and the lower resolution ones is in the glare – the reflection of surrounding light from the display. The lower resolution displays were one generation newer, and the significant reduction in glare made them easier to read. However, the reduction in glare did not make up for the lack of resolution, and readers preferred the XGA units.

The field of view of the display has less impact on the Lectrice than it does on a laptop, mainly because on Lectrice the natural reading position is directly in front of the screen. Many readers commented about the lack of uniformity in the viewing angle. In landscape orientation, the Lectrice has equal viewing angles to the sides. In portrait mode, the viewing angle is asymmetric on the left and right sides, because Lectrice's LCDs are designed for laptop use. The keyboard limits the angle the user can see from below on a laptop. As a result, LCDs designed for this market are biased towards viewing from above. This property of the display is more of a quirk than a problem and shows up in practice only when several people are trying to see the same book.

The example illustrations on the previous pages show the improvements from using anti-aliased fonts rather than black and white. In practice, smoothing the fonts makes long reading sessions more pleasant. Therefore, it is essential that the display support a few gray levels — experience from the Virtual Paper project suggest that between four and sixteen levels are sufficient. It is important that either the range of gray levels are uniformly spaced (this is not always the case on LCDs), or the algorithm should be tuned to the available levels.

## 3.3 *Ergonomics*

One of the major goals of a virtual book is to make on-line reading more comfortable. As discussed earlier in this document, using a monitor on a desktop to read electronic documents forces readers to sit in a fixed position. One of the main benefits of the virtual book device is to free readers to be able to make small adjustments to their posture. This capability allows readers to relax and to focus on documents, rather than on the delivery technology.

### 3.3.1 *Physical Characteristics*

A reading device must have certain physical characteristics for users to be comfortable. This section discusses them in approximately the order a reader will typically notice them.

Seeing a virtual book on a table gives a potential user a good idea of its use; it is approximately the same size as a pad of paper and is clearly a device for viewing things because it is dominated by the display. As discussed in Section 3.2.1, the ability to display a complete page at close to 1:1 size forces the screen to be close to 8.5" x 11". In practice, most documents have margins that can be stripped, so a smaller screen, with the case of the device providing the margin, is both practical and makes the device the appropriate size.

The first attribute that a user notices when picking up a Lectrice is its weight. At over five pounds, it is too heavy. Experiments performed at the beginning of the Virtual Book project used a clipboard — weighted in half-pound increments — to judge potential users' reactions to weight. An informal survey suggested that the ideal weight is less than three pounds, but that the users react in the same way to four pounds as they do to three. As the weight is increased above four pounds it becomes less and less acceptable. We believe these limits reflect the way the unit is used. The ideally weighted virtual book would allow users to hold the device in their hands almost indefinitely. Above the maximum weight that allows sustained hand-held operation, the device has to rest on a support (typically a knee or tabletop) during use. Increasing the weight beyond this point matters less, until it is uncomfortable to pick up the virtual book with one hand.

The thickness of the device (for a given weight) matters less when it is being used than when it is in transit. In Alan Kay's original plan for the Dynabook [KG1977], he wanted it to be thin enough that a user could carry other things in the same hand as the device. This observation is very astute, and helps determine the maximum thickness for a virtual book. Experience with Lectrice suggests that at 1.75", it is too thick, although it is just possible to carry a Lectrice and something else under one's arm. Thickness also matters when the device is put in a briefcase or bag for transport.

A virtual book will be used for an extended period and is quite likely to be held in the lap. To remain comfortable it is important that the power dissipated by the device be low enough not to annoy the reader. Lectrice's 15 Watt power dissipation was comfortable only on cold winter mornings. In retrospect, the initial goal of 10 Watts seems to be a reasonable upper bound on virtual book power dissipation.

The industrial design for Lectrice required many discussions about the tactile feel of the device. Characteristics such as size, shape, thickness, margin width, button location, and even paint type are important since the readers hold their virtual books for long periods of time. The hard lines and sharp edges that are typically found on thin laptop computes are unpleasant to hold for long reading sessions. In order to encourage use of the device, the "first contact" impression is also important. The combination of rubber buttons, contoured edges, and rubberized paint on a plastic case gave an acceptable result.

Other features of the physical packaging — relating primarily to style rather than comfort — are not addressed here. For example, wrapping a virtual book in a leather binder contributes to the owner's professional image. Such factors will become more important, since information appliances are in part seen as fashion accessories, rather than dreary computing devices.

### *3.3.2 User Interface Characteristics*

There are three main characteristics in the user interface that are important to allow continuous reading. These are the low cognitive effort required to drive the interface, the lack of distracting user interface elements on the screen, and the responsiveness. The Lectk interface described previously reflects these requirements.



Low cognitive effort involves more than just being easy to use. After a short learning period, common actions have to be performed unconsciously. The reader wants to focus exclusively on the document being read, so it is important that they can perform simple navigation — like flipping forward or backwards by one page — without being distracted from the document. Lectrice’s buttons are crucial to achieving this characteristic, activating them to change pages is simple enough that no conscious thought is required. The positioning of the most common “forward” button as the top one on either margin makes it easy to find without thought.

The need to help the reader stay focused also motivates the lack of distracting user interface elements. During normal use, the display shows only the material being read. It was surprising what disturbed the flow of reading. Even the presence of a small box containing the page number was enough to trip the eye during reading. It did not matter that the page number box did not cover any text and was tucked out of the way in the corner of the screen, users still wanted it removed from the interface because it distracted them from the text.

Fast page turning is important to prevent reader frustration. In order to emulate flipping through a paper document, it seems necessary to be able to turn pages at about 5 pages per second. Page turning during sustained reading can be a little slower, but must still be prompt. In retrospect, it would have been interesting to experiment with scanning through a document by flipping pages that were rendered at a lower resolution. If readers use the shape of document features (such as the pattern of paragraphs on a page rather than their actual content) to locate information, then it might be possible to increase flipping speed by reducing rendering time. On the other hand, if readers perform keyword spotting on the fly, then rendering pages at low resolution would not be useful. It is also possible that the display update bandwidth would be the limiting step on modern hardware, so reducing resolution may not improve page turning speed.

