Department of Computer Science

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Hamilton, New Zealand

Realistic electronic books

by

Veronica Liesaputra

This thesis is submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy in Computer Science
at The University of Waikato

October 2010

Abstract

People like books. They are convenient and can be accessed easily and enjoyably. In contrast, many view the experience of accessing and exploring electronic documents as dull, cumbersome and disorientating. This thesis claims that modelling digital documents as physical books can significantly improve reading performance. To investigate this claim, a realistic electronic book model was developed and evaluated.

In this model, a range of properties associated with physical books—analogue page turning, bookmarks and annotations—are emulated. Advantage is also taken of the digital environment by supporting hyperlinks, multimedia, full-text search over terms and synonyms, automatically cross referencing documents with an online encyclopaedia, and producing a back-of-the-book index. The main technical challenge of simulating physical books is finding a suitable technique for page turning that is sufficiently realistic, yet lightweight, responsive, scalable and accessible. Several techniques were surveyed, implemented and evaluated. The chosen technique allows realistic books to be presented in the Adobe Flash Player, the most widely used browser plug-in on the Web.

A series of usability studies were conducted to compare reading performance while performing various tasks with HTML, PDF, physical books, and simulated books. They revealed that participants not only preferred the new interface, but completed the tasks more efficiently, without any loss in accuracy.

Acknowledgements

This thesis would not have been possible without the help of so many people.

I especially would like to thank my chief supervisor, Ian Witten, for his continuous support during my doctoral study. He was always accessible and willing to discuss my research. He taught me how to question thoughts and express ideas clearly and simply. As a result, my research life has been smooth and rewarding for me.

I am also grateful to my second supervisor, David Bainbridge, for his encouragement and guidance at different stages of my research. Thank you for reading the final draft of the thesis and making so many helpful comments and corrections.

A huge thanks to those who have read countless revisions of this manuscript and offered advice: David Milne, Michael Walmsley, Anthony Blake, Kathleen Paris, Gay Main, Anna Huang, Shaoqun Wu, Rob Acksyn, and my sister Maggie Liesaputra. I have been very lucky to do my PhD with Ian's other students. Thank you for listening to my progress and practice presentations.

I would like to acknowledge the support provided by all my colleagues at the Digital Library Laboratory and European Media Lab, especially to Dave Nichols whose help and valuable suggestions have been an important contribution to this study.

This research is funded in part by Google. I would like to thank them for the generous scholarship. This thesis would not have been possible without their financial support.

Last but not least, I am greatly indebted to my family. My sister has consistently encouraged me, listened to my complaints, and gracefully accepted the many times I have said that I need to work on my thesis today. To my parents, thank you for educating me and giving me unconditional support to pursue my interests. I dedicate this thesis to my family, without whose support it could never have been attempted, let alone finished.

Contents

List of Figures

List of Tables

1

Introduction

The electronic book industry is burgeoning. Libraries and other content providers are digitising documents to enable people around the world to access them online. For example, the Million Book project has digitised more than a million books. The British Library has digitised more than 100,000. The Open Content Alliance and Google are creating online collections of tens of millions of volumes accessible through the Internet Archive and Google Books, respectively. These projects enable full-text retrieval over vast libraries.

Manufacturers are competing to produce state-of-the-art reading devices that are light and mobile, with high resolution displays and features. Barnes and Noble's Nook, Amazon's Kindle, and the Sony Reader are typical examples. Recent technological advances have improved the quality and readability of the screens (Adler et al., 1998; Marshall and Ruotolo, 2002). Touch-screen displays provide users with more flexible interaction possibilities. The International Digital Publishing Forum (2010) reports that electronic book sales revenue tripled from in 2009 to 2010. Current electronic document representations can offer many facilities that are denied to their paper counterparts, and text can be delivered and updated far more quickly and cheaply.

With these developments, users are increasingly willing to read on screen. However, people still consume far more paper than they used to (Crawford, 1998; Gorman and Crawford, 1995; Nicholas et al., 2008). They prefer to print the document when they want to read it intensively. According to McKnight et al. (1991)

and Rouet et al. (1996), this problem stems from the complexity of electronic book applications, which place more emphasis on the invention and implementation of tools than on the user's ability to interact with them. Over nearly three decades, researchers have modelled various document representations, most of which were claimed to have glowing potential, but there are no studies that compare user behaviour and performance of the new representations with conventional ones. This thesis aims to combine the advantages of physical and electronic documents to enhance on-screen reading experience and performance, and compare it with existing representations and the original physical form itself—the humble book.

1.1 Challenges of reading on screen

According to survey responses collected by Adler et al. (1998), Barnard (1999), Davidson et al. (1997), Denoue and Vignollet (2000), Fidler (1998), Kilgour (1998), Marshall et al. (1999), O'Hara and Sellen (1997), Roush and Schaul (2000), Sellen and Harper (1997), and Wearden (1998), the ideal electronic reading device should be affordable, portable, durable, easy to use, readable, comfortable, have large storage facilities and a long battery life, provide stylus-based or hand-gesture input, and be connected to the Internet. Technological advances in recent years have meant that digital display media, such as the Kindle, Sony Book Reader and iPad, are getting closer to support all of those features mentioned. But, the underlying document representation and interaction have not evolved. The authors of the surveys have invariably recommended that researchers focus on finding the best format to represent book digitally.

The survey results also highlight that participants want an electronic book application that is quick and easy to open, displays both text and multimedia, and allows a user to easily navigate, search, annotate, bookmark and share multiple documents. However, to simplify the problem, the application need not necessarily support sharing, handling of multiple documents, correct for any hardware defects, or be displayed in a specialised reading device. This section summarises the two challenges of digital reading considered in this thesis.

Accessibility

When reading an electronic text, people usually have to wait for the document viewer to be launched and the text to be downloaded. Coupled with network delays, the overall time required to access a lengthy article can be distractingly long (McKnight and Dearnley, 2003). For example, Malama et al.'s (2005) and Nicholas et al.'s (2008) studies reported that participants dislike Adobe Reader because of the slow start-up delay.

Not every user has their own computer. They must share with others, and cannot install applications or store things on the shared machine. They have to save articles on a removable storage device, or re-download them for each reading. Although an electronic book application may be free of charge, it has to compete with other applications for disk space—often, only the most common applications are installed. Inexperienced computer users also tend to shy away from downloading and installing new programs, and people may not be able to access documents that require a special viewer, such as Adobe Reader.

Document representation and interaction

Books have been supporting people's reading activities since 100 A.D. Their physical appearance provides a frame of reference for navigation, because relative positions in the text, such as *before* and *after*, have tangible physical associations. Even before opening a book, readers can likely guess its logical structure, subject matter, size, age and usage. The physical nature of paper allows effortless, informal and unselfconscious interaction (Dillon, 1992; O'Hara and Sellen, 1997). For example, readers can write in the margin as they read, fold the page to compare non-adjacent pages or to bookmark it, riffle through pages while browsing, and move the book around to improve legibility, adjust perspective or speed up handwriting. This interactivity supports critical thinking and text comprehension. Because the human nervous system correlates hand movement with the eye, it is much easier to read something that is held in the hand than something that just lies on the table (Strassmann, 1985).

These conventions and skills are transferable between most physical document types, but consistent conventions for online reading have not yet emerged (McEneaney, 2000). There is a wide range of interfaces for interacting with electronic text. Mastering all the functions in one application does not guarantee that users can employ the same techniques in others (Niederhauser et al., 2000). People must first learn how to operate the system before they can become comfortable and immersed in the reading experience. Although a portable reading device with a touch-screen display can create a richer and more tactile experience, digital interactions are still interruptive and deflect users from their reading tasks (Adler et al., 1998; Marshall and Bly, 2005; O'Hara and Sellen, 1997). Furthermore, the lack of physical aids and standards in the text structures cause users to frequently become lost in the document (Conklin, 1987; Edwards and Hardman, 1989; McDonald and Stevenson, 1998). They do not know where they are or where to go next. Compounded by factors such as reading ability and prior knowledge, a given electronic environment may increase a user's cognitive reading load and reduce his or her reading performance (Gordon et al., 1988; Marchionini and Shneiderman, 1988; McKnight et al., 1989; Niederhauser et al., 2000).

1.2 Motivation

An application has a good interface if it builds on its users' mental model of the task and speaks their verbal and visual languages (Nielsen, 1998b; Pakdel, 2006). A system becomes more intuitive to learn and easy to use by utilising real-world concepts and techniques that readers already understand. Moreover, people are very good at managing and memorising objects spatially. For example, it does not take long for someone to find a bottle of milk when they go to a supermarket that they visit frequently. These arguments suggest that simulating electronic documents as books could enable users to transfer their experience with physical books, reducing their cognitive load when navigating and acquiring information from documents. It also provides a familiar frame of reference for new users.

Other researchers reject the book metaphor in electronic reading (McKnight et al., 1991; Nielsen, 1998a; Rowland et al., 1997; Shneiderman, 1998). According to them, electronic documents should be designed to take advantage of their digital form by providing capabilities that elude their paper counterparts. The book metaphor limits the potential of electronic books and leads to poor design. Although it is a good idea to emulate a document's logical structure as it exists on paper, there seems little need to retain detailed aspects, such as mimicking page turning or displaying the stack of pages underneath either opened leaf.

There is enough evidence, both anecdotal and from formal user studies, to suggest that the usual HTML or PDF presentation of documents is not always convenient for the reader, regardless of the reading devices (Kearsley, 1988; Malama et al., 2005; O'Hara and Sellen, 1997; Parsons, 2001). On the other hand, it is quite clear that while many book models have been prototyped and demonstrated, none are in routine use in today's digital libraries. Given the remarkable historical success of the book form, and the fact that the superiority of HTML or PDF presentation is at least questionable, why are physical book models not more widely available in digital libraries? This thesis asked whether this is purely technological or there is any proven usability issues.

1.3 Thesis statement

This thesis claims that

> *Modelling digital documents as physical books can significantly improve reading performance over conventional document representations.*

A virtual book simulation involves modelling the act of page turning, the appearance of the turned page, the stack of pages on either side of the book, and the book's spine at different animation frames. This research addresses the question of whether such a simulations is worthwhile in practice by comparing the performance of the simulated book with conventional electronic formats (HTML and PDF) and with physical books. Each format is evaluated in terms of efficiency, effectiveness and reading experience. Section 1.4 further describes how the claim of this thesis is investigated. Readers are encouraged to experience the ideas developed in this research by reading the thesis in the simulated book format at http://www.nzdl.org/Books/Thesis/index.html.

This research does not aim to replace printed books. Paper offers significant advantages over digital technologies for certain activities, and people will continuously use both electronic and physical documents according to their tasks (Sellen and Harper, 2001). However, to successfully diminish our reliance on paper, the screen-based reading experience and performance needs to be improved.

Based on the two reading challenges presented in Section 1.1, this thesis investigates the best document representation and interaction for a user to read a single document as a book on a standard desktop or laptop with minimal time for downloading, installing and starting up the application. This project has deliberately excluded any hardware issues related to special reading devices, such as iPad, Kindle or Sony Book Reader. It is assumed that the subject's reading experience and performance will be further enhanced if the proposed document application were augmented with missing functionalities, such as the ability to share and collaboratively interact with multiple documents, or displayed on an ideal digital reading device.

1.4 Research questions and methodologies

To investigate whether a reader can access and use information in a document better when the book metaphor is utilised, this project posed the following research questions:

Q1) Does reading experience and performance significantly improve when documents are simulated as books?

There are two main perspectives on electronic book research: system-centred and user-centred (Shneiderman, 1998). The former focusses on technical challenges. System-centred researchers rarely conduct field studies, or even laboratory studies, to check the usability of their proposed techniques. This is the dominant approach in electronic book research (Benest, 1990; British Library, 2006; Card et al., 1996, 2006; Internet Archive, 2007; Janssen, 2005; Mangiaracina and Maioli, 1995; Narayanan and Hegarty, 1998; Rouet et al., 1996). The latter is concerned with the appropriateness of the tools for end users, the skills required to use the proposed system, its effect on people's behaviour, the problems users are likely to experience, and readers' responses. User-centred researcher takes into account users' needs, intentions and environment. They rarely focus on the technical features of the system (Chu et al., 2004; Crestani and Ntioudis, 2001; Landoni et al., 2000; Marshall et al., 1999).

It is important to design applications that are simple, and easy to use, learn and understand (Shneiderman, 1998). Thus, this project followed the user-centred approach. To investigate the utility of simulating realistic books in an digital environment, participants were recruited and asked to perform specific reading tasks, such as browsing, searching, annotating and bookmarking, with HTML, PDF, physical book and the simulated book model. Their performance and experience with each format was analysed and compared.

If the simulated book significantly improves a user's information gathering ability and reading experience compared with conventional representations, the hypothesis of this thesis holds. The performance of each document representation was evaluated according to three criteria: *efficiency*, the amount of time it takes to complete a tasks; *effectiveness*, the number of tasks completed successfully; and *user experience*, the reader's subjective opinion of the system's pleasantness, sense of engagement, ease of use, ease of learning, simplicity, intuitiveness, and overall usefulness.

Q2) Which features of paper and electronic books are essential to be incorporated into simulated books?

When reading a physical book, people make use of various cues such as the table of contents, back-of-the-book index, section headers, page numbers, text layout, stack of page edges and page size to help them locate information in the book and understand the size, structure and content of the text (Severinson et al., 1996). Their reading performance and behaviour may also be influenced by the way they interact with the document. Annotation and navigation on paper was relatively effortless and smoothly integrated with reading.

With many electronic document applications, various cues and interactions in paper books have been substituted. For instance, a scroll bar is shown instead of the stack of page edges, a back-of-the-book index is replaced with a full text search function, a glossary is substituted with an electronic dictionary, a bookmark is only displayed in a bookmark panel and not on the page, and buttons and scroll bar sliders, rather than page turning, are used to navigate. Digital environments also provide great added value over its printed form, such as searching, hyperlinking, editing, magnifying, automatic key phrase detection and multimedia.

To identify which aspects of physical and electronic documents are essential and which features of paper books are more easily emulated digitally, the reading behaviours of participants during the usability studies mentioned in Q1 were observed. This project then analysed how people interact with each document format, their reasons for choosing one format over another, the obstacles they encountered, and their strategies for overcoming them.

Q3) Is it possible for simulated books to be routinely used in practice?

When simulating a realistic book, there is a trade-off between accuracy, interactivity and scalability. The more accurate the model, the more system resources will be required and the longer it will take to compute. As described in Section 1.1, a document's accessibility and interactivity affect people's willingness to utilise an electronic book application. For a book model to be deployed in practice, its page turning simulation should: look sufficiently natural that no noticeable visual artefacts are produced; be responsive in real time; be able to handle books with many pages; allow additional features such as searching and annotation to be incorporated; and not require complex or rarely installed applications for display.

If there is a model that satisfies all of those criteria and the use of the book metaphor significantly enhance people's reading experience and performance, it will be possible for a simulated book to be widely used in practice. To find such model, this thesis surveyed, implemented and compared the performance of various page turning techniques. Although it would be quite feasible to extend each of these techniques to deal with other types of book binding, for practical reasons, this project only simulates documents as hard-bound or paperback books.

1.5 Contributions

This thesis makes the following contributions:

1. A comprehensive survey of computer graphic techniques for simulating the

act of page turning. Each technique is discussed, implemented and evaluated in terms of computation time, space complexity, and visual fidelity.

2. Two open-source applications that are usable within today's Web browsers:

    - *Realistic Books* is an interactive paginated electronic document representation that combines the advantages of online documents—full text search, hyperlinks, animation and multimedia—with the look and feel of printed books, providing page turning, random-access navigation bookmarks, highlighting and annotation.
    - *Bookmaker's Workbench* is an interactive system intended to facilitate the process of making Realistic Books. It incorporates many features, including a text mining option that automatically identifies significant key terms, links them to their corresponding Wikipedia articles and display background information about the terms from the article. It has the ability to include synonyms automatically into the full-text search capability provided, and generates a table of contents and a back-of-the-book index automatically.

3. A series of usability studies that:

    - compares participants' browsing and searching performance and behaviour in HTML, PDF, Realistic Books and physical documents
    - investigates participants' personalising experience and behaviour in PDF, Realistic Books and printed articles
    - evaluates the usability of the Bookmaker's Workbench.

The following papers have been published during this investigation:

- A paper that briefly describes the overview design and evaluation methods for this research (Sections 1.2 and 1.4):
    - Liesaputra, V. (2007) Realistic Books project: To develop and evaluate a three-dimensional representation of a book, an encyclopaedia and a personal digital library. *JCDL'07 Doctoral Consortium*, Vancouver, Canada.

- Publications that discuss in detail various page turning models and compare their performances (Chapter 3):
  - Liesaputra, V. (2007) Turning the page of an electronic book. In *Proceedings of the New Zealand Computer Science Research Student Conference*, Hamilton, New Zealand.
  - Liesaputra, V. and Witten, I.H. (2009) Computer graphics techniques for modelling page turning. *International Journal on Digital Libraries*, *10*(2), pp. 93–121.
- The following articles describe how users can navigate within Realistic Books (Section 4.2) and produce a Realistic Books (Section 5.2). They also report participants' browsing performance with HTML, PDF, Realistic Books and physical documents (Section 6.3.4):
  - Liesaputra, V., Witten, I.H. and Bainbridge, D. (2007) Lightweight realistic books: The Greenstone connection. In *Proceedings of the Conference on Digital Libraries JCDL'07*, p. 502, Vancouver, Canada. *Best demonstration award*
  - Liesaputra, V. and Witten, I.H. (2008) Seeking information in realistic books: A user study. In *Proceedings of the Conference on Digital Libraries JCDL'08*, pp. 29–38, Pittsburgh, USA.
  - Liesaputra, V. (2009) Creating and evaluating realistic books. In *Proceedings of the New Zealand Computer Science Research Student Conference*, Auckland, New Zealand.
  - Liesaputra, V., Witten I.H. and Bainbridge, D. (2009) Creating and reading realistic electronic books.*IEEE Computer*, *42*(2), pp. 72–81.
- Papers that analyse people's behaviour and performance when they are searching for information in HTML, PDF, Realistic Books and printed materials (Section 6.4.2):
  - Liesaputra, V., Witten I.H. and Bainbridge, D. (2009) Searching in a book. In *Proceedings of the Conference on Digital Libraries ECDL'09*, pp. 442–446, Corfu, Greece.

– Liesaputra, V. (2010) Finding information in a book. In *Proceedings of the New Zealand Computer Science Research Student Conference*, Wellington, New Zealand.

1.6 Thesis structure

This project is designed to investigate the research questions presented in Section 1.4. The remainder of this thesis is structured as follows.

To make reading on the screen successful, researchers must first understand the affordances of paper books and how people perform the physical act of reading (Marshall and Bly, 2005). Chapter 2 discusses the historical development of document systems. Document systems have evolved considerably in terms of physical form, content, text structure and layout, and writing style. This chapter first reviews how this change affects people's reading strategies and behaviour. Next, it summarises various research findings on usability issues of navigation and personalisation in digital environments. The chapter concludes with some aspects of both printed and electronic documents that could enhance people's reading performance.

While not necessarily the most efficient action, page turning is an ingrained gesture that is not easily reclaimed in an online environment. With it, document contents are displayed more immediately and continuously, and readers may gain a better understanding of the material (Just and Carpenter, 1980). Chapter 3 examines in detail various techniques to simulate the act of page turning. Each has been re-implemented using the Java programming language. They are then compared with each other in terms of the time and memory required to simulate a page turn, and the visual appearance of the turned page. This chapter highlights the possibility of simulating a sufficiently realistic and interactive page turning mechanism that requires only a small system overhead and is usable within today's Web browsers.

Inability to easily navigate and annotate electronic documents is one of the main reasons why people prefer to print their documents. They are not smoothly integrated into a user's reading activities. Often, readers must divert their attention from reading the document content to moving the mouse pointer to click on a button, and then back again. The repeated need to refocus interferes with the reading process (Murata, 1996). Taking this into account, Chapter 4 describes how a document is presented as a Realistic Books, and the functionalities that are supported. The user's interaction is designed so that readers rarely have to move their focus away from the book and click a button to complete a task.

Even if users could easily understand and interact with an electronic document that utilises the book metaphor, if that was the only feature offered, it would be unlikely to succeed (McKnight et al., 1991). If there were no advantage over paper documents, people would not read electronic books. Chapter 5 explains how documents are converted into Realistic Books in the Bookmaker's Workbench. To enhance people's reading performance, the Workbench uses Wikipedia as a comprehensive knowledge base to automatically mark key phrases in the text, provide on-demand definitions and search related pages in the book. Bookmarks can be automatically added to pages containing the start of a section, an annotation or an image, offering immediate access to particular portions of the text.

Because there is inevitably a cognitive overhead associated with any new format, it is important to consider the appropriateness of this format and the method used to convert documents to it (Barker et al., 1994). Chapter 6 presents the criteria, procedures and results of the user studies conducted in this research. These studies were designed to investigate the utility of the book metaphor in the electronic environment, user perceptions of the features supported by Realistic Books and the Bookmaker's Workbench, and the effect the metaphor has on people's reading behaviour. This chapter also reports the time required by the application to respond to user actions for various book sizes at different screen resolutions.

Conceptually, electronic books are an attempt to overcome the limitations of paper books, and incorporate a series of added value features. However, in reality, many paper affordances and interactions are lost in the digital setting (O'Hara and Sellen, 1997). Chapter 7 provides a concluding discussion of the lessons learned from the usability evaluation of the proposed book models versus conventional document representations, and analyses how successfully the hypothesis of this thesis has been addressed. This chapter also proposes possible improvements and directions for future work.

2

Books and Reading

Reading is the ability to make sense of written symbols. People use the symbols to recover information from their memory and construct an interpretation of the writer's message (Mitchell, 1982). Despite its quotidian familiarity, reading includes a range of activities, is done for various purposes, and is embedded within many other document-based activities. People read for different reasons and in different ways: locating information, checking facts, acquiring new knowledge, analysing text, recreation. Studying a textbook is not like reading a science fiction novel.

Adler and van Doren (1972) and Schilit et al. (1999) characterise reading along two dimensions: the reader's engagement with a text, and the number of documents involved, as shown in Table 2.1. Reading a single document involves within-document navigation, while reading multiple documents also involves filing, sorting and inter-document navigation. Multiple-text reading activities are outside the scope of this research.

Reader engagement with a text varies from passive to active (Adler and van Doren, 1972). Passive reading is often associated with reading for leisure or entertainment, as when reading novels, poem and e-mails. It requires less thought and effort than active reading. Active reading involves not only reading the document, but also critical thinking, learning and decision making. For example, while a student is studying a textbook, he or she might mark some pages of the book, and add their own commentaries to the text.

Like any other human activity, reading has a long history. Up until the 9^{th} Century, it was usually performed aloud, in groups or individually, intensively from beginning to end (Cipolla, 1969)—and was mostly passive. Since then, it has become a silent and solitary activity, and documents are studied critically and extensively. This shift in reading behaviour is what motivated changes in conventions for text format, page layout and book design into the form that modern readers regard as universal and immutable.

To be able to create an electronic document application that can enhance users' reading performance, it is important to investigate the best document appearance and services for supporting a user's reading activities. A good place to start addressing this research question is by examining the historical development of how people read, described in Section 2.1, and its effect on the development of the paper book's shape, physical properties and text layout, explained in Section 2.2. Section 2.3 reviews how document are presented in electronic form. Section 2.4 compares features of electronic document representations with ones of paper books. The objective of this review is to deduce design principles for the electronic document application developed for this thesis, both from traditional book design and from electronic book studies and applications.

Reader's engagement / Number of documents	**Passive reading**	**Active reading**
Single text	Enjoying a novel, reading a poem	Studying a textbook, reviewing a proposal
Multiple text	Surfing the Web, reading e-mails	Researching a problem, surveying a topic

Table 2.1: Categories for reading activities (Adler and van Doren, 1972; Schilit et al., 1999)

2.1 The act of reading

Writing was invented for administrative records and monumental displays, like remembering how many cattle were being transported to a certain place, the names of the buyer and seller, or rituals to worship gods (Coulmas, 1991). Ancient people did not utilise writing for creative expression. Books were seen as the medium for preserving verbatim sacred knowledge or the teachings of venerated personalities without human oral mediation. They could be passed to other people who lived in different areas, or to descendants.

People who could write rose quickly through the ranks of society (Michalowski, 1993; Roux, 1992). Others, who could not write, would dictate the text they wanted to write to a scribe. Similarly, only a few people could read; thus, public readings were common. People would gather around and listen to the voice of a scribe reading a manuscript out loud. Because religion was the chief motivator of literacy, priest scribes were society's first readers and writers, followed by elite scholars and lay celebrants.

Up until the 5^{th} Century B.C., reading was essentially a passive activity (Harris, 1989). Scribes wrote down text that was dictated to them and read it aloud on sacred occasions. By the end of 400 B.C., reading was no longer a simple memory aid, but a medium for conveying, interpreting and creating information. Scribes would sometimes incorporate outstanding events or add their own comments about a text. These commentaries would then become a scripture when the scribes gained popularity. Actual literary texts also began to appear. Although readership was still small and socially restricted, reading circles started to appear in Rome at the beginning of 200 B.C. In which, authors would exchange poems (Lichtheim, 1973).

As the book's physical form and text arrangement became clear, simplified, easy to read and more appealing to readers, as explained in Sections 2.2.1 and 2.2.2, in 900 A.D. reading started to change from a public act to a silent, private one (Cipolla, 1969). Silent reading introduced an uncensored relationship between readers and the text. People could read in confidence without any interruptions for questions or comments from other people. They could inspect a concept directly at leisure, cross reference and compare it with other books, and critically evaluate it. By the 12th Century A.D., teachers asked their students to sieve through the content of each document they read (Goldschmidt, 1943; Martin, 1994). They were told to investigate the limitations and definitions of every concept mentioned in the book. People were soon discussing their interpretations with their colleagues rather than passively accepting the interpretations of scribes. The book was seen as something objective, not just the preserved voice of someone speaking.

Gutenberg's invention of the moveable-type printing press transformed the way in which information was used and distributed. In the early 14th Century A.D., manuscripts were revered as treasures. There was often only one copy, which was expensive, and accessible solely by aristocrats, bishops and patricians. Each passive reader heard not only the content of the text but was told how to interpret it according to the prescribed conventions. With the printing press, publishers could produce copious amounts of these texts faster and cheaper than scribes, bringing books within reach of commoners. In 1450 A.D., only one printing press was operating in all of Europe. By 1500 A.D., 1700 presses and 250 printing centres had published 27,000 known titles in more than 10 million copies (Reinitzer, 1983).

More affordable books meant that people did not treat them as holy objects any more. Readers started to annotate documents and dog-ear pages to signal where they had left off reading. Scholars were expected to add their knowledge to the documents they read. The act of passive listening and reading was diminishing and changing to active silent reading (Cipolla, 1969). Readers began assessing and interpreting their reading materials according to their personal criteria. While reading, they underlined, highlighted, scribbled, and inserted comments and bookmarks in the text to help them understand, memorise, summarise and organise the material for later review and retrieval. External tools like dictionaries and encyclopaedias were often used to comprehend unfamiliar concepts or terms. Readers might read multiple articles to build a greater understanding of a problem. Author were considered only as guides who showed invisible audiences a variety of reading paths.

Previously, because there were only few manuscripts available, readers had time to read documents intensively from beginning to end, re-reading each section multiple times in purposeful contemplation. Starting from the end of the 16^{th} Century A.D., numerous books became available cheaply, and people could afford to purchase several books. Because there were too many potentially useful documents and not enough time to read them all, behaviour shifted from intensive to extensive reading (Birkerts, 1994; Darnton, 1989; Levy, 1997). Instead of focusing on the content of a single document, readers would access a variety of books that cover a given topic. They would scan and browse each document in search of relevant information to determine whether it was worth reading in its entirety. Section 2.2.2 describes how each change in people's reading behaviour resulted in a change in the way in which authors presented their ideas.

2.2 History of printed books

Are books adequately represented by their content, independent of their form? For many people, physical appearances cannot be neglected. Depending on the genre, time, and place, readers have come to expect certain books to look a certain way. For example, novels are often paperback and written in a linear structure; textbooks usually have hard covers and are written in a hierarchical structure. The book form conveys abundant information using well established conventions that everyone is accustomed to and can interpret intuitively. This section describes the evolution of the physical and textual characteristics of the book that are now universally recognised.

2.2.1 Evolution of the book form

The conventions for layout and design of a book are the legacy of a long period of evolution. Their familiarity renders a book's graphical elements and functionality practically invisible. The book form or "codex," the form of books we use everyday, was a technological innovation that significantly accelerated the transfer of knowledge.

The first material used for portable writing was the clay tablet of the Sumerians in 3500 B.C., shown in Figure 2.1(a). Because they were generally intended for administrative records, such as to help farmers remember the number of goats in a field, tablets were usually either square or oblong, about 3 inches across, and designed to fit comfortably in the reader's palm. A document may comprise several tablets, perhaps kept in a leather pouch or wooden box in a predetermined order for sequential retrieval (Diringer, 1953). For documents that were to be erected and consulted as works of reference, the text was written on much larger surfaces. For example, the Middle Assyrian Code of Laws from 1200 B.C were 67 square feet, written in columns on both sides.

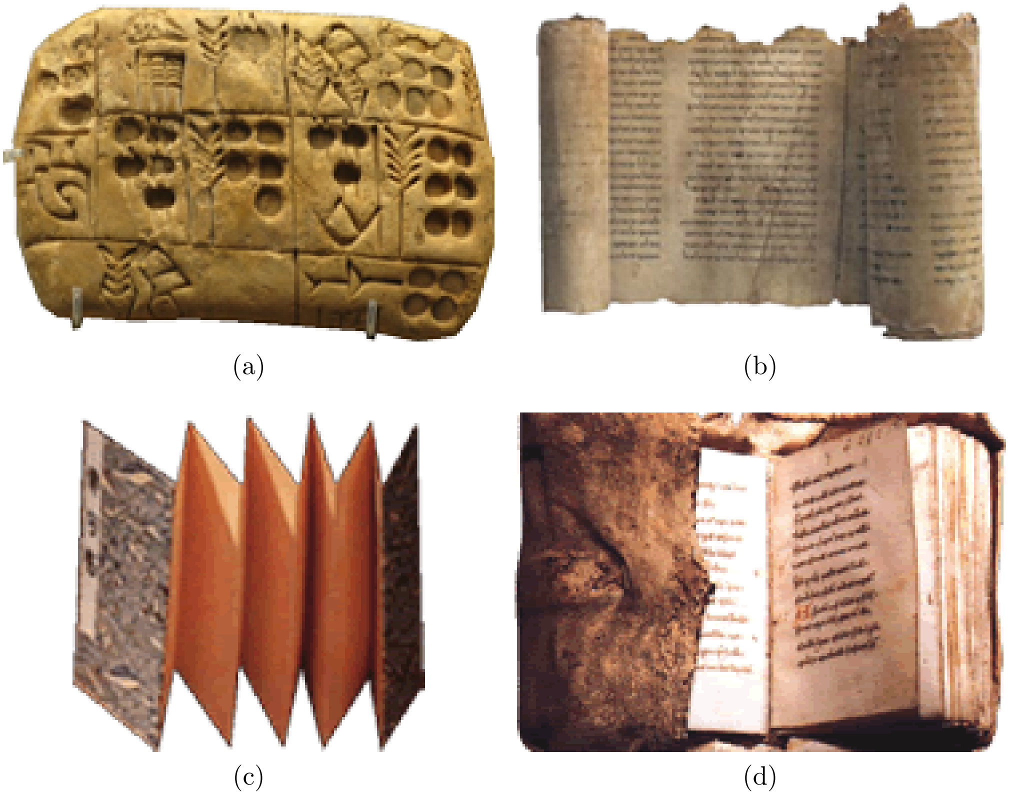
Scroll

Around 2400 B.C., Egyptians began to write on scrolls made up of many papyrus plants, stripped and connected with glue created from the sap of the plant (Baines, 1983). Papyrus sheets were used because they are thin, lightweight, and flexible. In other places, animal skins were used for parchments. Because they were too brittle to be folded, they were rolled into scrolls for storage. Figure 2.1(b) shows an example. At 135 feet, the Great Harris Papyrus, composed by King Ramesses IV, is the longest scroll ever found in Egypt (Dunn, 2006). In China, bamboo and silk were used as writing materials. Depending on size, scrolls were stored in individual wooden boxes with labels, or on open shelves with an identification tag showing at the end of each one so that it could be easily identified.

Scrolls were viewed sequentially, one frame at a time, suitable for oral reading. So important was this continuity that the Athenians raised a statue for the person who had invented a glue for fastening sheets of parchment or papyrus (Walsh, 1892). Nevertheless, reading a scroll was not a simple matter. The reader had to use both hands to roll and unroll continuously while reading. Similarly, to locate a specific passage, it is necessary to spool back and forth through the scroll. This frequent unrolling and re-rolling make scrolls liable to damage. To alleviate these problems, long texts were copied onto several short scrolls.

Concertina

To overcome the limitations of scrolls, concertina or accordion books were developed. They are intermediate between scrolls and codices. Figure 2.1(c) shows an



(a) (b) (c) (d)

Figure 2.1: Ancient devices for portable writing: (a) Sumerian clay tablet (Choi, 2010), (b) scroll (Easley, 2008), (c) concertina (Fuller, 2006) and (d) codex (Drucker, 2006)

example of a concertina book where one or more sheets were glued into one and folded onto itself. The book had hard covers to protect the pages.

This format was preferred over the scroll. It resembles an ordinary book when folded. Readers can randomly access any page, without unrolling and rolling, to search for a passage, and read the document by flipping each fold. Unfolded concertina book are essentially scrolls, so those who are comfortable with scrolls could use them in the same way. When folded they are smaller than scrolls: easier to handle while reading and store afterward. Julius Caesar folded rolls into concertina books to make it easier to send orders to his troops (Suetonius, 2000).

Codex

The codex or book form, shown in Figure 2.1(d), was developed after concertinas. The term *codex* is taken from a Latin word meaning the trunk of a tree, or something made from wood. Around 200 B.C., Greeks and Romans began to use wax tablets backed with wood, metal or ivory, and sometimes connected with cords (often looking like a three ring binder), to write orders or notes (Diringer, 1953). Each tablet could be re-used by melting the wax.

Around 100 A.D., Romans substituted parchment, which was lighter, more durable and resistant to insects and humidity, for the panels in codices. Although some codices were made of papyrus, parchment was cheaper and more easily foldable (Reed, 1972; Roberts, 1954). It was cut into sheets, written on both sides, folded and arranged together in the proper sequence. Heavy wooden boards were used as covers to keep the parchment from curling and protect the pages from damage. By the end of 300 A.D., codex books sported elaborate covers and often included a personal dedication or poem to mark a special occasion. Codices were stored lying flat on a bookshelf.

Initially, codices were intended to be small and compact so they could be read in one hand, and transported easily. This had the unforeseen consequence that texts that were forbidden by the Roman authorities could be hidden away in readers' clothes (Diringer, 1953; Reed, 1972; Roberts, 1954). Equally, with one hand holding the book, the other could write in it. The margins of the page allowed readers to add glosses, annotations and commentaries. Due to their popularity, codex books grew in size and thickness. This prompted innovations in information handling: text could now be organised according to its contents, in books or chapters, transcending the limited capacity of a scroll (Section 2.2.2).

Books in codex format were easy to make, shortening the production process and encouraging book circulation. By the end of the 4^{th} Century A.D., the standard form of documents had largely changed from scrolls to codices. The codex became the preferred format because it allowed people to write on both sides of the material and provided easy access to any part of the text. Page edges gave readers an excellent overall sense of a document. Every page had the same size and layout, which often communicated the type of document and where to find important information. Fixed layouts promote spatial memory and help readers locate information they have seen before.

Pages could be numbered, allowing readers easier access to the sections. Codex books used less material, were easier to read and store, and were portable and searchable. They protected the content better, and allowed thicker pigments for decoration or illustrations. They were cheaper than scrolls and concertinas. Their only real disadvantage was that readers could not view more than two pages at once. Codices quickly took over as the standard document format, and concertinas were relegated to maps and posters.

Paper was cheaper than papyrus and parchment (Mair, 2002). It came to the Western world around 1100 A.D., and by the mid-1400s A.D., it was replacing parchment nearly everywhere. The trend was further accelerated by the advent of the printing press, for which only paper provided a suitable writing material. King François I (1527) decreed that there should be only three standard paper sizes for all of France: folio, quarto and octavo, named according to the number of times a page was folded. People who disregarded this rule were imprisoned. Other countries soon followed suit.

Larger codices used more material, so they were more expensive to produce and bind. For example, a folio sized Bible written on vellum needed the skin of 250 calves (Thomas, 1975). To reduce production costs and selling price, publishers reduced their books to quarto or octavo. Scribes would try to maintain elegant proportions between the book's width, height and thickness (Haines, 1965). Depending on paper quality, many books had a 21:34 width to height ratio. The maximum thickness of an octavo volume was 450 to 650 pages.

In the early 17th Century, publishers started to create paperbacks books (Sutherland, 1993). They were generally written on lower quality paper, with a paper binding. For example, Nicolas Oudot, a bookseller in France, produced the *Bibliotheque bleue* series of small format books covered with a blue paper binding, and sold them for a pittance. The idea proved successful and others followed.

2.2.2 Text structural arrangement

New generations of readers imposed new demands on the written medium, which in turn dictated how the text was organised. Unlike scrolls, people can easily access any page in a codex. Readers began to demand the text organised so that they could read it swiftly, understand the content easily, and skim through to find a specific passage readily. Punctuation, text styles, page layout and the separation of letters into words, sentences, paragraphs and chapters were developed to make it easier for readers to extract the information conveyed in a text. Over the centuries this evolved into the form that we are all familiar with.

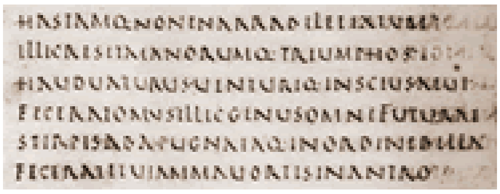
Continuous text

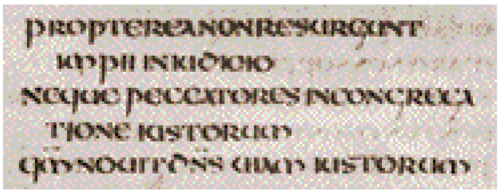
Initially, manuscripts were written in *scriptio continua* (Gaur, 1984; Rutherford, 1905), as illustrated in Figure 2.2(a). There was no capitalisation, spaces, punctuation marks, or any other external indications of structure. Because scribes often knew by heart the text they were transcribing, they needed hardly any visual aids. It was the reader's responsibility to use their literary judgement to gauge the appropriate location for pauses in the text. Before reading the script aloud to the public, readers practised multiple times with their instructors. They memorised the text, understood where the breaks were, and determined the appropriate pronunciation and expression to be used. They would then use the script as a cue sheet only.

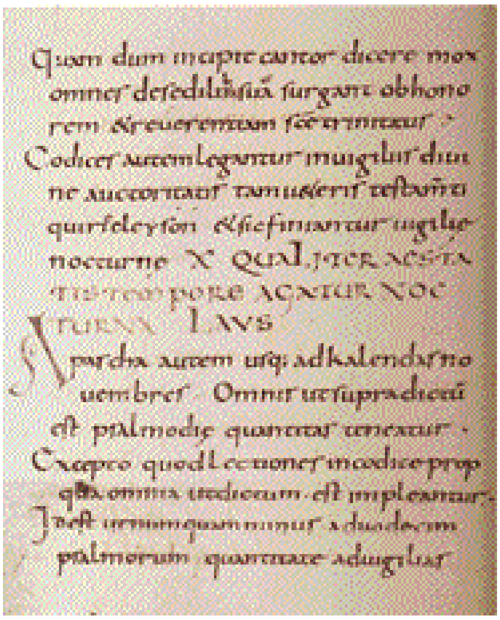
The use of scrolls was well suited to this style of writing because people always read the documents continuously; they rarely jumped to different sections of the text. Nevertheless, the absence of word divisions often led to errors of interpretation. For example, Donat was criticised by his colleague when reading the phrase "collectam *exilio* pubem," a people gathered for exile, as "collectam *ex ilio* pubem," a people gathered from Troy (Parkes, 1993).