### **3.4 Markup**

Paper is a malleable medium, and the ability to annotate paper documents is one of its major advantages. The pen on Lectrice provides a natural way to perform this function in the electronic representation. The only document annotation mechanism that was implemented in Lectk is to provide simple yellow overlays (similar to PostIt™ notes) that allow electronic ink to be captured in to image files. This user interface component is of limited use, because the notes are not anchored in the document on which they are written: they stick to the screen rather than the page. However, readers of drafts of large technical documents have used the notes to make comments for the author. These notes had to include page and paragraph references as well as the comments. This proof-of-concept indicates that devising a more complete annotation architecture would be well worthwhile. Section 4.4 briefly explores some of the benefits and challenges of such a system.

## **4 Surpassing Paper**

The Virtual Book project focused primarily on making the physical and cognitive effort of on-line reading approximately the same as reading paper. Two hypotheses motivated this goal. The first was that LCD technology was crossing thresholds that would allow flat-panel displays to be used for sustained reading; the results described in section 3 tend to support this hypothesis. The second is that once the experience of reading electronic documents is sufficiently similar to that of paper, most people will choose to read on-line documents instead of hardcopy. This hypothesis was not universally accepted in 1994, when the project started, and is still contentious. At that time, the World Wide Web was in its

infancy and the “Internet” was not part of the popular lexicon. The incredible growth of the Web that happened in parallel with the Virtual Book project publicized the benefits of electronic documents. The trend towards putting documentation on-line is clear to most information consumers, so they understand the motivation for having a virtual book.

This section explores the ways that electronic media surpass paper. While a virtual book could provide all of these advantages, only a few were actually implemented on the Lectrice prototype. An actual product could have many more of the capabilities described below.

#### ***4.1 Navigating Electronic Documents***

A number of facilities allow readers to navigate quickly through paper documents: examples include the table of contents, outlines, the index, the glossary, and footnotes. Electronic media can faithfully reproduce these facilities, augment them, and provide completely new navigational aids. For example, Adobe’s portable document format (PDF) and their popular Acrobat document reader implement a number of standard navigational features, such as a table of contents view and the ability to jump to a page in the document by entering its page number. Acrobat supplements these with mechanisms that can be supported only in the electronic form; these include the thumbnail view of the pages of the document and hyperlinks within and out of the document.

The table of contents in an electronic document provides a good example of how to augment a navigational facility. On paper, the table of contents simply maps a document’s organizational structures to page numbers, requiring the reader to search physically for a desired page. An electronic table of contents might look exactly the same as the paper version, but it can streamline navigation by performing an automatic page search. On Lectrice, tapping the pen on a page number in the table of contents causes the desired page to appear on the screen. The electronic table of contents can also be displayed in other forms, such as the Lectk thumb indexes described in Section 2.5.1.

Linking numbers to pages is a form of hypertext, a presentation style that is now familiar to everyone who uses the Web. Similar uses of hypertext extend other traditional navigational mechanisms. Consider linking actual pages to category headings in an outline, index page numbers, and bibliographic references. Not only are these extensions useful, they are convenient. On Lectrice, a reader can navigate through documents using just a few strokes of the pen.

Hypertext links are just one of the mechanisms for associating information in electronic documents. In a sense, every word in an electronic document can become active, giving the reader access to relevant information. For example, tapping on a word might indicate a request for the word’s definition, retrieved from an on-line dictionary. For appropriately formatted novels or plays, tapping on a person’s name could reference the *dramatis personae*; a place name might bring up a relevant map. Another pen gesture could request the book to search for a word, giving the reader access to a full concordance for every word in the document. Links and searches also allow electronic media to provide fast and easy access to entire libraries.

The flexibility of an electronic display can also help a reader navigate through a document. A virtual book can simultaneously display multiple views of a document. Possible views include the main body of a document, an outline, bookmarks, marginal notes, and bibliography entries. The thumbnail views in Lectk are a particularly interesting mechanism. They give the reader an overview of the pages in a document and allow quick, random access to the pages.

As discussed in Sections 2.5.1 and 3.2, the challenge to the virtual book designer is providing access to navigation aids without sacrificing the quality of sustained reading. People do not want to spend precious screen real estate on these features when they are simply reading a document. An electronic book might

be able to display many kinds of navigational aids, but it must not clutter the screen when the reader wants to focus on the main body of a document.

On a virtual book, it is sufficient to provide a single button — or pen gesture — so that a reader can indicate the desire to access extended functionality. A similar observation is also true of the pen itself. While navigating, the pen is useful every few seconds. After finding the desired page, a typical reader switches to sustained reading and puts the pen down (or stores it in Lectrice's pen holder). This transition is why the pen was rejected as the exclusive device for turning pages.

For years, the advocates of electronic reading media [Nelson1965] have described the benefits of hypertext, and the dramatic changes that it causes. Instead of reading a narrative linearly, hypertext makes it easy for a reader to follow a self-directed path through a document, or set of documents. Experience with Lectrice suggests that the "hypertext revolution" of a shift to entirely self-directed reading will not happen. Instead people will alternate between self-directed browsing and more traditional, linear reading. It is therefore important that the user interface of a reading appliance smoothly supports both kinds of access pattern.

## 4.2 *Transporting electronic documents*

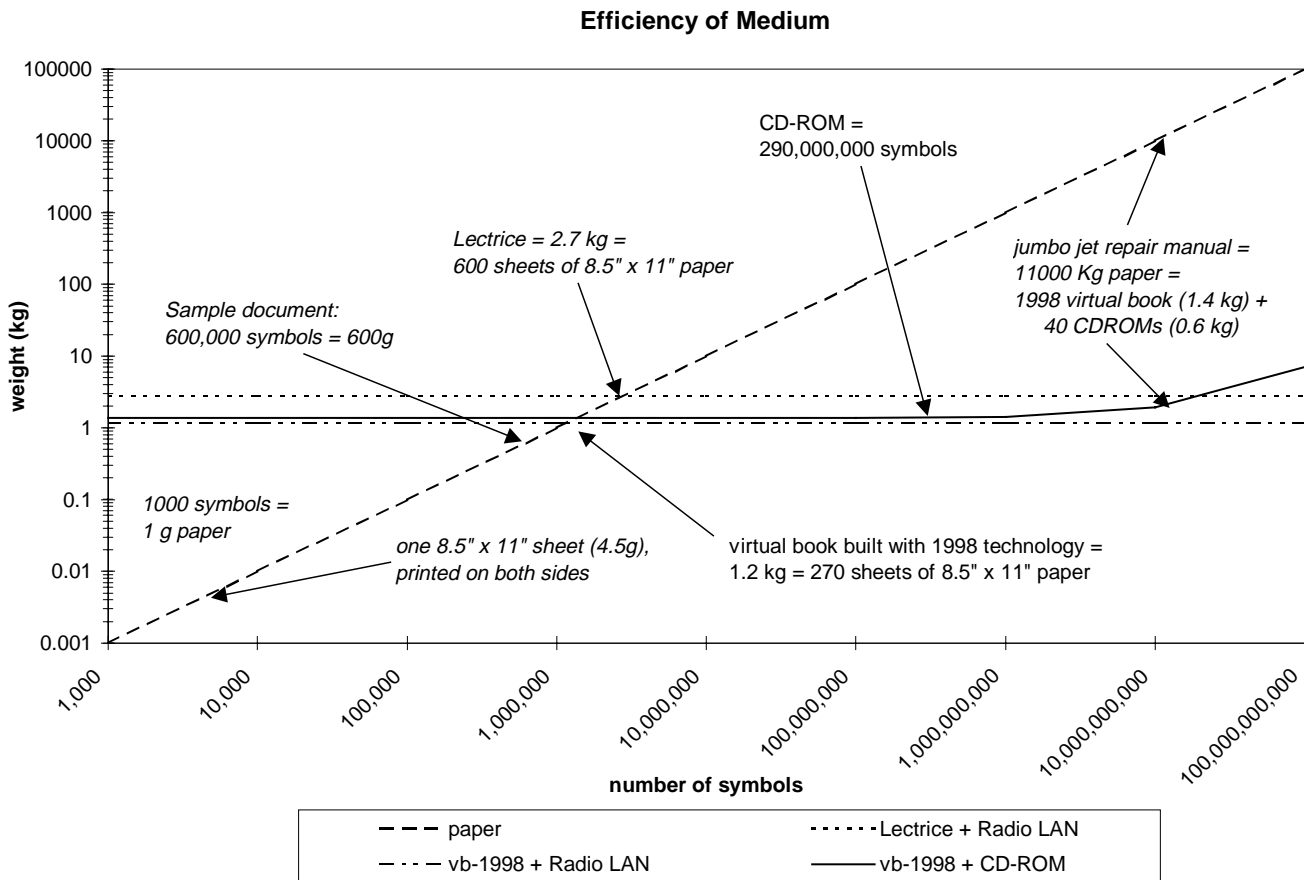
One of the benefits of portable reading appliances is that they allow a reader to conveniently transport many more documents than is possible in hardcopy. One way is to actually carry the documents in (or with) the device. Alternatively the device can be used to access a document library over a network. A combination of these methods is also possible: some documents can be carried, with supplemental ones and updates supplied over a network. This section considers the tradeoffs between transporting hardcopy, carrying electronic documents and relying on network access.

It is instructive to consider the information content versus the weight for the alternative ways to transport documents. This comparison is just one way to measure the efficiency of a medium. The study consists of an informal "back of the envelope" estimate of the number of symbols (an abstraction of letters) and the weight in kilograms for paper and three implementations of a virtual book. To express the information content, a "symbol" is defined to be roughly equivalent to a letter but is expanded to take account of diagrams and pictures. In line with the informality of the calculations, the old adage that "a picture is worth a thousand words" is applied literally to compute the number of symbols in a document.

The document characteristics used in these calculations were derived from a single technical document, which was carefully chosen as representative of the class of documents being analysed. Printed double sided on paper it is 150 sheets and weighs 600g. The document contains 500,000 text symbols (mostly in 10 point Times Roman typeface), aggregated into 90,000 words. The average word (string of connected symbols) is between five and six symbols; therefore, each illustration is considered to contain five thousand symbols. The document has 20 drawings (considering only figures with significantly more information than the text that they contain), equivalent to about 100,000 symbols. So the document has a total of 600,000 symbols. To calculate the weight of the electronic forms of the document it was encoded in Adobe's PDF format, which uses LZW compression to use minimal storage space.

Figure 16 shows the graph of weight versus symbols using the ratios found in the sample document. Both axes of the graph are logarithmic. The horizontal axis ranges from a short memo (one thousand symbols) to the size of a jumbo jet repair manual (more than ten billion symbols). The vertical axis goes from one gram (about one quarter of a standard sheet) to hundreds of tons. Four lines are plotted. The first is for Xerox 4024 DP paper (20 pound, 8.5 by 11 inches, printed on both sides). The second shows Lectrice with the RoamAbout radio LAN interface attached. The other two are based on the estimated weight of a

virtual book built in current technology, one assuming that the device has a CD-ROM drive and sufficient CDs to hold all the symbols, the other assuming a current radio LAN interface.



**Figure 16: Information content against weight.**

Although they are informal, the approximations yield some interesting rules of thumb. First, the diagonal line with large dashes shows that paper (Xerox 4024 DP is standard printer/copier paper) is a very efficient medium for small numbers of symbols. According to the metric, paper can carry about one thousand symbols per gram. At 75 grams per square meter, each sheet weighs about 4.5 grams, and can carry 4500 symbols. It is very hard for any electronic device to compete with a single sheet of paper. Paper is amazing in its flexibility: if one wanted to carry just a single symbol, it would be possible to cut out a 0.001 gram quantity with an ordinary pair of scissors. At the other extreme, a mechanic who wanted to carry a paper version of the jumbo jet repair manual would be unable to carry 11 metric tons of paper even using a forklift.