Phrase separated narratives

To help those with poor reading skills, around 500 B.C. the Greeks divided text into sections of different lengths, inserted a dividing stroke called *paragraphos* (or


(a)


(b)


(c)


(d)

Figure 2.2: Ancient text writing styles: (a) *scriptio continua* (Pratesi, 1946), (b) *per cola et commata* (Parkes, 1991), (c) word separated (Plenkers, 1906) and (d) sectionalised (Bertschi, 1520)

wedge) before each paragraph, and used punctuation marks to mark the end of sections (Parkes, 1993). Vowels were introduced to help readers identify syllables more swiftly within rows of uninterrupted letters. From the end of 200 B.C. to the beginning of 200 A.D., changes of topic were sometimes indicated by paragraphing, where the first line of the text was projected into the margin as shown in Figure 2.2(b).

Scripts were written in *per cola et commata* style, the text being divided into lines that have coherent meaning and represent a phrase, clause or sentence (Wingo, 1972). This made it easier to search for a passage. Each line was treated as a small paragraph by projecting the first letter into the margin, and readers were expected to take a breath at that part. Sometimes the break might be as short as taking a sip of water, or as long as taking a nap. People usually folded each segment of their concertina books at these line breaks.

Word separated manuscript

Words were frequently separated by a decorative character, like an ivy leaf, to form a special intraword (Sandys, 1919). Although these helped distinguish separate words, they were not as rapidly detected visually as a large blank space.

As the codex became commonly used as the standard document format, around 700 A.D. Irish scribes inserted blank spaces into their manuscripts, and introduced a series of punctuation marks, such as full stop, comma and semicolon (Parkes, 1991). The use of punctuation marks was light and haphazard. They were used interchangeably to indicate brief pauses rather than the syntactic structure of the sentence. Similarly, the spaces did not necessarily correspond to either units of meaning or rhetorical pauses (Levin and Kaplan, 1970). To preserve the *per cola et commata* style, major space was inserted every twenty characters, and minor space was erratically inserted between those characters.

Scribes also introduced minuscule letters, shown in Figure 2.2(c), which were far easier to read than the earlier font styles (Bouma, 1973). There were three different levels of height: ascenders (for letters like b and k), standard (a) and descenders (p). Used in combination with spaces, minuscule font created a graphic silhouette for each word, converting each one into a recognisable unit. Scribes combined upper and lower case letters so that people could visually distinguish important text more rapidly. Uppercase was reserved for headings and entity names; other text was written in lowercase.

Realising that the use of blank space aids reading most effectively when it corresponds to a unit of sound or meaning, scribes tried various different combinations of location and blank space size (Dennison, 1906; Lowe, 1914). At the beginning of the 8th Century, they decided to insert major spaces exclusively between words, and minor spaces between either words or morphemic syllables.

Although this form was not perfect, it was easier to read than *scriptio continua*, providing readers with points of reference for orienting eye movements within lines of text. People could identify words and obtain preliminary information about their meaning. They could recognise words, punctuation, and other signs simultaneously.

In early 1100 A.D., blank spaces was inserted only after words, and minor spaces within a word block were eliminated (Rothkopf, 1971; Saenger, 1990). Page breaks served the reader in a manner analogous to word separation. Punctuation marks had become an integral part of the sentence pattern, not just an indicator of rhetorical pauses. This pattern of spacing and punctuation is standard in today's printed books.

Sectionalised documents

In the 13th Century, narratives were divided into chapters—from the Old French word *chapitre* meaning heading or category—and sections (Parkes, 1976; Reynolds et al., 1983). These represented a separate unit of adventure, experience or lesson, not just a place for readers to take a break from oral reading. Each chapter was preceded by a short text that announced or summarised its content, as shown in Figure 2.2(d), written in red ink so that people could readily distinguish them. Over time, these short texts became chapter titles, and were numbered. By the 15th Century, a numbered list of section headers, or table of contents, was added to the beginning of each document to give readers a condensed summary of the themes, structural elements and text order within the volume as a whole (Parkes, 1976). Gradually, the summary text was removed from the beginning of each chapter and the table of contents, as shown in Figure 2.3.

Each page of a book was numbered to help readers retrieve specific information and to aid bookbinders assemble it. To further help people locate entries in manuscripts, the Italian scholar Papias divided pages into zones and labelled them with letters of alternating sizes placed in the margin (Parkes, 1976; Roberts and Skeat, 1983; Saenger, 1996). Reference books, such as dictionaries, encyclopaedias, and Bibles, had a thumb index where a series of half-circular notches were cut into the fore-edge of the book to indicate sections and facilitate quick

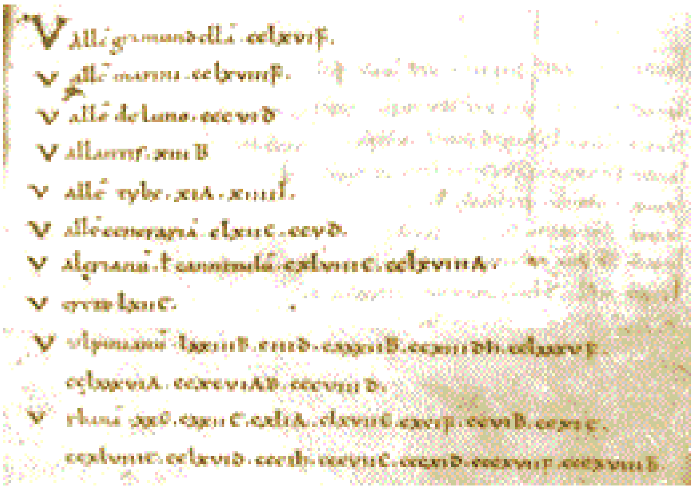


Figure 2.3: Example of an ancient table of contents (di Cantino, 1200)

reference. Small tabs of thicker paper, card, or fabric stamped with letters, words, or numbers were tipped in at the thumb point to strengthen the page.

Early subject indexes were limited to personal names or the occurrence of words in the text, without any controlled vocabulary. They were inserted at the beginning of the book. One of the earliest indexes for the Bible in 1239 A.D. consisted of words from the text and the chapters in which they appeared (Weiberg, 2000). Indexes were alphabetically ordered by the 16th Century. They also included subdivisions of concepts and references to related terms. Section titles or keywords, and the accompanying page numbers, functioned as pointers into the book.

Each page had a repeated running heading to provide a means of locating information and navigating through the book (Parkes, 1976). The first letter of each paragraph was written in a coloured uppercase character. *Scholia*, or marginal notes, often filled the margins with commentaries and relevant external information. Glosses were inserted between lines of text, translating or explaining foreign or obscure words. Blank pages were added at the beginning or end of the book so that readers could make annotations or create their own indexes. When these paratextual features were added, the book became a dynamic knowledge system, organised and structured for various routes of access (Saenger and Heinlen, 1991).

To ensure that more text could fit into a quarto or octavo page, Aldus Manutius

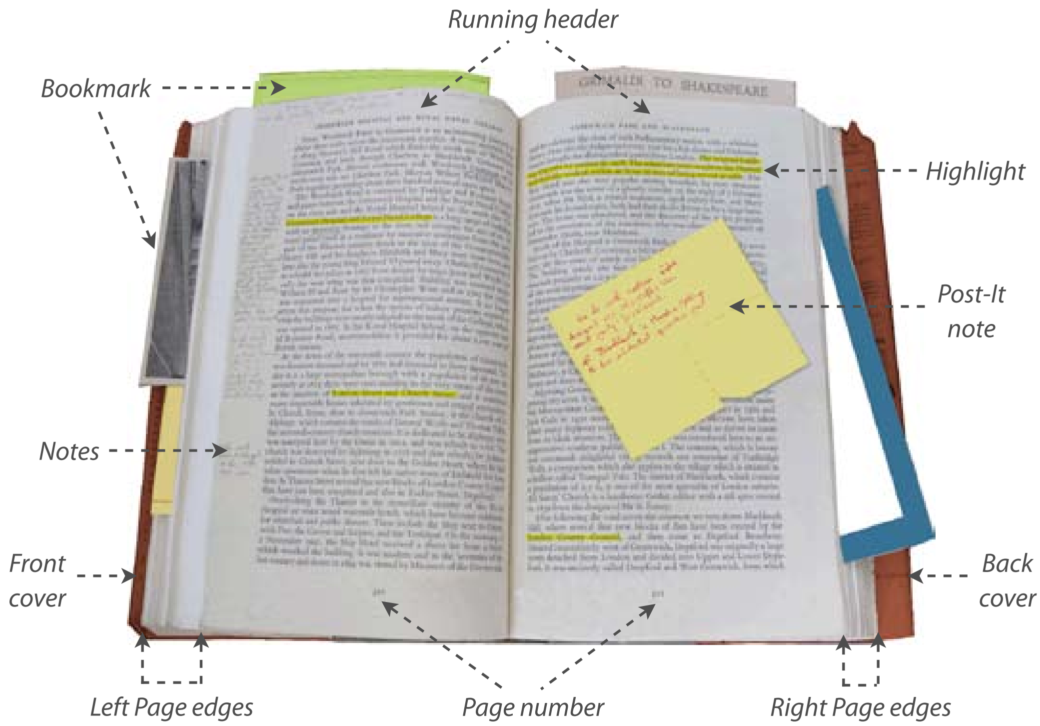


Figure 2.4: Orientation and personalisation features in a modern paper book

chose to print entire manuscripts in cursive or italic fonts designed by Francesco Griffo in 1600 A.D (Ullman, 1960). To increase ease of reading, he eliminated ornateness, abbreviations, ligatures and suspensions. Claude Garamond later created a font that blended capitals, lowercase and italics (Morrison, 1973), making sure that each character, though small, was still clearly legible.

By the early 20$^{\text{th}}$ Century, readers demanded a quick summary of every important topic in the document, and they were impatient to turn elsewhere (Birkerts, 1994; Darnton, 1989; Levy, 1997). As a result, authors started to keep text to a minimum; headings and illustrations to a maximum. A study by Berkenkotter and Huckin (1993) showed that in scientific journals, more and more authors inserted their experimental results into the abstract and introduction sections, and opted to put their research sections later on in the article.

The development of the physical appearance and textual characteristics of the current book form, illustrated in Figure 2.4, has been motivated by the need to find a format that is economical, user-friendly, and enhances people's reading performance. Each feature was gradually incorporated to help the reading process. The aim of this thesis is to investigate whether incorporating these features in an electronic document can further improve performance.

2.3 Electronic documents

At the most basic level, a book is just a series of words. It is how these words are presented that makes the book readable and appealing. Metaphors have been widely used to facilitate learning of computer applications (Benyon et al., 1990; Carroll and Mack, 1985; Hammond and Allison, 1987). They enable users to transfer their existing knowledge about the model, limiting the cognitive overhead of learning a new system.

Curiously, the evolution of computer output has paralleled the development of the book format. Paper rolls were used for early printers—scrolls. Later, Line printers used fan-fold paper, perforated so that it could be folded, boxed, and perused more easily—concertinas. Today, people print on pages and staple them together into books. The parallel is not confined to print technology. Over the last decade, many electronic document representations have emerged, and the evolution from scroll to codex representation can be discerned here too.

Scroll Early text display monitors scrolled; so do webpages, originating in the early 1990s. Examples can be found in text repositories maintained by Project Gutenberg and articles in Wikipedia. In this format, a document is represented as a long continuous page with horizontal and vertical scroll bars. People can change the page size dynamically according to the space available. In the HTML document, shown in Figure 2.5, the user is utilising the full-text search function and an annotation plug-in in their Web browser to search and annotate the document.

Concertina Adobe Reader, Microsoft Word, and ReadUp (Janssen, 2005) represent documents in paginated concertina-like views, as displayed in Figure 2.6. They segment text into sections or pages, and preserve the logical structures, text formatting and page layout of the book. Because they can either scroll or

Figure 2.5: Example of an annotated HTML document

page through the document, readers enjoy both advantages and disadvantages of scrolling and paging views.

Stack of pages Barnes and Noble's Nook (Barnes and Noble, 2009), Amazon's Kindle (Amazon, 2009), Sony Reader (Sony, 2009), XLibris (Schilit et al., 1998), Microsoft Reader, and Hyper-Text Book (Crestani and Ntioudis, 2001) present documents one page at a time. This is like viewing a concertina electronic book format without the ability to scroll from one page to another. Figure 2.7 shows two examples of how pages are displayed in this representation.

Codex Many researchers have started to embed the book metaphor into the electronic environment. As illustrated in Figure 2.8, documents are presented as double-page spreads with a stack of page edges on either side of the opened

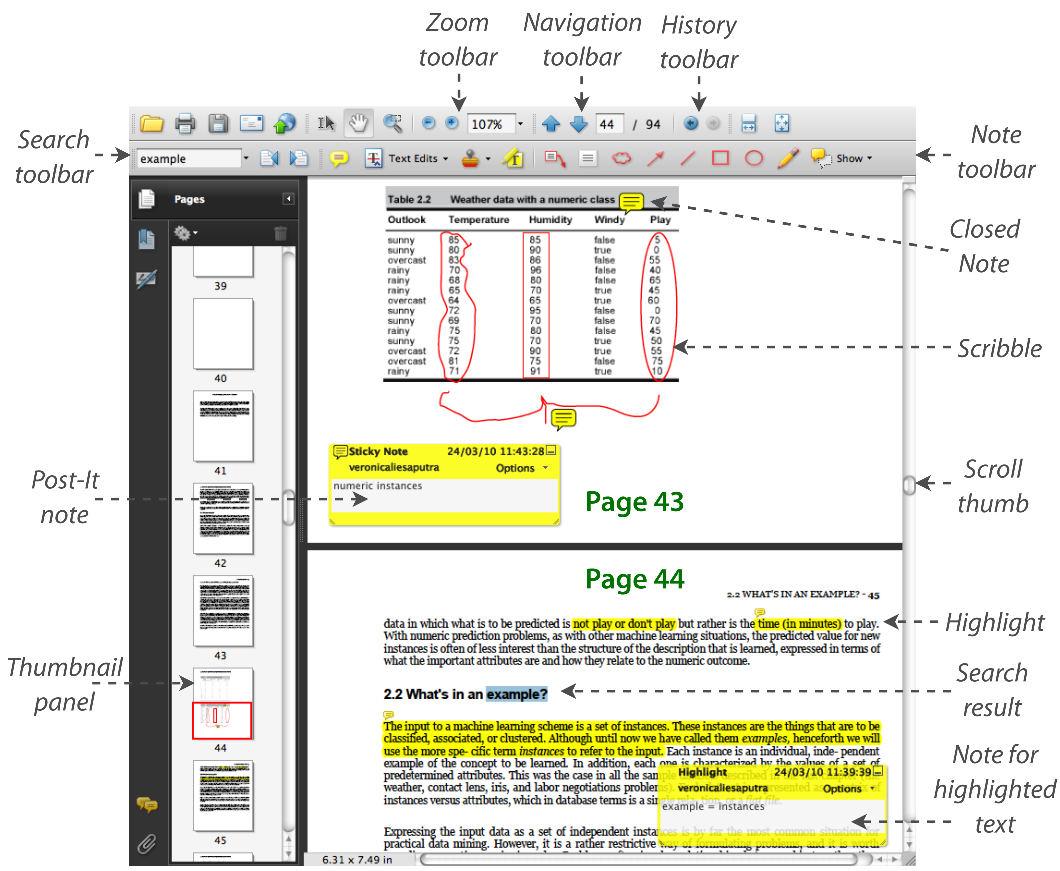
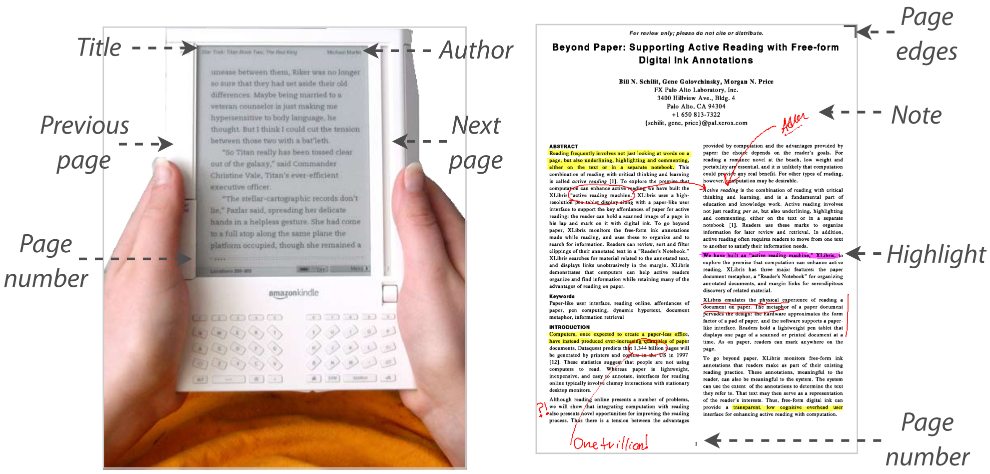


Figure 2.6: Example of an annotated PDF document

page. The appearance and handling of physical books are simulated either in a two-dimensional environment such as Book Emulator (Benest, 1990), Visual Book (Landoni et al., 2000), BookReader (Internet Archive, 2007), Turning the Pages (British Library, 2006) and iBooks (Apple, 2010), or in a three-dimensional environment—WebBook (Card et al., 1996), 3D Book Visualizer (Chu et al., 2003, 2004) and 3Book (Card et al., 2004a,b). In the latter environment, books can be rotated around the x, y and z axis, as shown in Figure 2.9.

2.4 Design considerations

To migrate people from paper books to on-screen reading, finding a good electronic representation is essential. As mentioned in Section 2.1, people utilise different



(a) (b)

Figure 2.7: Electronic book examples that uses the stack of pages metaphor: (a) Kindle (Amazon, 2009) and (b) XLibris (Schilit et al., 1998)

strategies for different types of text and reading goals. The four main issues when presenting documents in the electronic environment are orientation, navigation, readability and personalisation.

2.4.1 Orientation

While reading a book, a reader's attention often wanders away from their current reading position. They might be pursuing interesting external links, going to a meeting, taking a telephone call, looking back at the table of contents or reading a different section of the book. In any case, it is important that people can easily access the last visited page and re-orientate themselves in the document.

To address the problem of returning to previously visited pages, an electronic book application could have a built-in backtracking facility, where all the recently visited pages during the current reading session are automatically stored and displayed in a navigation history list (Monk, 1990). It could remember the last page read, and set the book to open at the last visited page when re-opened. XLibris, WebBook and Adobe PDF are some applications that implement this functionality.

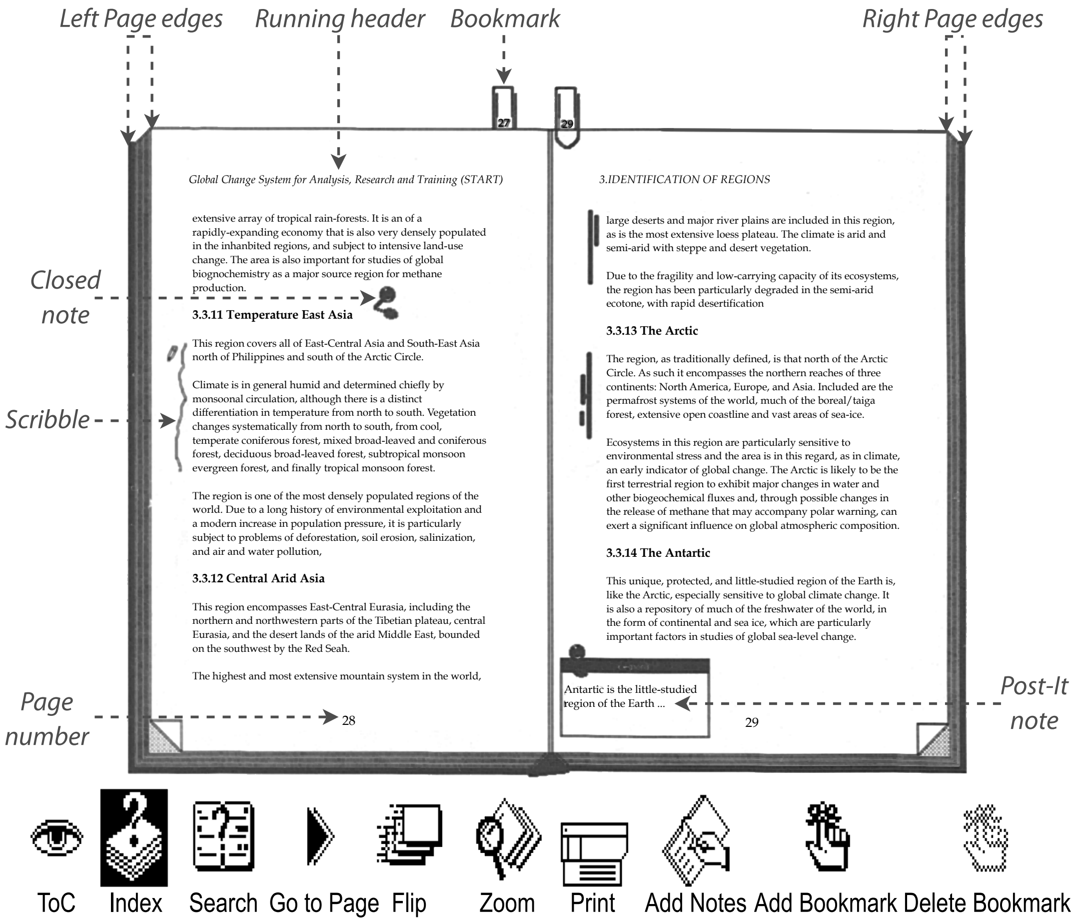


Figure 2.8: Visual Book (Landoni et al., 2000)

With physical books, when readers are interrupted they insert a bookmark (e.g., slip of card, or in the case of a momentary interruption, a finger) where they have left off and simply resume reading at that point. This is possible because a large part of their field of view is devoted to orientation cues like running headers, page numbers, the stack of page edges, and section titles. These cues work together to give readers a strong sense of context, their current position in the document and indicate its length (Conklin, 1987; Crestani and Ntioudis, 2001; Hammond and Allison, 1989). Although electronic books often display running headers, page numbers and section titles, the stack of page edges are replaced by a solitary scroll bar or a page number.

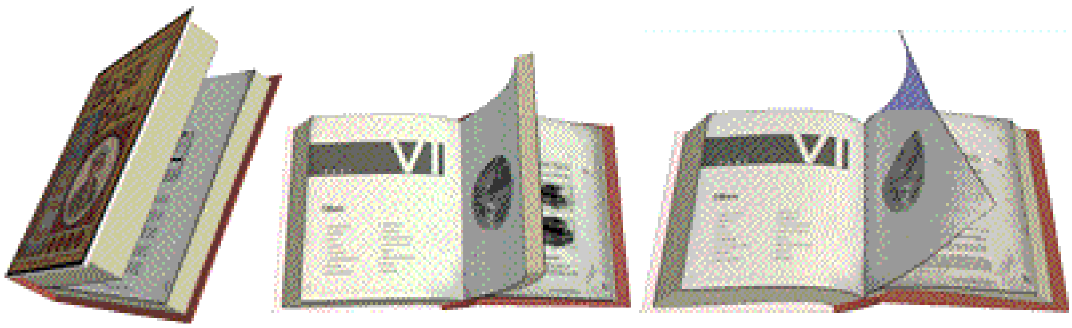


Figure 2.9: 3D Book Visualizer (Chu et al., 2004)

Scroll bar

Instead of page edges, in scrolling formats readers rely on a scroll bar slider. Because in electronic documents scrolling has been used longer than paging, readers may prefer it on the basis of familiarity. Mills and Weldon (1986) found no significant difference between scrolling and paging. On the other hand, Parsons (2001) found that users were not dissatisfied with the scroll format. Scroll bar sliders are usually fixed in size. They give only an approximate idea of the current location in the text. Moving the slider one pixel up or down completely changes the screen contents. Readers must press the Page Up and Down keys, or Up and Down arrow keys, to ensure that they do not skip sections of the document. Because it is hard for them to appreciate the overall organisation of the text, they can easily get lost in the flow of information.

A study by Goldsborough (2000) reveals that 90% of people viewing a web page do not read past the first screen of text. Because users are not comfortable reading scrolled documents on screen, they do not read every word—they merely skim the text looking for interesting headings or pictures. This is further supported by Bernard et al. (2002), Piolat et al. (1998) and Schwarz et al. (1983), who found that readers required significantly less time to find information in long articles with paging than with scrolling.

With pagination, documents are divided into pages that provide both absolute and relative positional cues. Users can rely on their visual memory to remember the location of items, because each page has a fixed layout. On the other hand, in a scrolling view they cannot reliably use the scroll bar slider as an indicator of an item's location. Scrolling weakens the spatial relationship and makes it difficult to read a document non-linearly (Lovelace and Southall, 1983; Olsen, 1994; Rothkopf, 1971). Paging is preferable for long documents.

Page edges

With paginated documents, systems generally present users with two numbers: the current page number and the total number of pages in the book. Although iBooks application displays a stack of page edges on either side of the opened book, these are static images and do not reflect the reader's progress through the book. Despite their apparent stack of pages metaphor, iBook users must rely on these two numbers to know where they are in the document and how long it is.

Not all users benefit from this information. Marshall and Ruotolo's (2002) and O'Hara and Sellen's (1997) experiments reported that subjects preferred analogue visual feedback, like page edges, to inform them where in the book they were and how long the book was. Numbers had little value to them because they could not easily visualise this information.

Instead of displaying the stack of page edges, XLibris provides small right-angled lines at the top left and right corners of the page to indicate the approximate number of pages preceding and following the current page, as shown in Figure 2.7(b). Although Marshall et al.'s (1999) usability study reported that participants liked the paper document metaphor and found it easy to interact with XLibris because of it, it was not specified whether this visualisation was an adequate replacement for the book's stack of page edges.

According to Landoni et al. (2001), the book metaphor should be respected. Indications of a reader's progress through the book should be accurate and visible. The fixed page layout, and the stack of page edges shown in an opened book helps people establish the spatial memory of an item's location within the book. When an article is presented a single page at a time, people lose the incidental exposure to the broader context that they obtain with a two-page spread (Marshall and Bly, 2005).

Crestani and Ntioudis (2001) evaluated users' accuracy and rapidity in satisfying their information needs using printed books and Hyper-Text books, a book application that uses the stack of pages metaphor. They conjectured that redesigning the appearance of Hyper-Text books to resemble physical books would further improve the users' performance.

Overall, experiments also show the book metaphor is well accepted and understood by participants, even those who are not regular users of computer technology (British Library, 2006; Chu et al., 2004; Landoni et al., 2000). People acquire incidental memory of the location of information just by flicking through the book. The visual representation helps them extract the document's logical structure.

2.4.2 Within-document navigation

While there are several ways to give readers a well-constructed overview of a book and facilitate direct access to places within it, less attention has been paid to more serendipitous and unselfconscious ways of navigating. Navigation is the way in which people get from one page to another. This might be through the act of page turning, by randomly flipping several pages, or by searching for specific information.

Page turning

Few electronic book applications add interactivity such as rotating and page turning. In order to go to the next or previous page, users select buttons that go forwards or backwards through the document. With this representation, the change of page is a discrete, rather than an analogue act (Benest, 1990). It takes the form of blanking the area and then drawing the page in front of the user. Though it is efficient, readers briefly lose contact with the text, and their interaction becomes interrupted. Furthermore, during the update, the elements of the page often appear on the screen at random, causing the eye to jump around the screen until the update is complete. Generally, the last component shown is at the bottom of the page. However, during the reading process, users focus their eyes on the top of the page, and as a result this navigation act interferes with the reading process.

During the navigation study performed by Marshall and Bly (2005), participants complained about the inability to turn pages in the electronic book application. Without page turning, they could not continue to see the text on the current page while subtly peeking ahead to the next page to get a foretaste of what is in store. Users of Visual Books suggested the addition of more interactivity, such as rotating and page turning, to further enhance the reading experience (Landoni et al., 2000). These findings are supported by comments made by the readers of Turning the Pages and the 3D Book Visualizer, which incorporate interactive page turning (British Library, 2006; Chu et al., 2004). They commented that the book metaphor and the page turning simulation were amazing, and they rapidly became absorbed in the content. Users enthusiastically appreciated the visualisations and interactive features provided. The new version of Adobe PDF and the recently released iBooks at the time of writing (2010) incorporate an interactive page-turning mechanism. iBooks even include three-dimensional non-linear surface distortion so that letters appear to curve around the page as it is turned.

According to McCusker (1998), the feeling of physically turning a page can be preserved simply by gradually refreshing it from top to bottom. By the time a reader moves their eyes from the last word of the previous page to the first word of the new one, the new page has already been uploaded, and continuity is not lost. Similarly, the developers of Book Emulator and BookReader also thought that analogue page turning is ineffective and inefficient (Benest, 1990; Internet Archive, 2007). Instead of asking readers to manually turn pages, they only need to click the *previous* or *next* button for the system to automatically turn the page. This animation is enough to maintain reading continuity with the text and to inform readers about their direction of travel (Benest, 1990). Furthermore, studies of people's behaviour while reading with a device like Kindle or Sony Reader, show that participants liked the fact that they do not have to turn pages.

Flipping several pages

Opening an electronic book should be at least as quick and easy as opening a paper one. With paper books, readers cannot immediately access a particular page number. They have to turn a block of pages to the approximate location, and flip several pages to reach the desired one. According to a user study performed by Landoni and Gibb (2000), some readers felt that these actions slowed down their reading process. In an electronic environment, they felt they should be able to access the page immediately, through the use, for example, of hyperlinks. However, people like to browse and find things by accident. Almost 82% of Olsen's (1994) participants considered the act of serendipity important while reading. Readers flip pages and scan through documents not only to locate information but also to get a sense of the whole text, and evaluate its content and structure.

Although in the concertina and stack of pages electronic book models users cannot riffle through pages as in paper books, the act of scrolling through page thumbnails might work just as effectively, because it provides readers with a global representation of the information in the document (Graham, 1999; Tufte, 1990). People can look ahead for distinctive figures, drawings, tables or a particular page layout while viewing the page they are currently reading. As a result, they can obtain a rough idea of the document's structure and content. Alternatively, a reader could press a button that automatically turned pages continuously (Benest, 1990; Landoni et al., 2001). The longer the button is pressed, the faster the page is turning. When released, turning ceases. Enabling readers to randomly access pages by clicking on the stack of page edges may also enhance their browsing experience.

Searching

Boyle and Snell (1988), and Fillion and Boyle (1991) suggested that other than these orientation cues, document readers should provide quick and easy access to the table of contents and back-of-the-book indexes because these special pages display an overview of the document structure. They provide context information for users to understand their current position in the book, the amount of information in each section, and what other parts of the book they might be interested to read next. These structures help readers plan a navigation course through the document (McKnight et al., 1991). With immediate access to the table of contents, people can easily re-orientate themselves within the document's logical structure, when they are interrupted.

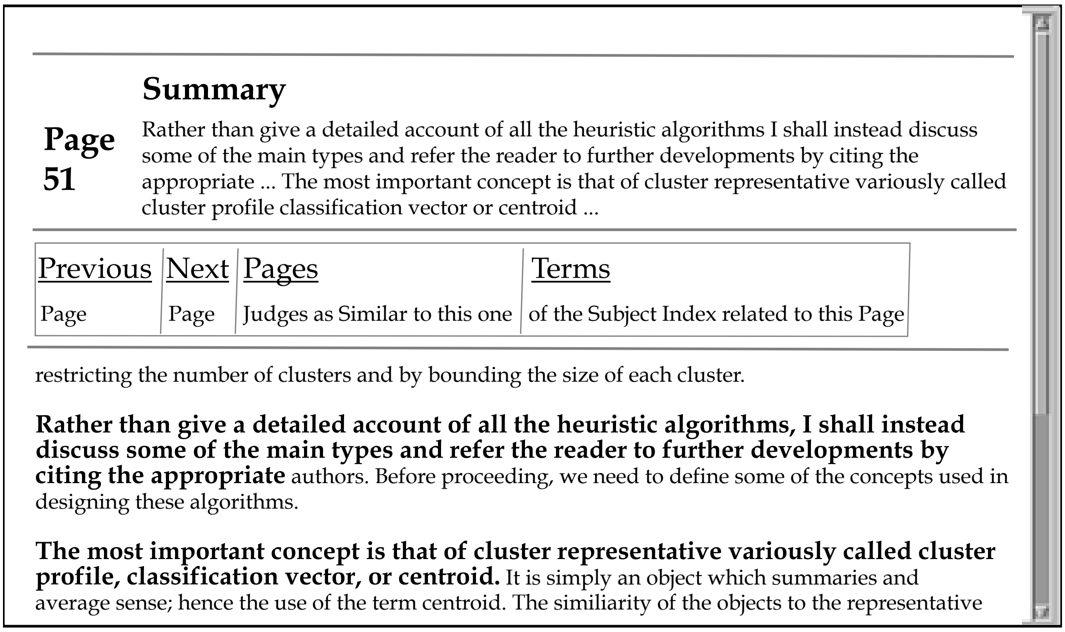
However, readers of electronic documents are generally provided with only two methods for locating information: the table of contents and the full text search function. Back-of-the-book indexes are rarely included. A study by Barker et al. (1994) compared user performance for finding information in two types of electronic books: one type with table of contents and back-of-the-book index included, and the other without. Participants found information more quickly and accurately with the books that include the table of contents and back-of-the-book indexes. This is in line with studies by Landoni and Gibb (2000) and Summerfield and Mandel (1999), who found that users utilised different techniques to locate information: back-of-the-book index for finding specific concepts, and table of contents for more general concepts. Participants also utilised the full text search facility, but commented that this did not replace a back-of-the-book index. Each entry in the back-of-the-book index and table of contents should be hyperlinked, and advanced search facilities should be offered to suit expert readers.

In contrast, a survey of electronic book features by Henke (2002) indicated that participants rated full text search to be more important than the table of contents and back-of-the-book index. In fact, few people mention the index as a desirable electronic book feature. A usability study by Ryan and Henselmeier (2000) also found that users preferred to use full-text search for finding information, but got more accurate results with a back-of-the-book index. They felt that the search facility was slightly more engaging than the index.

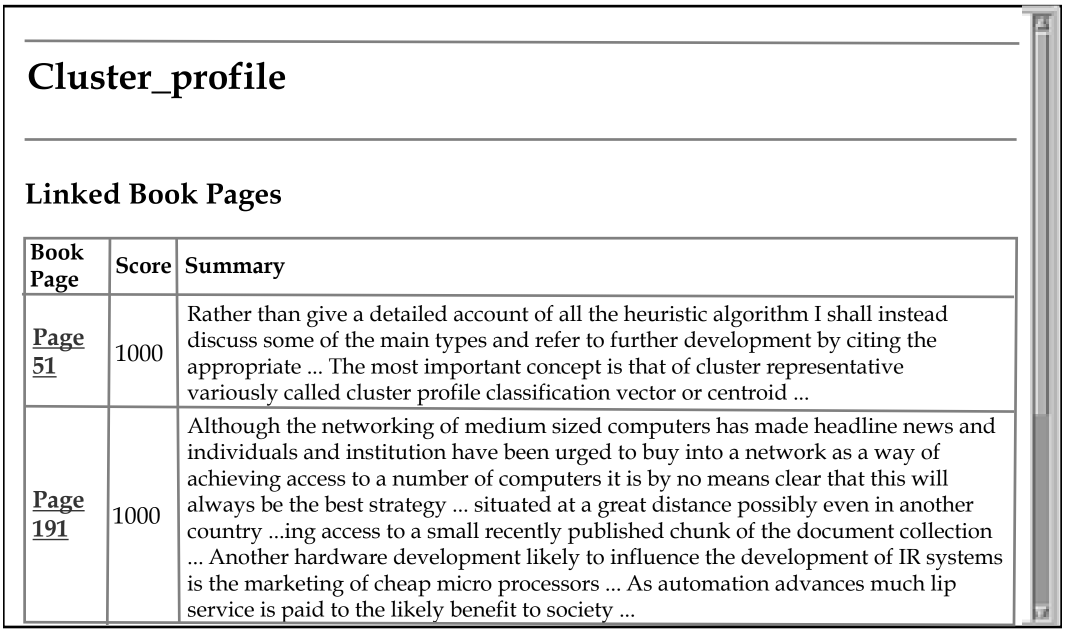
To further help users locate relevant information, 3Book's back-of-the-book index is reorganised every time people enter a query (Card et al., 2004b). When users enter a search term, they see only the index entries that are relevant to it; all other entries are hidden. If a back-of-the-book index is not available, Crestani and Ntioudis (2001) suggest that an electronic book application should automatically compute and display how each page and term is related to other pages and query terms, as shown in Figure 2.10.

2.4.3 Readability

As described in Section 2.2, many presentation aspects of printed books, such as page dimensions, margins, fonts, text size, line width, leading space between lines and character spacing, have been thoroughly developed to make paper books more



(a)



(b)

Figure 2.10: Example of a page in Hyper-Text Book (Crestani and Ntioudis, 2001): (a) the page and (b) list of pages relevant to the term

readable. Although incorporating all these features will close the gap between conventional books and electronic ones, users still get a different experience from reading online documents (Gould et al., 1987; Muter and Maurutto, 1991).

Size, type and quality of a computer screen affects reading speed, number of pauses, concentration time, and the decision to read or skip material (Burbules, 1998; Muter, 1996). People tend to skip more because they cannot read comfortably for a long period of time from a screen. They experienced eyestrain and fatigue from reading on-screen. For a screen display to approach print quality, it should have high resolution, high contrast and minimal glare. Many researchers have sought ways to increase electronic text readability.

Screen quality

Nook, Kindle and Sony Reader utilise the E-Ink Vizplex electronic paper display to deliver a crisp black and white screen that requires no backlighting and reflects light like ordinary paper. This technology eliminates the glare associated with other electronic displays, and is fully readable in direct sunlight for long periods of time. But the refresh rate is slower than desktop or laptop computers, animation or video files cannot be displayed, and readers cannot flip quickly back and forth between pages (Martinez, 2010). Because of their limited capability, many people feel that these devices are too expensive to buy for the value they provide, and prefer to use a laptop or desktop computer (Williams, 2000).

Screen size

Most physical books are portrait rather than landscape oriented. This is the the opposite to that of monitors for PCs or laptops, and consequently readers cannot comfortably fit the full page on the screen (Dillon, 1992). Users must move the document around to be able to see all components of the page. If the size of the document is scaled to fit the screen, the page content may not be legible. Furthermore, the act of zooming in and out of the book is commonly rated as cumbersome and difficult (Waycott and Kukulska-Hulme, 2003).

One solution is to allow readers to alter the font size so that the text is large enough to be read comfortably for long periods of time. But in a paging interface, this changes the text position and the total number of pages in the book. Text might be split across several pages, causing readers to turn back to previous pages more often (Dillon, 1992). Users find it a nuisance to move away from their current page to obtain navigational information, such as section headings. Small screen displays lead to more navigation, and are not suitable for lengthy material (Marshall and Ruotolo, 2002; Waycott and Kukulska-Hulme, 2003).

Text comprehension

Many researchers have reported that the reader's background knowledge facilitates text comprehension and learning (Chiesi et al., 1979; McNamara et al., 1996). The more readers know about the domain of a text, the more likely they are to comprehend it. Their knowledge might be about the language, the world in general, or the domain specific situation. To increase comprehension, books should provide background information that explains relationships between ideas in the text that may be left unstated in the main body of the document (McKeown et al., 1992; McNamara et al., 1996). This additional information is generally found at the back of the book: appendices, glossaries and indexes. Glossaries can be viewed as a mini dictionary in which readers can find definitions of terms in the document. Indexes provide keyword access to the document's content.

Electronic documents can enhance comprehension by linking words and phrases in the main text to the glossary (Landoni et al., 2001). Readers who do not understand the meaning of a term in the text, can click its hyperlink, and are taken straight to the term's definition. Having viewed it, they can return to the text. Unlinked terms are not listed in the glossary, and users have to consult external sources for a definition.

People often prefer to read foreign texts online rather than on paper because they can immediately look up dictionary definitions or language translations (Crane, 2002; Davis, 1989). On the other hand, the use of active links to glossaries or references can be confusing and distracting (Marshall et al., 1999; Wilson et al., 2003). Readers might find it unclear which items were hyperlinks to glossaries and references, and which were hyperlinked to other sections of the book or to external documents. When readers already understand all terms used in the text, they need not consult the glossaries and might be distracted by these superfluous links.

Skim reading

Surveys of people's reading behaviour conducted by Liu (2005) indicate that over 70% of participants employ keyword spotting and picture reading. People tend to read selectively and non-linearly (Eveland and Dunwoody, 2001). They look for section headings, keywords and short key sentences to decide whether a passage is worth reading.

To help readers skim easily, researchers have experimented with ways of presenting automatically detected keywords, phrases or sentences. In 3Book, when a search term is entered, the system automatically highlights potentially relevant keywords or key sentences (Card et al., 2004b). In Wikipedia, important terms are linked to pages that explain them. Similarly, in the skimming mode of XLibris, salient words or phrases stand out in darker shades of grey, and common words fade away in lighter shades of grey (Schilit et al., 1998). The darker the term, the more important it is.

However, the use of highlighted or different coloured words and phrases to call attention to important pieces of text received a mixed response from participants of Wilson et al.'s (2003) user study. Some felt that it helped them locate important passages faster; others felt that the text became unattractive and distracting.

Historical integration

With many electronic book applications, readers do not have a sense of how old a document is (Hill et al., 1992). In some systems, like BookReader and Visual Book, readers are presented with the image of each page of a paper document. All visual features, such as text format, page layout, decolorization and coffee stain, are evident. However, pages cannot include audio, video or animation. In 3D Book Visualizer, pages that have been accessed more than others appear dirtier because each time they are read a few random stains are added to give the book a well-thumbed look.

2.4.4 Personalisation

Personalisation of books is generally private and idiosyncratic, and done in order to help people learn, assimilate and remember the text (Anderson and Armbruster, 1984; Bennett, 1994; Marshall, 1997). Often only the annotator understands why they have highlighted text, written comments on the title page or scribbled some page numbers inside the back cover. Although these marginalia are only intended for them, annotators try to make them as legible and as coherent as possible so that they can still understand them in future. The most basic marks are vertical lines along the side margin, underlines, asterisk, check and exclamation or question marks. Multiple marks are used to heighten the effect. Annotating and bookmarking are an integral part of traditional reading practices that have been culturally transmitted since the Middle Ages. Adler et al.'s (1998) study on how people read in their daily life, observed that 47% of participants' reading time is spent on annotating the document.

Given a limited amount of margin space, annotators must limit their comments to fit on the page. An extreme example is the legend of French mathematician Pierre de Fermat, who declared, in a note published posthumously in 1670, that he had cracked a classic problem but could not record the proof because the margin was too small (Singh, 1998). (It is not clear whether his statement was serious.) People generally continue their notes on the next page or a page with more blank spaces adding a mark like "continue to page 5" to remind them where to go. Another common practice is to turn the book 90° and write comments along the margin instead of across it. Some readers prefer to attach a piece of paper or a Post-It note to the page and write their notes there.

Annotation

Online annotation and bookmarking systems are meant to give electronic documents the same personalisation facilities as paper documents. Many such systems have been built. Some, such as ComMentor (Roscheisen et al., 1995), ScreenCrayons (Olsen Jr. et al., 2004), CritLink (Yee, 2002) and Adobe PDF, allow readers to write at arbitrary places in the document without selecting text first, while others, such as Third Voice (Zeichick, 1999), Annotea (Kahan, 2001), Annotator (Ovsiannikov et al., 1999) and Kindle, only permit annotations once text is highlighted. Some display annotations and bookmarks on the same page; others on a separate page. Some support free-form drawing, highlighting and text comments. Yet in contrast to the richness of personalisation practices in printed books, personalisation of online documents is rare. Among 113 participants in Liu's (2005) survey, over 75% annotated or highlighted paper documents more frequently than electronic ones. People prefer to print, and annotate the paper version (Dayton, 1998; Liu, 2005; Ovsiannikov et al., 1999). Online annotations are distracting and require more effort and practice than a pencil or highlighter (Marshall, 1997; McKnight, 1997; O'Hara and Sellen, 1997). They are not seamlessly integrated with a user's reading activities.

Most keyboard and mouse interfaces for online annotation are cumbersome. People must select a button, and draw with a mouse or type on the keyboard. Nevertheless, there are advantages to the online form. Being in digital form, users are freed from page constraints and can write as much as they want. Hyperlinks can be inserted, providing easy access to relevant information within or outside the document. In paper books, readers generally do not know which pages have been annotated: they must flip through, looking for pages with markings, Post-It notes or clippings. On the contrary, in electronic documents, all annotations can be displayed in one location, allowing readers to review them and navigate quickly between them. Furthermore, hand-held reading appliances with touch screen displays free up one hand, and can imitate the ease and flexibility of annotation on paper (Marshall et al., 1999).

Online annotations can be easily searched, modified, moved or deleted. Although readers sometimes develop their own personal system in which symbols and pen colours have different meanings, electronic annotation systems should not display an array of possible symbols, colours and pen types (Marshall, 1997; Obendorf, 2003; O'Hara and Sellen, 1997). Rather, they should show only the most essential tools. People tend to annotate with whatever is at hand, and presenting too much choice disrupts the reading activity and adds overhead. Online annotation should interrupt reading as little as possible.

If users frequently annotate their electronic documents, their annotations can be utilised to trigger other actions. For example, Baldonado et al. (2000) created a system that sends users an e-mail message containing the annotated phrase whenever they type an annotation beginning with the word “remind.” Highlighting a citation in an article triggers the system to search for a copy in the library and send it to the reader; highlighting a passage of text causes it to seek related documents (Golovchinsky et al., 1999).

Bookmarks

Buchanan and Pearson (2008) report that readers prefer bookmarks to appear as tabs beside the page edges, instead of being listed in a drop-down menu, as in Web browsers, or displayed in an ever-present vertical panel located alongside the page display, as in Adobe’s PDF Reader. The tabs provide stronger spatial cues about the position of the bookmark relative to the reader’s current position in the document. Nevertheless, participants also expressed concern about the limited screen space when bookmarks were displayed as tabs or a sidebar. Listing bookmarks in a menu or sidebar, only provides readers with the bookmark’s page location and title, which is insufficient to facilitate many forms of information access (Cockburn and Greenberg, 1999).

2.5 Summary

According to Kennedy (1984) and Scholes (1989), reading is an intertextual process undertaken by active readers to join their mind more intimately with the mind of the book’s author. Active reading involves activities such as finding related materials, moving from one text to another, critical thinking and learning, and annotating text.

This chapter first considered books in their traditional paper form. In order to identify and extract properties that address the needs of readers, the historic development of the physical and logical structures has been reviewed. Next, we considered how to support or enhance each aspect of the paper book in the electronic environment. To evaluate the hypothesis of this research, an electronic document application prototype, called Realistic Books, has been developed according to this analysis.

This is not the first book simulation application ever devised. Reviewing the literature, as done in this chapter, it is clear that a great deal of attention must be paid to the user interface, mainly in terms of how to translate the codex metaphor onto the screen in order to satisfy the reader's needs. Chapter 4 describes how documents are presented in the Realistic Books system, and explains the services that are available to readers. Chapter 6 reports on how this system affects reading performance and subjective experience, and how it performs compared with conventional book representations such as HTML, Adobe PDF and printed books.

3

Techniques for Modelling Page Turning

Turning the page is a mechanical part of the cognitive act of reading that we do literally unthinkingly. Yet actually producing a computer graphics implementation for modelling page turning is a challenging undertaking. Even the simplest methods are not trivial, and the more sophisticated ones involve complex physical and mathematical models. This chapter provides a detailed survey and explanation of several computer graphics techniques for modelling the act of turning a page, and compares their performance in terms of computation time, space complexity and visual fidelity.

The British Library's "Turning the pages" (British Library, 2006) is a pioneering project that aims to provide a reading experience closely resembling a real book. Readers view a screen showing a double-page spread of what looks like a physical rather than an electronic book. By wiping a finger across the touch-sensitive screen they metaphorically pick up a page and turn it. Pages look three-dimensional; the book's binding moves slightly in sympathy as a page is turned; the left and right stack of page edges indicate how far through the book you are.

In order to accomplish this, photographs have been taken at several intermediate points during each page turn—so that what is displayed is a stored photo, not an artefact computed from a model of the book. There are many images—for

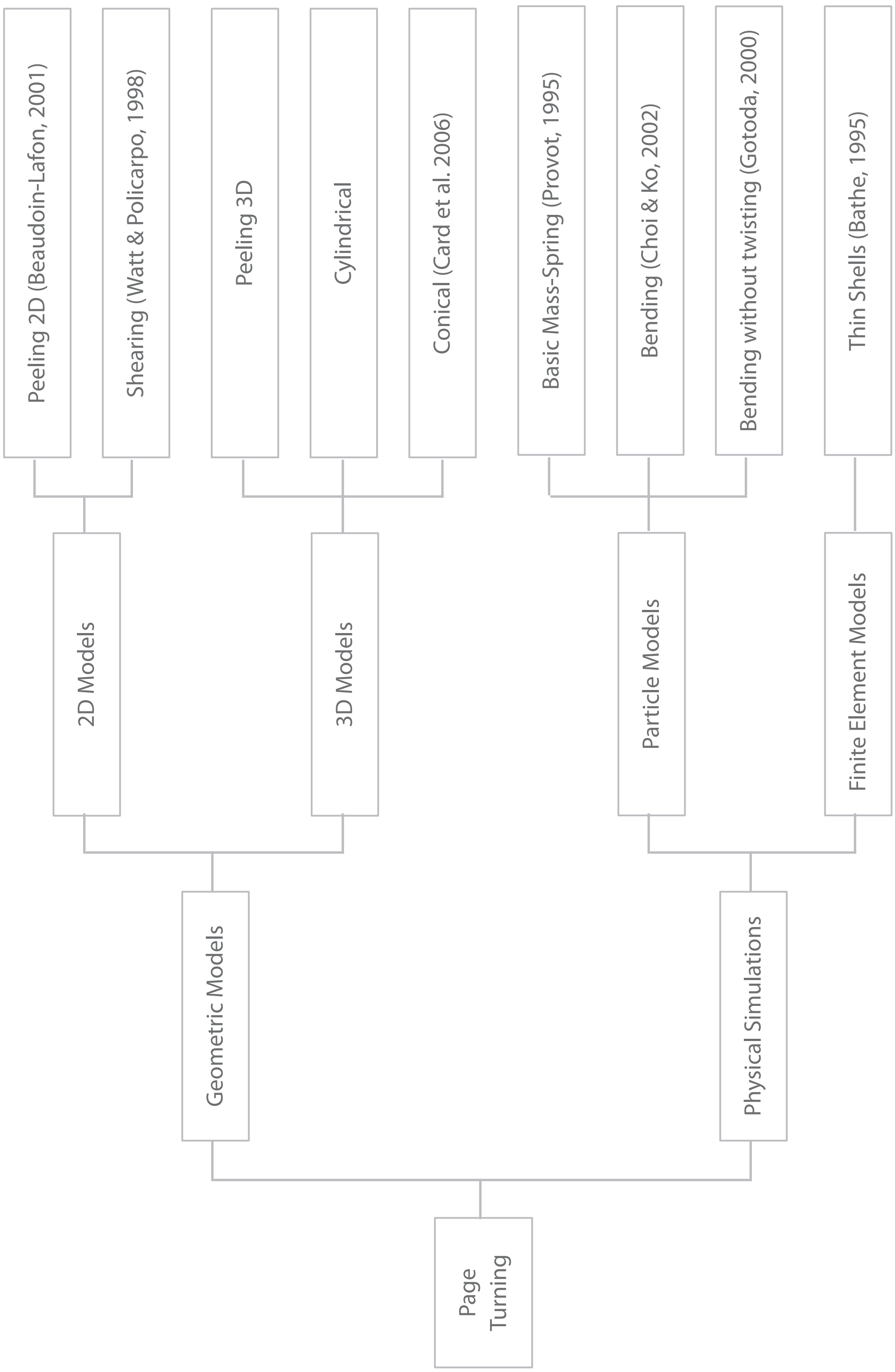


Figure 3.1: Page turning simulation

example, one version of the system consumes 300 MB for only twenty book pages (this includes zoomed-in versions of each page, and accompanying audio). The system is constructed using Macromedia Director. The simulation is compelling, and users rapidly become absorbed in the book itself, turning pages unthinkingly. However, the main drawback is that a stop-animation sequence for every page-turn must be painstakingly photographed in advance, for each book. This is not feasible as a delivery mechanism for online digital libraries.

Figure 3.1 shows several more practical techniques that have been proposed for modelling the act of turning a page (Bathe, 1995; Beaudouin-Lafon, 2001; Bhangal, 2004; Card et al., 2006; Choi and Ko, 2002; Gotoda, 2000; O'Shell, 2006; Provot, 1995). There is a basic distinction between geometric and physical simulations. The former group ignores the physical properties of paper and defines appearance by a set of geometric equations, each of which relates to a particular user action. For example, one action represents turning the page from its lower corner, while another might simulate the paper being folded. Of course, there is an infinitude of possible user actions and it is impossible to know them all in advance: in practice, one picks a limited set and restricts users accordingly. For example, this chapter only explains the case when a user grasps the bottom corner of the page and turns it from the right to the left side of the book; the algorithms are readily adjusted for the more general case where the user grasps the top right corner or (less usually) any point down the right-hand edge or even the top or bottom edge. It is also assumed that initially the page lies flat on the xy plane.

Section 3.1 describes a simple two-dimensional geometric model, called *peeling*, that allows the lower right-hand corner to be moved and shows the paper as though it were creased flat, the position of the crease being adjusted to follow the motion. The standard graphics technique of *shearing* provides an alternative two-dimensional method of turning a rigid page, but it is very straightforward and is not explained here (see, e.g., Watt and Policarpo, 1998). To complete the page turning survey, this research shows how the basic idea of peeling can be easily extended to three dimensions. Section 3.2 describes two further three-dimensional geometric techniques that wrap the paper around a cylinder or cone. In this project, the former is created by simplifying the conical model developed by Card et al. (2006).

Physical models are more realistic than geometric ones but considerably more demanding to compute. There are two types of physical simulation: particle and finite element. The former group uses mass-spring models of the interactions between the particles comprising the paper and are described in Section 3.3. The masses and springs form a mesh, and the forces are summed over the mesh points and then integrated over time to obtain the velocity and acceleration of these points. This section describes first a model that copes only with paper that has cloth-like flexibility; then it allows the paper to bend in more natural curves but still suffers from an unnatural twisting effect; and finally a more sophisticated version that rectifies this twisting problem is presented.

A more comprehensive physical modelling technique is the finite element method that divides the paper into small elements and calculates the force, velocity, and acceleration for each element. This is described in Section 3.4. It is a rather complex operation, though for paper it is simpler than a full finite element analysis because the material forms a thin shell that need not be split into elements in the direction perpendicular to the paper surface.

Finally, Section 3.5 compares the methods mentioned above in terms of measured computation time and space complexity taken by an implementation of each one. The visual fidelity comparison of the illustrations, which are generated through our implementation of each model, are presented as well.

3.1 Two-dimensional geometric model

Imagine turning over the lower right-hand corner of a page and creasing it flat to reveal a triangular-shaped region of the page beneath—a “dog-ear”—with a corresponding triangular region that shows the text on the reverse side of the page. Imagine creating a sequence of successively larger dog-ears. This would be difficult physically (and would make a creased mess of the page), but is trivial in a computer model—and not messy at all. As the motion continues the triangle grows and becomes a quadrilateral when it eventually subsumes the top left-hand corner of the page.

Figure 3.2 shows this peeling effect. Although the underlying model is entirely

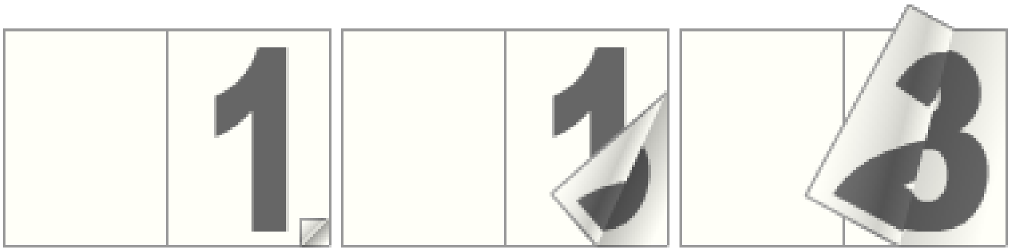


Figure 3.2: Page turning using the peeling form of two-dimensional geometric model

two-dimensional, visual details have been added to simulate the effect of a smooth bend rather than a sharp crease: some shading on the bend and some shadowing just beneath it. It is easy to perform the computation in real time as the page is turned. The method was proposed by Beaudouin-Lafon (2001) to handle overlapping windows, and has been used in other page-turning projects (Bhangal, 2004; O'Shell, 2006).

Although this simple model does not look terribly realistic in the static pictures of Figure 3.2, it is surprisingly effective in practice. The reader defines the path of the corner of the page as it turns: in effect they gesture with the mouse or touch-screen and the corner of the page follows instantly—whether the motion is straight across to the left, or directly upwards, or even up and to the right. There is complete freedom to move the corner of the page (within the physical constraint imposed by not tearing the paper), and the crease and visual shading details follow instantly. The simulation is satisfyingly reactive.

This technique involves partitioning the page into three sections: the visible portion of the page being turned, part of the obverse side of the page that the turn has made visible, and the part of the following page that has been revealed. These regions are shaded differently in Figure 3.3. The region formed by the crease (dark grey) can either be triangular, as in Figure 3.3(a), or quadrilateral, as in Figure 3.3(b). The area revealed (light grey) has exactly the same shape reflected in the axis formed by the crease. In these figures the page's initial position is the rectangle $ABCD$, and the act of turning has moved the lower right corner $\boldsymbol{C}$ to position $\boldsymbol{P}$. The creased region is either the triangle PRS in Figure 3.3(a) or the quadrilateral $PQRS$ in Figure 3.3(b). The location of points $\boldsymbol{Q}$ (if applicable), $\boldsymbol{R}$

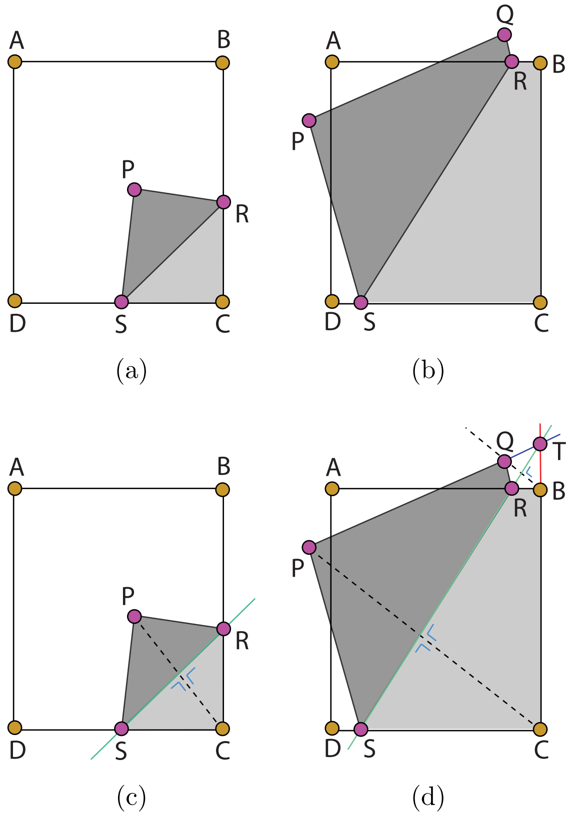


Figure 3.3: The appearance of the page: (a) when $\boldsymbol{R}$ intersect BC (b) when $\boldsymbol{R}$ intersect AB. (c) and (d) shows the geometry of the page at situation (a) and (b), respectively.

and $\boldsymbol{S}$ are calculated from the position of point $\boldsymbol{P}$.

Figures 3.3(c) and 3.3(d) show the geometry of the two situations. The crease is the perpendicular bisector of the line PC. The triangular configuration of Figure 3.3(c) occurs when this bisector intersects the right-hand edge BC of the original page; the quadrilateral configuration of Figure 3.3(d) occurs when it intersects the top edge AB. This allows points $\boldsymbol{R}$ and $\boldsymbol{S}$ to be calculated. In Figure 3.3(d) $\boldsymbol{Q}$ is obtained by drawing a line from corner $\boldsymbol{B}$ parallel to CP, noting where it intersects the crease line (point $\boldsymbol{M}$), and producing it for an equal distance to locate point $\boldsymbol{Q}$—effectively ensuring that the crease line is a perpendicular bisector of QB as well as of PC.

The paper imposes a physical constraint on where the page corner $\boldsymbol{P}$ can be,

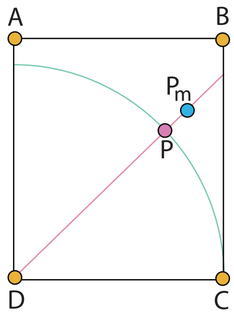


Figure 3.4: Mapped position of the mouse

for the distance PS cannot exceed the length of CD. This constrains $\boldsymbol{P}$ to lie within the arc shown in Figure 3.4; otherwise the page would be torn from the spine. If the user moves the mouse outside the circle (say to position $\boldsymbol{P_m}$) it is silently mapped to a point $\boldsymbol{P}$ on the arc. This feels perfectly natural because the system provides instant feedback by drawing the crease and filling in the contents of the region. Note that although this section has described the technique using a portrait rather than a landscape layout, and turning from the lower right corner of the page instead of the upper right corner, it is straightforward to extend it to cover more general cases, including ones where the user picks up the page along an edge rather than at a corner.

The computational complexity of this technique is constant, independent of any grid size, and in practice the system responds instantaneously to user actions (see Section 3.5.1 for more details). The method can be implemented entirely in Adobe Flash. Three images are maintained: the facing and reverse side of the page being turned, and the facing side of the page beneath. To show the creased region, the page's obverse image is rotated to position its lower corner at $\boldsymbol{P}$ and upper corner at $\boldsymbol{Q}$ (if applicable), and a mask whose shape corresponds to the creased region is applied. To reveal the facing image of the page beneath, a mask shaped like the triangle RCS or quadrilateral $RBCS$ is applied.

A shadow effect can be created using a transparent bitmap image that includes appropriate shading for the revealed part: it is rotated to place its centreline along SR and masked appropriately. Shading can be applied on both sides of the crease, not just to the page beneath. In addition, a small shadow can be applied to the top of the visible edge of the page being turned. These subtle effects, which can be seen by examining Figure 3.2 closely, enhance the perception that the page is being turned in three dimensions. In addition, pages can easily be made slightly transparent so that readers see a hint of the page underneath.

A visual shortcoming of the peeling effect is that the creased area PRS is *exactly* the same as the revealed area CRS (or, in the quadrilateral case, $PQRS$ is exactly the same as $CBRS$). In a true three-dimensional page-turning situation the extension of the paper in the z direction will cause the turned area to be smaller than the revealed area. The three-dimensional bending of the page can be simulated by adding a z-value to each point $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{S}$, and utilising a spline function (e.g., Watt and Policarpo, 1998) to make the bend look curvaceous instead of sharply creased. This makes the turned area smaller than the revealed area. The z-value of $\boldsymbol{P}$ can be assigned using a simple heuristic, and suitable values for $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{S}$ calculated from it. In practice this necessitates segmenting the paper into a regular grid of size $m \times n$ (m and n differ in proportion to the paper's aspect ratio). The use of a spline function raises the time complexity of the technique from constant to $O(n^2)$.

3.2 Three-dimensional geometric model

In the two dimensional model of Section 3.1 the paper is creased and visual effects are incorporated to make the crease look reasonably realistic. An alternative simulation of how the paper bends can be created in three dimensions by shaping the turned page as a cylinder or cone. There are many possible ways of doing this. One is to wrap the page around an imaginary cylinder or cone that rests on the xy plane, touching it along the y axis (which corresponds to the spine of the book). The radius of the cylinder, or the angle of the cone and the position of its apex on the negative y axis, are varied to provide different degrees of curling.

As in the previous technique, the user controls the page turn by moving the point $\boldsymbol{P}$ that corresponds to its lower right-hand corner. In the cylinder model only the horizontal position (x-coordinate) of $\boldsymbol{P}$ can be controlled, and the page has exactly the same curvature all the way along the y axis. In the conical model both the x and y value of $\boldsymbol{P}$ can be controlled, and the position of the cone's apex is adjusted to fit (Card et al., 2006). In both cases the height (z-value) of $\boldsymbol{P}$ is calculated according to a simple heuristic. Of course, if a three-dimensional input device were available the user could control the height of $\boldsymbol{P}$.

The first stage of the computation is to determine the parameters of the cylinder or cone from the position of $\boldsymbol{P}$. In the second stage, the page is modelled as a uniform mesh of $m \times n$ points and as it is turned it is necessary to calculate where each point maps to on the cylinder or cone.

3.2.1 Cylindrical model

Imagine wrapping the paper around a cylinder of radius r. At the beginning, the radius is effectively infinite and the page lies flat on the xy plane. The cylinder axis is located along the y axis. As the turn proceeds, the cylinder's radius is gradually reduced until the midpoint is reached. This causes the page to curl around the cylinder, lifting its right-hand edge from the plane. From the midpoint onward, the radius gradually increases and the cylinder axis moves down and to the left, uncurling the page, until the radius becomes infinite when fully turned and the page lies flat on the left-hand side of the book.

Referring to Figure 3.5, let us assume that a user has moved the bottom right

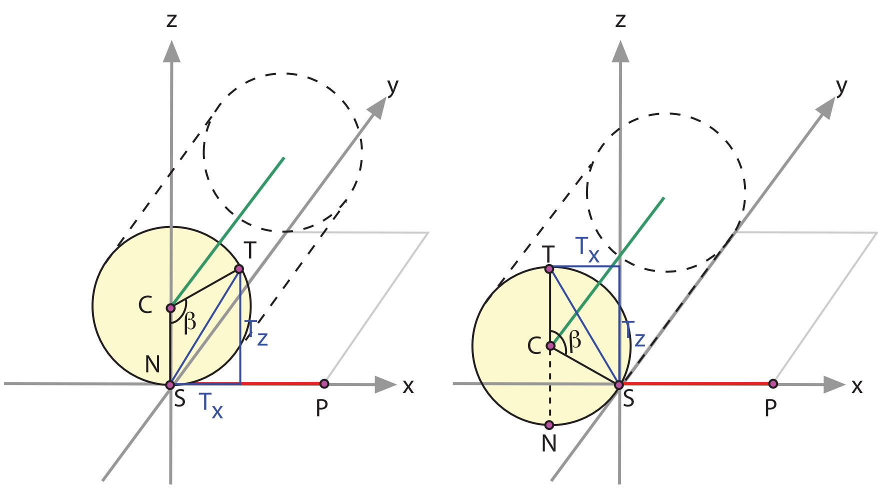


Figure 3.5: Mapping a point to a cylinder

corner of the page $\boldsymbol{P}$ to position $\boldsymbol{T}$. Based on the position of $\boldsymbol{T}$, define a cylinder of radius r whose circular base passes point $\boldsymbol{T}$ and intersects the y axis at point $\boldsymbol{S}$, where the length of the arc ST equals the line SP. Point $\boldsymbol{C}$ is the centre of the circle that lies on the cylinder's axis. If $\boldsymbol{T}$ has not passed the midpoint, the angle between CN and CS is 0. Otherwise, the angle between CN and CS is $180° - \beta$, where β is the angle between CS and CT. The x-value of each mesh point is mapped to the cylinder, so $O(n)$ operations are involved.

The result of this wrapping is rather stilted, and the page appears unnaturally rigid. The turning path parallels the x axis, which is not usually appropriate for single pages but can be useful when an entire multi-page section of a book is turned at once.

3.2.2 Conical model

Readers seldom turn a page with a path that is parallel to the x axis. They generally lift the corner of the page and move it diagonally upward until it reaches the page midpoint, then diagonally downward from the midpoint onwards. Thus, only the grabbed corner should be lifted ahead of the mesh. To achieve this, Card et al. (2006) use a conical model instead of a cylindrical one. The shape of the cone is defined by its angle θ and apex location $\boldsymbol{A} = (0, A_y, 0)$, i.e., it is always anchored to the y axis. Different page trajectories can be obtained by varying these values.

Assume the reader has moved the bottom right corner of the page $\boldsymbol{P}$ to position $\boldsymbol{T}$. Define a cone of radius r whose circular base passes through $\boldsymbol{T}$ and intersects the y axis at the point $\boldsymbol{S}$ where the length of the arc ST equals the line SP, as shown in Figure 3.6. As in the cylindrical model, if $\boldsymbol{T}$ has not passed the midpoint, the angle between CN and CS is 0; otherwise it is $180° - \beta$. Point $\boldsymbol{C}$ is the centre of the circle that lies on the cone's axis. Given the angle β between point $\boldsymbol{T}$, the centre of the cross section, and point $\boldsymbol{S}$, $\boldsymbol{T}$ can be obtained by first rotating $\boldsymbol{S}$ around the line $x = 0, z = r$ through an angle of β to the new position $\boldsymbol{S}' = (S'_x, S'_y, S'_z)$, and then rotating it around the line $y = S_y, z = 0$ through an angle of θ. The radius r of the circle at each cross section varies, so the deformed page is defined by mapping each mesh point onto the cone. The computational

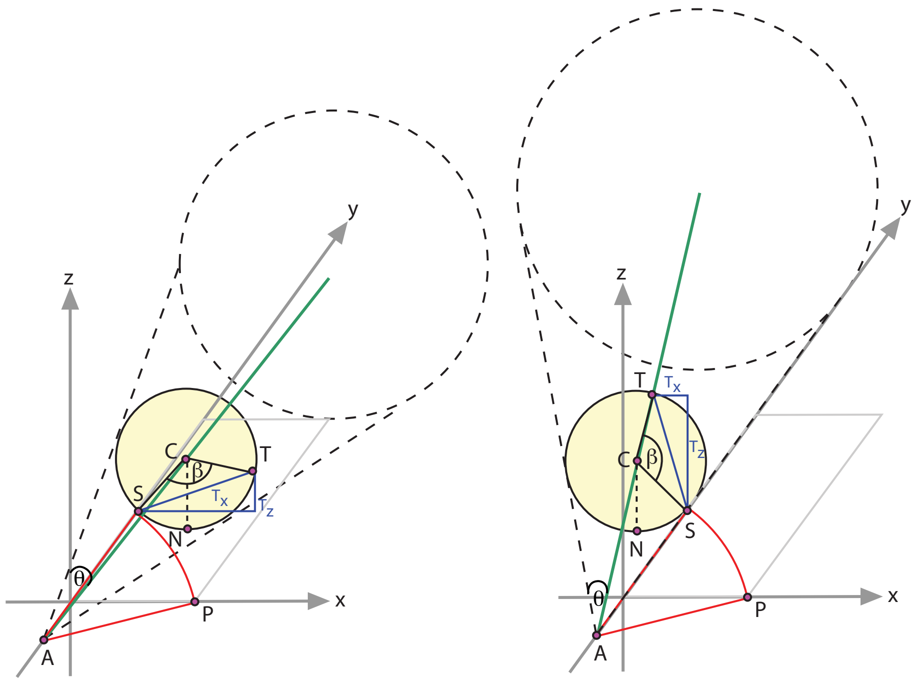


Figure 3.6: Wrapping a page around a cone

complexity of the algorithm is $O(n^2)$.

3.3 Particle models

In particle models, the paper is simulated by dividing the material into a grid of small elements and modelling each element as though it were a single "particle." The accuracy of the simulation is determined by the size of the grid: the result is more accurate with more elements, but needs more computation. Such models are often used in computer graphics to simulate cloth, such as flags waving in the breeze or folds of clothing draped upon a person's body, and even hair (Choi and Ko, 2002; Gotoda, 2000; Provot, 1995). For page-turning, a sheet of paper is divided into $m \times n$ small patches on a two-dimensional rectangular mesh. In reality the patches can be non-planar, which is what gives paper its mechanical properties, but here this is ignored and each patch is treated as though it were atomic.

In this model, the mechanical properties of the paper are simulated by a lattice of springs rather than the elasticity of the material itself. These springs, which are mass-less, model the interactions between the particles that comprise the paper. The force that the springs exert on each particle is calculated: this is called the "internal" force. "External" forces include the user's page-turning action, constraints imposed by the book's spine, and the force of gravity. Both types of force are summed and the result is used to determine the deformation of the paper at each given point in time. This kind of physical simulation has three major components: mesh representation; forces, both internal and external; and time integration.

Three different particle models are considered in this section. The first is for paper that flexes in a way that resembles cloth more than paper (Provot, 1995), the second allows more natural curved bends to develop (Choi and Ko, 2002), while the third incorporates a more sophisticated model of bending based on dihedral angles between constituent planes (Gotoda, 2000). In each case, this section discusses the mesh representation, the forces applied and exerted, and how results are obtained by integrating the system of equations over time.

$\boldsymbol{p}_{ij}$ is the particle at grid point i, j, where $i = 1, 2, ..., m$ and $j = 1, 2, ..., n$. These positions depend on the current time h, but to simplify the notation this dependence is not shown explicitly. The particle has a mass of m_{ij}. It is acted upon by the external forces of gravity and, depending on the particular values of i and j, the user's page-turning action and the spine constraints. The total external force is denoted by $\boldsymbol{F}_{ext}(\boldsymbol{p}_{ij})$:

$$\boldsymbol{F}_{ext}(\boldsymbol{p}_{ij}) = \boldsymbol{F}_{grav}(\boldsymbol{p}_{ij}) + \boldsymbol{F}_{user}(\boldsymbol{p}_{ij}) + \boldsymbol{F}_{spine}(\boldsymbol{p}_{ij}) \ . \tag{3.1}$$

Two kinds of internal forces act on the particle: a damping force and the force exerted by all the springs attached to it. The internal force is designated by $\boldsymbol{F}_{int}(\boldsymbol{p}_{ij})$:

$$\boldsymbol{F}_{int}(\boldsymbol{p}_{ij}) = \boldsymbol{D}(\boldsymbol{p}_{ij}) + \boldsymbol{F}_{spring}(\boldsymbol{p}_{ij}) \ . \tag{3.2}$$

The damping force $\boldsymbol{D}(\boldsymbol{p}_{ij})$ is calculated from the damping constant and the node's velocity, while the spring force $\boldsymbol{F}_{spring}(\boldsymbol{p}_{ij})$ is computed from the displacements of the node and those nodes to which it is attached, and each spring's stiffness constant.

The sum of the external and internal forces on the particle is

$$\boldsymbol{F}_{ij} = \boldsymbol{F}_{ext}(\boldsymbol{p}_{ij}) + \boldsymbol{F}_{int}(\boldsymbol{p}_{ij}) \ . \tag{3.3}$$

The value of this at time h is used to determine the particle's position at time $h + \Delta h$. This operation is done on each particle in the grid, so $O(mn)$ operations are involved.

3.3.1 Basic particle model

In the simplest type of particle model, developed by Provot (1995), nodes are connected with three types of mass-less springs as shown in Figure 3.7: stretch, shear and bend springs. They produce a force that depends on their length and stiffness constant. The stiffness is the same for all springs of a given type.

1. Stretch springs connect $\boldsymbol{p}_{ij}$ with $\boldsymbol{p}_{i\pm1,j}$ and $\boldsymbol{p}_{i,j\pm1}$.
2. Shear springs connect $\boldsymbol{p}_{ij}$ with $\boldsymbol{p}_{i\pm1,j\pm1}$.
3. Bend springs connect $\boldsymbol{p}_{ij}$ with $\boldsymbol{p}_{i\pm2,j}$, $\boldsymbol{p}_{i,j\pm2}$ and $\boldsymbol{p}_{i\pm2,j\pm2}$.

These springs are responsible for simulating the paper's stretch, shear and bend resistance. Stretch springs define the resistance of the material to elongation or compression in the horizontal and vertical directions. Shear springs define the resistance in diagonal directions. When the paper is bent, particle $\boldsymbol{p}_{ij}$ moves closer to its next-nearest-neighbour particles, and the paper's bend resistance is calculated from the distance of $\boldsymbol{p}_{ij}$ to eight of its next-nearest-neighbours.

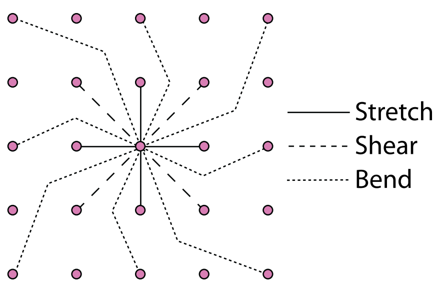


Figure 3.7: Simple mass-spring model

Effect of forces

The stiffness constant of the spring linking particles $\boldsymbol{p}_{ij}$ and $\boldsymbol{p}_{kl}$ is denoted by $K_{ij,kl}$, and the rest length of the spring, which is its initial length at time $h = 0$, by $\boldsymbol{l}^0_{ij,kl}$. The length of the spring at time h is $\boldsymbol{l}_{ij,kl} = \boldsymbol{p}_{kl} - \boldsymbol{p}_{ij}$. According to Hooke's Law, the total spring force on $\boldsymbol{p}_{ij}$ is given by

$$\boldsymbol{F}_{spring}(\boldsymbol{p}_{ij}) = \sum_{(k,l)\in R} -K_{ij,kl}(|\boldsymbol{l}_{ij,kl}| - |\boldsymbol{l}^0_{ij,kl}|)\frac{\boldsymbol{l}_{ij,kl}}{|\boldsymbol{l}_{ij,kl}|} \ , \tag{3.4}$$

where the sum is over R, the set of nodes that are connected with $\boldsymbol{p}_{ij}$.

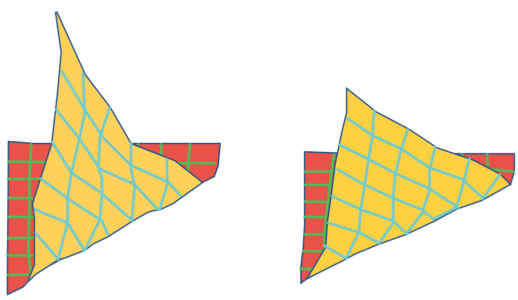
The damping force for node $\boldsymbol{p}_{ij}$ is

$$\boldsymbol{D}(\boldsymbol{p}_{ij}) = -K_d\dot{\boldsymbol{p}}_{ij} \ , \tag{3.5}$$

where K_d is the damping coefficient of the paper and $\dot{\boldsymbol{p}}_{ij}$ is the velocity of $\boldsymbol{p}_{ij}$.

Limit strain

The elasticity of real materials is non-linear. Once a certain degree of elongation has been reached, they become very stiff extremely rapidly when stretched further. If the applied load is high the material will rupture; however, this thesis will not attempt to model ruptures. Figure 3.8(a) shows the appearance of a piece of



(a) (b)

Figure 3.8: The appearance of the page: (a) without limit straining (b) with limit straining

paper that has been stretched unreasonably, this problem can be fixed by limiting the amount of strain on each spring.

The amount of elongation is defined as

$$\tau = \frac{|\boldsymbol{l}_{ij,kl}| - |\boldsymbol{l}^0_{ij,kl}|}{|\boldsymbol{l}^0_{ij,kl}|} \; . \tag{3.6}$$

It is limited to a predetermined constant τ_c (normally set to 10%). If the elongation threatens to exceed τ_c the nodes are brought closer together to bring it back to this value. This is done by moving the both particles towards their mid-point if they are both loose, while if one is fixed—e.g., if it lies on to the book's spine—the other is moved closer to it. The result is shown in Figure 3.8(b), where each spring has been limited to a predefined maximum elongation.