Although it does not compare with a single sheet of paper, the electronic format becomes competitive in the range of information that people typically carry. The graph suggests that virtual books become lighter than paper somewhere in the 300-600 page range. It is not hard to find people carrying this amount of paper in their briefcase: a couple of technical or business documents and a couple of magazines can easily land right in the middle of this range. With a battery and a radio frequency network card Lectrice

weighs 2.7kg. Users found this configuration heavier than they would like while they were reading, but they did appreciate being able to “carry” a virtually unlimited number of documents for the same weight as four documents similar to the sample one. A virtual book built with commodity technology available in 1998 (vb-1998 + Radio LAN) — represented by the line with dots and dashes — would weigh about 1.2kg. This device would be an acceptable weight during reading. As well as being lighter, the newer virtual book would also be thinner than Lectrice: about 2.5cm compared to 4.5cm. Current technology makes the hypothetical device similar in both weight and thickness to two copies of the sample document. Assuming adequate radio network coverage, both of these devices could provide access to documents as large as the jumbo jet repair manual.

Unfortunately, it is not yet possible to rely on radio networks, especially in harsh environments such as the airport tarmac, hangars or factory floors. A virtual book with a CD-ROM reader, rather than a radio network, might be more suitable. The CD-ROM reader would be slightly thicker and heavier (by about 200g), making this virtual book weigh 1.4kg, assuming that it is built in 1998 technology. On the graph, the solid line represents such a device (vb-1998 + CD-ROM). With just one CD, this virtual book would be able to carry up to almost 300 million symbols, equivalent to 300kg or 70,000 sheets of paper. Over this limit, the need to change disks makes use a bit more cumbersome. The jumbo jet mechanic would need a set of 40 CD-ROMs (weighing 600g) to hold the entire manual. The other advantages of the electronic format, such as the ease of searching this mountain of documentation, should more than make up for this inconvenience. Since the end of the Virtual Book project, DVD-ROM products have become available with similar cost and weight to CD-ROMs, but five times the storage capacity. These would reduce the number of disks required for the jumbo jet manual to 8, weighing just 120g.

### ***4.3 Publishing electronic documents***

Assuming the willingness of readers to accept electronic formats, every step of the publication process (authoring, editing, production, replication, storage, delivery, archiving, and disposal) can be performed without consuming or transporting paper. The front end of the process is already largely electronic. As the weight and cost of virtual books and their media decrease, the relative cost of paper will continue to increase. It will be both cheaper and more convenient to acquire books in their electronic form. The success of Amazon.com [[www.amazon.com](http://www.amazon.com)] has started to demonstrate that readers are willing to buy books online. Several companies have recently announced their intentions to produce electronic books. These trends suggest that virtual books will allow electronic media to replace many paper documents.

While virtual books remove a primary impediment to electronic publication, a number of challenges still remain. The ease of duplicating electronic documents reduces publishing costs by orders of magnitude, but also makes it much easier to violate copyright laws. Ease of duplication, combined with digital media’s resistance to signal degradation, also makes archival easy—for the lifetime of an electronic format. Unfortunately, the rapid improvement in technology makes digital media and formats notoriously short-lived. Any long-term, electronic archival scheme requires a strategy for converting to new formats as they emerge. Given the improvements in physical storage density, upgrading to a new format can often be justified just by decreasing the long-term cost of storage. (How many CD-ROMs would it take to hold all of the data ever stored on punch cards?) On balance, the physical benefits of digital media far outweigh the impediments to electronic publication.

#### **4.4 Beyond Print**

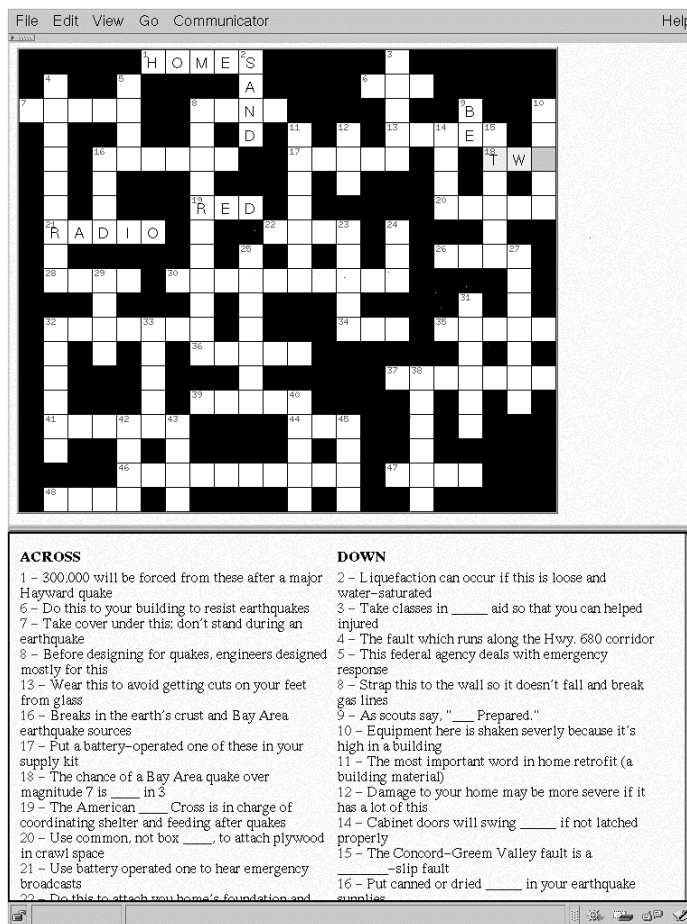
Virtual books—and other portable information appliances—do not merely emulate paper documents, but can extend them by making use of the dynamic nature of the electronic medium. Some of the new capabilities will consist of straightforward combinations of traditional text documents with other media. Others allow the document to adapt to the reader.