Integrating over time

Given each node's position $\boldsymbol{p}_{ij}$, velocity $\dot{\boldsymbol{p}}_{ij}$ and total force $\boldsymbol{F}_{ij}$ at time h, the position $\boldsymbol{p}'_{ij}$ and velocity $\dot{\boldsymbol{p}}'_{ij}$ at time $h + \Delta h$ can be determined. According to Newton's law, the acceleration at time $h + \Delta h$ is

$$\ddot{\boldsymbol{p}}'_{\boldsymbol{ij}} = \frac{\boldsymbol{F}_{ij}}{m_{ij}} \; . \tag{3.7}$$

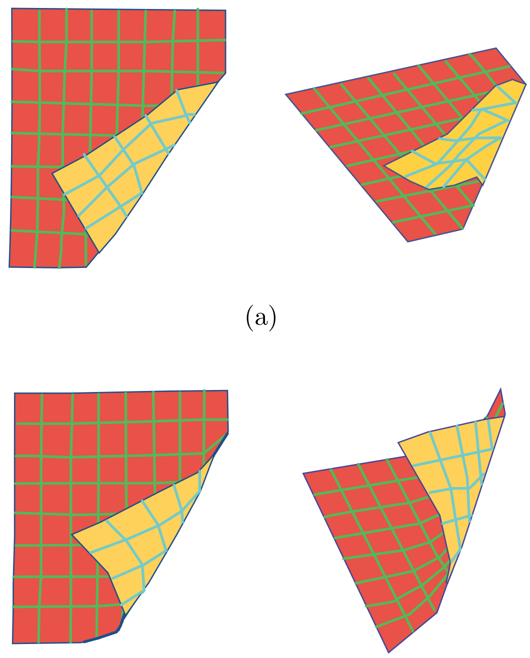
Using explicit Euler time integration,

$$\begin{aligned} \dot{\boldsymbol{p}}'_{ij} &= \dot{\boldsymbol{p}}_{ij} + \ddot{\boldsymbol{p}}'_{ij}\Delta h \; , \quad \text{and} \\ \boldsymbol{p}'_{ij} &= \boldsymbol{p}_{ij} + \dot{\boldsymbol{p}}'_{ij}\Delta h \; . \end{aligned} \tag{3.8}$$

These equations are used to update the nodes' position and velocity at every time step.

3.3.2 Bending

This section describes a more advanced mass-spring model developed by Choi and Ko (2002) that can be bent into smooth curves. Paper is strongly resistant



(a)

(b)

Figure 3.9: The appearance of the page: (a) using the simple mass-spring model of Section 3.3.1; (b) using the improved model of Section 3.3.2

to stretching or shearing, but has little initial resistance to compression. However, when a certain degree of compression has been reached, it rapidly becomes stiffer under further compression. To simulate this a more sophisticated damping, bending and compression model is needed.

Figure 3.9(a) shows how unnatural bending can look with the previous model. The page behaves in a manner more akin to cloth than paper, with insufficient resistance to the bending force—it crumples rather than bending stiffly and smoothly as paper does. In contrast, Figure 3.9(b) shows a smoothly contoured bend that was produced by the model described in this section.

While it would be perfectly possible to use the same explicit time integration that was developed in the last section for this new model, it would be much slower. In order to make the system respond quickly to the user's interaction, semi-implicit time integration is used instead. This is more stable than the explicit integration procedure, so large time steps can be used to reach the equilibrium state quickly.

Each particle is connected with others in the mesh in the same manner as before, and again all springs of the same type have the same stiffness constant. However, different formulae are used to calculate the spring forces. Stretch and shear springs only affect the paper's resistance to elongation in the vertical, horizontal and diagonal directions, while bend springs affect its resistance to bending and compression in these directions.

Effect of forces

Let $K_{ij,kl}$ denote the stiffness constant of a spring connecting $\boldsymbol{p}_{ij}$ to $\boldsymbol{p}_{kl}$, and $\boldsymbol{l}^0_{ij,kl}$ be the spring's initial length at time $h = 0$. The length of the spring at time h is $\boldsymbol{l}_{ij,kl} = \boldsymbol{p}_{kl} - \boldsymbol{p}_{ij}$. The total spring force on $\boldsymbol{p}_{ij}$ is given by

$$\boldsymbol{F}_{spring}(\boldsymbol{p}_{ij}) = \sum_{(k,l)\in R_s} \boldsymbol{s}_{ij,kl} + \sum_{(k,l)\in R_b} \boldsymbol{b}_{ij,kl} \ , \tag{3.9}$$

where the sums are over R_s, the set of nodes connected to $\boldsymbol{p}_{ij}$ by stretch and shear springs, and R_b, the set of particles connected to $\boldsymbol{p}_{ij}$ by bend springs.

The force exerted by the stretch and shear springs is the same as before (equation (3.4)) for elongation but zero for compression:

$$\boldsymbol{s}_{ij,kl} = \begin{cases} -K_{ij,kl}(|\boldsymbol{l}_{ij,kl}| - |\boldsymbol{l}^0_{ij,kl}|)\frac{\boldsymbol{l}_{ij,kl}}{|\boldsymbol{l}_{ij,kl}|} & : \ |\boldsymbol{l}_{ij,kl}| \geq |\boldsymbol{l}^0_{ij,kl}| \\ 0 & : \ |\boldsymbol{l}_{ij,kl}| < |\boldsymbol{l}^0_{ij,kl}| \end{cases} \tag{3.10}$$

The force exerted by the bend springs is

$$\boldsymbol{b}_{ij,kl} = \begin{cases} 0 & : \ |\boldsymbol{l}_{ij,kl}| \geq |\boldsymbol{l}^0_{ij,kl}| \\ -K_{ij,kl} f_b(\frac{|\boldsymbol{l}_{ij,kl}|}{|\boldsymbol{l}^0_{ij,kl}|})\frac{\boldsymbol{l}_{ij,kl}}{|\boldsymbol{l}_{ij,kl}|} & : \ |\boldsymbol{l}_{ij,kl}| < |\boldsymbol{l}^0_{ij,kl}| \end{cases} \tag{3.11}$$

This equation models the non-linear nature of bend resistance. The shapes of buckled paper is close to a circular arc. This particle model approximates the arc's length by the spring's rest length $|\boldsymbol{l}^0_{ij,kl}|$ and expresses the curvature as a function of the distance between the particles. Choi and Ko (2002) approximated the curvature function f_b as a fourth-order polynomial,

$$f_b(s) = -a_4 s^4 + a_3 s^3 - a_2 s^2 + a_1 s - a_0 \ ,$$

where positive numeric values for the coefficients a_4, a_3, a_2, a_1 and a_0 were derived from the physical deformation constants that characterise the material in question.

In the previous model the same damping constant is used for stretch, shear and bend (equation (3.5)). However, paper resists stretching and shearing much more than it resists bending, and this requires higher values for the these two damping constants than for the bend damping constant. This was not done in the previous model because bending looked unrealistic anyway, and reducing the bend damping constant causes in-plane oscillations. To inhibit these requires a more sophisticated damping model than the one developed in the previous section. If $K_{d_{ij,kl}}$ is the damping constant of the spring linking particles $\boldsymbol{p}_{ij}$ and $\boldsymbol{p}_{kl}$, the total damping force on $\boldsymbol{p}_{ij}$ is

$$\boldsymbol{D}(\boldsymbol{p}_{ij}) = - \sum_{(k,l)\in R} K_{d_{ij,kl}} (\dot{\boldsymbol{p}}_{ij} - \dot{\boldsymbol{p}}_{kl}) \ , \qquad (3.12)$$

where R is the list of nodes that are connected to $\boldsymbol{p}_{ij}$ by any type of spring.

Integrating over time

In this model, semi-implicit time integration with a second-order backward difference formula is used to calculate the new position of particle $\boldsymbol{p}_{ij}$ with mass m_{ij}. Let $\boldsymbol{p}_{ij}$ and $\dot{\boldsymbol{p}}_{ij}$ respectively be the position and velocity at time $h - \Delta h$, and $\boldsymbol{p}'_{ij}$ and $\dot{\boldsymbol{p}}'_{ij}$ the position and velocity at time h. Given this information, the new position $\boldsymbol{p}''_{ij}$ and velocity $\dot{\boldsymbol{p}}''_{ij}$ at time $h + \Delta h$ can be calculated as follows:

$$\begin{aligned} \Delta \boldsymbol{p}''_{ij} &= \frac{1}{\Delta h}\left(\frac{3}{2}\boldsymbol{p}''_{ij} - 2\boldsymbol{p}'_{ij} + \frac{1}{2}\boldsymbol{p}_{ij}\right) \ , \quad \text{and} \\ \Delta \dot{\boldsymbol{p}}''_{ij} &= \frac{1}{\Delta h}\left(\frac{3}{2}\dot{\boldsymbol{p}}''_{ij} - 2\dot{\boldsymbol{p}}'_{ij} + \frac{1}{2}\dot{\boldsymbol{p}}_{ij}\right) \ . \end{aligned} \qquad (3.13)$$

These quantities must also satisfy the vector equality

$$\begin{pmatrix} \Delta \boldsymbol{p}''_{ij} \\ \Delta \dot{\boldsymbol{p}}''_{ij} \end{pmatrix} = \begin{pmatrix} \dot{\boldsymbol{p}}''_{ij} \\ \frac{\boldsymbol{F}''_{ij}}{m_{ij}} \end{pmatrix} \tag{3.14}$$

and

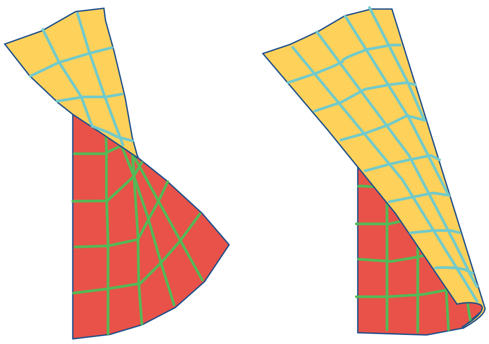
$$\boldsymbol{F}''_{ij} = \boldsymbol{F}'_{ij} + \sum_{(k,l)\in R} \left(\frac{\partial \boldsymbol{F}''_{ij}}{\partial \boldsymbol{p}''_{kl}} \Delta \boldsymbol{p}''_{ij} + \frac{\partial \boldsymbol{F}''_{ij}}{\partial \dot{\boldsymbol{p}}''_{kl}} \Delta \dot{\boldsymbol{p}}''_{ij}\right) , \tag{3.15}$$

where R is the list of nodes that are connected to $\boldsymbol{p}_{ij}$ by any type of spring.

3.3.3 Bending without twisting

Previous sections modelled the resistance of the paper to bending using springs that connect each particle with its next-nearest-neighbours. This model is simplistic because these springs are insensitive to changes in the dihedral angles between faces. The particle model described in this section takes into account these angle changes by modifying the way the bending force is calculated. This model was developed by Gotoda (2000).

The effect can be seen in Figure 3.10, which shows a page using the model in the previous section and the model that will be developed in this section. Clearly, the latter looks more realistic, because the twist in the edge nearest to the viewer



(a) (b)

Figure 3.10: The appearance of the page: (a) using the model from Section 3.3.2; (b) using the improved bending model from Section 3.3.3

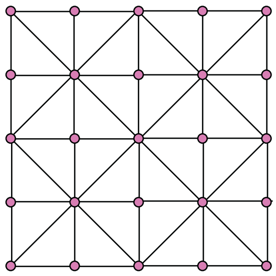


Figure 3.11: Mesh representation

has been eliminated.

In this model, the paper is discretized into a set of right-angled triangles, as shown in Figure 3.11. Initially the paper is flat, and $z = 0$ for each particle. In order to calculate the internal forces on the triangles instead of on each particle, slightly different formulae are used to determine the material's resistance to stretching, shearing and bending.

Effect of forces

The improved bending model is better expressed in terms of energy rather than force; the latter is calculated as the spatial derivative of energy. As before, three types of energy—stretch, shear and bend—are used to characterise the difference between the undeformed configuration and the deformed one. The internal energy working on each particle $\boldsymbol{p}_{ij}$ is

$$\boldsymbol{E}_{spring} = \sum_{x \in T} \boldsymbol{E}_{stretch}(x) + \sum_{x \in L} \boldsymbol{E}_{shear}(x) + \sum_{x \in B} \boldsymbol{E}_{bend}(x) \ ,$$

where T is the set of all triangles containing particle $\boldsymbol{p}_{ij}$, L is the set of all edges that are connected to particle $\boldsymbol{p}_{ij}$, and B is the set of all edges that are connected to particle $\boldsymbol{p}_{ij}$ and shared by two triangles. The stretch component relates to the triangle's area, the shear component to the length of its edges, and the bend component to the dihedral angle with adjacent triangles; these three components are discussed further below. The total internal force at particle $\boldsymbol{p}_{ij}$ is obtained by differentiating the energy:

$$\boldsymbol{F}_{spring}(\boldsymbol{p}_{ij}) = \frac{\partial \boldsymbol{E}_{spring}}{\partial \boldsymbol{p}_{ij}} \ .$$

Stretch energy is produced by the triangle's resistance to stretching or compression, which corresponds to changes in the triangle's area. Let A^0 be its initial area ($h = 0$) and A^h be its area at time h. The stretch energy for the triangle is

$$\boldsymbol{E}_{stretch}(x) = -K_{stretch}A^0(1 - \frac{A^h}{A^0})^2 \ , \tag{3.16}$$

where $K_{stretch}$ is the stretch stiffness constant.

Shear energy is produced by the triangle's resistance to shearing, which corresponds to changing the length of its edges but not its area. The shear energy for the triangle is the sum of the shear energies for each of its edges. Consider the edge e. Let $\boldsymbol{l}^0$ be its initial length (at time $h = 0$) and $\boldsymbol{l}^h$ its length at time h. The shear energy that works on the edge is

$$\boldsymbol{E}_{shear}(x) = -K_{shear}|\boldsymbol{l}^0|(1 - \frac{|\boldsymbol{l}^h|}{|\boldsymbol{l}^0|})^2 \ , \tag{3.17}$$

where K_{shear} is the shear stiffness constant.

Bending energy is produced by altering the mean curvature between two triangles that share the same edge. The bending energy for the triangle is the sum of the bending energies for each edge that borders a neighbouring triangle. Figure 3.12 shows two triangles T_1 and T_2 that share a common edge e. Their initial (flat) position is shown unshaded; the shaded triangles are leaning upwards, towards the viewer. The angle between the triangles at time h is

$$\theta^h = \tan^{-1}\left(\frac{\sin\theta^h}{\cos\theta^h}\right) \ . \tag{3.18}$$

The sine and cosine can be expressed in terms of $\boldsymbol{n}_1^h$ and $\boldsymbol{n}_2^h$, the unit normal vectors to T_1 and T_2 at time h, and the length $\boldsymbol{l}^h$ of the edge e at time h through

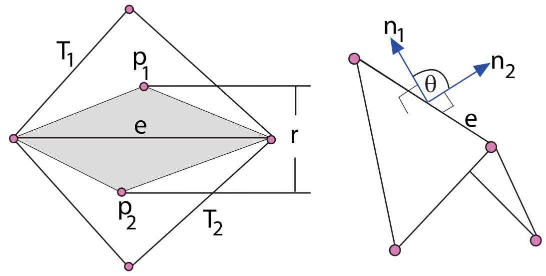


Figure 3.12: Components to calculate the mean curvature

the dot and cross products:

$$\begin{aligned} \sin\theta^h &= (\boldsymbol{n}_1^h \times \boldsymbol{n}_2^h) \cdot \frac{\boldsymbol{l}^h}{|\boldsymbol{l}^h|} \ , \quad \text{and} \\ \cos\theta^h &= \boldsymbol{n}_1^h \cdot \boldsymbol{n}_2^h \ . \end{aligned}$$

t_1 and t_2 denote the heights of triangles T_1 and T_2, measured from the shared edge e to the opposite vertex. The initial curvature span is the distance between the two triangles' centres of gravity in the initial position, which is $r = \frac{1}{6}(t_1 + t_2)$. The bending energy of edge e is

$$\boldsymbol{E}_{bend} = -K_{bend}\frac{|\boldsymbol{l}^0|}{r}(\theta^h - \theta^0)^2 \ . \qquad (3.19)$$

The same formulae described in equation (3.12) above is used to calculate the damping force exerted on particle $\boldsymbol{p}_{ij}$ through its interaction with particle $\boldsymbol{p}_{kl}$.

Integrating over time

Newmark implicit time integration is used for this model. This method gives the new position and velocity of particle $\boldsymbol{p}_{ij}$ at time $h + \Delta h$ as

$$\begin{aligned} \boldsymbol{p}'_{ij} &= \boldsymbol{p}_{ij} + \Delta h \dot{\boldsymbol{p}}_{ij} + \frac{\Delta h^2}{4}(\ddot{\boldsymbol{p}}_{ij} + \ddot{\boldsymbol{p}}'_{ij}) \ , \quad \text{and} \\ \dot{\boldsymbol{p}}'_{ij} &= \dot{\boldsymbol{p}}_{ij} + \frac{\Delta h}{2}(\ddot{\boldsymbol{p}}_{ij} + \ddot{\boldsymbol{p}}'_{ij}) \ . \end{aligned}$$

The acceleration at time $h + \Delta h$ is given by

$$\ddot{\boldsymbol{p}}'_{ij} = \frac{\boldsymbol{F}_{ij}}{m_{ij}} , \tag{3.20}$$

where m_{ij} is the mass of $\boldsymbol{p}_{ij}$.

3.4 Finite element method

The finite element method is a general technique for simulating any physical material (Bathe, 1995). The material is divided into a regular grid of small elements in three-dimensional space whose size determines the accuracy of the simulation. Such models are used to help design and develop products in many fields of engineering, particularly for investigating trade-offs between maximising stiffness and strength and minimising weight, materials and cost. For example, one might investigate how a car deforms under stress such as crash. In such situations, the finite element method allows entire designs to be constructed, refined, and optimised before any manufacturing takes place.

In the special case of page turning, the quasi-two-dimensional nature of paper means the page can be simulated using a two-dimensional rectangular mesh rather than a full three-dimensional grid, dividing it into $m \times n$ small patches, as shown in Figure 3.13. However, in the finite element method patches are not modelled as particles, as they were in the previous section: instead, their dimensions and shape are taken into account. The forces on a patch—including gravity, forces exerted

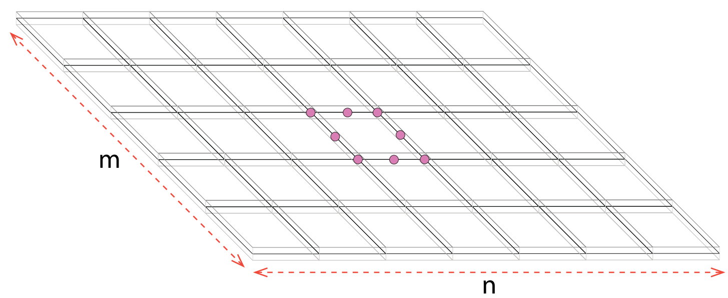


Figure 3.13: Dividing a sheet of paper into $m \times n$ elements

by neighbouring patches, and the force the reader applies to turn the page—produce stress which deforms the patch, and the finite element method calculates the amount of deformation. Note that although the grid is two-dimensional, the thickness of the paper is not ignored: each patch does have thickness, which varies across the patch.

Modelling paper as a thin shell, represented by a two-dimensional rectangular mesh of patches whose thickness varies, is a special case of the general finite element model of a three-dimensional solid, and is slightly easier to tackle than the general case (Bathe, 1995). The first task is to establish a geometric mapping between the patches themselves and the global coordinate system (Section 3.4.1). Ultimately this method considers the physical forces on each patch—the stress—and works out the distortion they cause—the strain. Strain is defined in terms of the spatial derivatives of a point's displacement, and calculating the stress-strain relationship involves differential equations. This requires a smooth, differentiable model of the patch; obtaining this is the second task (Section 3.4.2). It is necessary to model the patches not just as two-dimensional surfaces but as having thickness; the third task (Section 3.4.3). Then a three-dimensional coordinate system needs to be established at each reference point of the patches (Section 3.4.4), and the displacement of the reference points is expressed in terms of these coordinates (Section 3.4.5).

In terms of local coordinates the relationship between stress (Section 3.4.6) and strain (Section 3.4.7) is simple: it is given by a deformation matrix (Section 3.4.8) whose components depend on the Young's modulus, Poisson's ratio, and shear correction factor of the material under consideration. The relationship, which involves first derivatives of displacement, must be mapped back into global coordinates in order for it to be solved. All the forces on each element are considered, and finally the differential equations that have been obtained are integrated (Section 3.4.9).

3.4.1 Local and global coordinates

Figure 3.14 shows a general element (Bucalem and Bathe, 1993). It is defined by eight reference points, four representing its corners and four representing the

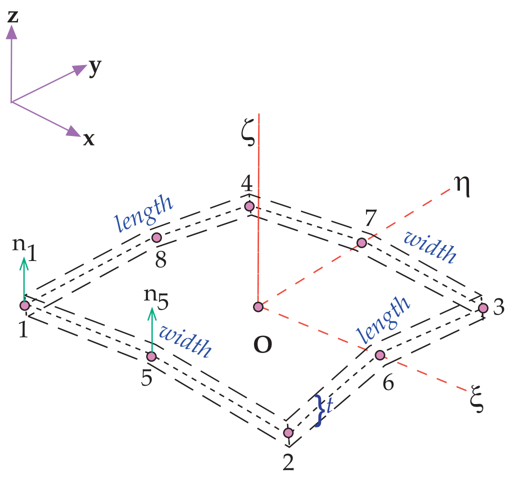


Figure 3.14: An element of the shell

mid-points of each side. In global (x, y, z) coordinates these reference points are $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$ and $\boldsymbol{p}_4$ for the first group and $\boldsymbol{p}_5, \boldsymbol{p}_6, \boldsymbol{p}_7$ and $\boldsymbol{p}_8$ for the second: each $\boldsymbol{p}_i$ has coordinates (x_i, y_i, z_i) in ordinary three-dimensional space. If thickness is modelled, the reference points are taken to be at the centre of the material and further parameters $t_1 \ldots t_8$ give the corresponding thicknesses, so that at each reference point the material of the paper extends from $-\frac{t_i}{2}$ to $+\frac{t_i}{2}$. The direction of this extension, $\boldsymbol{n}_i$, is normal to the paper at point $\boldsymbol{p}_i$.

Figure 3.15 shows the local coordinate system that is defined for each patch. Any point within the element is given by coordinates (ξ, η, ζ), where all coordinates lie between -1 and $+1$. ξ and η give the extent in directions corresponding to the width and height of the paper, respectively; while ζ is the direction perpendicular to the paper, and corresponds to its thickness. Reference point $\boldsymbol{p}_1$,

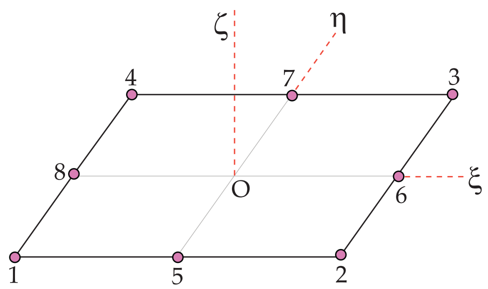


Figure 3.15: Midsurface of the shell in local coordinates

i	(ξ_i , η_i , ζ_i)	i	(ξ_i , η_i , ζ_i)
1	(-1 , -1 , 0)	5	(0 , -1 , 0)
2	(1 , -1 , 0)	6	(1 , 0 , 0)
3	(1 , 1 , 0)	7	(0 , 1 , 0)
4	(-1 , 1 , 0)	8	(-1 , 0 , 0)

Table 3.1: Local coordinates of the eight reference points

which lies in the midsurface of the paper, has local coordinates $\boldsymbol{\pi}_1 = (-1, -1, 0)$, while point $\boldsymbol{p}_3$ has local coordinates $\boldsymbol{\pi}_3 = (1, 1, 0)$. The upper surface of the paper at $\boldsymbol{p}_3$ is $(1, 1, 1)$, while the lower surface is $(1, 1, -1)$. Table 3.1 shows the local coordinates of the eight reference points.

Each patch is fully defined by the position in space of its eight reference points. The coherence of the assembly of patches is ensured by making the three points that define one side of a patch correspond to the three points that define the adjacent side of the neighbouring patch in the obvious manner, all the way along both the height and the width of the paper.

3.4.2 Defining a smooth surface

This section addresses the problem of how to define from this grid of reference points a smooth and continuous surface for each patch (Bucalem and Bathe, 1993). In fact, it is not necessary to fit a smooth surface in order to present the patch on the computer screen: for that purpose the discrete grid of reference points is entirely adequate. Rather, a smooth surface is needed because the analysis in subsequent subsections in terms of stress and strain involves spatial derivatives of the surface.

Given a point on the midsurface of a patch whose local coordinates are $\boldsymbol{\pi} = (\xi, \eta, 0)$, what is its global coordinate vector $\boldsymbol{p} = (x, y, z)$? The vector $\boldsymbol{p}$ will be expressed as a weighted sum of the vectors $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_8$ that represent the 8 reference points of the patch, where the weights $w_1(\xi, \eta), w_2(\xi, \eta), \ldots, w_8(\xi, \eta)$

reflect the position of the point $\boldsymbol{\pi}$ in local coordinate space:

$$\boldsymbol{p} = \sum_{i=1}^{8} w_i(\xi, \eta)\boldsymbol{p}_i \ . \tag{3.21}$$

For example, if $\boldsymbol{\pi}$ corresponds with one of the reference points, the weights will be 1 for that reference point and 0 for the other seven points.

If the patch were planar this would be a simple matter of linear interpolation. However, in general the points $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_8$ do not lie on a plane in three-dimensional space. To account for this, it is necessary to define eight non-linear interpolation functions $w_i(\xi, \eta)$ with the following properties:

1. $\sum_{i=1}^{8} w_i(\xi, \eta) = 1$ for any values ξ, $\eta \in [-1, 1]$
2. $w_i(\xi_i, \eta_i) = 1$ for each reference point, $i = 1, 2, \ldots, 8$
3. $w_i(\xi_j, \eta_j) = 0$ if $j \neq i$.

It turns out that while linear interpolation is inadequate to model a non-planar patch, quadratic weighting functions can be constructed that satisfy these properties. Figure 3.16(a) shows a suitable function $w_1(\xi, \eta)$ for reference point 1 of the patch: as can be seen, it satisfies properties 2 and 3 with $i = 1$. This function can be constructed by taking the component shown in Figure 3.16(b) and subtracting the components in Figure 3.16(c) and 3.16(d). Figure 3.16(b) is a bilinear surface that satisfies property 2 at reference point 1, and satisfies property 3 at points 2, 3, 4, 6 and 7 but but violates it at points 5 and 8; Figure 3.16(c) is a parabolic correction that restores property 3 at point 5; and Figure 3.16(d) does the same for point 8.

The bilinear function in Figure 3.16(b) has the equation

$$\frac{1}{4}(1 - \xi)(1 - \eta) \ .$$

This alters the function's value for reference point $\boldsymbol{\pi}_1 = (-1, -1, 0)$ as desired, and evaluates to 0 at any reference points with either $\xi = 1$ or $\eta = 1$—that is, points $\boldsymbol{\pi}_2, \boldsymbol{\pi}_6, \boldsymbol{\pi}_3, \boldsymbol{\pi}_7, \boldsymbol{\pi}_4$. It also has the undesired effect of evaluating to 0.5

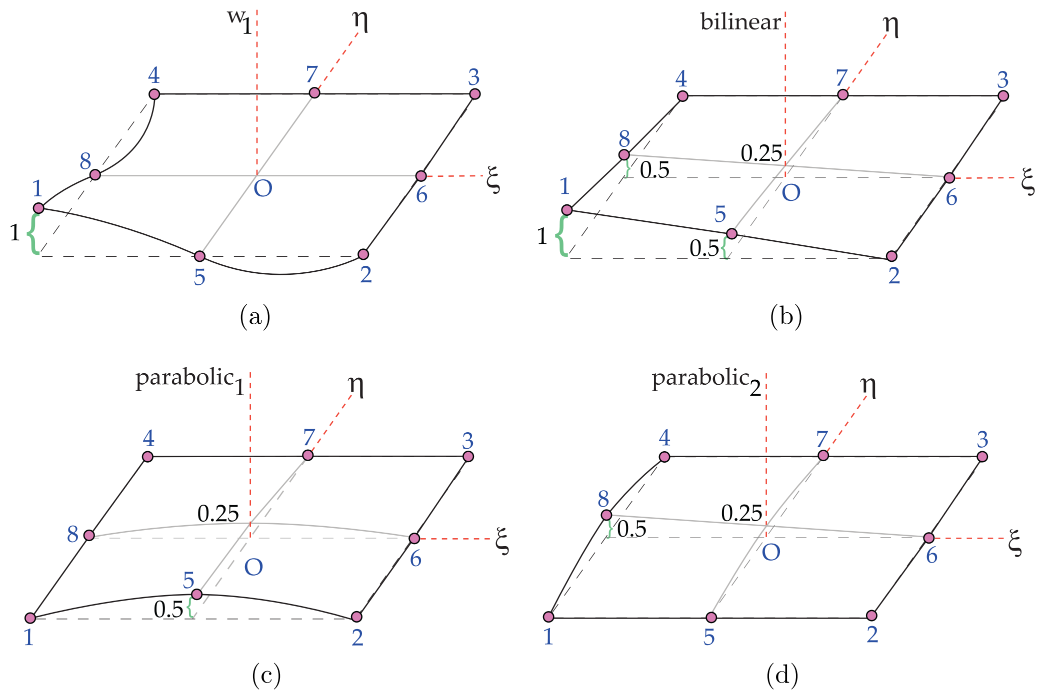


Figure 3.16: Creating (a) the interpolation function w_1 by adding three functions: (b) *bilinear*, (c) *parabolic*$_1$, and (d) *parabolic*$_2$

at points $\boldsymbol{\pi}_5$ and $\boldsymbol{\pi}_8$, as shown. The parabolic functions in Figures 3.16(c) and 3.16(d) have the equations

$$\frac{1}{4}(1-\xi^2)(1-\eta) \quad \text{and} \quad \frac{1}{4}(1-\xi)(1-\eta^2)\ .$$

These both evaluate to 0 at points $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_6, \boldsymbol{\pi}_3, \boldsymbol{\pi}_7$ and $\boldsymbol{\pi}_4$, and 0.5 at $\boldsymbol{\pi}_5$ and $\boldsymbol{\pi}_8$ respectively. Thus the overall effect of raising to 1 the weighting function's value at reference point $\boldsymbol{\pi}_1$, shown in Figure 3.16(a), is accomplished by,

$$\begin{aligned} w_1(\xi,\eta) &= \mathit{bilinear} - \mathit{parabolic}_1 - \mathit{parabolic}_2 \\ &= \frac{1}{4}(1-\xi)(1-\eta)(-1-\xi-\eta)\ . \end{aligned}$$

A weighting functions for the other three corners $\boldsymbol{\pi}_2, \boldsymbol{\pi}_3$, and $\boldsymbol{\pi}_4$ are accomplished in similar manner.

The weighting function for the mid-point of a side is simpler, requiring only a

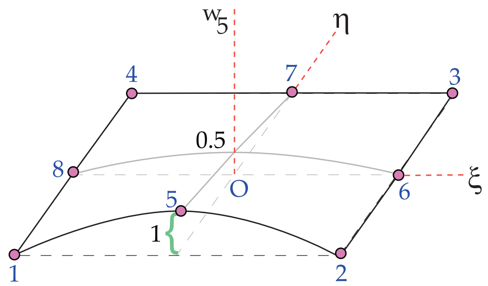


Figure 3.17: Interpolation function w_5

single parabolic model as shown in Figure 3.17 for $\boldsymbol{\pi}_5$.

In summary, the weighting functions that have the value of 1 at a reference point and 0 at the other seven reference points are

$$w_i(\xi,\eta) = \begin{cases} \frac{1}{4}(1+\xi\xi_i)(1+\eta\eta_i)(-1+\xi\xi_i+\eta\eta_i) & i \in [1,4] \\ \frac{1}{2}(1-\xi^2)(1+\eta\eta_i) & i \in \{5,7\} \\ \frac{1}{2}(1+\xi\xi_i)(1-\eta^2) & i \in \{6,8\} \end{cases} \tag{3.22}$$

Not only do they satisfy properties 2 and 3 above, but it can be shown algebraically that the eight functions sum to 1 for any values of ξ, η, thus satisfying property 1 too. Furthermore, the values of all eight interpolation functions along any edge are defined purely by the three reference points that constitute that edge; thus (x, y, z) coordinates of any point along an edge will be the same as for the neighbouring patch, making the entire surface continuous. However, the first derivative across an edge is *not* continuous, which means that creases in the paper such as dog-eared pages—or even a crumpled ball of paper—can be modelled.

3.4.3 Inside the shell

The discussion so far has focused on points on the midsurface of the shell. To deal with general points inside the shell, $\boldsymbol{\pi} = (\xi, \eta, \zeta)$ with $\zeta \neq 0$, vectors normal to the shell at each of the reference points need to be established, as shown in Figure 3.18.

In the beginning, each element of the shell is represented by the coordinates of the top and bottom surfaces at each of the eight reference positions, rather than the reference points themselves. Then, at each reference position, the actual reference point is calculated by averaging the coordinates of the top and bottom surfaces, and the direction of the normal vector is determined by taking the difference between the top and bottom points and normalising to a unit vector, as illustrated in Figure 3.18. As the configuration of the shell evolves in time the direction of each of the eight normal vectors is updated as described in the next subsection.

Given the normal vector, an arbitrary point $\boldsymbol{\pi} = (\xi, \eta, \zeta)$ inside the shell is transformed to $\boldsymbol{p}$ in (x, y, z) coordinate as follows:

$$\boldsymbol{p} = \sum_{i=1}^{8} w_i(\xi, \eta) \left(\boldsymbol{p}_i + \zeta \frac{t_i}{2} \boldsymbol{n}_i \right) \ . \tag{3.23}$$

The first component inside the bracket, $\boldsymbol{p}_i$, corresponds to equation (3.21) above representing the midsurface of the shell. The second component, involving ζ, translates the displacement along the ζ axis back into (x, y, z) coordinates.

3.4.4 The normal-vector coordinate system

The normal vectors will play a crucial part in this model. For example, in Section 3.4.6 it is assumed that the material of the paper is completely flexible and offers no resistance to external forces in the direction perpendicular to it. Because of this, and to assist in updating the normal vectors from one configuration of the paper to the next, a coordinate system at each reference point is established based on the direction of the normal at that point.

The normal-vector coordinate system at reference point i is denoted by $\hat{x}_i, \hat{y}_i, \hat{z}_i$. The new z axis is in the direction of the normal to the surface at that reference point, $\boldsymbol{n}_i$. The new x axis is in a direction perpendicular to both this new z axis and the *old* y axis, that is, the y axis of the original global x, y, z coordinates. This direction is $\boldsymbol{e}^y \times \boldsymbol{e}_i^{\hat{z}}$, where $\boldsymbol{e}^y$ is the unit vector along the global y axis and $\boldsymbol{e}_i^{\hat{z}}$ is the unit vector along the new z axis, which is in the same direction as $\boldsymbol{n}_i$. (A degenerate case arises when the normal $\boldsymbol{n}_i$ coincides with the direction of the global y axis, in which case the new x axis is set to be in the direction of the

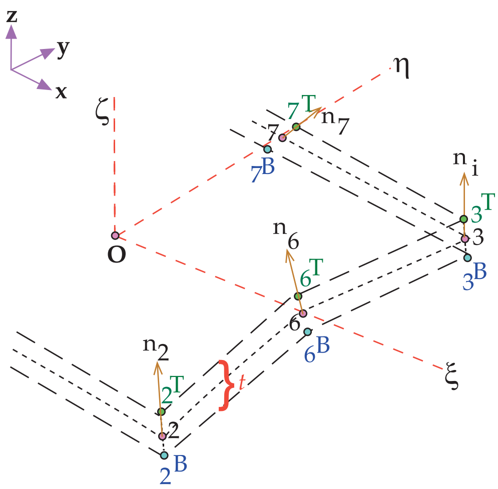


Figure 3.18: Normal vector of the shell's midsurface

old z axis.) Having established the new z and x axes, the new y axis is set to be perpendicular to them both, that is, in the direction $\boldsymbol{e}_i^{\hat{z}} \times \boldsymbol{e}_i^{\hat{x}}$.

To update the normal vector $\boldsymbol{n}_i$ from one moment to the next, it is assumed that the new normal is the result of rotating $\boldsymbol{n}_i$ through an angle α_i around the $\hat{x}_i$ axis and an angle β_i around the $\hat{y}_i$ axis. The new normal vector is

$$\begin{aligned} \boldsymbol{n}'_i &= \boldsymbol{e}_i^{\hat{x}} \sin\beta_i + \boldsymbol{e}_i^{\hat{y}} \cos\beta_i \sin(-\alpha_i) + \boldsymbol{e}_i^{\hat{z}} \cos\beta_i \cos(-\alpha_i) \\ &= \beta_i \boldsymbol{e}_i^{\hat{x}} - \alpha_i \boldsymbol{e}_i^{\hat{y}} + \boldsymbol{e}_i^{\hat{z}} \ , \end{aligned} \tag{3.24}$$

assuming that the rotations are small. $\boldsymbol{n}'_i$ will be the $\hat{z}$ direction of the normal-vector coordinate system that is used for the next step.

The angles α_i and β_i are central to the updating procedure, and will figure prominently in what follows.

3.4.5 Displacement

If the position of point $\boldsymbol{p}$ at time h is $\boldsymbol{p}$, and at time $h+\Delta h$ is $\boldsymbol{p}'$, its displacement from time h to time $h+\Delta h$ in the x, y and z directions is

$$\boldsymbol{d} = \boldsymbol{p}' - \boldsymbol{p} \ .$$

By substituting the value of $\boldsymbol{p}$ according to equation (3.23),

$$\boldsymbol{d} = \sum_{i=1}^{8} w_i(\xi,\eta)\left\{(\boldsymbol{p}_i' - \boldsymbol{p}_i) + \zeta\frac{t_i}{2}(\boldsymbol{n}_i' - \boldsymbol{n}_i)\right\} .$$

If $\boldsymbol{d} = (u, v, w)$, then the entire displacement of node i from time h to $h + \Delta h$ is characterised by the following 5-vector:

$$\boldsymbol{q}_i = (u_i, v_i, w_i, \alpha_i, \beta_i) ,$$

where u_i, v_i, and w_i are the displacement of $\boldsymbol{p}_i$ in the x, y and z axis directions respectively, and α_i and β_i are the rotation angles around the $\hat{x}$ and $\hat{y}$ axes.

Given the displacement vectors $\boldsymbol{q}_i$ for each of the nodes of a surface patch, the displacement vector $\boldsymbol{d}$ for a general point on the patch at position (ξ, η, ζ) is

$$\begin{aligned}\boldsymbol{d} &= \sum_{i=1}^{8} w_i(\xi,\eta)\left\{(\boldsymbol{p}_i' - \boldsymbol{p}_i) + \zeta\frac{t_i}{2}(\boldsymbol{n}_i' - \boldsymbol{n}_i)\right\} \\ &= \sum_{i=1}^{8} w_i(\xi,\eta)\left\{\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} + \zeta\frac{t_i}{2}(\beta_i \boldsymbol{e}_i^{\hat{x}} - \alpha_i \boldsymbol{e}_i^{\hat{y}})\right\} \\ &= \sum_{i=1}^{8} \boldsymbol{w}_i \boldsymbol{q}_i , \end{aligned} \tag{3.25}$$

where the interpolation matrix for reference point $\boldsymbol{p}_i$ is

$$\boldsymbol{w}_i = w_i(\xi,\eta)\begin{bmatrix} 1 & 0 & 0 & -\zeta\frac{t_i}{2}l_i^{\hat{y}} & \zeta\frac{t_i}{2}l_i^{\hat{x}} \\ 0 & 1 & 0 & -\zeta\frac{t_i}{2}m_i^{\hat{y}} & \zeta\frac{t_i}{2}m_i^{\hat{x}} \\ 0 & 0 & 1 & -\zeta\frac{t_i}{2}n_i^{\hat{y}} & \zeta\frac{t_i}{2}n_i^{\hat{x}} \end{bmatrix}$$

and $\boldsymbol{e}_i^{\hat{x}} = (l_i^{\hat{x}}, m_i^{\hat{x}}, n_i^{\hat{x}})$, $\boldsymbol{e}_i^{\hat{y}} = (l_i^{\hat{y}}, m_i^{\hat{y}}, n_i^{\hat{y}})$.

Now the geometry analysis of the surface element or patch has been completed. To summarise: each patch is situated in three-dimensional space by the coordinates of its eight reference points $\boldsymbol{p}_i$. A local coordinate system is defined for the patch which allows any point within it to be defined by coordinates (ξ, η, ζ). The entire volume of the patch is swept out as these coordinates range from -1 to 1.

The third coordinate corresponds to the thickness of the patch, and $\zeta = 0$ defines the midsurface. Eight of the nine combinations of the values -1, 0, 1 for each of ξ and ζ define the eight reference points $\boldsymbol{p}_i$ (the ninth, $(0, 0)$, is the centre of the patch). Any point $\boldsymbol{\pi} = (\xi, \eta, \zeta)$ within the patch can be expressed in terms of its global coordinates $\boldsymbol{p} = (x, y, z)$ using the transformation of equation (3.23). This transformation also involves the normal vector $\boldsymbol{n}_i$ to the patch at each reference point.