A prime benefit of the dynamic nature of the electronic version of documents is that they can be quickly formatted to fit the capabilities of the desired display medium and the reader's preferences. Just as paper-based books range in size from small pocket diaries to large blueprints, virtual books can come in many sizes. Paper documents are designed to match their final size, but this is not necessary in the electronic version. If the document is represented in a markup language (such as SGML or HTML), rather than a page description language (like PostScript or PDF), both layout and rendering can be done to match the display. The same document can be moved between different sized devices and it will continue to be legible.

Dynamic formatting of documents also gives the ability for them to change to suit the needs of the reader. When presented with a Lectrice displaying a document with very a small typeface, one executive asked, "Where is the bifocal option?" The motivation behind his question was clear: virtual books should be able to adapt a document to the needs of the reader. On Lectrice, a simple button press increased the magnification of the document, and satisfied his request. It turns out this dynamic behavior is common. When using the Acrobat reader to browse through a document, people typically work in the standard view with all the toolbars and navigation aids showing. When readers make the transition from browsing to sustained reading, they typically switch to full screen mode (making the appearance very close to Lectk). In a sense, the document changes to meet the immediate needs of the reader

To some extent, the familiarity and dominance of paper hinders dynamic reformatting of electronic documents. Graphic designers still primarily target paper when they design documents and they think in terms of this static medium, carefully planning the arrangement of text and graphics. This traditional approach to document design affects electronic documents in two ways. First, documents originally produced on paper and later scanned are hard to resize. Second, conventional design ideas impact digital document formats. HTML, originally defined as a markup language, now contains a number of constructs (like fixed size frames) that make resulting documents more rigid, compared to documents without such constructs. As electronic media and information appliances proliferate, the effects of paper-oriented design will decrease. In the near term, however, virtual books must credibly render traditional documents, as well as demonstrating the value of documents that can adapt to different media.

Dynamic rendering is only a simple example of the possible interaction between readers and documents. In addition to modifying how a document's text is presented, readers can interact with the content itself. For example, as the pen touches each box in an electronic crossword puzzle, the virtual book can display the associated clue. When used for instruction rather than entertainment (as is the puzzle shown in Figure 17), puzzles might correct students' answers as they are entered.



**Figure 17: Filling out a Web based crossword.**

Active textbooks [GDL1993, BN1996] are a more persuasive example of the benefits of interactive documents. Such documents allow teachers and students to work together, for example viewing and manipulating the state of algorithm animations. Teachers can use active textbooks to demonstrate concepts to students. After class, students can work at their own pace replaying the lecture material, using the animation to understand concepts, and doing related homework. While these methods have been in common use in university computer science departments for years, wide deployment of virtual books will allow these techniques to be used as early as elementary school.

The animation used in active textbooks is a form of multimedia. Audio and video can also be used in presenting an electronic document. A popular demonstration on Lectrice simultaneously plays music and displays the program notes associated with the recording. The use of mixed media on Web pages further supports the view that multimedia provides a powerful way to communicate the message of a document. Web-based catalogs and product brochures make effective use of multimedia: videos, animations, three dimensional models and audio clips are all used to entice customers. Unfortunately, it is easy to misuse the power of electronic documents. The Web has all too many examples of documents that needlessly blink, throb, spin, and talk. The challenge is to use the new mechanisms judiciously, improving the reader's experience, rather than drawing attention to the format itself.

To some extent, paper is already an interactive medium. Readers annotate documents for their own use, for communicating with colleagues, or for the benefit of the original authors. On paper, this task is easy but destructive. For example, many students use highlighting as a study tool, but corrupt their textbooks in the process. Electronic documents can allow markup without damaging the document. A student

could highlight or make marginal notes in a virtual textbook, and then enable or disable the visibility of the annotations, as desired. Even though they were just a simple proof-of-concept, Lectk's PostIt™-style boxes proved useful for recording annotations about documents.

Annotations on electronic documents need not be limited to textual markup. It would be possible for a reviewer to record verbal comments and attach them to a document. An icon on a reviewed page could indicate a related audio—or even video—annotation, and allow the author to play it back. An annotation system intended for collaboration could collect all types of comments from a number of reviewers, and present them all to the author. Authors (or other interested readers) might choose to view a document with no comments, with markups from a single reviewer, or with annotations from multiple editors. In the latter case, the system could differentiate annotations from different reviewers with the use of color or other visual techniques.

While the potential applications are persuasive, much work on annotating electronic documents remains. For example, anchoring annotations is not trivial when documents are dynamically rendered. Consider the following scenario: a reviewer circles a word on an HTML document. A simple annotation system might draw the corresponding digital ink at the coordinates of the word on the document window, oblivious to the underlying content. When the author attempts to look at this annotation, the HTML renderer could use completely different fonts on a different type of display. The coordinates associated with the annotation's digital ink would then be invalid, and placing it usefully would be impossible. This scenario is just one example of the problems facing digital annotation. Experience with Lectrice suggests that solving these problems and producing a credible annotation system would be worth the effort.

## 5 Related Work

The Virtual Book project aimed to integrate a number of technologies: high-quality display, pen input and handwriting recognition, wireless data communication, low-power design, and the Virtual Paper document rendering and browsing software. Several of these technologies were developed by vendors in the laptop computing domain over the preceding five years, and the state of the art in these areas could be observed at any local computer store. Interaction with a number of component vendors helped resolve many of the design decisions for Lectrice.

The concept of a portable information appliance was in the public domain long before the project started. The purpose of this section is not to review the entire history that preceded the Virtual Book Project but to touch on highlights. Greater detail can be found in the literature. For example, Brad Myers recently published "A Brief History of Human-Computer Interaction Technology" with many useful references [Myers1998].

Although many research projects and products predated the Virtual Book, a few visionaries had a particularly strong impact. For the foreseeable future, Vannevar Bush's article *As We May Think* [Bush1945] will continue to spark the imagination of computer designers. In this article, Bush proposed "a future device for individual use:"

"A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.