The normal vector $\boldsymbol{n}_i$ is used to define a coordinate system at reference point i which will be deployed in the analysis below. As the patch moves from one configuration to the next under the influence of forces discussed in subsequent sections, it is necessary to update the coordinates of each reference point and the orientation of the normal vector at each one. The key to the updating procedure is the displacement vector, a five-dimensional vector $\boldsymbol{q}_i$ which includes the displacement of the reference point in the x, y, and z directions as well as the angular displacements around the two axes of the normal-vector coordinate system that are perpendicular to the normal at the reference point. Given any point within the patch expressed as (ξ, η, ζ) in local coordinates, and the five-dimensional displacement vectors $\boldsymbol{q}_i$ for each reference point, equation (3.25) shows how to calculate the corresponding displacement of the point in global (x, y, z) coordinates. These displacements occur as the result of the forces that the next section discusses.

3.4.6 Stress

Stress ($\boldsymbol{\sigma}$) measures the distribution of the internal forces in the shell, and determines the material's resistance to deformation under the pressure of external forces ($\boldsymbol{b}$). The stress at any point within the shell—and here a general point $\boldsymbol{p}$ within a surface element is considered, not a reference point $\boldsymbol{p}_i$—is defined as the force per unit area over a small area (ΔA) of the body. Formally,

$$\boldsymbol{\sigma} = \lim_{\Delta A \to 0} \frac{\boldsymbol{\Delta b}}{\Delta A} .$$

Stress can be resolved into normal stress and shear stress (Mindlin, 1951; Reissner, 1945). The former measures the force perpendicular to the surface, while the latter measures the forces that act parallel to the surface. For example, σ_x is a stress perpendicular to the yz plane, while σ_{xy} and σ_{xz} are stresses in the yz plane that are parallel to the y and z axes respectively. A total of six stresses act at every point within the shell, and they are expressed as a column vector $\boldsymbol{\sigma} = [\sigma_x, \sigma_y, \sigma_z, \sigma_{xy}, \sigma_{xz}, \sigma_{yz}]^T$.

The effect of stress is to deform the body under consideration. The amount of deformation is called *strain*, and is expressed as the change in length per unit of the original length. Each of the six stresses at a point (more precisely, at a vanishingly small area) cause a corresponding amount of strain, where strain is a six-dimensional vector that corresponds to the spatial derivatives of the point's displacement $\boldsymbol{d} = \boldsymbol{p}' - \boldsymbol{p}$ discussed earlier:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \epsilon_{xy} \\ \epsilon_{xz} \\ \epsilon_{yz} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix} \boldsymbol{d} \ .$$

The relationship between the degree of stress and the amount of strain it causes is given by Hooke's law of elasticity, which states that the amount by which a body is deformed—the strain—is directly proportional to the stress that cause the deformation. The stress-strain relation can be expressed by the equation

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\epsilon} \ , \tag{3.26}$$

where $\boldsymbol{D}$ is the deformation matrix, discussed below. First, however, let us consider the problem of calculating the strain itself.

3.4.7 Calculating the strain

The strain is obtained by taking partial derivatives of $\boldsymbol{d}$ with respect to the global coordinates x, y and z, as shown above. Equation (3.25) expresses the displacement at any point within a patch in terms of its local (ξ, η, ζ) coordinates and the eight vectors q_i that characterise the displacement of each reference point of the patch. To take the derivatives of this equation with respect to ξ, η and ζ is a simple matter. In order to calculate the strain, partial derivatives with respect to x, y, and z must be expressed in terms of partial derivatives with respect to ξ, η and ζ.

This is done by applying the chain rule in the following form:

$$\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} = \boldsymbol{J}^{-1} \begin{bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \zeta} \end{bmatrix} , \tag{3.27}$$

where $\boldsymbol{J}^{-1}$ is the inverse of the Jacobian matrix that relates local coordinate derivatives to global ones:

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} . \tag{3.28}$$

$\boldsymbol{J}$ can be determined by taking derivatives of equation (3.23) above, which gives an explicit transformation from local coordinates to global ones. The details can be found in Appendix A.

The upshot is that the strain $\boldsymbol{\epsilon}$ at any point can be obtained from its local coordinates ξ, η and ζ and the displacement vector $\boldsymbol{q}_i$ at each of the patch's eight reference points. The operation can be summarised in matrix form as

$$\boldsymbol{\epsilon} = \sum_{i=1}^{8} \boldsymbol{B}_i(\xi, \eta, \zeta)\boldsymbol{q}_i , \tag{3.29}$$

where $\boldsymbol{B}_i(\xi, \eta, \zeta)$, the *strain-displacement matrix*, can be obtained explicitly using the above techniques, for each reference point i.

3.4.8 The deformation matrix

If a body is compressed it tends to expand sideways; on the other hand, if it is pulled the material contracts laterally (Mindlin, 1951; Reissner, 1945). These effects are quantified by the deformation matrix $\boldsymbol{D}$ that characterises the relationship between stress and strain expressed by equation (3.26).

The deformation matrix depends on three constants that characterise the material under consideration: Young's modulus of elasticity E, Poisson's ratio ν, and the shear correction factor K. Young's modulus represents the stiffness of the material. Poisson's ratio captures the relation between deformation in the direction of the stress and deformations in directions perpendicular to it (this thesis assumes that the material is isotropic).

The shear correction factor is used to model a shear strain that is not uniform throughout the material, and allows the strain to alter linearly across its thickness. Typically the values of $E = 1,370\ N\ cm^{-1}$, $\nu = 0.33$ and $K = \frac{5}{6}$ are used for paper with mass density of 80 m^{-3} (Bathe, 1995).

Assuming that the shell does not resist any deformation in the $\hat{z}$ direction, the stress normal to its surface is zero. In the normal-vector coordinate system, the deformation matrix at point $\boldsymbol{\pi} = (\xi, \eta, \zeta)$ is

$$\widehat{\boldsymbol{D}} = \begin{bmatrix} \frac{E}{1-\nu^2} & \frac{E\nu}{1-\nu^2} & 0 & 0 & 0 & 0 \\ \frac{E\nu}{1-\nu^2} & \frac{E}{1-\nu^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{E}{2(1+\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{EK}{2(1+\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{EK}{2(1+\nu)} \end{bmatrix}. \tag{3.30}$$

This matrix must be transformed to give a deformation matrix $\boldsymbol{D}$ in the global coordinate system. The elements of the transformation matrix $\boldsymbol{Q}$ are obtained from the direction cosines of the $\hat{x}, \hat{y}, \hat{z}$ coordinate axes at point $\boldsymbol{p}$ measured in the x, y, z coordinate directions. $\boldsymbol{Q}$ transforms the strain vector in local coordinates $[\epsilon_\xi, \epsilon_\eta,\ \epsilon_\zeta, \epsilon_{\xi\eta}, \epsilon_{\xi\zeta}, \epsilon_{\eta\zeta}]^T$ into the strain vector in global coordinates $[\epsilon_x, \epsilon_y, \epsilon_z, \epsilon_{xy}, \epsilon_{xz}, \epsilon_{yz}]^T$. Again, the details can be found in Appendix A.

3.4.9 Effect of forces

The deformation of any element is determined by the external and internal forces that work on it. By using the principle of conservation of energy, the forces can be translated into the displacement of the element at time $h + \Delta h$. This section explains the principles, the details are given in Appendix A.

When external forces are applied to a body, their potential energy (δE_T) is converted into kinetic energy (δE_K), corresponding to internal forces of inertia and damping, and potential energy (δE_P), corresponding to stress. Conservation of energy is expressed by equating internal and external energy:

$$\delta E_K + \delta E_P = \delta E_T \ . \tag{3.31}$$

There are three different types of external forces to which each element of the shell is subjected. *Body forces* ($\boldsymbol{F}_B$)—such as the force of gravity—apply throughout the element. *Surface forces* ($\boldsymbol{F}_S$) may be applied at the surface of the shell—for example, the upper and lower surfaces of the left edge of the paper are held rigidly in place by forces exerted by the book's spine. *Nodal forces* ($\boldsymbol{R}_N$) are applied at individual reference points, for example by the user's thumb and forefinger on the corner of the page as it is turned. These three examples—gravity, the spine, and the user's effort in turning the page—are the only external forces applied during the page turning simulation required for this thesis, although the model permits more general body, surface, and nodal forces to be applied.

Potential energy equals force times displacement. Each force must be integrated appropriately. The body force is integrated with respect to the element's volume, yielding $\boldsymbol{R}_B$. The surface force is integrated with respect to its surface area, yielding $\boldsymbol{R}_S$. Both these forces, along with $\boldsymbol{R}_N$, are multiplied by the displacement to compute the energy.

The potential energy done by external forces is calculated by multiplying each vector by the displacement vector and summing:

$$\delta E_T = (\boldsymbol{\delta q})^T (\boldsymbol{R}_B + \boldsymbol{R}_S + \boldsymbol{R}_N) \ .$$

There are three types of internal forces. *Inertial force* is caused by the element's reluctance to accelerate or deaccelerate. In accordance with Newton's second law, it is given by $\rho \boldsymbol{w}\ddot{\boldsymbol{q}}'$, where ρ is the element's mass density. *Damping force* results from friction within the deforming material, and is proportional to the element's velocity: $\kappa \boldsymbol{w}\dot{\boldsymbol{q}}'$ where κ is the element's damping coefficient. *Stress* measures the material's resistance to deformation due to the external forces and the element's initial configuration, and can be written as $\boldsymbol{DBq}' + \boldsymbol{\sigma}_0$. All three effects must be integrated over the volume of the element. This gives

$$\delta E_K + \delta E_P = (\boldsymbol{\delta q})^T(\boldsymbol{M}\ddot{\boldsymbol{q}}' + \boldsymbol{C}\dot{\boldsymbol{q}}' + \boldsymbol{K}\boldsymbol{q}' + \boldsymbol{S}_0) \ .$$

The matrices $\boldsymbol{M}, \boldsymbol{C}$ and $\boldsymbol{K}$ are the element's mass, damping and stiffness matrices, $\boldsymbol{S}_0$ is the integrated initial stress $\boldsymbol{\sigma}_0$. If the initial position of the paper is flat, $\boldsymbol{\sigma}_0$ can be omitted altogether. For a more general case, where the paper might already be bent, $\boldsymbol{\sigma}_0$ is not zero and is calculated in the same way as $\boldsymbol{\sigma}$.

The law of conservation of energy is expressed by equating δE_T with the sum of δE_K and δE_P. Cancelling $(\boldsymbol{\delta q})^T$ and moving $\boldsymbol{S}_0$ to the other side yields the second-order differential equation

$$\boldsymbol{M}\ddot{\boldsymbol{q}}' + \boldsymbol{C}\dot{\boldsymbol{q}}' + \boldsymbol{K}\boldsymbol{q}' = \boldsymbol{R}_B + \boldsymbol{R}_S + \boldsymbol{R}_N - \boldsymbol{S}_0 \ . \tag{3.32}$$

This is the fundamental equation that must be solved in order to determine how the position of the element evolves under the applied forces.

This equation is solved using Newmark implicit time integration (Bathe, 1995; Newmark, 1959). Given the position $\boldsymbol{q}$, velocity $\dot{\boldsymbol{q}}$ and acceleration $\ddot{\boldsymbol{q}}$ of the element at time h, the new values $\dot{\boldsymbol{q}}', \ddot{\boldsymbol{q}}'$ at time $h + \Delta h$ are

$$\begin{aligned} \ddot{\boldsymbol{q}}' &= \frac{4}{\Delta h^2}(\boldsymbol{q}' - \boldsymbol{q}) - \frac{4}{\Delta h}\dot{\boldsymbol{q}} - \ddot{\boldsymbol{q}} \ , \quad \text{and} \\ \dot{\boldsymbol{q}}' &= \dot{\boldsymbol{q}} + \int_h^{h+\Delta h} \ddot{\boldsymbol{q}} \, dh \ . \end{aligned} \tag{3.33}$$

The nodal displacement at time $h+\Delta h$, $\boldsymbol{q}'$, can be calculated based only on terms at time h by substituting the value of $\dot{\boldsymbol{q}}', \ddot{\boldsymbol{q}}'$ from equation (3.33) into equation

(3.32), and bringing all other terms to the right-hand side.

$$\boldsymbol{q}'\left[\frac{4}{\Delta h^2}\boldsymbol{M}+\frac{2}{\Delta h}\boldsymbol{C}+\boldsymbol{K}\right] = \boldsymbol{M}\left(\frac{4}{\Delta h^2}\boldsymbol{q}+\frac{4}{\Delta h}\dot{\boldsymbol{q}}+\ddot{\boldsymbol{q}}\right)+\boldsymbol{C}\left(\frac{2}{\Delta h}\boldsymbol{q}+\dot{\boldsymbol{q}}\right)+ \boldsymbol{R}_B+\boldsymbol{R}_S+\boldsymbol{R}_N-\boldsymbol{S}_0\ . \quad (3.34)$$

To summarise the entire algorithm to simulate the behaviour of the element:

1. Initialise $\boldsymbol{q}^0 = \dot{\boldsymbol{q}}^0 = \ddot{\boldsymbol{q}}^0 = 0$
2. Select the time step Δh
3. Form the matrices $\boldsymbol{M}, \boldsymbol{C}, \boldsymbol{K}, \boldsymbol{S}_0$.
4. For each time step:
   (a) Calculate the load $\boldsymbol{R}'_B, \boldsymbol{R}'_S$.
   (b) Calculate the displacement $\boldsymbol{q}'$ at time $h + \Delta h$ using equation (3.34).
   (c) Calculate the acceleration $\dot{\boldsymbol{q}}'$ and velocity $\ddot{\boldsymbol{q}}'$ at time $h + \Delta h$ from equation (3.33).

3.5 Comparison of methods

Table 3.2 shows the essential features of the page turning techniques that have been surveyed in this chapter. It contains entries for the two variants of the *peeling* technique, basic peeling and the use of a 3D correction to simulate foreshortening of the paper due to its extension in the z direction; two *geometric wrapping* techniques that involve wrapping the paper around a cylinder and cone respectively; the basic *mass-spring model* and two variants that attempt to simulate more natural bending and twisting respectively; and the *finite element* method, which models the paper as a 3D shell whose thickness is represented by a single layer of elements. The measured time and space complexity reported here are the worst time and space complexity for computing the page turning simulation. It does not include the time and memory used by the 3D Java software to draw the appearance of the turned page.

3.5.1 Time complexity

The computation time required for each method is largely dictated by the size of the grid, where applicable. Most methods use rectangular grids. However, the first method needs no grid, and the third mass-spring variant uses a triangular grid in order to resist twisting by constraining dihedral angles between neighbouring triangles. Grid elements in the finite element method are three-dimensional, but are arranged in a two-dimensional array because paper is thin enough for a single layer of elements to adequately model its substance.

A complete page turn with each method is simulated. It involves many intermediate postures between the beginning, where the page lies flat on the right-hand side, and the end, where it lies flat on the left-hand side. For the first four methods, which do not involve iteration, Table 3.2 shows the average time taken to compute the configuration of the paper at each posture. For the last four methods, it shows the average number of iterations to reach a posture, and the average execution time per iteration. The execution time taken by these methods to compute the configuration of the paper at each posture is the product of these two numbers. The time taken to completely turn a page can be calculated by multiplying this product by the number of in-between postures—say 20 in-betweens for a fast $\frac{2}{3}$-second page turn animated at 30 frames/second.

In each case the lower right-hand corner of the page was constrained to move along the same arc. Each point is specified as a position on the grid. The number of possible positions with different x-values is twice the grid size. Times are shown for grids of size 10×10 (18 postures between the paper lying flat on the right side and flat on the left side) and 100×100 (198 in-between postures). The smaller grid gives a rough idea of what the page turn looks like, while the larger one produces a very good visual approximation of the page turn. In practice, one would not use a square grid but tailor it to the aspect ratio of the paper. For instance, using an 8×11 or 9×11 grid for US letter paper. Implementations of each page turning model described in this chapter are not optimised and could no doubt be improved by more careful attention to computational details. The code was written in Java, and timings were measured on a 2.8 GHz Intel Pentium 4 processor.

Technique	Method	Mesh	Time integration	Iteration/position		Time/iteration (secs)		Time complexity
				10×10	100×100	10×10	100×100	
Peeling								
two dimension	produce dog-ear effect by simple geometric reflection	-	-	1	1	5×10^{-5}	5×10^{-5}	$O(1)$
with 3D correction	adjust dog-ear with heuristic z-value and spline correction	rectangle	-	1	1	0.016	5.96	$O(n^2)$
Geometric wrapping								
around cylinder	map grid to cylinder whose radius and axis vary	rectangle	-	1	1	0.001	0.31	$O(n)$
around cone	map grid to cone whose angle and apex vary	rectangle	-	1	1	0.016	6.44	$O(n^2)$
Mass-spring model								
basic	calculate spring lengths from Hooke's law and apply limit straining	rectangle	explicit Euler	349	27	0.016	6.38	$O(n^2)$
with bending	apply Hooke's law for elongation, and calculate curvature for compression	rectangle	semi-implicit backward difference	386	30	0.015	6.38	$O(n^2)$
bending, no twisting	calculate dihedral angle between neighboring triangles for bending energy	triangle	implicit Newmark	450	33	0.017	6.94	$O(n^2)$
Finite element								
3D shell model	use stress-strain relationship to derive the deformation from applied forces	3D patch	implicit Newmark	500	-	6.1	-	$O(n^4)$

Table 3.2: Summary of page turning techniques

The two-dimensional peeling technique does not use a grid and is essentially instantaneous; it takes 50 μs to compute a posture. The second method, which incorporates a correction for foreshortening in the z direction, does require a grid, and this increases computation time from $O(1)$ to $O(n^2)$ for an $n \times n$ grid. Table 3.2 gives values of 16 ms for the 10×10 grid and 5.96 s for the 100×100 grid, which are indeed in the order of 100 and 10,000 times the values for the $O(1)$ peeling technique (to within a factor of 3 and 12 respectively).

The time taken by geometric wrapping is $O(n)$ for the cylinder and $O(n^2)$ for the cone. The difference is because in the former case each element of the paper has the same position for all values of z, while in the latter the position depends on the z-value, as described in Section 3.2. The time taken for the cylinder is 1 ms for the 10×10 grid and 312 ms for the 100×100 grid; the increase is rather more than the factor of 10 that is expected for an $O(n)$ method but does include the time required to output the coordinates of all the points, which adds a small $O(n^2)$ component. The time taken for the cone is 16 ms for the 10×10 grid and 6.44 s for the 100×100 grid.

For the physical simulation models in the lower half of the Table 3.2, stability depends on the value chosen for the integration time step. If this is too large the simulation will be unstable—the paper may vibrate, or, worse still, the computation may blow up and produce infinite values. Although the explicit time integration technique used for the first of these four physical models is the simplest method, it is less stable than the implicit methods that are used for the remaining three, and therefore requires smaller time steps. However, in order to compare the execution time based on the complexity of the model itself, the same time step value of $\Delta h = 20$ ms is used, which works well for all physical models.

For each method, the number of iterations required for the 100×100 grid is far smaller than for the 10×10 one because the previous posture of the page, which is used as a starting point for iteration, is far closer to the current posture on the finer grid. Recall that 18 in-between postures are computed for the coarse grid and 198 for the fine grid.

Table 3.2 gives the *average* number of iterations. In physical models the number of iterations required to reach a specified posture varies. More accurate models simulate the stiffness properties of the paper better, which means that there is little change from one iteration to the next. The largest change from one iteration to the next occurs when force is first applied to the page, and therefore only a small number of iterations are needed to reach the first given posture. When the page approaches its equilibrium posture near the end of the simulation, the page deformation reduces, the amount of change in each iteration decreases, and more iterations are required to reach a posture. In practice, the number of iterations ranges from about 200 steps below the average to 300 steps above it for the 10×10 grid, and from about 10 steps below to 20 steps above for the 100×100 grid.

In terms of the grid size, the complexity of the mass-spring models is $O(n^2)$. For all three of them, the time taken for a complete 20-frame page turn animation is about $400 \times 16 \times 20$ ms ≈ 2 min on a 10×10 grid, and about $30 \times 6.5 \times 20$ s ≈ 1 hour on a 100×100 one.

The finite element method is far more complex, and grows with the fourth power of the grid size. A complete 20-frame page turn animation takes $500 \times 6.1 \times 20$ ms ≈ 17 hours even on the crude 10×10 grid, and the simulation for the larger 100×100 grid was not performed because it would have taken in the order of 10^4 times as long.

Note that in a practical implementation many speed-ups are possible. The integration step size could be carefully tuned. Moreover, one could cache the results of a single page turn (or, perhaps, a selection of different page turns) and texture-map the actual contents of each page onto the cached geometric surface. For example, Chu et al. (2003) used this technique for their page turning simulation.

3.5.2 Space complexity

Table 3.3 shows the space complexity of the techniques, along with the vector representation they use. In all cases the initial positions of the four corners of the page must be stored.

Geometric models define the appearance of the page by a set of equations, each relating to a particular user action. This chapter only considers the case where a user picks the bottom corner of the page and turns it from right to left. The page's appearance at any point is obtained by entering the current position

Technique	Stored vector	Space complexity
Peeling		
two dimension	(x, y)	$O(1)$
with 3D correction	(x, y, z)	$O(1)$
Geometric wrapping		
around cylinder	(x, y, z)	$O(1)$
around cone	(x, y, z)	$O(1)$
Mass-spring model		
basic	(x, y, z)	$O(n^2)$
with bending	(x, y, z)	$O(n^2)$
bending, no twisting	(x, y, z)	$O(n^2)$
Finite element		
3D shell model	(u, v, w, α, β)	$O(n^2)$

Table 3.3: Space complexity of page turning models

of the bottom corner of the page into the appropriate geometric equation. Thus the space complexity is $O(1)$.

In physical models, the paper is simulated by dividing its material into $m \times n$ elements, and the new position of an element in the grid at time $h + \Delta h$ is calculated from the element's current position at time h. In the mass-spring model, each element is treated as a particle whose position is given by a 3-vector (x, y, z). At each time step its position at time h must be stored; thus the worst space complexity is $O(n^2)$.

In the finite element method the size and shape of the elements are taken into account. Each element is defined by eight reference points, and at each time step the displacements of all reference points at time h are stored. The displacement of $\boldsymbol{p}_i$ at any given time is represented by a 5-vector $(u_i, v_i, w_i, \alpha_i, \beta_i)$, where u_i, v_i, and w_i are the displacement of $\boldsymbol{p}_i$ in the x, y, and z directions, and α_i and β_i are the rotation angles around the x and y normal-vector coordinate axes. Again the space complexity is $O(n^2)$.

3.5.3 Visual fidelity

Figures 3.19 and 3.20 show the visual appearance of the page produced by each method when the lower right-hand corner of the page is placed at three different positions. To achieve this in a geometric simulation, the program is given the desired (x, y, z) coordinates of the page corner—in the case of the two-dimensional peeling method, just the (x, y) coordinates. For physical models, it involves applying a sequence of forces to the lower right-hand corner of the page that are sufficient to bring it to the appropriate point.

Figure 3.19 shows representative samples of the visual effects produced by the geometric techniques of peeling and wrapping. Figure 3.19(a) is clearly inferior to 3.19(b) because, as explained in Section 3.1, the creased dog-ear has exactly the same area as the part of the page beneath that has been revealed, which is obviously unnatural. Nevertheless, when used in a reactive real-time page-turning simulation users do not, in our experience, notice this deficiency. The geometric wrapping models in Figures 3.19(c) and 3.19(d) do not fare so well. The cylindrical model (Figure 3.19(c)) looks rather unnatural. The first and third images of the conical model (Figure 3.19(d)) suffer from the fact that the cone's axis is constrained to lie on the negative y axis, and the page is therefore hardly curled at all. No doubt different variants of this model could produce more realistic effects.

Some more radical differences can be seen in the images in Figure 3.20 produced by the physical methods—three mass-spring models of increasing complexity and the finite element method. Because the basic mass-spring method does not model the page's resistance to bending forces, the page crumples in a way that resembles cloth rather than paper, weighed down by the force of gravity. This can be seen in Figure 3.20(a). In Figure 3.20(b) the page is twisted unnaturally, an effect which—although it looks plausible—is impossible to achieve with ordinary paper. The third model tries to correct this twist and overall the simulation looks reasonably natural, but a slight twist can still be discerned in the third image of Figure 3.20(c). This twisting behaviour does not appear in the finite element method (Figure 3.20(d)), which looks quite natural.

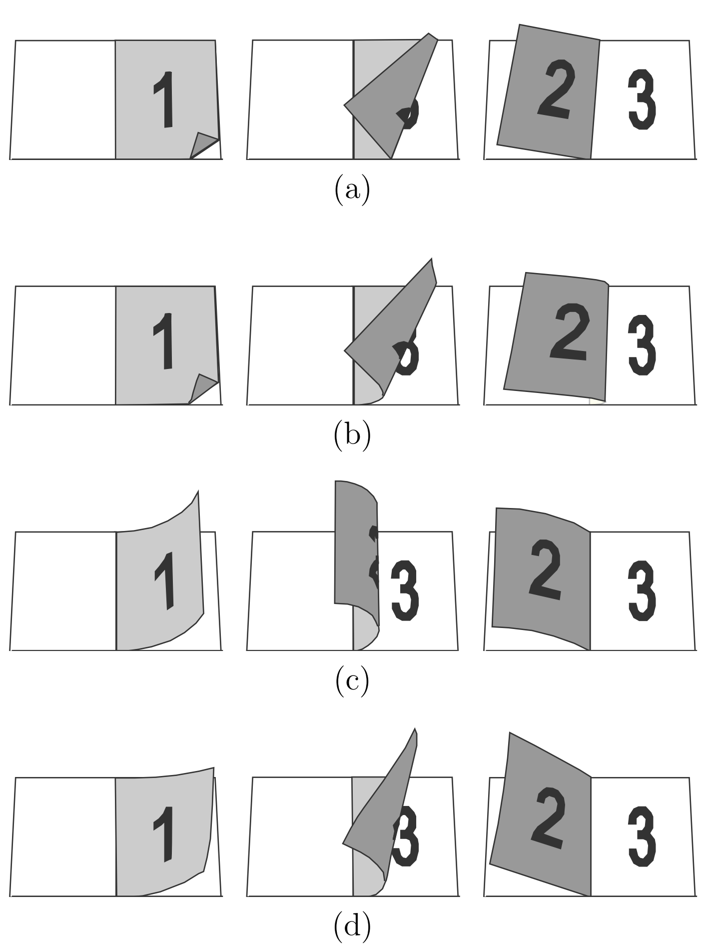


Figure 3.19: Geometric page turning simulation using: (a) Peeling 2D, (b) Peeling 3D, (c) Cylindrical, and (d) Conical models

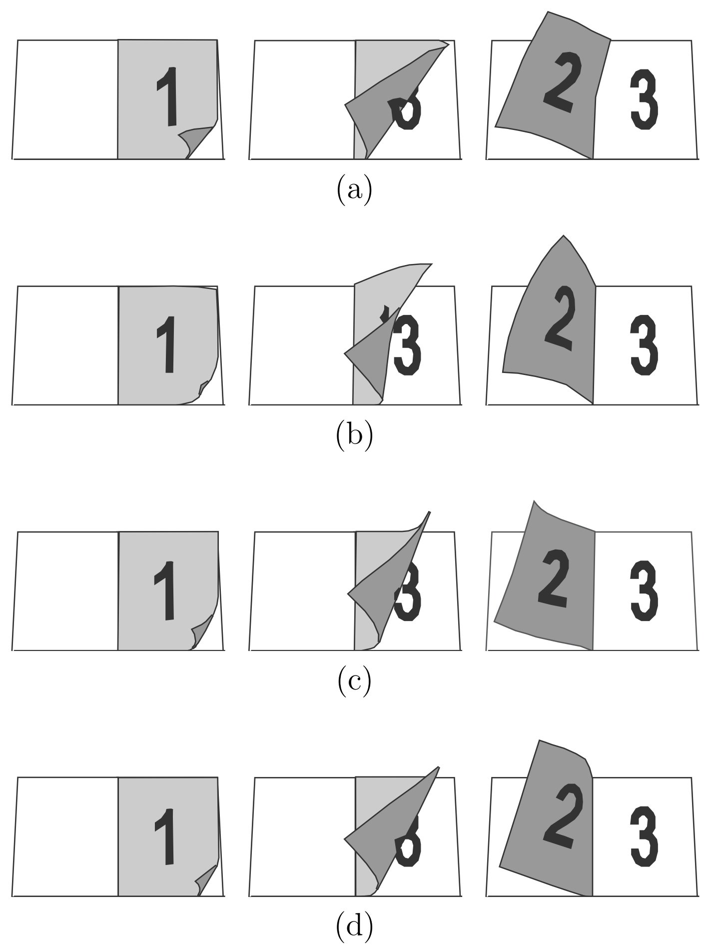


Figure 3.20: Physical page turning simulation using: (a) Basic mass-spring, (b) Bending, (c) Bending without twisting, and (d) Finite element method models

3.6 Summary

The simple act of turning a page is quite a complicated thing to simulate. The variety, intricacy, and complexity of possible ways of simulating this fundamental act of reading is virtually unknown. Descriptions of these algorithms are scattered around the literature, and vary greatly in style and prerequisite knowledge. This chapter has presented eight different page turning models with different geometric and physical underpinnings. Each model has been implemented and compared on the basis of execution speed, space complexity and visual appearance.

Differing degrees of visual accuracy can readily be observed when the simulation results are placed side by side as in Figures 3.19 and 3.20. However, these effects generally go unnoticed when the images are viewed individually, particularly when they are animated under the control of a user who is focused on the page content. For a book using ordinary rectangular flexible pages, the simplest model—two-dimensional peeling as in Figure 3.19(a)—seems adequate for most applications. Although it sacrifices 3D realism, the appearance can be improved by adding visual details to simulate the effect of a smooth bend rather than a sharp crease—some shading on the bend and some shadowing just beneath it—as illustrated in Figure 3.2.

The peeling model is inadequate when showing the visual effect of several pages being turned together; the cylindrical and conical models are more suitable for this. If the paper's rigidity and/or shape must be varied, or if users are allowed to make creases—for example, folding corners down into dog-ears—geometric models are inadequate and mass-spring models should be used. In a situation with unlimited computing resources where visual accuracy is more important than interactivity, the finite element method will produce the most realistic simulation for any type of paper under any condition. But a heavy price is paid in computation time.

As explained in Section 1.4, for a realistic electronic book to be deployed in practice, it must not only look sufficiently realistic, but it must be highly responsive, scalable, and not require complex applications for display. Therefore, in this thesis, to investigate whether book models with realistic page turning offer real advantages, a lightweight Adobe Flash-based application, called Realistic Books, that uses two-dimensional geometric page turning technique was implemented. Chapter 4 describes this application in detail. To determine whether the new format can enhance a user's reading experience over conventional document representations like Adobe PDF readers, HTML browsers and physical books, a set of task based evaluations were performed and are presented in Chapter 6.

4

Reading with Realistic Books

This thesis argues that an electronic document application that simulates the physical representation of a book will have significant advantages over conventional document representations, whether electronic or physical. Realistic Books is an application that has been developed to investigate the argument. It was designed according to the guidelines specified in Chapter 2.

As mentioned in Section 1.4, for an electronic book application to be used widely, it must be able to run on conventional computing platforms, and not require an rarely installed software to display it. Realistic Books is a reading application that is built using Adobe Flash. Flash seems particularly appropriate because it is widely pre-installed; it minimises both download time and compatibility issues (Millburn Survey, 2010). For these reasons, it is a popular way to present rich media contents to users in Web pages. The limitations and trade-offs it entails are described in Section 4.4. Realistic Books communicates with a PHP application server to save a user's annotations.

A part of simulating a realistic physical book metaphor is modelling the act of turning a page or a block of pages, and page riffling. Chapter 3 reviews different techniques for simulating these navigation activities. To simulate an analogue page turn, a simple two-dimensional technique can be utilised to create a sufficiently natural simulation that is responsive, scalable and can be easily implemented in Flash. However, a three-dimensional model is required to simulate the other two navigation acts. If this model were to be implemented in Realistic

Books, the simulation would become less responsive and scalable.

Once the page-turning act has been simulated, many more details must be attended to before arriving at a compelling model of a realistic book (Chu et al., 2004). These include modelling the interaction between the pages and the covers, the effects of metadata, and tools for supporting various reading activities such as searching and annotating. These details are crucial to the realisation of a complete electronic book application. Chapter 5 explains in detail how users can define the logical structure of the book, the layout of the page, and the list of reader services.

The interface for Realistic Books consists of two parts: a graphical simulation of a book's physical properties and a mixture of paper and computer-related services provided to satisfy the generic requirements of readers. The first section of this chapter discusses the various components of the book model. The next three sections explain the functionalities that Realistic Books offers. The final section describes technological issues related to the production of Realistic Books.

According to the dictionary, a page is a leaf of a book. However, a leaf actually contains two pages—one on either side of the leaf. Throughout this chapter we

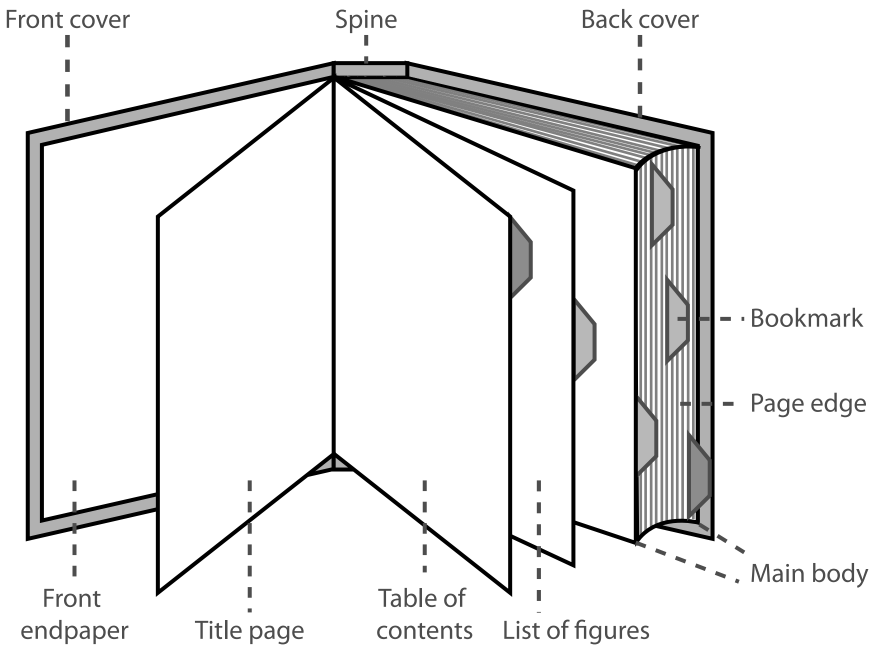


Figure 4.1: Parts of a book

use “paper” to refer to a double-sided page: recto and verso. For example, if the currently visible pages on the book is Page 4 and Page 5, the verso side of the left paper is Page 4 and its recto side is Page 3. Similarly, the recto side of the right paper is Page 5 and its verso side is Page 6.

4.1 Book representation

Each element of a Realistic Books is controlled by the mark-up specified in the book’s template, described in Chapter 5. All books have a front and back cover, and body text. Usually there are many more structural parts. Figure 4.1 shows the major elements of a book.

This section first describes how the contents of a book are displayed in Realistic Books and defines parameters that can be set to modify the presentation. Next, this chapter presents the model used to simulate various portions of the book, both when it is lying still and when a page is turned. Then we turn to other components in the Realistic Books interface.

4.1.1 Structure of a book

The front matter of a Realistic Books comprises of front endpaper, title page, abstract, foreword, preface, acknowledgement, table of contents, and list of figures. Each element starts on a right-hand page. The table of contents and list of figures can be automatically generated from the book’s metadata, as described in Section 5.2. Each entry is hyperlinked: clicking on it opens the book at the relevant section or figure. Except for blank pages, every page in the front matter is numbered by a lowercase roman numeral.

Next comes the main body of the book. Illustrated in Figure 4.2, the page can contain text, image, video and audio. Each chapter starts on a right-hand page. Headings and subheadings are stylised according to the mark-up used in the book’s template. By default, they are written in boldface, in a larger font size than the main text. Chapter titles are centred on the page, and section titles are left-aligned in a smaller font size. Image captions are shown below the image in smaller letters.

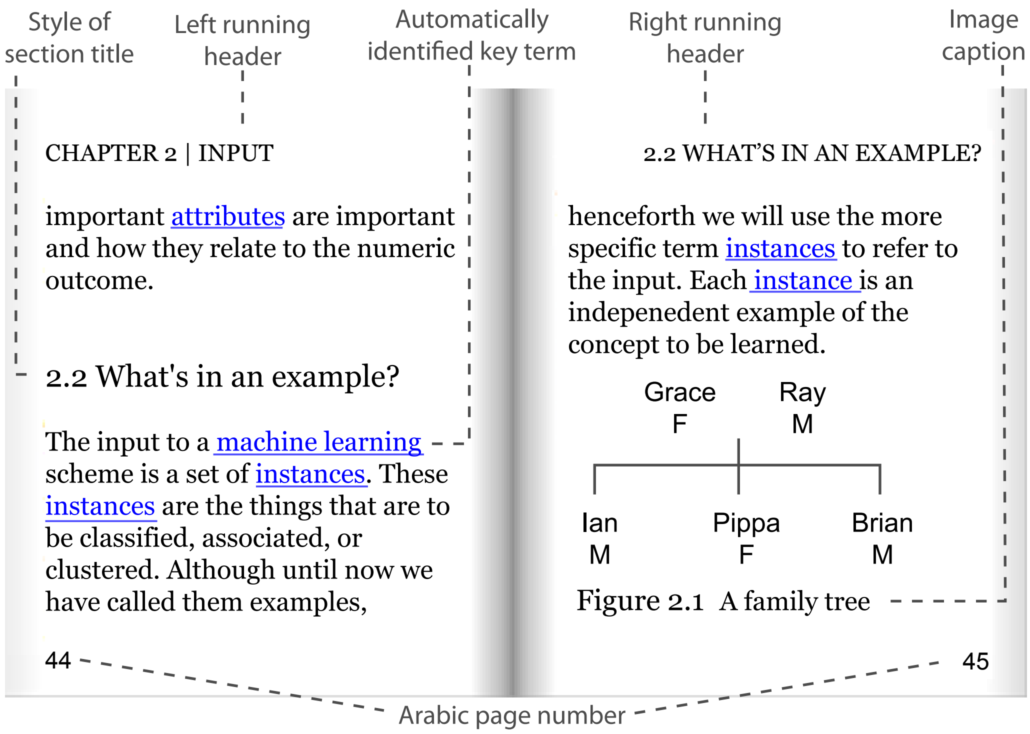


Figure 4.2: Typical page layout for the main body in a Realistic Books

A running header on each left-hand page displays the chapter title, and on each right-hand page the current section title. All the text in a running header is capitalised. Running headers are omitted from the first page of each chapter. Starting from the main body, each page of the book (except blank pages), has an Arabic page number.

The content of the book may have been subjected to a process that identifies key phrases, hyperlinking them to corresponding articles in Wikipedia. This process is explained in Chapter 5. Hovering over a term reveals its term definition; clicking opens its Wikipedia article in a Web browser.

The book's back matter comprises of appendices, bibliography, glossary, index, and rear endpaper. Each element starts on a right-hand page and has the same appearance as the main body. The index page may be automatically generated from the book's text.

For presentations, any given page can have several versions or "slides." The *animation* buttons on the control bar, or the arrow keys, move between slides.

By default, the beginning of each chapter and subsection, the table of contents, the list of figures and the index pages are bookmarked. Bookmarks are shown as tabs that give visual cues to the structure of the book—an ever-present table of contents. They provide users with convenient access to those pages. Section 4.2.2 describes how readers can utilise tabs while browsing books.

4.1.2 Geometric book model

Now that we have examined the structure of a Realistic Books, it is time to describe the model used to simulate a book's physical properties.

A book can be presented closed, with the front or back cover in view, or open. Figure 4.3(a) shows an opened book in Realistic Books. When the book is open, a two-page spread is displayed. The reader's position in the text is indicated by the thickness of the stack of pages on either side. Figures 4.3(b) and 4.3(c) illustrate the geometric model of the opened book from the side and top view, respectively.

In Realistic Books, flat parallelograms are used to represent the visible pages, the stack of page edges and the spine of the book. As described in Section 3.1, although curved shapes may appear more realistic, flat parallelograms enhance text readability and ensure constant time complexity during the book simulation. This model is updated at the end of each complete page turn. Table 4.1 lists the parameters that can be specified in the book's template file to define the appearance of each component.

Covers

Books in Realistic Books may have hard covers or paper covers. Changing the *CoverType* parameter not only changes the way the covers are turned, but also changes the size of the covers and the appearance of the book's front and rear endpapers.

As illustrated in Figure 4.3(c), the width and height of the cover is given by $PageWidth + (2 \times BookMargin)$ and $PageHeight + (2 \times BookMargin)$, respectively. *BookMargin* is the distance between the border of the page and the border of the cover, that is, the vertical distance between $a'b'$ and AB and the horizontal

Parameters	Description	Default value
Pages:		
PageWidth	width of a page	465 px
PageHeight	height of a page	600 px
PageType	type of papers used for the book's pages	flexible
PageColour	base colour of the book's pages	white
LeftMargin	left margin of the page	50 px
RightMargin	right margin of the page	50 px
TopMargin	top margin of the page	50 px
BottomMargin	bottom margin of the page	50 px
Covers:		
CoverType	type of papers used for the book's covers	hardcovers
CoverColour	base colour of the book's covers	white
BookMargin	distance of the page from the cover	10 px
Page edges:		
xOffset	horizontal offset between each paper	0.5 px
yOffset	vertical offset between each paper	0.1 px
Bookmarks:		
TabCutoff	number of bookmarks fit along a page edge	22
BaseBookmarkColour	the bookmarks default colour	blue
TabStyle	styles of the bookmark	coloured & elongated
SnapTo	bookmark targets	chapter & subsection
Page flipping:		
FrontPause	pause time before turning the front cover	700 ms
BackPause	pause time before turning the back cover	700 ms
TurnPause	pause time before turning the book's pages	700 ms

Table 4.1: Modifiable parameters for visualising a book

distance between JK and $j'k'$. If the book is paperback, *BookMargin* is set to one pixel; otherwise, it is ten pixels by default.

The base colour of the book's cover is set to the colour defined by the *CoverColour*. If an image is provided for the front cover, *CoverColour* is set to the average colour found on the edges of the image according to the algorithm described in Appendix B. The default settings can be overridden by specifying a colour for *CoverColour* in the book's template.

Spine

The spine of the book is represented by the parallelogram $b'i'l'c'$ in Figure 4.3(c). It is drawn by connecting the front and back covers, which are represented by the parallelograms $a'b'c'd'$ and $i'j'k'l'$ respectively. The base colour of the spine is the same as the covers (i.e., *CoverColour*).

Pages

In Figure 4.3(c), parallelograms $EFGH$ and $FMNG$ represent the two pages of the book that are currently visible to the user; parallelograms $ABCD$ and $IJKL$ represent the page content area of the book front and rear endpapers. Their size is defined by *PageWidth* and *PageHeight*. By default, the aspect size is set to correspond to the ratio of an A4 paper. *LeftMargin* and *TopMargin* represent the left and top margins of the page content from the page edge.

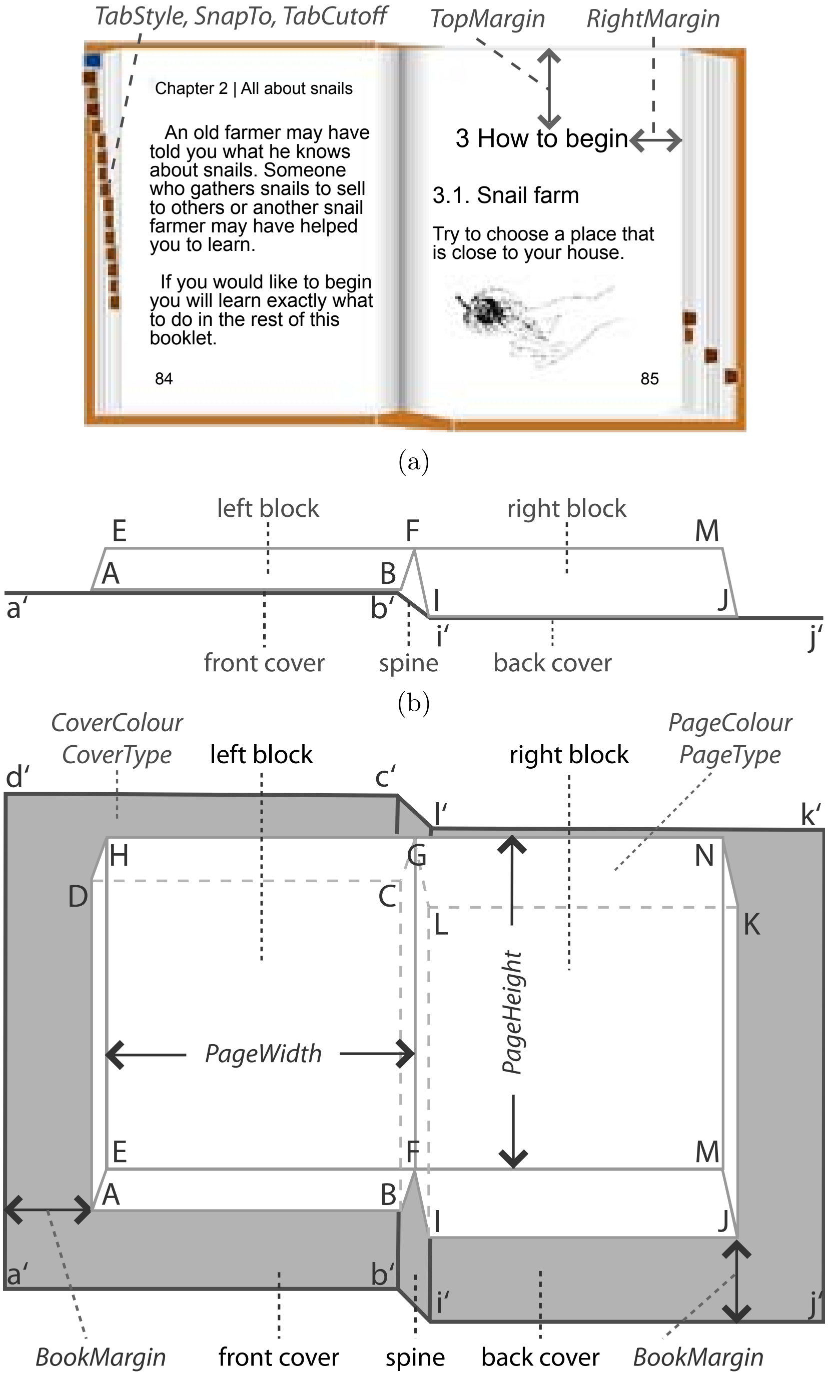
The pages in Realistic Books may be flexible or rigid. Rigid pages are turned as though they were made from stiff cardboard; flexible pages are turned like ordinary paper. Section 3.1 describes the algorithms in detail. *PageType* controls which type of paper is used.

The base colour of a page is specified by *PageColour*, which is white by default. If an image is used and the *PageColour* has not been set, the algorithm explained in Appendix B is used.

Stack of page edges

Figure 4.4 shows an enlarged version of the geometric model for the book's left and right blocks. The stack of pages on the left and right side of the book are represented by the left and right blocks $ABCD\_EFGH$ and $IJKL\_FMNG$ respectively. These connect at the line FG vertically, located at the centre of the book.

As in the previous figure, parallelograms $ABCD$ and $IJKL$ represent the content area of the front and rear endpapers. Parallelograms $EFGH$ and $FMNG$ represent the two papers on the top of the book that contain the currently visible pages. For example, if the currently visible pages are Pages 4 and 5, the verso



(c)

Figure 4.3: An opened book in Realistic Books: (a) the book display, and its enlarged geometric model from the (b) side and (c) top view

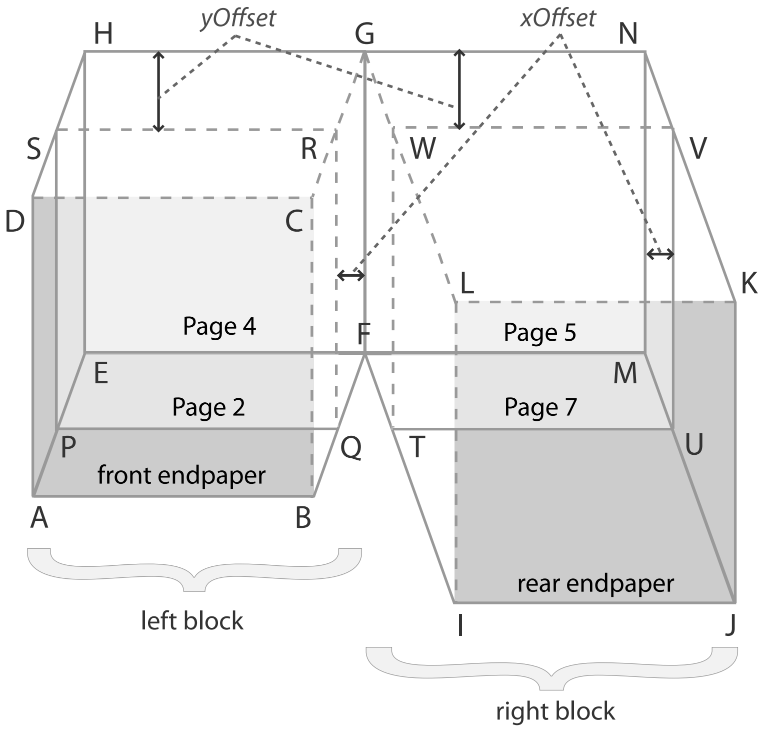


Figure 4.4: Geometric model for the stack of page edges

side of the left paper $EFGH$ is Page 4 and its recto side is Page 3. Similarly, the recto side of the right paper $FMNG$ is Page 5 and its verso side is Page 6. Parallelograms $PQRS$ and $TUVW$ represent papers that are located just below $EFGH$ and $FMNG$ respectively. The verso side of $PQRS$ is Page 2, and the recto side of $TUVW$ is Page 7.

The thickness of the left block is the vertical distance between AB and EF, which is the number of papers on the left side of the book. Similarly, the thickness of the right block, the vertical distance between IJ and FM, is the number of papers on the right side of the book. The width of the base of a block, AB and IJ, is the width of an individual page, that is *PageWidth*. The height of the base, AD and JK, is *PageHeight*.

When the book is opened, the vertical offset in the block is the vertical distance between WV and GN, and between SR and HG; and the horizontal offset in the block is the horizontal distance between PS and EH, and between UV and MN. Their values are defined by *yOffset* and *xOffset* respectively.

These page edges can be clicked to give direct access to another part of the document, opening the book to the page at a corresponding proportional location in the document. Given a mouse position m_p, to calculate the number of papers that need to be turned, first determine which parallelogram contains the point m_p. For example, if m_p is located in parallelogram $AEHD$, divide the horizontal distance between m_p and the line EH by *xOffset* to give the number of papers that need to be turned from the left block. If m_p is located in parallelogram $ABFE$, divide the vertical distance between m_p and the line EF by *yOffset* to give the number of papers need to be turned from the left block. The same method can be used when m_p is located in parallelograms $MJKN$ or $IJMF$.

Bookmarks

Bookmarks are placed in predefined positions along the right or left page edges. The first one is placed at the top, the second at a small distance below it, and so on down the page edge. The height of a bookmark is $\frac{PageHeight}{TabCutoff}$, where *TabCutoff* represents the greatest number of bookmarks that can fit along the edge of the page without overlap. Once this cutoff is reached, the starting position is reset back to the top of the page edge, as shown in Figure 4.3(a). Through experimentation with several books, the default cutoff value is 22 bookmarks.

Readers can change *TabStyle* and *SnapTo* variables at runtime to change the bookmark style and targets. For example, in Figure 4.3(a) tabs at the start of each front or back matter component are wider and darker than tabs at the start of each chapter. The range of colour gradation is calculated based on the values of *CoverColour* and *PageColour*. Section 4.3.2 explains the method in detail, and describes the bookmark styles and targets that are available in Realistic Books.

4.1.3 Poses during a page turn

A book may be presented closed, with the front or the back cover in view, or open, with a double-page spread in view. These are the two distinctive states of the book during the simulation. So far, the static appearance of each element of the book on each of this static poses has been described. A further step in book simulation is to define the appearance of each element when a page is turned, transitioning from one pose to another.

There are only four possible page turning actions: turning the front or back cover to open and close the book, and, when the book is open, turning the left-hand page to the right and vice versa. In all the book's poses, only the content of the two topmost papers ($EFGH$, $FMNG$, $PQRS$ and $TUVW$ in Figure 4.4) are visible to the reader. For all other papers, they can only see the edges. This makes it unnecessary to store the entire book's content in main memory. During the book simulation, only the content of these four parallelograms is needed.

Because all pages in the book have the same base colour, horizontal and vertical page edge textures can be used to represent the left and right page blocks. For a hardcover book, the front and rear endpapers are bigger than the other pages. When the book is opened, as in Figure 4.3(a), the outer edge of the book spine, front and rear endpapers are visible. In the book simulation, they are drawn as parallelograms $b'i'l'c'$, $a'b'c'd'$, and $i'j'k'l'$ respectively, with *CoverColour* as their colour. During automatic page turning, Realistic Books waits *FrontPause*, *BackPause* and *TurnPause* milliseconds before turning the front cover, the back cover and a page in the book, respectively.

Turning the covers

If readers turn the front cover ($FMNG$) of the book in Figure 4.3(c), the recto and verso page content of $FMNG$ and the recto page content of $TUVW$ are loaded. Here, parallelogram $a'b'c'd$ is equal to $FMNG$. A vertical page edge texture is applied to parallelogram $UJKV$, and a horizontal page edge texture is applied to parallelogram $IJUT$. When the book is completely open (i.e., $FMNG$ is on the left side), the rear endpaper $i'j'k'l'$ and the right block $IJKL\_TUVW$ are translated upwards, as displayed in Figure 4.5(a). If the book is then closed (i.e., $FMNG$ has moved from the left to the right side of the book), the rear endpaper $i'j'k'l'$ and the right block $IJKL\_TUVW$ are translated downwards—reversing the sequence in Figure 4.5(a). A similar mechanism is used for turning the back cover. Figure 4.5(b) shows the movement of each book component when the book is opened from the back cover.

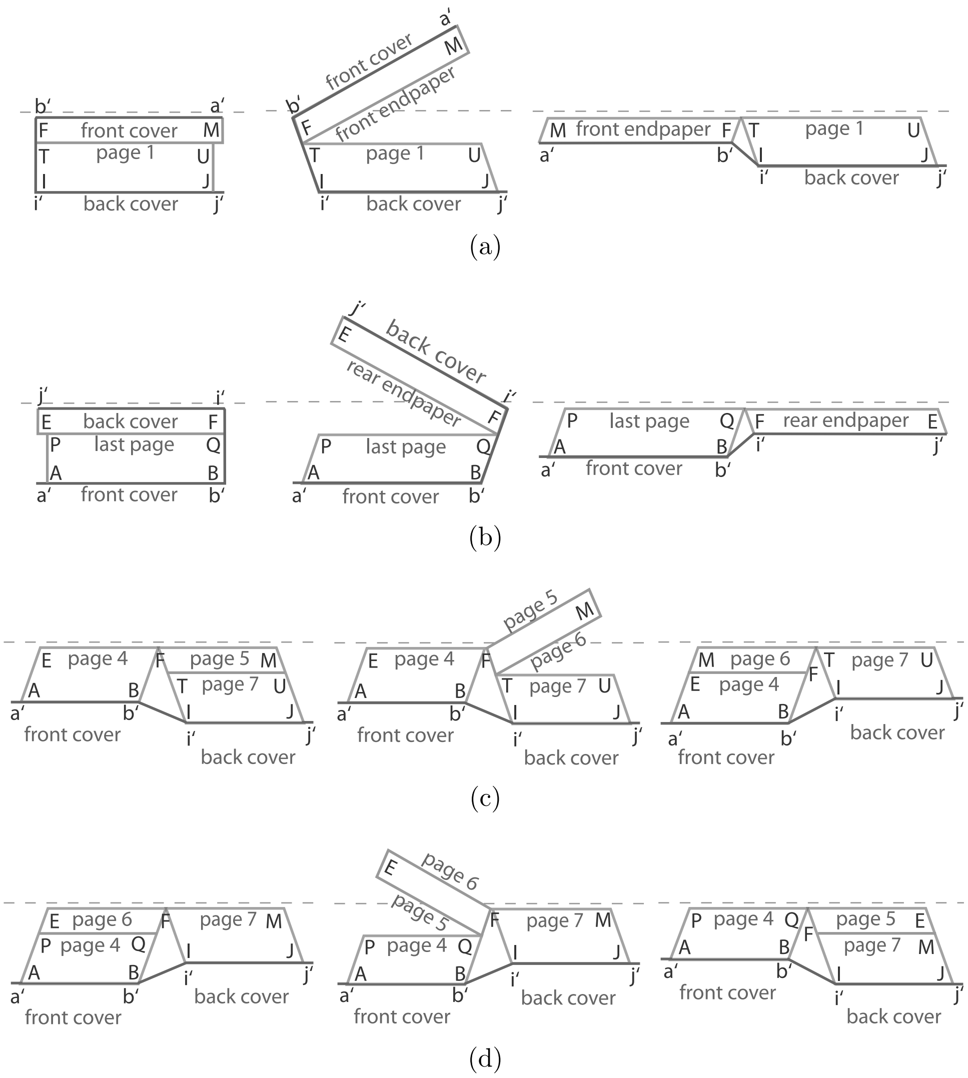


Figure 4.5: Side view of the book model when (a) the front cover, (b) the back cover, (c) a right page, or (d) a left page is turned.

Turning the pages

If the reader turns the right page ($FMNG$) of an open book to the left side, the recto and verso page content of $FMNG$, the recto content of $TUVW$, and the verso content of $EFGH$, are loaded. If the left page ($EFGH$) is the front endpaper, the parallelogram $a'b'c'd'$ and the left block $ABCD\_EFGH$ are not drawn. Otherwise, vertical page edge textures are applied to parallelograms $AEHD$ and

$UJKV$, and horizontal page edge textures are applied to parallelograms $ABFE$ and $IJUT$. As shown in Figure 4.5(c), once the user has completely turned the parallelogram $FMNG$ to the left side (i.e., $FMNG$ is on top of $EFGH$), the position of each block is updated: $a'b'c'd'$ and the block $ABCD\_EFGH$ are moved downwards; $i'j'k'l'$ and the block $IJKL\_TUVW$ are moved upwards. Likewise, when the left page ($EFGH$) is turned. Figure 4.5(d) displays the sequence of the book poses during the page turn.

4.1.4 Book interface

As mentioned in Chapter 2, reading is a complex and multifaceted activity. People do more than just read a document from the beginning to the end: they usually underline, highlight, scribble, add comments, add bookmarks to the text while reading to help them understand and organise their reading for later review and retrieval. They might want to find specific information in the book, and consult dictionaries or encyclopaedia to comprehend unfamiliar concepts or terms.

Figure 4.6 shows the Realistic Books interface. The screen is divided into three areas: the book space, the reader's tools and the preview area. The largest part of the screen is dedicated to the book itself. Previous sections have described how electronic documents are presented as books. The following sections explain the reading services that are provided to satisfy the generic requirements of readers as discussed in Chapter 2. These services are designed to follow the book metaphor and the features of normal book consultation as closely as possible.

The reader's tools give direct access to commonly used functions such as navigation, search, zoom and animation. The preview area displays either thumbnail images of each page or short snippets of all annotations in the book. Both areas are optional and may be suppressed during the book's creation; also different toolsets can be chosen as explained in Chapter 5.

4.2 Within-document navigation

Navigation is how people move from one page to another. Obviously, it is important for users to be able to find their way around a document easily. As described

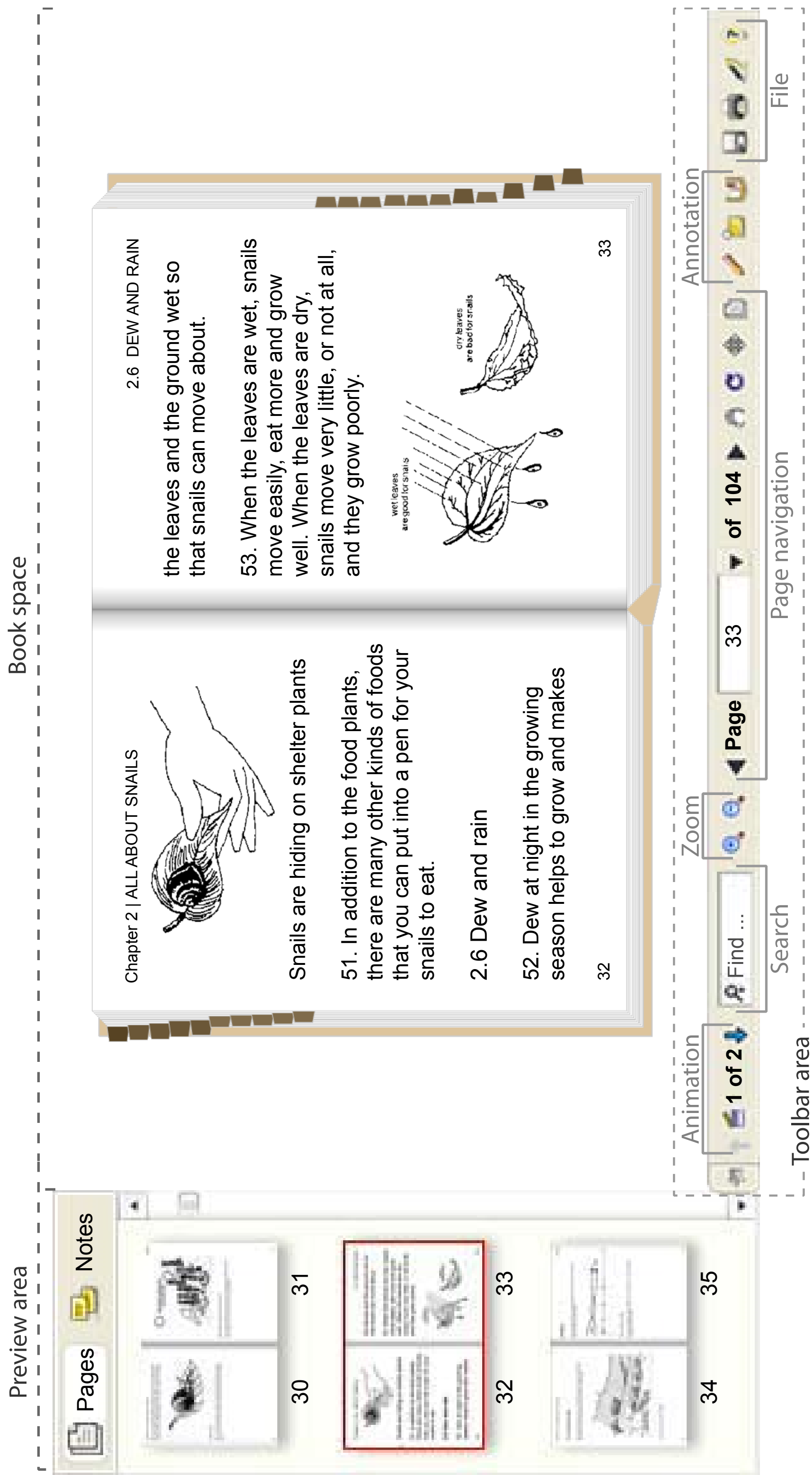


Figure 4.6: Workspace in Realistic Books

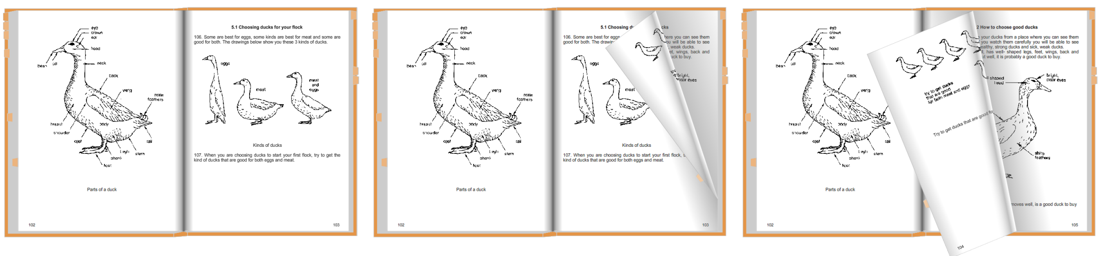
in Chapter 2, within-document navigation is usually a combination of browsing and searching. Browsing involves scanning through the document casually to find something of interest. People may access pages directly, linearly, hierarchically from the table of contents, or transversally from internal or external hyperlinks. Searching involves examining the document thoroughly for particular information. This section describes various ways a user can move around the book, and explains the search functions supported.

4.2.1 Page turning

By simulating the physical act of page turning in an electronic document reader, users can subtly peek ahead to get a foretaste of what is in store. They retain contact with the text because the interaction is continuous. Realistic Books utilises the two-dimensional peeling and shearing techniques of Chapter 3 to simulate the act of turning flexible and rigid pages respectively, as illustrated in Figure 4.7.

Readers grasp a page by pointing at a corner—or anywhere along the top,

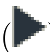
(a)



(b)

Figure 4.7: *Raising Ducks: How to Begin* book: (a) opening the front cover, and (b) turning a page

right, or bottom edge—and depressing the mouse button. The page follows the movement of the cursor. It can be moved anywhere, within the physical constraints imposed by not tearing the paper, and the visual details follow instantly. When the button is released, the page either falls back to its original position or continues to turn until it lies at rest in the turned position, depending on whether the page has been turned pass the centre of the book. Although better control can be obtained by using a touch-panel instead, even with a mouse all this feels perfectly natural, as confirmed by the user evaluation reported in Chapter 6.

The system provides alternatives to the mouse gesture for turning pages. Pages turn when the right-hand page is clicked, the right arrow key in the keyboard is pressed, or the *next page* button () on the toolbar is clicked. The analogous actions work on left-hand pages. A *keep-flipping* button () turns pages continually. Clicking the book, pressing the escape key in the keyboard, or clicking the *hand* button on the toolbar stops the flipping. The *options* button () allows users to select an appropriate flipping speed.

4.2.2 Non-sequential navigation

Except perhaps for novels, the strategy of starting at the beginning and stopping when the book comes to the end is rare indeed: people skip about. Realistic Books provides several mechanisms for non-sequential navigation.

Readers can jump to the beginning of any chapter or section by clicking its table of contents entry. Similarly, they can jump to a figure by clicking its list of figures entry. They can click on internal and external hyperlinks to jump around the book and onto the Web respectively.

As with physical books, readers can pick up any page edge and open the book there. Hovering over the page edges reveals the page number as mouseover text. When the book is open, the inside border of both covers is visible, and clicking them closes the book (in either direction). Users can also select or type the page number in the toolbar to jump to a specific page.

Through a configuration option described in Section 4.3.2, bookmarks can be displayed as tabs that protrude from the pages to mark the beginning of each chapter and section, or, alternatively, the location of figures, creating an ever present table of contents or list of figures. Hovering over a bookmark reveals its associated page number and the chapter/section title or figure caption. Clicking it, opens the page the bookmark is set to.

A document overview is also provided. The reader can display the thumbnail view of the document, as shown in Figure 4.6 by clicking on the thumbnail icon () on the toolbar. In this display, each page is shown as a small numbered thumbnail image. The currently visible pages are highlighted. Users can switch to a different page by clicking on it. This is often a useful way to find a particular page containing distinctive graphics.

While consulting a book, it may be useful to be able to return to previously visited locations. For instance, when reading a scholarly paper, readers often flip briefly to the list of references when citations occur in the text. In Realistic Books, whenever a user performs a non-sequential navigation, the previously visited page is bookmarked. Clicking on it, the pages turn back to the previous location.

When the reader jumps to a specific page, pages are turned automatically, one by one, if the target location is just a few pages away from the current page; otherwise the book opens immediately at the appropriate point. In the book options menu, users can change this behaviour. They can specify whether pages are turned automatically one by one or immediately regardless of how far away the target page is.

A notable problem for online texts is homogeneity: most online texts always look the same (Nielsen and Molich, 1990). This makes it difficult for users to get a sense of location. In Realistic Books, running heads, page numbers, bookmark tabs, table of contents, and page edges provide context and orientation clues to the users. The stack of page edges on both sides of an open book indicate visually the length of the document and the reader's position in it. Users can easily know where they are in the book, and acquire incidental knowledge of the location of information by relating it to its physical position in the document.

4.2.3 Text search

A prime advantage of electronic documents is the ability to search for text in their contents. Realistic Books provides a search mechanism modelled on the incremental text search mode of GNU Emacs (Stallman, 2000). This section shows how users perform text searches and describes the algorithm used to find their search terms in the book.

User interface

Searching is another way for users to navigate non-sequentially through a book. The back-of-the-book index and table of contents are the two main methods of locating pertinent information in a physical book. Realistic Books provides both of these, as well as a full-text search function.

There are two ways of invoking a text search. First, users can type search terms into a box in the toolbar. As they type, the program scans the entire book. Readers can also invoke a text search by selecting some text, right mouse clicking to display the context menu, and selecting the *Search this text* option. This makes it easy to search for other occurrences of a word or phrase in the document.

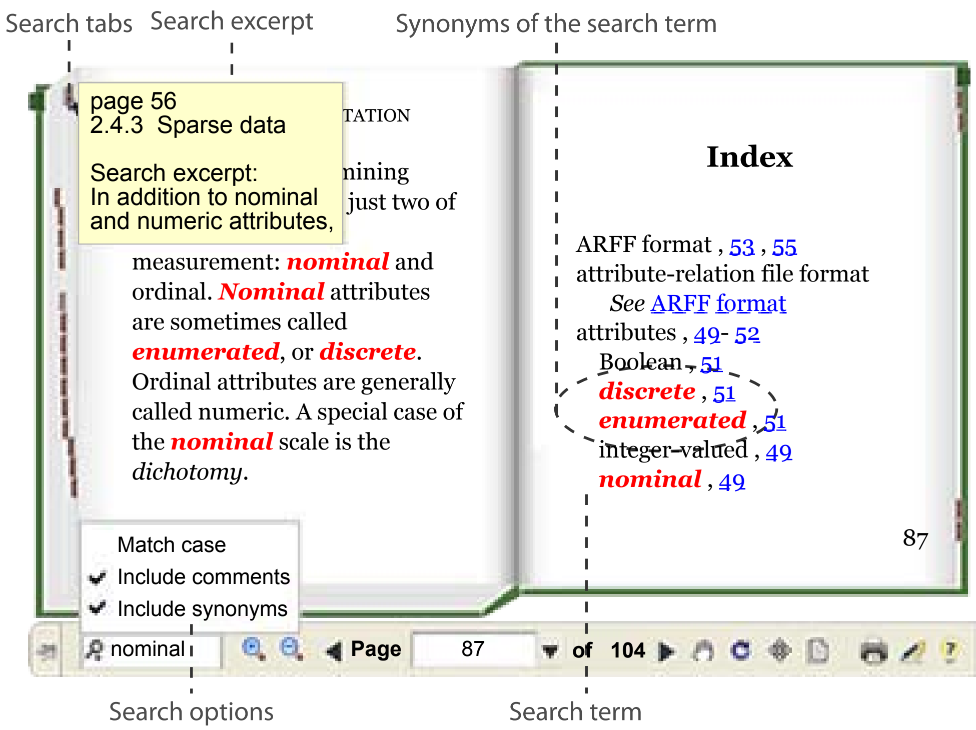


Figure 4.8: Text search interface

If there is a match, all bookmark tabs except for the ones that point to the table of contents, list of figures, back-of-the-book index and previously visited page are hidden and replaced by tabs on all pages containing the search terms. Mousing over a tab shows the page number and an excerpt from the first matching fragment on the page, as illustrated in Figure 4.8. If the user clicks on the tab, the document opens to that page and matching passages are highlighted. Pressing the enter key takes readers to the matching page closest to their current position. If there are no matching passages, the box turns red and displays *no matches found*.

There are three searching options: *Match case*, *Include annotations* and *Include synonyms*. They are invoked by pressing the *magnifier* button in the *find* box. The latter two are set by default.

Incremental text search algorithm

String search algorithms match a search term provided by the user against every string in the document. Incremental search does partial matching as the term is being typed. The results are narrowed down for each key that is pressed.

During the search there are three conditions that the algorithm must satisfy:

1. If there is no search term, the search result contains all pages and strings in the document.
2. When a new character is pressed, the result includes all pages containing strings that match. If no strings in the document match, the search result is empty.
3. If the new character is added to the end of the search term, the result is a subset of the previously shown result.

When a user enters a search term and the *Include synonyms* option is on, a table is consulted to find all synonyms of the search term that occur in the book. These are included in the list of search queries. If the search term has no synonyms, the search term is the only element of the list. These terms can be a word or a phrase. For example, a search for “Obama” will automatically find pages containing the phrase “44th President of America”—and vice versa.

When a search term cannot be found in the index table, a string matching operation is performed. Because this operation is slow for a lengthy document, the matching process ceases once the first occurrence of the string is found. For pages that are visible, matching continues until all occurrences of the string on the page have been found.

4.3 Personalisation

Personalisation is as old as the practice of reading itself, and is an integral part of it. Readers create their own personal view of the book to help them understand the content and recall detailed facts such as prior analysis and relations to other documents. Realistic Books supports two kinds of personalisation tool: annotation and bookmarks. Both are superimposed upon the page without altering the original copy of the book. They can be created in any order and removed when they are no longer required. Users usually do not need to press any button to switch from one personalisation tool to the other. Annotation tools include free-form drawings, text highlighting, and Post-It notes. Any changes made will be automatically saved every five minutes and before a reader exits the application.

4.3.1 Annotations

Annotation is a useful technique for adding information to existing documents, which is frequently identified as a valuable part of the process of learning. According to O'Hara and Sellen (1997), the smooth integration of annotating with reading should be a priority for any system meant to support active reading. The user's reading activity should not be disrupted by the act of creating annotations. They should be able to annotate directly on the page, and their annotations should stand out visually from the source text. People sometimes personalise their annotations, where symbols and pen colours mean something to them. The advantages of annotating directly on the page are convenience, visual search and putting comments within the document context.

To annotate, people can scribble, insert images, type comments, resize, move, or delete the note. There are three ways of inserting a comment. The easiest way

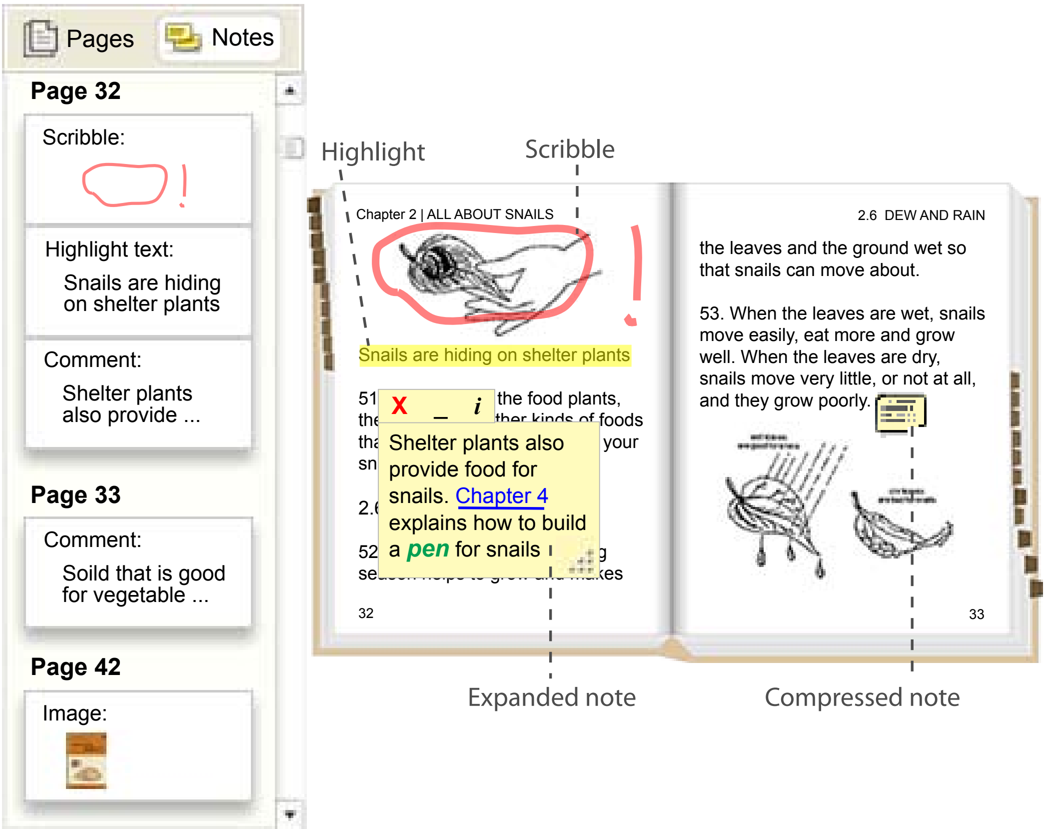


Figure 4.9: Realistic Books interface with the list of annotations displayed

is to use the mouse pointer to indicate where the note should appear and start typing text. Selecting a portion of text on the page automatically highlights it. Comments can be inserted by clicking the highlighted text. Otherwise, users can select the *note* button ( ) in the toolbar and click a location on the page. To create a free-form drawing, users press the *pencil* button ( ) in the toolbar and draw anywhere on the page by depressing the mouse. Clicking the *hand* button ( ) in the toolbar or the escape key leaves the drawing mode and returns to the reading mode. Users can specify the formats of the line and the text, and switch between the note's compressed and expanded mode.

Through a configuration option described in Section 4.3.2, bookmark tabs can be displayed that protrude from the pages to mark the location of annotations. Hovering over a tab reveals its associated page number and an excerpt from the note. When users click the tab the document opens to the corresponding page.

Readers can also have a list of annotations displayed, as shown in Figure 4.9, by clicking on the *list of notes* button ( ) on the toolbar. Each note's excerpt is grouped based on page number. The currently selected note is highlighted. Clicking on a snippet takes the reader to the appropriate page and selects the note.

4.3.2 Bookmarks

Bookmarks are used as landmarks to help people navigate to previously visited locations and through the search space. In Realistic Books, there are six types of bookmark: section, picture, note, search, clip and user. Each bookmark is displayed as a small tab that protrudes from the right or left stacks of page edges. Hovering over a tab reveals its associated short note and page number. When users click the *options* button ( ) in the toolbar, they can specify which type of bookmarks should snap to the book.

The first five bookmarks are automatically generated by the application. They mark the beginning of a chapter or a section, the location of a figure or an annotation, a page containing the search term, and a previously visited page. Clicking the *add bookmark* button ( ) on the toolbar automatically inserts a new bookmark on the right-hand page and places it in the first available position along the page edge.

Although all bookmarks have the same shape and height, as specified in Section 4.1.2 the bookmark's width and colour varies depending on its type. For instance, the higher the section level, the darker and wider it is. Appendix C shows the algorithms used to define the colour and width of a bookmark. To ensure that the colour of each bookmark matches the colour scheme used in the book, if *CoverColour* is not white, the bookmark's base colour is calculated based on the book's *CoverColour* and *PageColour*. Otherwise, it is set to blue. During the book creation process, the value of *BaseBookmarkColour* can be specified.