It consists of a desk, and while it can presumably be operated from a distance, it is primarily the piece of furniture at which he works. On the top are slanting translucent screens, on which material can be projected for convenient reading. There is a keyboard, and sets of buttons and levers. Otherwise it looks like an ordinary desk."



This vision stayed in the realm of science fiction for a number of years. By 1968, futuristic information appliances had changed from desk-sized devices to tablet-size ones. The left-hand picture of Figure 18 shows a scene from *2001: A Space Odyssey* [Kubr1968]. In this classic movie, the astronauts watch video programs and read messages on devices about the same size and shape as Lectrice.<sup>8</sup>



**Figure 18: Virtual books as imagined by others.**

Thirty years after Bush’s article, Alan Kay and Adele Goldberg published their work on the “interim Dynabook” [KG1977]. In effect, they had invented a desk-sized system that could perform many tasks, including much of the functionality of the memex. This paper — published in the mid-1970’s — described the desired form-factor and functionality of future Dynabooks:

“Imagine having your own self-contained knowledge manipulator in a portable package the size and shape of an ordinary notebook. Suppose it had enough power to outrace your senses of sight and hearing, enough capacity to store for later retrieval thousands of page-equivalents of reference materials, poems, letters, recipes, records, simulations, and anything else you would like to remember and change.”

Given the capabilities of components that are now produced in volume, Lectrice may very well be the last “interim Dynabook.” As discussed in Section 4.2, it is now possible to produce a Dynabook-like product with the required size and weight.

One decade after the Dynabook project, Jerry Kaplan started the GO Corporation, which developed the AT&T EO Personal Communicator and the Penpoint Operating System [CS1991]. GO significantly advanced the state-of-the-art in pen-based computing. For example, Penpoint inspired Lectrice’s text entry mechanism, described in Section 2.4.2.3.

More recent research also influenced specific details of Lectrice’s design. Kantarjiev and others at Xerox PARC published their experiences with the X Window System in a wireless environment shortly before the beginning of the Virtual Book project [Kant1993]. James Kempf wrote about integrating handwriting recognition into Unix at about the same time [Kempf1993].

Several contemporary research projects investigated devices similar to virtual books. The Infopad Project at UC Berkeley built a similar client with a slightly different focus [Barr1994]. Infopad assumed permanent network connectivity, resulting in a design

“in which there is no user accessible computation in the portable pad, which not only relieves the portable unit of the requirement to support a general purpose operating systems (e.g. Windows NT,

<sup>8</sup> The small, white logo on the bottom, right corner of each tablet reads “IBM.”

Unix, etc.), but also eliminates the need for expensive and power hungry mass storage devices, costly memory and, implicitly, a high speed general purpose processor sub-system.”

Depending on a network for all functionality can also be a disadvantage. Lectrice’s stand-alone mode, which not possible in the Infopad design, proved to be quite useful, particularly when the device was used away from the office. Currently the limitations of wireless networks make it impossible to get high-speed network access everywhere. Even if these were overcome, there are likely to be restrictions on access – for example it is not unreasonable for a company to deny network access to a visiting competitor.

The Berkeley researchers reached slightly more optimistic conclusions than the Virtual Book project:

“The resistance to reading large amounts of text on a computer screen, instead of paper, is due in large part to the inconvenience of the usual fixed desktop placement of the screen. However, when multimedia data formats are incorporated into a document the use of paper becomes obsolete, in spite of its historical significance and continuing proponents.”

Experience with Lectrice confirms the former statement: a mobile screen eliminates resistance to reading on-line. The latter conclusion is a bit overstated. While information appliances may eventually make paper obsolete in business and academic settings, paper will continue to exist in a variety of forms. It would be hard to imagine a world without engraved invitations, blueprints, or charcoal sketches!

Researchers at the FX Palo Alto Laboratory prototyped a virtual notebook using the Fujitsu Stylistic 1000, a tablet personal computer running Microsoft Windows 95 and Pen Services [SWS1997]. Their prototype, called Dynamite, accepted both handwritten and audio input. The Dynamite system was used to investigate user interfaces for taking notes. A virtual book designed for professionals would certainly benefit from note-taking tools like the ones developed for Dynamite.

Methods for interacting with virtual books will continue to change as handwriting and speech recognition software matures. Experiments with Lectrice used only contemporary input devices: pen gestures and buttons for navigation, handwriting recognition for a single-stroke alphabet, and remote speech recognition for simple command and control. Researchers at the Center for Human-Computer Communication at the Oregon Graduate Institute have also been examining multimodal (combined pen and audio) interfaces [ODK1997], making optimistic assumptions about recognition technology. They have been using tablet devices in user studies that investigate the way that people use speech and writing to interact with computers. Their studies show that people prefer to alternate between pen and speech input methods, rather than using just one or the other. Their conclusion of the importance of multimodal interfaces is supported by the observation that people switch back and forth between using the pen and buttons when browsing the Web using Lectrice.

For a special issue of Wired Magazine in 1995, Lunar Design produced concept designs for tablet-sized information appliances [Lunar1995]. The middle picture in Figure 18 shows one of their whimsical tablets. Knight-Ridder Information Design Lab also prototyped packaging design and interfaces for devices like Lectrice. Although Knight-Ridder did not prototype a complete appliance, the lab did produce a video demonstrating the package design and the use of a simulated tablet newspaper [Martin1995]. The right-hand picture of Figure 18 shows the mock-up used in the video.

Viewed in context, Lectrice is one member of a large family of information appliance prototypes. It was the first such device tuned for sustained reading, and its use showed that on-line reading can be comfortable and convenient. The concluding section of this report discusses why this functionality — combined with technology and interfaces from other research projects — will soon be in the hands of many readers.