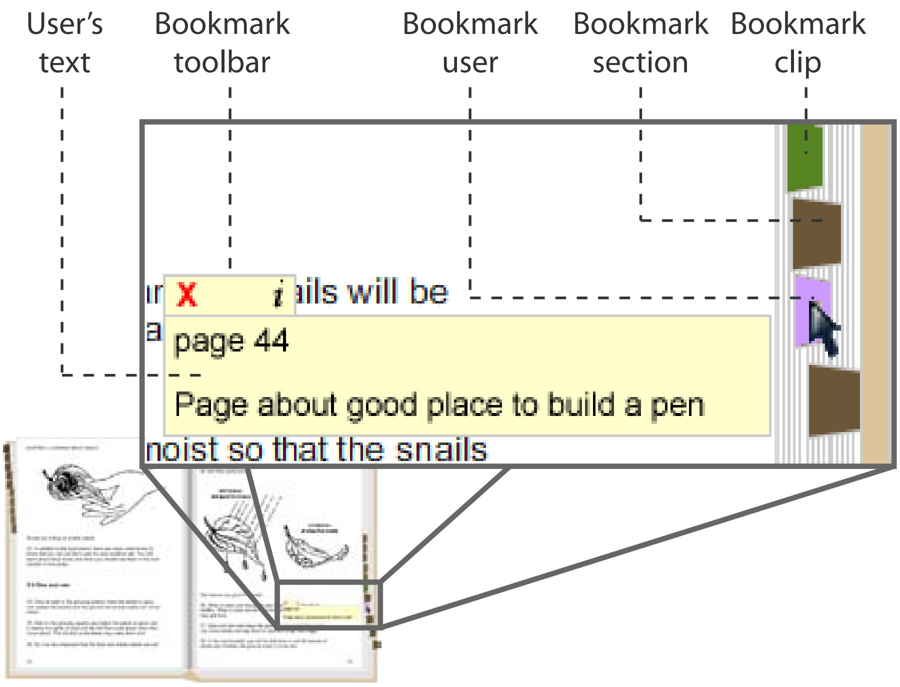


Figure 4.10: Enlarged view of bookmarks in Realistic Books

4.4 Limitations

Adobe Flash proved to be very effective in the development of the Realistic Books interface. However, it was sometimes inadequate for certain aspects of the implementation. This section describes the two main limitations encountered.

HTML text formatting

Adobe Flash only recognises a subset of HTML tags and attributes. For example, it does not recognise the *title* attribute in a link tag, or the *background-colour* attribute. Because the *title* attribute is commonly used for showing the mouse over text of a hyperlink, Realistic Books creates a transparent rectangle that covers each hyperlink containing a *title* attribute. When a user hovers over a rectangle, its corresponding mouse over text appears. This approach is also used to change the background colour of selected text. As mentioned in Section 4.3.1, to highlight text, semitransparent rectangles are created on top of the selected text.

The size and position of the rectangles are derived from the size and position of the bounding boxes of the text returned by Adobe Flash. Unfortunately, these bounding boxes are sometimes inaccurate, and so the rectangles created inside the Realistic Books may not cover the correct portions of the text.

The output of Flash's HTML renderer is non-deterministic; i.e., two identical pieces of HTML text may be rendered differently. The spacing and alignment of the rendered text may change when the text is reloaded or the mouse is hovered over the text. These changes may alter the position of the line break and the height of the text. Thus there are portions of text that may be located outside the page. To alleviate this problem, explicit line breaks are inserted into the text during the pagination process. Although the spacing and alignment on each line of text may change, the explicit line breaks guarantees that the position of the line break and the height of the text do not change. Chapter 5 describes this process in detail.

Loading of files

As explained in Section 4.1.2, when *CoverColour* and *PageColour* parameters are not defined, their colours are derived from the average colour found on the edge of the image used in the front cover and the title page, respectively. Realistic Books only performs this operation once Adobe Flash has signalled that the front cover and title page images are completely loaded into the book. In some cases Adobe Flash erroneously generates this signal before the loading of the image is complete, leading to incorrect *CoverColour* and *PageColour* values being set.

Adobe Flash caches any files loaded into any Flash applications in a Web browser. These cached files are refreshed once all the cached files in Web browser are deleted. Because users rarely delete cached files from their Web browsers, the Realistic Books application needs to ensure that Adobe Flash always loads the newest version of the book's template file by appending the URL of the file with a randomly generated number and a timestamp parameters (e.g., if the URL is "*book.htm*," the new URL is "*book.htm?cachebuster=75*")

4.5 Summary

This chapter has described the prototype called Realistic Books which has been developed for evaluating the hypothesis of this research. The system, which is based on Adobe Flash, is usable within a web browser. Design principles for the prototype—considered in terms of the type of interface, styles of interaction and reading facilities provided to the users—follows the design guidelines listed in Chapter 2.

Realistic Books maintains many features of their printed counterparts—page edge, page turn, annotation, bookmark—and enriches it with several tools which are only made possible because of its electronic form, such as searching facilities, hyperlinks and multimedia. The user interaction style is based on direct manipulation, which provides an immediate and natural understanding of the system functionalities. People can use the *animation* buttons to move between various versions of a page. To increase readability, they can select the *toolbar hide* button to hide the toolbar, utilise the *zoom* button to change book's magnification level and select the *move* button to change the position of the book within the web browser. When the book is made from an HTML or PDF document, users can view the source file in their Web browser or PDF readers, and print the document in its original form. If the users print the book directly from the Realistic Books program, the document will be printed just as it appears in the book.

5

Constructing Books

Publishing a printed book is a major undertaking. Each of many components—size, format, binding, page layout, placement of chapters and sections, tables of contents and figures, subject index, and so on—must be carefully designed to ensure an attractive product, whose content can easily be assimilated and whose readers can easily find information and navigate around.

The conventional publication process for printed books has three stages: editorial, design and production (Mogel, 1996). In the *editorial* stage, editors help finalise the book's content. In the *design* stage, the structure and visual aspects of the book are determined. The book designer chooses details like page size and format, cover design and binding method, margin size, typeface, font size, colours, style of illustrations and tables—anything that affects how the page looks. In the *production* stage, the table of contents, subject index and bibliography are inserted and the manuscript is formatted according to the designer's specification. Proofreaders read each page carefully and examine its layout in a cycle of checking and correction; then the proofs are handed off to the printer. Self-publishing authors often perform one or more of these tasks themselves.

The process of producing high-quality electronic books involves much the same steps. Furthermore, for online reading to be generally acceptable, electronic book applications must provide features that increase reader performance by transcending the affordances of paper books (Dillon et al., 1989). Of course, such applications will only find wide usage if it is easy to convert conventional computer-readable documents into the special form they require (Egan et al., 1989); in this thesis, this is the role of the Bookmaker's Workbench.

Users interact with the Workbench to transform electronic documents into Realistic Books. Implemented within the Adobe Flash environment, it communicates with a PHP-based web server to save the books and any changes made to them. It addresses the design and production stages, but not the editorial processes. The term *book designer* is used to denote the various players involved during the last two phases of the publishing process, such as typesetters, indexers and proofreaders.

A book in the Realistic Books application is constructed from one or more electronic files that are uploaded by the designers. Its logical structure, layout, content, physical properties and reader services are all defined in a template file. The book construction process involves two steps: automated document processing and customisation. Sections 5.2 to 5.4 explain how a template file for the book is automatically translated into the book itself. Section 5.5 describes how designers can modify the template file according to the reader's needs, the application domain, and the anticipated reader requirements.

5.1 Workbench interface

The Bookmaker's Workbench interface has the five areas shown in Figure 5.1: book space, reader tools area, preview area, repository and designer tools area. The first three are components of the Realistic Books application, and were explained in Chapter 4. They show how the document is presented, the available reading tools, and thumbnail images of each page or short snippets of the annotations. The repository area holds the raw material from which the book will be created—all the video, audio, text and image files that the designer has uploaded. Designer tools allow designers to customise visual aspects of the book such as its dimension, fonts and page layout, and select suitable reading services that it will incorporate.

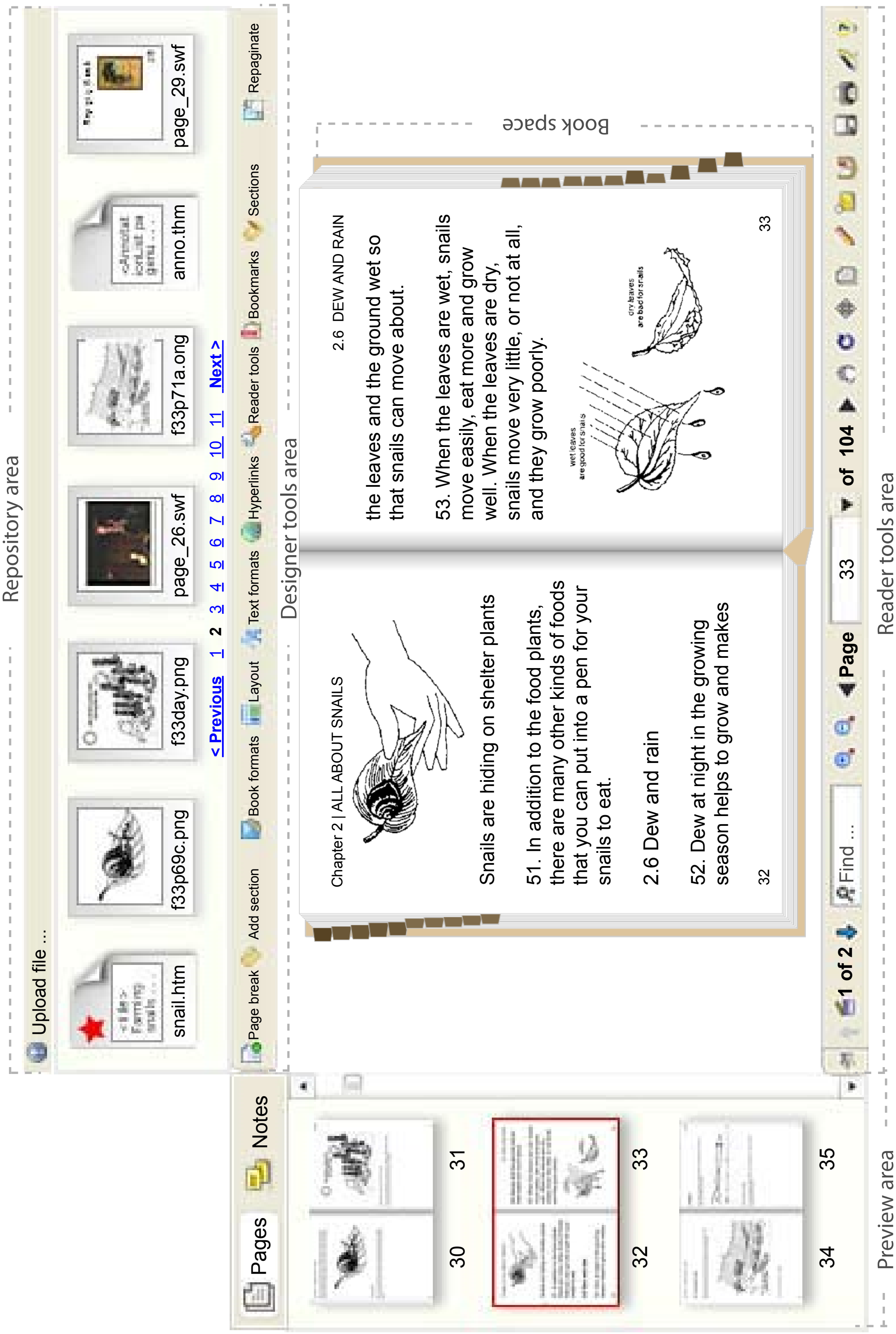


Figure 5.1: The Bookmaker's Workbench interface

5.2 Automated document processing

Realistic Books are generated under the control of a template file written in XHTML. This file defines the sequence of pages that constitutes the book and the services to be provided to readers, along with any associated metadata such as page size, margins, title and where the main text starts (Table 4.1). It may also contain text, with HTML mark-up, that will go into the book.

Figure 5.2 illustrates the process of creating a Realistic Books. There are three phases: acquisition, automated document processing and customisation. The book itself does not exist (and therefore does not appear on the display of Figure 5.1) until it has been generated during the *automated document processing operation* described below. A common way of proceeding is to first make an initial version of the book and then customise it by adding new content to existing pages, and adding new pages. These facilities are described in Section 5.5.

Acquiring source material

Designers use the *upload* button () to add to the repository area every file that contains source material for the book. When a PDF file is uploaded, each page is saved as a vector page image in Adobe Flash format (*.swf*). Similarly, audio or video files that are recognised by Adobe Flash are wrapped into a media player and stored as *swf* files. All these images are displayed in the repository area shown in Figure 5.1, with names such as *page_28.swf* and *page_29.swf* that are assigned by the system.

The template file

As noted earlier, the complete structure of a book is specified in a template file. It is marked with a red star in the repository area. Any text document couched in a standard mark-up language such as XML or HTML (or even a plain text ASCII file) can serve as the book's template. Designers run a conversion script on HTML or ASCII files to convert them to the required XHTML format. If no template is specified, one is automatically generated. By default, it will create a

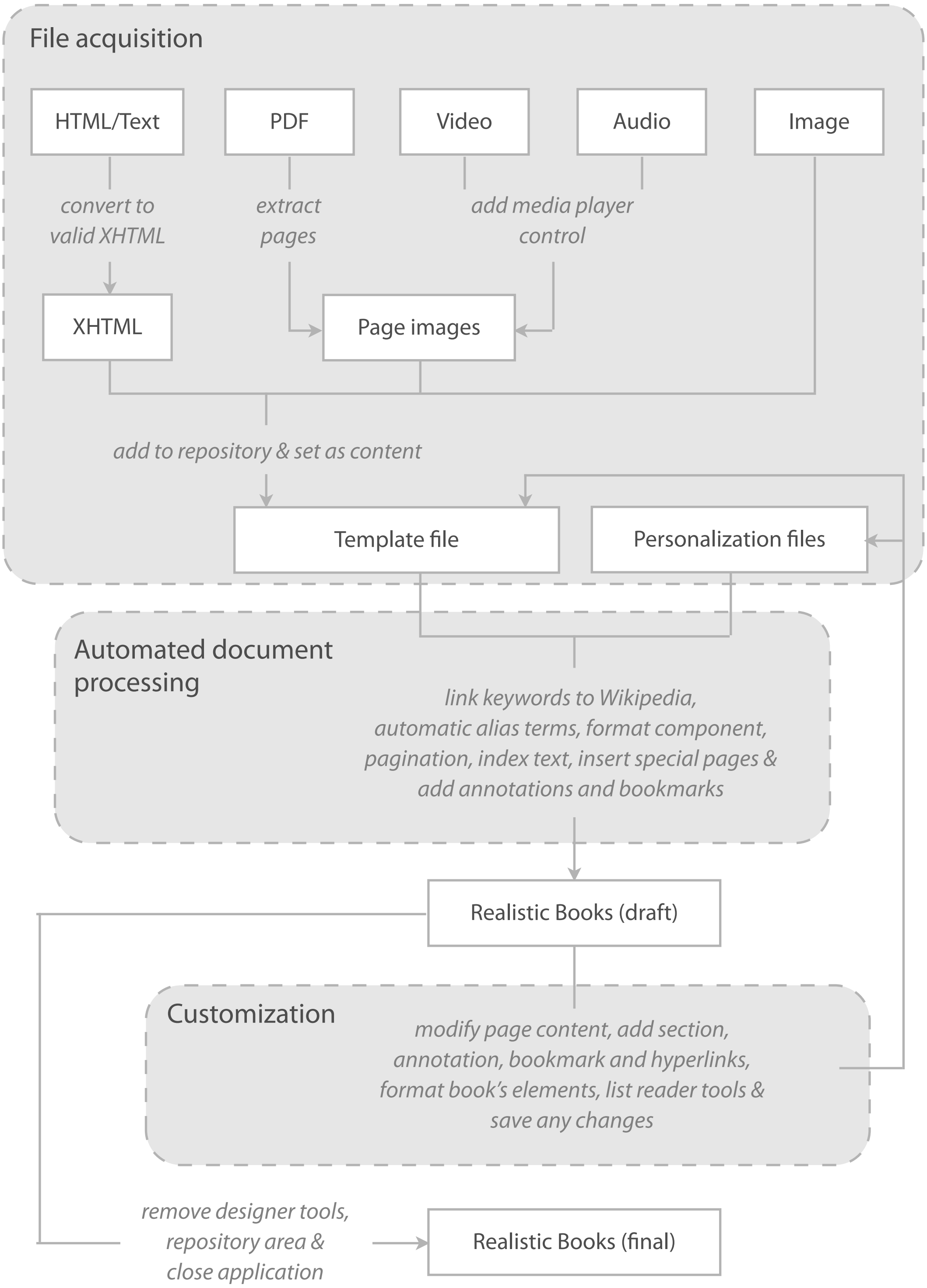


Figure 5.2: The construction process for Realistic Books

```
<Description>
|   <Metadata name="CoverColour">0xcccccc</Metadata> --------------.
|   <Metadata name="StartNumbering">3</Metadata>              Book appearance &
|   <Metadata name="PageSize">450x600</Metadata>              Reader services
|   <Metadata name="AllowPrint">false</Metadata>  -----------------'
</Description>
<title>Learning about raising rabbits in the pen</title> ---------- Book title
<Section>
|   <Description>
|   |   <Metadata name="Title">Introduction</Metadata> -------- Section title
|   </Description>
|   <p>A good way for you to improve your lives is by raising -----------.
|   <a href=”rabbitInfo.htm”>rabbits.</a></p>                    HTML texts
|   <img width="354" src="1.png" height="160" title=”family” />---------'
|   <Section filename=”SectionPage;SectionPage2”>
|   |   <Description>
|   |   |   <Metadata name=”Title”>Raising rabbits</Metadata>
|   |   </Description>
|   |   <p>Rabbit meat is good for you. It also tastes good.</p>
|   |   <newpage /> ----------------------------------- Page breaks
|   |   <p>Rabbit are clean, quiet and small.</p>
|   |   <newpage filename=”ExtraPage;ExtraPage2.png” /> -------- Page Images
|   </Section>
</Section>
```

Figure 5.3: Example of a book’s template file

hardcover book and use all the files in the repository area as the contents. This is a common way of generating an initial version of a book, which will then be customised.

If a book designer chooses to work with an existing Realistic Books, all files specified in its template file are automatically uploaded to the repository area. Figure 5.3 shows an example of a template file.

Personalisation files

Associated with any book may be reader-created information such as annotations and bookmarks. These are stored in personalisation files; books have different personalisation files for each reader. If working with an existing book, the designer

```
<AnnotationList pagenum="4">
|   <Annotation      id="0" x="-89.8" y="30.8" width="117.95"
|                    height="66.05" colour="0xFFF9BC"
|                    transparency="100" isResized="false">
|   |   <Text>
|   |   |   <p><font face="Arial">types of rabbit</font></p> ----- Post-It note
|   |   </Text>
|   </Annotation>
</AnnotationList>
<AnnotationList pagenum="6">
|   <Annotation      id="1" x="-198.15" y="-158" width="221.65"
|                    height="38" colour="0xFFF9BC"
|                    transparency="0" isResized="false">
|   |   <Line      orig="(0,0)" pointlist="(0,0);(220.675,0);  -------- Highlight text
|   |              (220.675,14);(0,14);(0,0)" minPt="(0,0)"
|   |              maxPt="(220.65,14)" alpha="50" thickness="2"
|   |              fill="true"  colour="0xFFFF00" />
|   |   <Highlighted>rabbit meat is good</Highlighted>
|   </Annotation>
|   <Bookmark   id="2" x="-205.5" y="34.1" colour="0xFFF9BC"  ---- Bookmark
|                    text="notes about rabbit meat" />
</AnnotationList>
```

Figure 5.4: Example of a reader's personalisation file

may choose to upload all its personalisation files to the repository area.

Each template and personalisation file is checked to ensure that it is a valid and well-formed XHTML file. Personalisation files, illustrated in Figure 5.4, are parsed as soon as the template file has been translated into a Realistic Books.

Automated document processing

Figure 5.5 illustrates in detail the process of translating the specified template and personalisation files into a Realistic Books. The system paginates any text in the template file, or in any other text documents that have been uploaded, using the algorithm described in Section 5.4. Link information is extracted and saved for later use (see Section 5.5).

The structure of the book is the next step to be determined. Sections are assigned a number and level (e.g., 1, 1.2, 7.1.9). Similarly, images are numbered according to their enclosing section identifier and order of appearance (e.g., Figure

1.1, Figure 4.1.3). A table of contents and list of figures are generated from the titles of the different sections and figures in the book respectively. Bookmark tabs are added to the beginning of sections and figures. Running headers and page numbers are also created automatically.

Predefined design styles are applied to particular elements of the book. For example, section titles are given a boldface style with an increased font size that depends on the level of the section.

Unwanted tools in the reader tools area are disabled. Annotations and bookmarks listed in the personalisation files are created and attached to the appropriate pages. Each occurrence of every word in the document is processed and stored in a full-text index.

People tend to read documents extensively, first determining whether a book is worth reading, and then focusing on portions of the text in order to locate the desired information (Birkerts, 1994; Darnton, 1989; Levy, 1997). Readers readily notice headings, illustrations, charts and tables, all of which stand out visually. They also scan the text for keywords and phrases. To facilitate this, the Bookmaker's Workbench is able to automatically identify significant key terms in the text and mark them visually. Wikipedia is used as a comprehensive knowledge base for this operation, and the visual mark takes the form of a hyperlink to the relevant Wikipedia article. Furthermore, when a reader hovers the mouse over a term, a short description of it pops up. A hyperlinked subject index can be automatically created by collating them together with hyperlinked references to the appropriate page (and page number). All this information is generated using the text mining technology described in (Milne and Witten, 2008). Section 5.3.2 explains this process.

Previous research has shown that one reason why people fail to locate the information they seek is that they describe it in terms that differ from the terminology used in the document (Furnas, 1985). One solution is to increase the number of ways in which each piece of information can be denoted. Section 5.3.3 described how the Workbench generates a synonym table automatically as a byproduct of the above-mentioned text mining operation. This table is consulted automatically

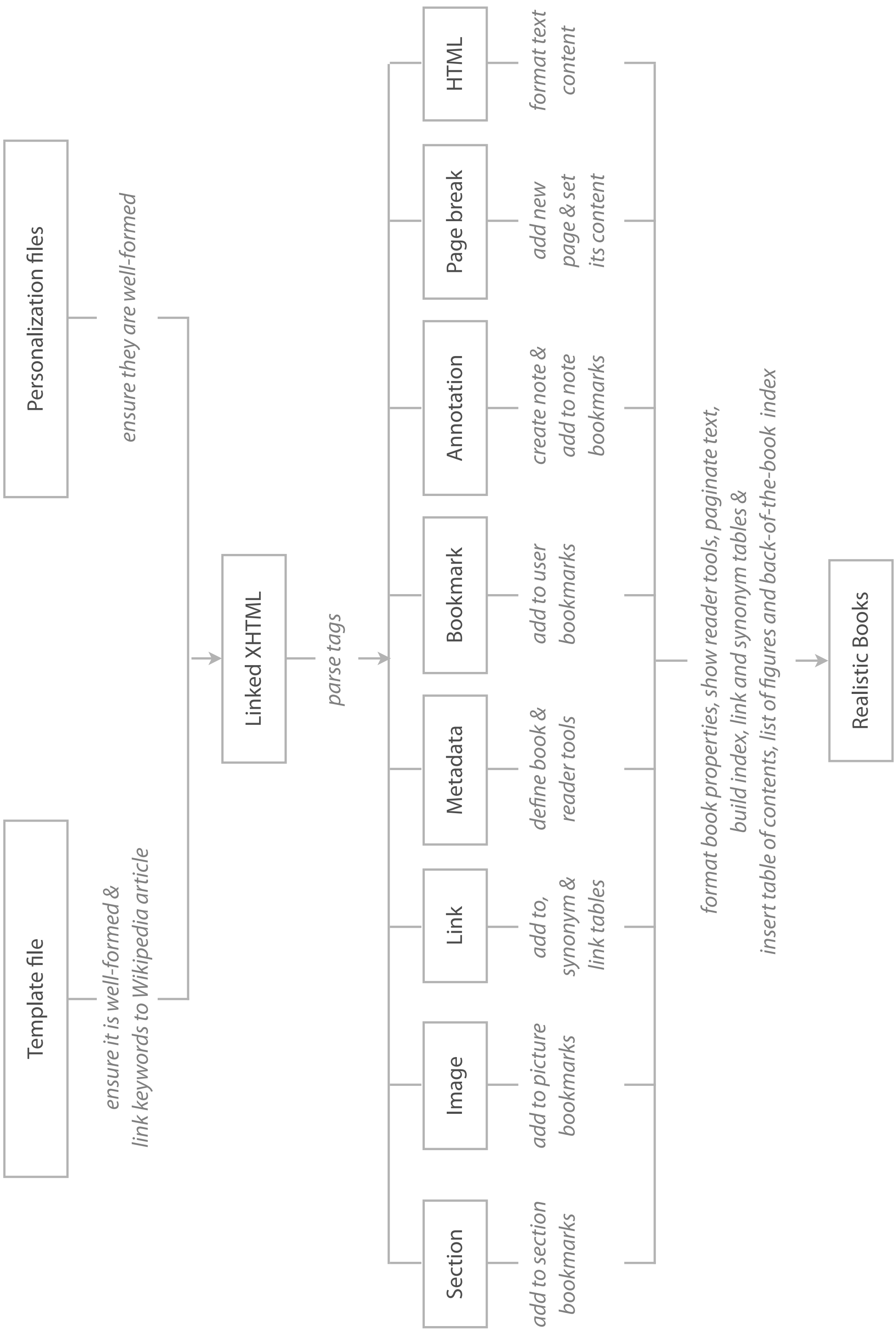


Figure 5.5: Process to automatically translate documents into a Realistic Books

during full-text search to increase the chance of getting a hit, even when the user utilises different terminology to that adopted in the book. Book designers can also manually add synonyms to the table as explained in Section 5.5.

Customisation

Once the document has been processed as described above, a Realistic Books is generated. The appearance of each component in the book space, reader tools areas and preview area are updated accordingly.

Designers can further customise the resulting book, as explained in Section 5.5. The designer toolbar contains a set of functions for modifying the book's appearance and the reading services associated with it—for instance, redefining its logical structure, associating different styles with section headers, and changing the page margins. After these changes, the template file is updated accordingly and the document is reprocessed according to the new specification. The cycle is repeated until the designer is satisfied with the appearance of each component of the Realistic Books.

5.3 Enriched document representation

The information explosion powered by the World-Wide Web has led to a huge increase in skim-reading (Goldsborough, 2000; Poynter Institute, 2000). People scan documents for keywords, tables and illustrations in order to obtain an overview of their contents. Only on finding something that catches their attention do they read in depth.

A reader's comprehension depends largely on their mental model of the text (Mandler and Johnson, 1977; Shneiderman, 1989). The amount of background materials that needs to be provided by a document depends on the reader's prior knowledge. It is impossible to select background material that suits the needs of both high and low knowledge users. Studies by Berkenkotter and Huckin (1993), Crane (2002), Dillon et al. (1989), Goldsborough (2000), and Murray (2002) show that when participants are unfamiliar with the subject domain or the language used in the text, they prefer reading on a computer over reading on paper because they can immediately look for additional information from an electronic dictionary or encyclopaedia, which allows them to read faster and understand the material better.

To support these activities, navigational, definition and semantic links can be automatically added to Realistic Books. *Navigational* hyperlinks can be automatically generated when section and image tags are specified in the template file. Each entry in the table of contents and list of figures is a navigational hyperlink that points to the beginning of a section or a figure. Page numbers indicate the location of each entry and the length of each section. In contrast, definition and semantic links are generated before the document is paginated, and they are listed in the automatically generated back-of-the-book index. A *definition* link points a reader to additional background information that explains the linked term, and mousing over it displays a short description of the term. These links make it easier for people to quickly determine the meaning of a word or a phrase. *Semantic* links facilitate rapid browsing of related topics in the document. When readers search for a concept or select it from an index, all pages mentioning the concept are marked with a bookmark tab, as illustrated in Figure 4.8. These tabs are the semantic links that connect related passages. Mousing over them shows the page number and an excerpt from the first matching fragment on the page.

Realistic Books uses Wikipedia as the basis for definition and semantic links. This section first provides a brief overview of Wikipedia and the organisation of its contents. Next, it shows how Wikipedia can be used as a resource to automatically generate definition and semantic links in any text document.

5.3.1 Wikipedia

Since its launch in 2001, Wikipedia has became the largest and most visited reference site on the Web. It is a free on-line encyclopaedia that allows people everywhere to write collaboratively. It contains 14 million articles in 260 languages, over 3 million of which are in English. Around 65 million people visit Wikipedia monthly. Because anyone can add or edit its content, the number of articles is growing rapidly, and many mistakes are corrected almost immediately. Wikipedia compares favourably in both coverage and accuracy with the Encyclopaedia Bri-

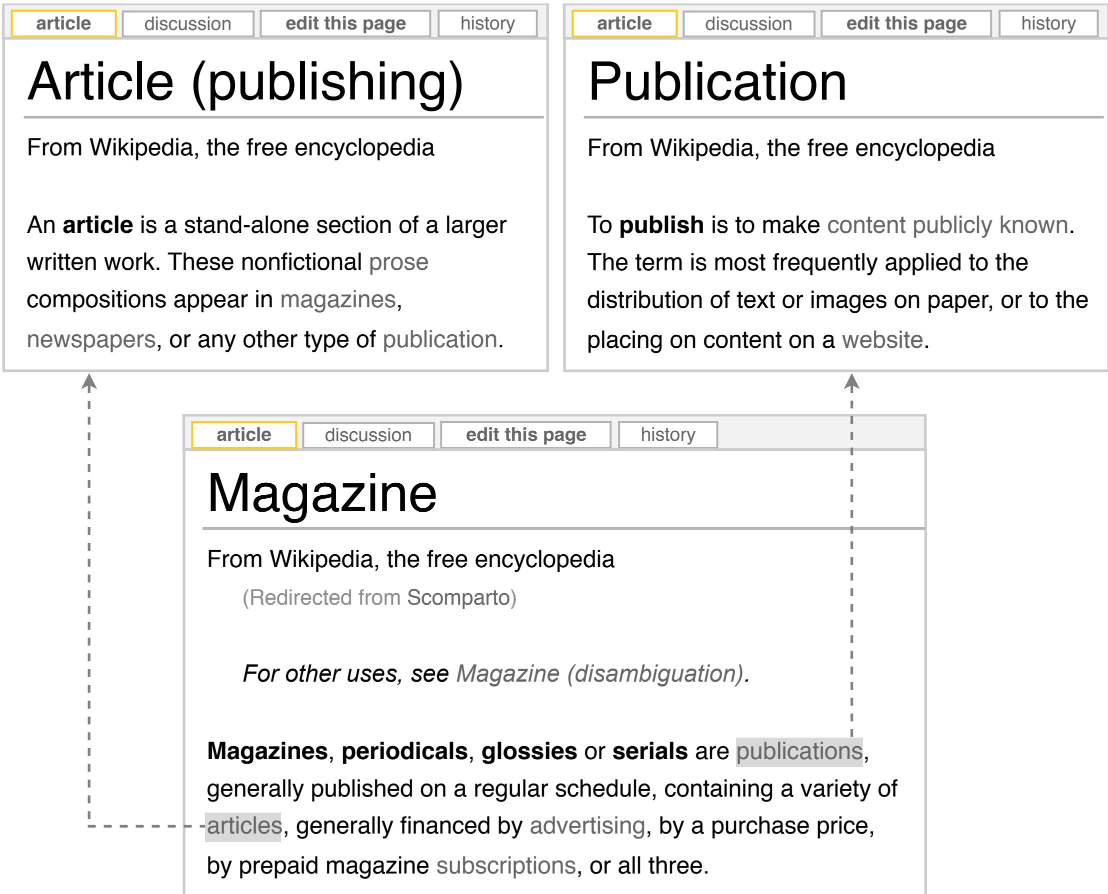
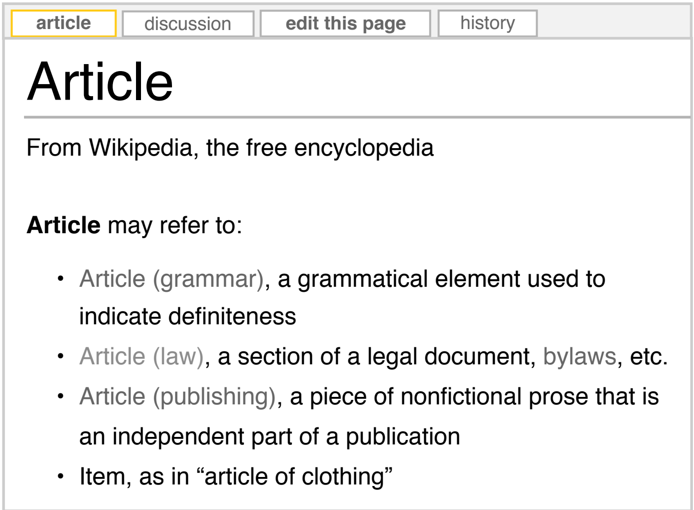


Figure 5.6: A sample Wikipedia page for *Magazine*, with links to related articles

tannica, an encyclopaedia written by acknowledged experts that is considered to be one of the most authoritative reference source in the English language (Giles, 2005).

Each article in Wikipedia defines and explains an entity, event or concept. At least one category is assigned to each article, and categories can be assigned to other categories. The category structure is a graph with a loose hierarchy. A Wikipedia article contains manually added links that connect salient terms to pages that explain them. These links point to other Wikipedia articles that provide users with a deeper understanding of the topic.

Figure 5.6 shows an example of a Wikipedia page. There is often more than one name to describe a concept—for instance, *magazine* is also referred to as *quarterly*, *serial magazine*, *scomparto*, amongst others. The most commonly used term



article | discussion | edit this page | history

Article

From Wikipedia, the free encyclopedia

Article may refer to:

- Article (grammar), a grammatical element used to indicate definiteness
- Article (law), a section of a legal document, bylaws, etc.
- Article (publishing), a piece of nonfictional prose that is an independent part of a publication
- Item, as in "article of clothing"

Figure 5.7: Disambiguation page for the term *Article*

(*magazine* in this case) is chosen as the article title, and readers are automatically redirected to it when they use one of the alternative names. Wikipedia uses the same technique to handle capitalisation and spelling variations, abbreviations, synonyms, plurals, technical terms, and common variants.

Words frequently have more than one possible meaning. For example, the term *article* can refer to a word that indicates the type of reference being made by a noun, as described in the Wikipedia article called *Article (grammar)*, or to a stand-alone section of a larger written work, as described in *Article (publishing)*. When authors add a link to a key term, they determine the meaning of the term based on the context where it occurs, and set the link target to the corresponding Wikipedia article. Furthermore, each ambiguous term in Wikipedia has a *disambiguation page*, illustrated in Figure 5.7, that lists meanings of the term from which the intended article can be selected.

5.3.2 Linking documents to Wikipedia

Definition links are intended to provide readers with immediate access to the meaning of a term in the document. To help users skim faster, only keywords are linked. Because Wikipedia is an open project and the largest online encyclopaedia,

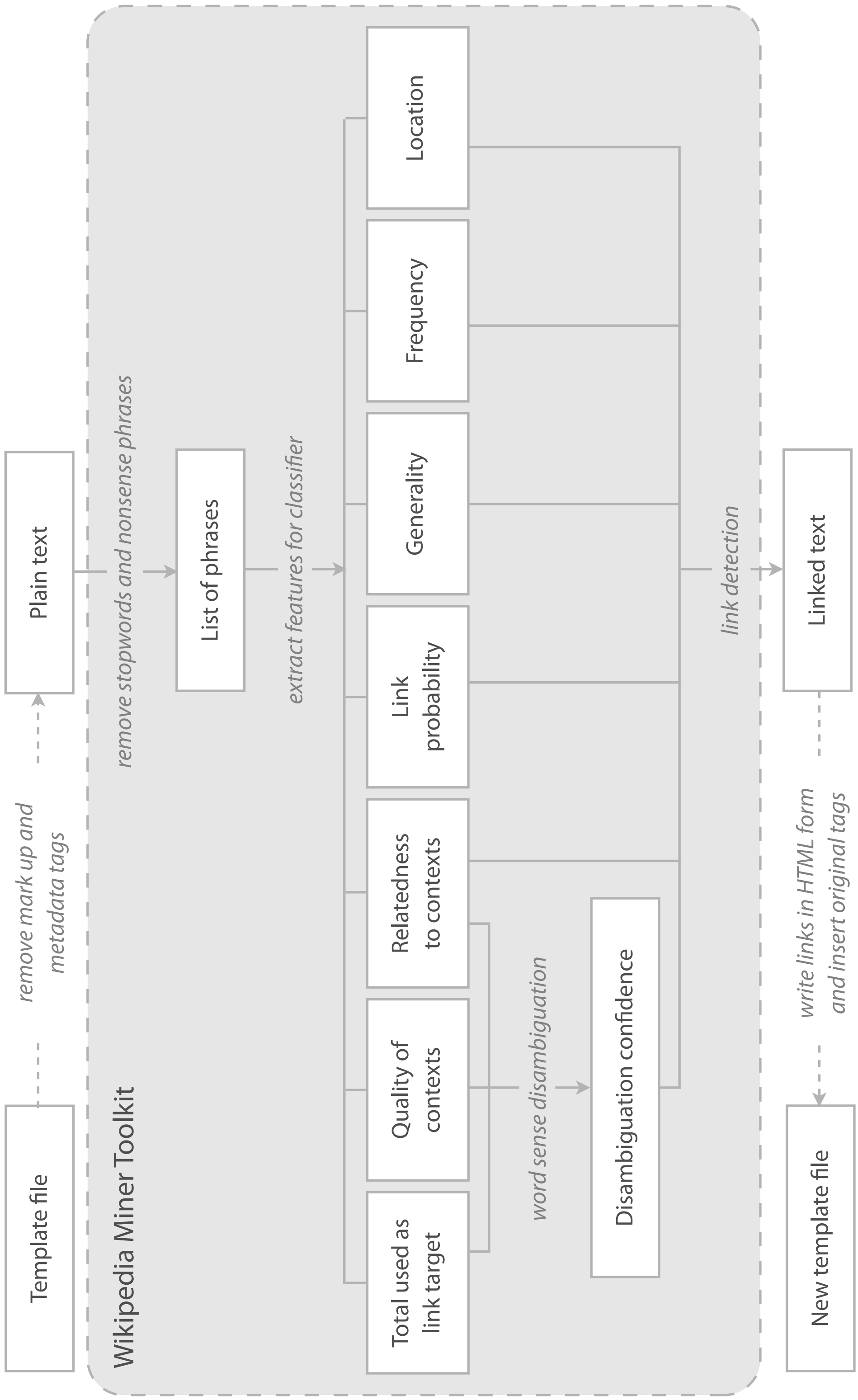


Figure 5.8: Process of automatically linking a book to Wikipedia

the description of terms can be acquired from Wikipedia.

In recent years, several researchers have utilised Wikipedia's semi-structured characteristics to facilitate the analysis of textual content for problems such as document classification (Gabrilovich and Markovitch, 2005; Wang et al., 2007), named entity disambiguation (Cucerzan, 2007), semantic relatedness computation (Gabrilovich and Markovitch, 2007; Krotzsch and Ponzetto, 2006), keyword extraction (Mihalcea and Tarau, 2004; Wan et al., 2007) and "wikification" (Mihalcea and Csmoai, 2007; Milne and Witten, 2008; Zhou et al., 2009). Wikification is the process of automatically extracting significant terms from a given text, and augmenting each term with a link to an appropriate Wikipedia article. The tools developed for this process can be used to automatically create definition links in any text document. The Workbench utilises the Wikipedia Miner toolkit developed by Milne and Witten (2008) because it performs the best of the other toolkits developed.

As mentioned in the previous section, each possible meaning of a term is represented by a single Wikipedia article. When adding a hyperlink to a term, Wikipedia authors have carefully selected the correct article that describes its intended meaning. Wikipedia Miner learns from these manually added links to disambiguate terms and to identify which terms in a document should be linked.

Figure 5.8 illustrates the process of automatically generating definition links. Given a book's template file, the Workbench removes any HTML and metadata tags and passes the clean text to Wikipedia Miner. The wikification process starts by removing stop words from the text and disambiguating remaining terms. For each possible sense of a term, Wikipedia Miner calculates the probability that a sense is correct based on the number of times the Wikipedia article for that sense is used as a link destination in Wikipedia, the quality of the context terms, and the semantic relatedness of the sense article with all of the context articles.