## 6 Conclusions

Over the course of the Virtual Book project, Lectrice was used by a number of researchers, served as a demonstration for customer visits, elicited comments at focus groups, and continued to be shown at trade shows through 1997. Lectrice became part of the day-to-day computing environment of the project members, and is still in use in 1998. This experience validated our hypothesis that it is now possible to build a device tuned for reading and browsing. Experience with Lectrice leads to a number of specific conclusions about virtual book technology and more general conclusions about the future of virtual books.

### 6.1 *Displays for virtual books*

Virtual books require well-designed user interfaces, low-power but high-performance hardware, and high density packaging techniques. All of these required elements are available or within the scope of a talented product development team. As predicted at the beginning of the Virtual Book project, the price of a suitable high-resolution display will determine the economic viability of the device. To allow technical and business documents to be read on such a display, users have two requirements: the display must be large enough to show a full page of information, and each character must be crisply formed. These requirements translate into thresholds that must be passed to ensure readability. The display needs to have a diagonal of at least 10 inches and a dot pitch of well above 100 dots per inch. The high-resolution (122dpi) Lectrice display was a little above this threshold. Readability is improved by anti-aliasing the fonts which only requires a few gray levels (a minimum of two in addition to black and white). However, most Web pages demand a color display. Displays with these characteristics have the readability required to comfortably read U.S.-letter (or A4) sized documents, and were used widely in the laptop PC market by the end of the project. Moreover, display manufacturers understand how to make higher resolutions, and future reading appliances can take advantage of their improvements as soon as they become available.

Readers were content to use Lectrice to study long technical documents for sustained periods. However, we discovered that users of notebook computers (and personal digital assistants, or PDAs) had a preconception that reading from an LCD is hard. Once they saw a high-resolution display they changed this opinion, but the preconception may present a barrier to adoption of virtual books.

### 6.2 *Important features of virtual books*

While there is no doubt that the high-resolution display is the most important component of a virtual book, a number of other factors are needed to make them successful. A virtual book should support both existing documents, that are formatted to be printed on paper, and dynamically formatted documents, designed for on-line use. The device must allow access to large quantities of documents, either over a network link or using a built-in storage device.

The packaging of a virtual book is of great importance because the device will be held for long periods. The weight must be less than 3 pounds, and it should be evenly distributed. To allow use in both landscape and portrait orientations the center of mass should be close to the center of the device. If only a single orientation is supported the center of mass should be slightly below the centerline. The case has to feel comfortable; one way to satisfy this requirement is to use a rubberized paint. In addition sharp edges should be avoided, this poses a challenge in the case design since it is hard to make a thin device that has curved edges. For some users the device will contribute to their professional image, and style must be factored in to the design.

Three primary features are needed in an interface to allow reading for long periods: after a very short training period, it must be useable without conscious thought; the display must show only the material being read with nothing to distract the reader's focus; and page rendering must be fast. Secondary elements include using visual cues to indicate page-turning direction, providing navigation aids and allowing annotations on the documents. A general UI observation is that for this type of device pop-ups are preferable to toolbars because screen real estate is not permanently allocated to the pop-up.

Buttons proved to be ideal for controlling the device during continuous reading, when most users did not want to hold the pen. One of the advantages of the device is that it allows the reader to shift their body position, this requires that the buttons be arranged symmetrically to allow easy use with whichever hand is most convenient. Supporting both landscape and portrait orientations requires symmetrical buttons on the other edges. The pen is the ideal control device during browsing. It is excellent for navigating hyperlinks, entering URLs and making control gestures. The preferred way to enter text with the pen is to write directly on the entry window. The pen interface was much improved by showing the 'ink' for a short time.

### 6.3 *The Global Library: the future of virtual books*

There is no better confirmation of the need for virtual books than the emergence of the Web. From the beginning, this research project focused on electronic documents, but at first, there was no dominant encoding for documents containing formatted text, hypertext, and images. By the time the prototype units became ready for sustained use, the most persuasive application was clear: Lectrice makes an excellent Web browser, primarily displaying documents encoded in HTML and PDF.

There is no longer any question that there is a vast, growing body of documents in electronic format. This library has both catalogs (*e.g.* Yahoo) and indexes (*e.g.* AltaVista). A significant percentage of business documents are now published as part of the library. The existence of this global library—and the other benefits of electronic media—will drive the proliferation of virtual books. As information appliances become more common, the percentage of electronic documents intended for consumers will grow: novels, reference books, newspapers, directories, and product brochures will all appear on-line.



**Figure 19: The team took Lectrice everywhere.**

Lectrice has many of the capabilities—rendering high-resolution graphics, playing audio and video, wired and wireless network connectivity—that will allow virtual books to subsume the role of many information appliances. The near-term challenge is to provide the right balance of cost, weight, and functionality. After using Lectrice for several years, it is hard to imagine reading electronic documents in any other way. Five years hence, virtual books — possibly in the guise of portable Web browsers — will be familiar information appliances in many offices and homes. We look forward to purchasing commercially available virtual books in the near future.

## 7 Acknowledgements

The authors carried out the majority of the work on the Virtual Book project. We would like to acknowledge the help of several colleagues in completing the system.

Andrew Birrell and Paul McJones built the Lectern system and document repository as part of the Virtual Paper project.

Chuck Thacker encouraged us to start the project and provided support throughout.

Sharon Perl suggested the name Lectrice

A team from Studio Red, led by Philip Bourgeois, did the industrial and mechanical design and produced the cases for the Lectrice.

Eric Brewer, visiting from the University of California at Berkeley, did the initial port of the Tcl/Tk runtime.

Ari Rapkin, visiting from the University of Virginia, built the framework for handwriting input, ported the CMU single stroke recognizer and did the first implementation of Grafitti™ on Lectrice.

Mike Looney, along with Chris Kryzan and Laurie Girand from Beyond Marketing Strategies, helped us organize and conduct focus groups and intensive interviews.

Christina Hepner-Brodie and Beebe Nelson, from the Center for Quality of Management, guided us through the Concept Engineering process to digest the results of the intensive interviews.

Mark Foster helped us understand what was possible in packaging for future virtual book products.

## 8 References

[AF1993] Thomas Levergood, Andrew C. Payne, James Gettys, G. Winfield Treese and Lawrence C. Stewart “AudioFile: A Network-Transparent System for Distributed Audio Applications” *Proceedings of the USENIX Summer Conference*, June 1993.

<http://gatekeeper.dec.com/pub/DEC/AF/af-usenix93.ps>

[Barr1994] B. Barringer, T. Burd, F. Burghardt, A. Burstein, A. Chandrakasan, R. Doering, S. Narayanaswamy, T. Pering, B. Richards, T. Truman, J. Rabaey, R. Brodersen; “Infopad: A System Design for Portable Multimedia Access”, *Calgary Wireless 94 Conference*, July 1994.

[Birrell95] Birrell, Andrew and Paul McJones. “Virtual Paper”. *Project Web Page, Digital Equipment Corporation Systems Research Center*. 1995-7.

<http://www.research.digital.com/SRC/virtualpaper/home.html>

[BN1996] Marc H. Brown and Marc A. Najork. “Collaborative Active Textbooks: A Web-Based Algorithm Animation System for an Electronic Classroom” *Report #142, Digital Equipment Corporation Systems Research Center*. May 31, 1996

<http://gatekeeper.dec.com/pub/DEC/SRC/research-reports/abstracts/src-rr-142.html>

[Bush1945] Vannevar Bush, “As We May Think” *The Atlantic Monthly*. July, 1945.

<http://www.isg.sfu.ca/~duchier/misc/vbush/>

[CS1991] Robert Carr and Dan Shafer “The Power of Penpoint” *Addison Wesley* 1991  
ISBN 0-201-57763-1

<http://www.utexas.edu/ftp/microlib/eo/html/Bibliography.html>

[GDL1993] Peter Gloor, Scott Dynes, and Irene Lee. “Animated Algorithms: A Hypermedia Learning Environment for Introduction to Algorithms / MacIntosh Format CD-ROM” *MIT Press, Cambridge, Massachusetts*, 1993.

[Kant1993] Christopher Kent Kantarjiev, Alan Demers, Ron Frederick, Robert Krivacic, and Mark Weiser. “Experiences with X in a Wireless Environment” *USENIX Mobile and Location-Independent Computing Symposium*. August, 1993. CSL-93-9 *Xerox Corporation Palo Alto Research Center* September, 1993.

<http://www.usenix.org/publications/library/proceedings/mobile93/kantarjiev.html>

[KG1977] Alan Kay and Adele Goldberg. “Personal dynamic media” *IEEE Computer*. March, 1977.

[Kempf1993] James Kempf “Integrating Handwriting Recognition into Unix” *USENIX Summer 1993 Technical Conference Proceedings*. June, 1993.

<http://www.usenix.org/publications/library/proceedings/cinci93/>

- [Kubr1968] Stanley Kubrick and Arthur C. Clarke "2001: A Space Odyssey" Distributed by: Metro-Goldwyn-Mayer Inc.
- [Li1993] X. Li and N. S. Hall, "Corner detection and shape classification of on-line handprinted Kanji strokes", *Pattern Recognition*, 26/9, 1315-1334 (1993)
- [Li1995] X. Li and D. Y. Yeung, "On-line handwritten alphanumeric character recognition using feature sequences", *Proceedings of International Computer Science Conference*, 197-204 (1995)
- [Li1997] X. Li and D. Y. Yeung, "On-line handwritten alphanumeric character recognition using dominant points in strokes", *Pattern Recognition*, 30/1, 31-44 (1997)
- [Lunar1995] "A Day in the Life: Monday, 19 October 2020" in Scenarios: 1.01. *A special edition of Wired Magazine*. Industrial Design and Alias work by Lunar Design. Photographs by James Porto Pages 92-99 October, 1995.
- [Martin1995] Teresa Martin, "The Tablet Newspaper: A Vision for the Future" *In CHI 95 Proceedings Videos*.  
[http://www.acm.org/turing/sigs/sigchi/chi95/Electronic/documnts/videos/tam\\_bdy.htm](http://www.acm.org/turing/sigs/sigchi/chi95/Electronic/documnts/videos/tam_bdy.htm)
- [Myers1998] Brad Myers, "A Brief History of Human-Computer Interaction Technology" *ACM interactions*. Vol. 5, no. 2, March, 1998.  
<http://www.cs.cmu.edu/~amulet/papers/uihistory.tr.html>
- [Nelson1965] Theodor H. Nelson, "The Hypertext." *Proceedings International Documentation Federation Annual Conference*, 1965.
- [ODK1997] Sharon Oviatt, Antonella DeAngeli, and Karen Kuhn "Integration and synchronization of input modes during multimodal human-computer interaction." *In Proceedings of Conference on Human Factors in Computing Systems: CHI '97*. New York. ACM Press.
- [Oust1991] John Ousterhout, "An X11 toolkit based on the Tcl language" *In Proceedings of the USENIX Winter Conference* January 1991.  
<ftp://ftp.scriptics.com/pub/tcl/doc/tkUsenix91.ps>
- [Palm1994] Palm Computing "Palm Computing Delivers Major Advance in Handwriting Recognition Technology." *Press Release Los Altos, Calif.* -- September 19, 1994.  
<http://www.grot.com/zoomer/zoomer-list/zoomer-list-3Q1994/0048.html>
- [Rubine1991] D. Rubine, "Specifying Gestures by Example" *Computer Graphics*, 1991. 25(4): p. 329-337.
- [SG1992] Robert Scheifler and James Gettys. X Window System (Third Edition). *Digital Press, Burlington, MA*, 1992.
- [SWS1997] Bill Schilit, Lynn Wilcox, and Nitin "Nick" Sawhney "Dynamite: A Dynamically Organized Ink and Audio Notebook." *In CHI 97 Conference Proceedings*, ACM Press, 1997, pp. 186-193.  
<http://www.fxpal.xerox.com/papers/sch97.pdf>