The next task is to identify which terms in the document should be linked. In Wikipedia Miner, a term is considered to be link-worthy if: it is used as a link in many Wikipedia articles; is closely related to the main topic of the document; is mentioned numerous times in the text; and there is a high confidence that the selected meaning of the term in the passage is correct. Furthermore, Wikipedia Miner prefers to add links to terms that readers are more likely do not know,

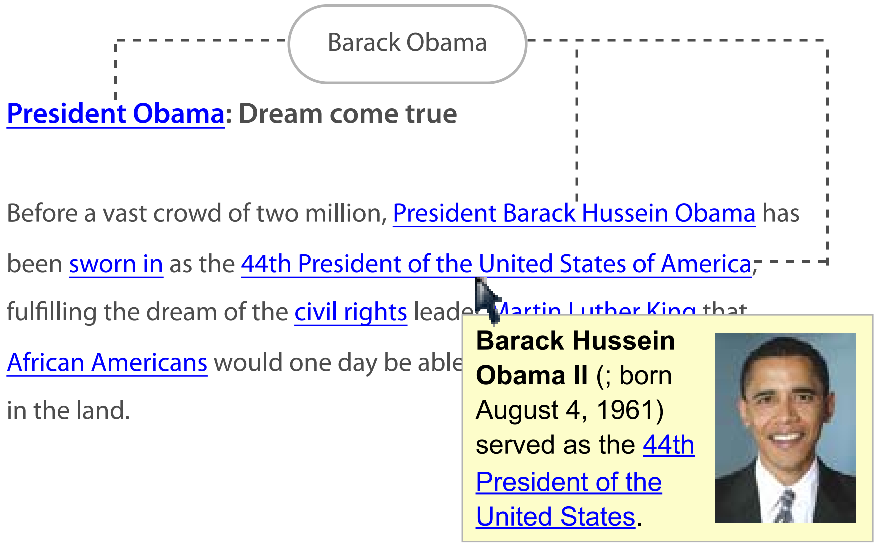


Figure 5.9: A news story excerpt with links to related Wikipedia articles

especially if they are located near the beginning or end of the document. Once Wikipedia Miner has finished identifying the important keywords in the text and linked them to the correct Wikipedia article, the annotated text is passed back to the Workbench. All Wikipedia links are converted into HTML hyperlinks tags. For each key term, the Workbench obtains the first paragraph and the accompanying image of the Wikipedia article, and sets them as the tooltip text for the link. This provides readers with immediate access to the definition of these keywords. Finally, the original HTML and metadata tags are inserted back into the new template file.

Figure 5.9 shows a text document that has gone through the wikification process and the tooltip display that is shown when a user hovers over a definition link. Section 5.5 shows how designers can customise the appearance of these hyperlinks.

5.3.3 Automatic aliasing

Most electronic text search features implement only a basic literal match, possibly with case-folding. Their success depends on the user typing in the right words to obtain the desired information. However, people use a variety of words to refer to the same thing. An evaluation conducted by Furnas et al. (1987) with over 500 participants found that any one individual usually thinks of at most six different terms to describe a topic, whereas more than 100 words may be suggested across all people.

This is exactly the problem faced by subject indexers when deciding on terms for a back-of-the-book index. Even trained indexers who abide by carefully formulated rules generate a wide range of different terms (Bates, 1986; Cooper, 1969; Marley, 1984). Ideally, subject indexes should contain all terms that readers might conceivably use when seeking information in the book. However, entries are not chosen based on the infinitude of possible search terms. Indexers strive for an index that is consistent in internal structure, logical, accurate, and can be defended amongst other indexers.

Good indexes bridge the author's perspective of the subject and the set of terms describing the book's contents with the range of keywords that readers might use in an enquiry. To make the connection, indexers use words that are not specifically mentioned in the document but are recognisable to the majority of readers. On average, these bridging terms represent 10% of the entries in an index (Weinberg, 1996).

When readers find a term in a book's subject index, they learn where the term and the topic it describes are mentioned in the book. In contrast, with full-text search in electronic documents, when readers search for a term, they will only know where the term is mentioned. Furthermore, electronic search facilities rarely include a bridging words list. The success rate for finding a match on the first few tries is 10–20% (Furnas, 1985). Because users often only look at the first few matches, it is important to improve the likelihood of finding the information on the first few tries.

Experiments by Furnas et al. (1987), Gomez et al. (1990) and Good et al. (1984) have shown that users find target information faster when many alternative terms have been assigned to the information. This can yield a success rate of 50–100%. Whether the words are acquired through manual or automatic indexing, or they are considered to be the most important by the experts, the source and quality of the words have little effect on the reader's searching performance. However, it is expensive and tedious to store every synonym of every word in the

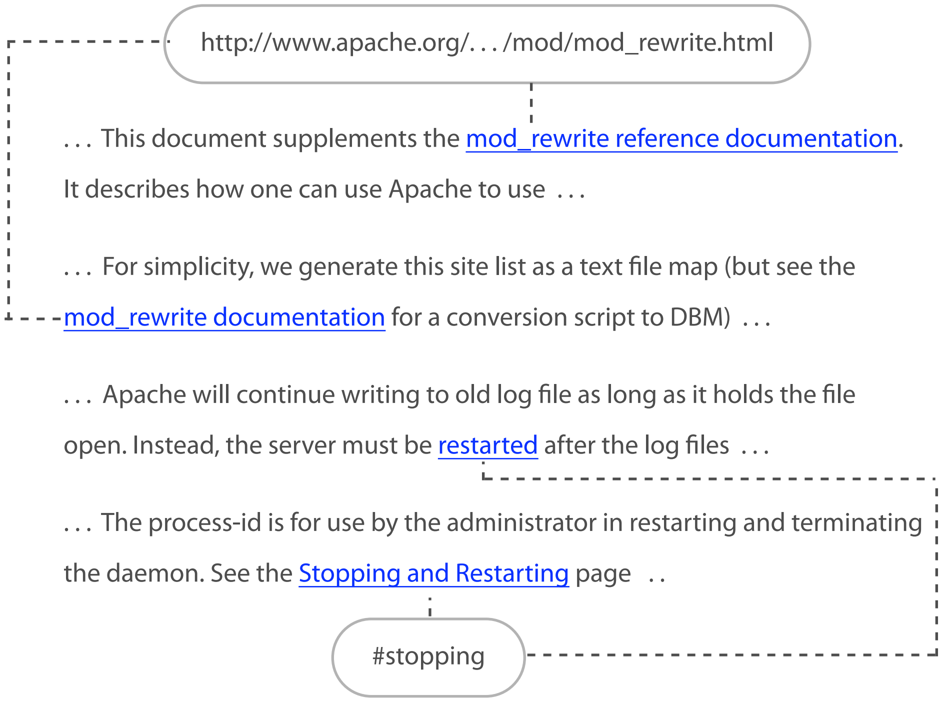


Figure 5.10: Excerpt of an Apache documentation file and the link destinations

document.

Another approach to this aliasing task is to have computers automatically analyse a document and assign each topic with a word that real readers most frequently use to identify the topic. The success rate of finding a topic with a search system that understands a single empirically optimised term for each topic in a document is almost two times better than a system that only understands a single access term chosen by the book designer for each of the topics (Furnas, 1985).

Based on these findings, we decided to use the definition links generated in Section 5.3.2 to build a synonym table automatically. Terms that are linked to the same Wikipedia article are considered to be synonymous. For example, Figure 5.9 shows a news story about the day President Obama took his oath of office, which has been automatically augmented with links to relevant Wikipedia articles. Because *President Obama*, *President Barack Hussein Obama* and *44th President of the United States of America* are all linked to the article *Barack Obama*, these four terms are considered to be synonyms. Although the phrase

Barack Obama is not used in the text, it is inserted into the synonym table because Wikipedia contributors choose it as the term for the topic. When a reader enters one of these terms in the *find* box, the search returns the location of the term and its synonyms—creating semantic links that connect related but distinct passages.

Book designers can also apply the same assumption to internal and external hyperlinks in the document that are not references to Wikipedia articles. With this option set, every word or phrase with the same link destination is treated as a synonym for the purposed of search. For example, in Figure 5.10 the terms *mod_rewrite reference documentation* and *mod_rewrite documentation* both target the same URL and are therefore considered synonymous for the purposes of search. The same is true for the words *restarted* and *stopping and restarting*.

5.4 Pagination

Pagination is a vital step in the conversion of electronic text formats such as HTML or plain text into a book, and is performed automatically by the Bookmaker's Workbench. It is defined as the process of laying out parts of a document into pages (Rubinstein, 1988). Of course, text cannot be divided into pages willy-nilly: for satisfactory results there are many typographical constraints that must be respected (Martin, 1989).

Criteria

Margins frame the content of a page. They save the text from damage, enable readers to flip pages, and annotate without obscuring the text. The objective of margin design is to enhance the book's utility and ensure that every pair of facing pages produces a pleasing aesthetic effect when the bound book lies open. Margins normally occupy up to 40% of the page, and the organisation of the text is often adjusted to prevent unacceptable situations arising during pagination as described below.

The content of facing pages should have approximately the same height. This makes it necessary to look ahead during pagination to ensure that the vertical extent of every left page matches that of its right-hand counterpart. To improve the layout of the document, paragraph breaking and pagination are usually performed in tandem

When inserting a page break, widows and orphans should be avoided. A "widow" occurs when just one (or perhaps two) isolated lines from a paragraph appear at the top of the next page. Likewise, an "orphan" occurs when just the first one (or perhaps two) lines in a paragraph appear at the bottom of a page. A title or header line that is left alone without an accompanying paragraph is also called an orphan.

Extra space within words can degrade the quality of text presentation and distract the reader's attention. Repeated hyphenations on consecutive lines should be avoided, as should hyphenating across a page break, which interrupts the flow of reading by forcing the reader to look ahead.

Tables, figures, footnotes, and marginal notes should be placed near their first point of reference, and preferably on the same page. Cross-references involving page numbers must be modified every time a change occurs due to repagination.

Paginating Realistic Books

Finding an optimal pagination for a document that satisfies a formal specification of constraints for the criteria sketched above is computationally expensive, although heuristic methods help greatly. However, implementing pagination algorithms is not the focus of this research, so the Bookmaker's Workbench applies just a few basic rules during the pagination process. If necessary, book designers can make further modifications manually.

First, a book's template file is parsed and the text is formatted according to the style described in the template—for example, section titles are given an appropriate typeface and bookmark tabs are attached to the relevant pages. Link source and destination targets are stored in the hyperlinks table, both for hyperlinks that appeared in the original document and ones added by the text mining processes. Line and page breaks are created according to the guidelines explained below. At the end of the pagination stage, the table of contents, list of figures, and back-of-the-book index are automatically generated, each entry being expressed as an internal hyperlink, and inserted into the appropriate position in the document.

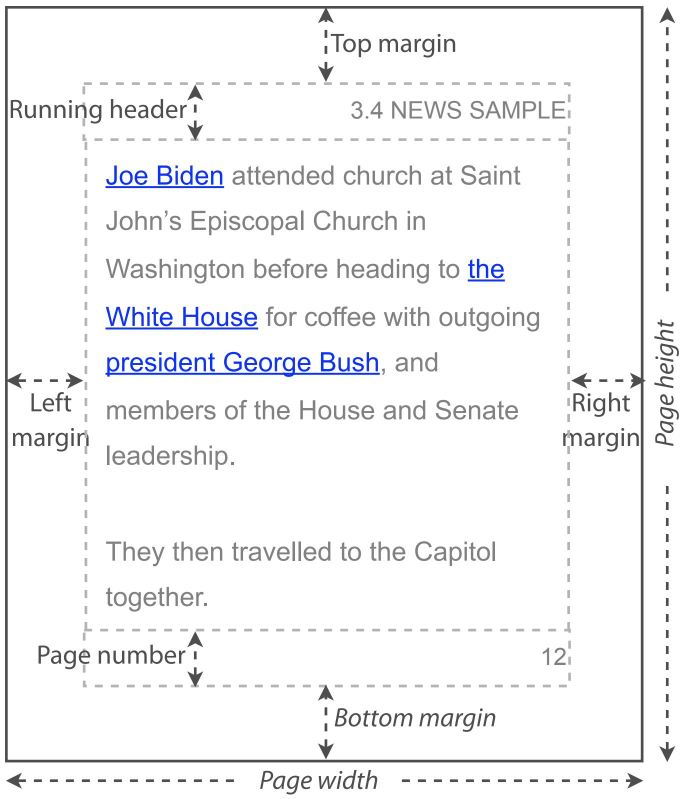


Figure 5.11: Page layout in Realistic Books

Figure 5.11 shows a typical page layout. By default, page margins are all set to 50 pixels, but Section 5.5 describes how all the relevant parameter values can be specified by the book designer. The size of the main content of the page is calculated based on those parameters, as follows:

$$\begin{aligned} ContentHeight &= PageHeight - TopMargin - BottomMargin - \\ &\quad RunningHeaderHeight - PageNumHeight \\ ContentWidth &= PageWidth - LeftMargin - RightMargin \end{aligned}$$

The body text of the document will never expand outside this region. A 12-point font is used for the page number and running header.

A page break is introduced before the beginning of each chapter, and every chapter begins on a right-hand (odd-numbered) page. If the end of a chapter falls on an odd-numbered page, a blank page is inserted to ensure that the next chapter starts on the next odd-numbered page.

When a paragraph does not fit into a page, orphan and widow rules are applied. Section titles are always located on the same page as the section's first paragraph. A paragraph is moved to a new page when fewer than $OrphanCount$ lines remain on the current one or when fewer than $WidowCount$ lines would flow on to the next page.

All pictures are automatically scaled so that their width and height do not exceed $ContentWidth$ and $ContentHeight$ respectively. A zoom facility allows readers to view images at their original size, to solve the problem of poor legibility at reduced sizes. If insufficient space remains on the current page for a figure, it is placed on the next one. Figure captions appear on the same page as the illustration.

In Adobe Flash, the same piece of text may be rendered on different occasions with slightly varying space and letter size. This means that the location of line breaks can change when the document is reloaded, altering the paragraph's height and requiring the entire book to be repaginated to ensure that the text does not overflow its allocated space. This is highly undesirable, because the pagination process is expensive; also, it gives books an unstable appearance. Consequently the Bookmaker's Workbench inserts explicit line breaks into the text to guarantee the same paragraph size whenever the document is reloaded.

Words are not hyphenated across line or page breaks. Successive words in a paragraph are inserted into a line until the text width exceeds $ContentWidth$. When a word does not fit, the line is broken and that word begins the next line. Because an explicit line break is inserted after each line, the text is unjustified. This has the side effect of avoiding rivers of space, because the space between words cannot exceed the space between lines.

During the line breaking process, full-text indexing is performed. All words in the book—and all their synonyms—are indexed and stored in the index table. Index terms are case-folded; however, stemming is not performed. The "redirects" that have been recorded manually in Wikipedia provide a far more reliable basis for full-text searching than any automatic stemming algorithm, and as noted in Section 5.3.3 these are incorporated into the synonym table and used during full-text search.

During pagination, any hyperlinks that occur in the manuscript are interpreted and recorded. Designers can create a link to a particular page in the book or to an external source, as described in Section 5.5.1. Figures, chapters, paragraphs, and sections can all be referred to by unique link identifiers rather than by page number, to cater for the dynamically paginated nature of Realistic Books.

Whenever the book is repaginated, the page numbers of link destinations may change but the link identifiers remain the same. During the pagination process, every link identifier and its associated page number are stored in a table. Whenever a reader clicks an internal link, the hyperlink table is consulted to determine the appropriate page number and the book is opened at that page. Whenever the book is repaginated, the table is updated. Designers do not need to manually update link destinations unless they refer explicitly to page numbers. Because the locations of user-created annotations and bookmarks are specified in terms of page numbers, they need to be moved manually to the appropriate page whenever new material is added to the book, as mentioned in Section 5.5.1.

5.5 Customisation

Once the document has been paginated, designers can use the tools provided in the Bookmaker's Workbench toolbar to manipulate the appearance of the book and alter the reader services to satisfy the needs of the targeted readers. For instance, they can change the page size, add new links, or alter the font used in the main text. These changes are made to the book template and take effect whenever the book is generated.

There are two types of change that designers need to be able to perform after the automated document processing phase: content organisation and document appearance. The first alters the contents of book pages, marks important components, and assembles the pages into a book. The chapter and section hierarchical structure that forms the basis of the common notion of a book is maintained, and the hypertext perspective is defined for online consultation. The second type of change incorporates pagination and visual features like title and hyperlink styles, and adds functions to make the document more useful. One advantage of electronic books over conventional ones is that different versions of the book can easily be produced for different classes of readers (e.g., large-font versions) without the need for manual reprocessing.

5.5.1 Organisation of content

The repository area stores the entire contents of a book, including text, images, and multimedia. Having acquired the necessary material, the designer assembles each page of the book by dragging files from the repository area and dropping them on to the appropriate page.

Inserting a text document appends it to the existing text in the page content area. Text is paginated according to the rules specified in Section 5.4. Appropriate typographical styles are applied to chapter or section titles, figure captions and hyperlinks. New pages might be added as a result of this process; they typically contain body text and the automatically generated running headers and page numbers.

When designers place an image on to a page, a configuration panel appears in which they can specify the caption and whether the illustration should occupy a full page. Once the figure has been added, the book is repaginated starting from the page containing it. If an image is not set as a full-page illustration, it is appended to the body text and scaled so that it, along with its caption, remains within the page text area. If insufficient space remains on the page, both illustration and caption are moved to the next one.

Full-page illustrations are centred on the page and proportionally scaled to fill it. The automatically generated running header and page number elements are suppressed. A single book page can contain more than one page image, in which case each image represents a different version of the page. Readers see one version at a time. Initially they see the first version, and can move to later ones (and then back to earlier ones) using the *animation* buttons on the toolbar or the arrow keys in the keyboard, as explained in Section 4.1. For example, pressing the down-arrow key replaces the current page image with the next added page image. Book designers achieve this effect simply by adding new full-page illustrations to the page.

Once designers have determined the sequence of pages and the content of each one, they can label parts of the document. They can add notes, drawings and scribbles. They can attach bookmarks to particular pages or passages. There are three types of bookmarks: section, illustration and user. The first two are automatically generated, while the third is defined by the designer, or by readers. Every bookmark has a unique identifier, target, tooltip text and colour.

If a portion of text is selected when a user-defined bookmark is added, the target of the bookmark is the selected text; otherwise, it is the page itself. A bookmark's *settings* button opens a configuration panel within which the bookmark's identifier, tooltip text, colour and visibility can be modified. Although neither readers nor designers can see a hidden bookmark, designers can create hypertext that links to its target.

Section and illustration bookmarks are automatically attached to pages containing section and image tags respectively. The former are named according to section number, while the latter are named according to figure number. The tooltip text is the section title and image caption respectively. Further text can be inserted manually by selecting the bookmark's *settings* button.

New chapters and sections can be created by clicking the *add section* button (), which opens a configuration panel within which the section's title, level, and visibility can be specified. Titles are formatted according to the section's level. Regardless of their visibility status, all sections are listed in the table of contents and marked with section tabs. Once they have been added, the book is repaginated, starting from the modified page.

Designers can insert links to bibliographic citations, or references to other chapters, sections and figures. New hyperlinks are created by selecting a portion of the text and clicking the *hyperlink* button (). If the text contains no other links a new link is created and a configuration panel opens in which its destination, tooltip text, status and type can be entered. The destination can be a URL, a page number, or a bookmark identifier. Tooltip text is displayed when the mouse pointer hovers over the link. If the link status is active and the destination is not empty, clicking on the link takes the reader to the link's destination.

As mentioned in Section 5.3, Realistic Books can contain three types of hyperlinks: navigational, definition and semantic. By default, all manually added hyperlinks are navigational. If designers set the link to be a definition or a semantic link, the linked term will be included in the synonyms table and in the automatically generated subject index. Clicking on any of these links takes readers to additional background information in other parts of the book or to external sources. Clicking on semantic links also shows the readers all the term occurrences in the book. All hyperlinks in the automatically generated subject index are semantic links. The appearance of the new hyperlink follows the design style defined for the specified link status and type.

5.5.2 Document appearance

Once the contents of each page, the sequence of pages and the hierarchical structure of the book have been defined, the next step is to define and format visual cues. Clicking the *options* button ( ) opens a panel in which designers can modify the appearance of the book, the list of enabled reader services, and the document layout.

Selecting the *document* button ( ) displays a configuration panel for specifying parameters used for visualising the book. These are shown in Table 4.1: they include the colour and the type of paper used for the covers and pages, the horizontal and vertical offset between each page in the stack of page edges that visibly flank the current double-page spread, the flipping speed and document magnification factor. When readers invoke the book, it can be configured to open initially at any particular location, which the designer specifies by page number or bookmark identifier.

The document configuration panel is where designers control the generation of the table of contents, list of figures and subject index. When selected, section and image tags in the book are used to create the first two, while definition hyperlinks and semantic hyperlinks are used for the third. Section 5.3 explained how these hyperlinks can be automatically generated.

Designers of electronic books strive to foresee the needs of their readers. By default however, the Realistic Books system provides a set of functionalities to satisfy a generic situation. To keep the appearance of the books as simple as possible, for the target audience, the features that are included for any particular book should be tailored to the anticipated need of readers. This can be achieved by pressing the *tools* button () which shows what tools can be provided to the readers—navigation, search, zooming, printing, annotation, bookmarking, and animation—and the associated checkboxes dictate which tools will be displayed. For example, the designer may disable the print, annotation and bookmark tools for copyright reasons. The size of the application file depends on the tools included in the book.

The *layout* button () also assists this design task, and brings up a panel in which designers can configure the page size, margins, allowable number of orphan and widow lines, and whether running headers or page numbers are shown. By default, the page aspect ratio is that of A4 paper: changing this affects all pages and the text is automatically repaginated. (Because the pagination process could take some time to finish, designers are given the choice of activating the process immediately or repaginating later.) Page breaks can be inserted manually, after which the document should be repaginated.

By default, the main text is assumed to begin at the first chapter, and page numbering begins at this point—designers can override this by specifying manually where the main text starts. Preceding pages are marked with Roman numerals, and succeeding pages with Arabic ones. Pages that are entirely blank are not numbered.

When a book is generated, predefined text formats are automatically applied to different elements of the text: running headers, page numbers, section titles, hyperlinks, and image captions. The designer changes these by opening up a configuration panel for the currently selected element in which its typeface, size, colour, line spacing, and paragraph alignment can be modified. The changes can be applied individually or to all instances of the element throughout the document.

When satisfied with the appearance of each component in the book and the set of services it provides, the designer saves the book, which updates the template file and produces a final version of the book from which the editor tools are suppressed.

5.6 Summary

The amount of available electronic documents keeps growing with the availability of huge online archives of electronic books, blogs, wikis and multimedia material on the Web. Studies have shown that the most popular reasons for using electronic books was for finding relevant content (Dillon et al., 1989; Olsen, 1994; Wright, 1993). Simply putting books online is not sufficient, an electronic document application should take advantage of the characteristics of the digital medium and go beyond the affordances of paper documents.

Although orders of chapters and sections in an article reflect a carefully designed way of learning, information presented in it does not need to be accessed in a linear way. Rather than writing multiple documents intended to appeal to distinct group of readers, hyperlinks that guides each reader through a series of concepts can be inserted into an electronic book (Egan et al., 1989; Engelbart and English, 1968; Nelson, 1967). However, people are confused when they cannot determine the length of the material or orient themselves around the document.

The physical design of a printed book provides frequent cues describing the reader's place in the document. This thesis argues that if the structure and features of physical books are preserved in the hypertext document, a user's reading experience and performance will be enhanced. The Realistic Books application was developed to investigate this hypothesis. Chapter 4 shows how an electronic document is presented as a Realistic Books and how readers can interact with it.

The Bookmaker's Workbench transforms an electronic document into a Realistic Books under the control of the book's designer. This chapter has explained the two stages involved in this transformation: automated document processing and customisation. First, text is paginated and can be augmented with hyperlinks that are automatically generated through the use of Wikipedia. Book designers can then add various features to the document, which may involve repagination, and provides readers with suitable reading tools to perform their tasks. The next chapter evaluates both the books themselves and the process of producing them.

6

Evaluation

At a time when the place of the book in the digital world is changing, it is important that the appearance, content and technology designs of electronic texts are explored thoroughly and delivered to the end users in a form that maximises their usability. As mentioned in Section 1.2, some researchers suggest that electronic books should be designed to be as similar as possible to paper books so that people can apply directly the reading skills they have already learned (Benest, 1990; Landoni et al., 2000; McEneaney, 2000). Others believe that training and practice can help users overcome their difficulties with screen reading, regardless of the document presentation (Kol and Schcolnik, 2000). However, both these claims need to be investigated further.

There is little quantitative empirical data evaluating whether a simulated book model can improve users' reading performance compared to other online document representations or to physical books. According to Landow (1996), the findings of the few studies that do compare reading on screen with reading on paper are inconclusive because participants were presented with electronic text that used smaller characters and page margins to that used in paper form. Furthermore, the tasks designed for these studies controlled so many variables that the resulting experimental tasks bore little resemblance to real world activities that people routinely perform while reading (Dillon, 1992). As a result, it was not clear which of the many features associated with the new online document representations evaluated might have been effective.

Chapter 4 introduced the conceptual model of Realistic Books, which maintains many features of printed books—page edges, page turning, annotations and bookmarks—and adds several electronic functions such as searching, hyperlinks and multimedia. Chapter 5 described a Bookmaker's Workbench for processing electronic documents to produce Realistic Books in a semiautomatic way, with an appropriate interface and reader services. The focus of this chapter is on evaluating the interface design and reader tools, to investigate the hypothesis of this thesis and define a set of best practice guidelines for designing electronic books. The evaluation also implies an indirect evaluation of the environment used for producing books.

One way to ensure that a new electronic book system actually improves people's reading performance is to compare performance using the new system with commonly used document representations like HTML and Adobe PDF, and with physical books, completing tasks that represent likely situations in which the system will be used (Good et al., 1984; Gould, 1988). The typographical features, logical structure and page layout of the electronic text should all be equivalent to the paper material. Moreover, evaluations must be designed to show not only whether the new system is better than the others, but (if so) to identify which of its properties are responsible for the improvement.

Reading is a highly practised activity, and includes a wide range of document-based activities that serve many different purposes (Dillon and McKnight, 1990; Levy, 1997; Marshall et al., 1999; McKnight, 1996; Schilit et al., 1999). For example, according to Paris et al. (1991), readers generally preview the document's contents first before reading it thoroughly. They then inspect the text to identify the main ideas, evaluate concepts and make inferences. Text inspection involves looking forward and backward in the article, and using context to understand difficult information. They look for section headings, page layout and special fonts to distinguish important information. Sometimes, key passages are highlighted, pages are bookmarked, and notes are added to the page. Finally, readers summarise what they have read.

To determine which features of Realistic Books support the readers' needs, a series of usability studies was conducted. It compared how people browse, find information, and personalise documents in Realistic Books, Web browsers, Adobe PDF Reader and printed forms. Evaluating electronic book readers such as Amazon's Kindle and Sony's Digital Book lie beyond the scope and resources of this research. However, this may not be a significant shortcoming, because their display resembles the PDF format—without its scrolling capability.

System responsiveness may influence a user's reading performance. This chapter first investigates the time required by the Realistic Books application and the Bookmaker's Workbench to complete a user's request. Then, the overall experimental design and criteria employed in the user studies are described. Finally, the results and the main conclusions drawn from the evaluation are reported.

6.1 System responsiveness

System responsiveness is the speed with which a system responds to a user's action, and is responsible for a good deal of user satisfaction and engagement (Hansen and Haas, 1988). When a user is in the middle of an action and has to wait for a specific reaction from the system, low responsiveness can be frustrating, introduce errors and reduce concentration. One source of engagement is the satisfaction of seeing the system react. All the timings were measured on a Windows computer with 2.8 GHz Intel Pentium 4 processor, and an Apple laptop with 2.2 GHz Intel Core 2 Duo processor.

Loading documents

When a user opens an electronic document in a Web browser, the browser loads all of the document's contents. However, with linearized form of PDF documents and Realistic Books, only the first few pages are loaded. As the user progresses through the document, the few pages following the currently viewed page are loaded. There are two ways of viewing a PDF document: through a Web browser's plug-in , and using the Adobe Reader application. The former method requires the plug-in to be loaded first before loading the document, and its start-up delay is about one second longer than the latter.

Figure 6.1 shows the average time taken by the Adobe Reader and Realistic Books applications to open books with a varying number of pages. To ensure that

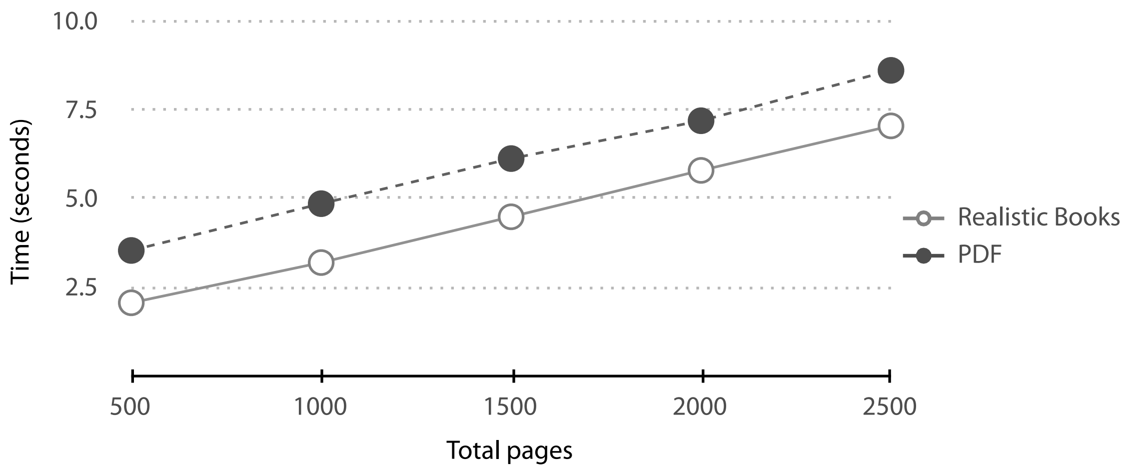


Figure 6.1: Average start-up delay for various numbers of pages

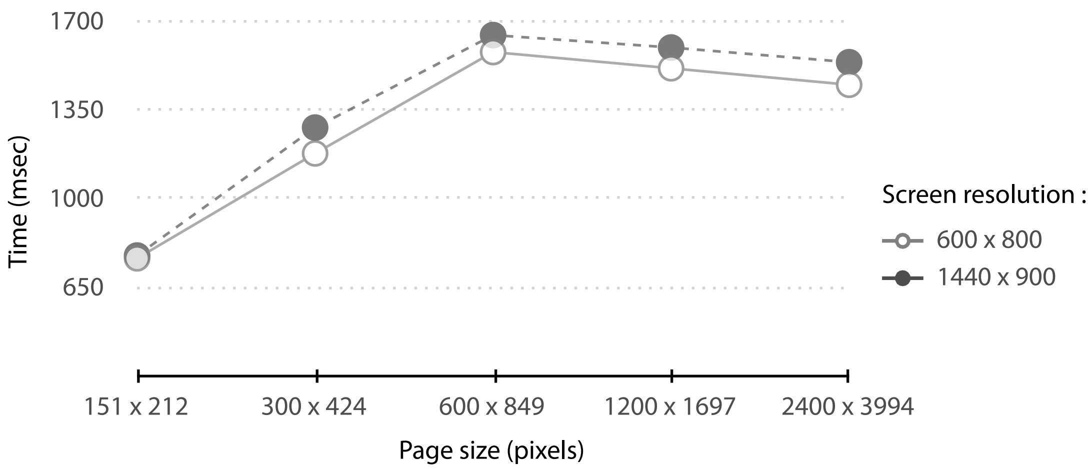
Page size (pixels)	Prepare (A)	Page turn (D)			Complete (E)	Total simulation time (A + D + E)
		Set position at a pose (B)	Draw pages at a pose (C)	Total page turn time (11 x (B + C))		
151 x 212	39	6	60	724	18	781
300 x 424	42	5	105	1212	14	1268
600 x 849	38	6	139	1600	13	1651
1200 x 1697	42	5	136	1546	14	1602
2400 x 3994	42	5	130	1489	13	1544

Table 6.1: Average page turning time (in msec) for various book sizes

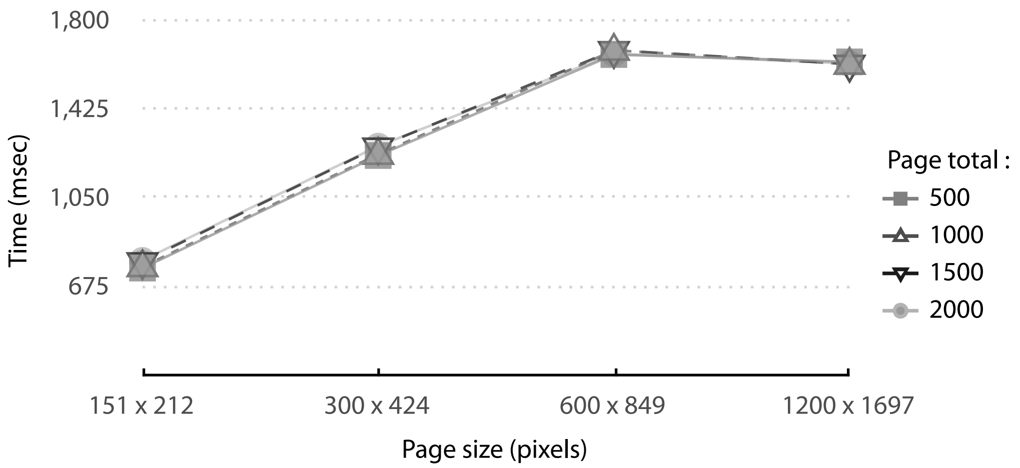
the times recorded are not influenced by the speed of the internet connection, all documents are viewed offline. Regardless of the document's length, the average start-up delay for PDF is about $1\frac{1}{2}$ seconds longer than that for Realistic Books. The average time required to load a PDF page is also half a second slower.

Page turning

As described in Chapter 4, a Realistic Books page turn involves three stages: *preparation*, showing the appearance of the book before the page turn and loading the four pages required for the simulation; *page turn*, computing and displaying the



(a)



(b)

Figure 6.2: Average time (in msec) to simulate a page turn : (a) in a 2000 page book at various screen resolutions, and (b) for various book sizes at 1440 × 900 screen resolution

appearance of those pages at each intermediate position; and *completion*, updating all the state variables. Table 6.1 shows the average time taken to complete each stage for five 2000 page books, and Figure 6.2 compares the average time required to perform this simulation at two screen resolutions and various total numbers of pages. The page turning speed is set to *medium*, i.e., there are 11 intermediate poses, and each page size follows the A4 page ratio of $1 : \sqrt{2}$.

Table 6.1 shows that the times taken to prepare the book for page turning

(Column A), to compute the appearance of a page at each pose (Column B), and to update the variables (Column E) are constant—independent of the screen resolution, the size of the page and the total number of pages. However, the time required by Flash to draw the book at each position (Column C) is influenced by the page size. The Realistic Books application relies on masking to simulate a page turn. As illustrated in Figure 6.2, the bigger the page, the longer it takes to mask. Because Flash only masks the parts of objects inside the user's viewing area (1100 × 1006 pixels), as shown in Table 6.1, the time required to display the book for each pose is no greater than the time required to mask the entire viewing area (140 msec), regardless of the page size.

Text searching

Table 6.2 shows the average time taken at each stage on five 2000 page books with no images or blank pages, when users entered their first search query. The book's page size was 600 × 849 pixels, with about 1100 words or 6300 characters per page. To test the worst case scenarios, both types search terms (indexed and non-indexed) can be found at least once in every page.

As explained in Chapter 4, incremental text search in Realistic Books involves two stages: *set page list* (Column G) and *highlight matched terms* (Column H). Initially the search page list contains all the pages in the book. In the first stage, all pages that do not mention the term are removed from the list. On the next stage, all term occurrences on each of the four pages loaded during the page turn

Search term (A)	Set page list (G)			Highlight matched terms (H)			Search total
	Find first location on a page	Index look up	Total	Find all location on a page	Update book	Total	
	(B)	(C)	(2000B + C)	(E)	(F)	(4E + F)	(G + H)
In index	—	1	1	—	38	38	39
Not in index (search for `e')	1	—	2000	12	38	86	2086

Table 6.2: Average search time (in msec) for five 2000 page books

simulation are highlighted, and bookmark tabs are added to each page in the search page list (Column F).

If the query terms are not listed in the index or synonym table, e.g., searching for the character *e*, Realistic Books tries to find the location of the term's first occurrence on each page in the list (Column B) and performs a string matching operation on each of the loaded four pages to find all term occurrences (Column E). As a user appends more characters to their query, the less time is required to perform both of these matching operations. If the search term is listed in the index table, the total time required to perform the search is equal to the time required to highlight all term occurrences and display the search bookmarks (Column F).

6.2 Usability study design

As mentioned earlier, the focus of the usability evaluation is on assessing the design interface and the reader services proposed in both the Realistic Books system and the Bookmaker's Workbench. A series of studies were conducted to verify that both applications can be understood and used easily by readers/designers, and to collect information for improving the current prototypes. This section describes the overall evaluation strategies employed.

Subjects came individually to each study and completed a profiling questionnaire that recorded age range, gender, language proficiency, and their experience with computers, web browsers and Adobe Reader. To ensure that they understood how to use all the functions that they might need, participants were asked to familiarise themselves with each document format. They were then explicitly shown any functions that they did not appear to know. Once ready, they were given lists of tasks and asked to perform them.

A within-subjects or repeated measured usability design was employed, where the same participants were tested under every possible condition (Dix et al., 2003; Nielsen, 1993). For example, half of the users performed Task A with physical books and Task B with Realistic Books; the other half performed Task A with Realistic Books and Task B with physical books, where Tasks A and B were similar but not identical. This evaluation model eliminates experimental bias

User study		Features
Browse	(Section 6.3)	Bookmark tabs made larger and page numbers added to the section's title in the Table of Contents
Search	(Section 6.4)	Thumbnail list added and the speed of the page turning animation increased
Personalise	(Section 6.5)	Automatic page turning through mouse clicking disabled on annotation mode

Table 6.3: Changes made to Realistic Books as a result of the user study

caused by individual differences between participants, and is therefore suitable for evaluating a system where learning is involved.

By testing subjects under every condition, participants were able to provide more accurate responses and make a better judgement on each of the document formats. This is less costly than between-subjects usability design because it minimises the number of participants. However, with this method, a participant's performance on the later task may be affected by the experience of having performed other tasks (Kinnear and Gray, 2006). To balance out this effect across the conditions, the order of the document representation was varied from person to person.

Participants

According to Dix et al. (2003), Lewis (1994), Nielsen (2000), Rubin (1994), Shneiderman (1998) and Virzi (1990), more usability problems can be identified when the usability tests are conducted several times with a smaller number of participants, rather than once with a larger number of participants. Between each test, the application prototype is redesigned based on the results of the previous test. For instance, the implementation of thumbnail list in Realistic Books. Initially, Realistic Books application did not provide the thumbnail list described in Section 4.2. Because participants from the two usability studies reported in Sections 6.3.4 and 6.4.2 requested this feature, the list is now displayed in Realistic Books. Table 6.3 displays the changes made after each study.

These researchers found that 80% of usability problems are revealed by the first four or five participants. After the fifth, the evaluation produces the same findings (Lewis, 1994; Nielsen, 2000; Shneiderman, 1998; Virzi, 1990). However, to discover all the problems in the system, Dix et al. (2003) and Rubin (1994) suggest that the test should be conducted with eight to ten participants. Additional subjects are less likely to reveal new information. Hence, in all of experiments described in this thesis there were at least eight participants for each treatment condition.

Obviously readers vary widely in how they understand a text and the way they process it. Relevant factors include background knowledge, reading skill, language proficiency, and intrinsic interest in the content. To help mitigate the effect of this variability, high school and university students aged 15–40 from a variety of disciplines were recruited. In order to bias the experiment *against* the Realistic Books format, the chosen participants had used computers extensively for several years, and were familiar with the HTML and PDF formats.

Criteria

Reading behaviour and performance can be assessed in terms of outcome and process measures (O'Hara and Sellen, 1997; Schumacher and Waller, 1985). Outcome measures focus on what readers obtain from the text, how usable the system is, and whether it provides functions that allow people to accomplish their tasks. Process measures assess how readers engage with the materials and are obtained by observing participants' reading behaviour.

ISO 9241–11 (1998) defines effectiveness, efficiency and satisfaction as the three main criteria for outcome measures. Benyon et al. (1990), Nielsen and Molich (1990), Quesenbery (2001), Shackel (1991), and Shneiderman (1998) further expand the ISO definition, making it more comprehensive by explicitly specifying items that need to be evaluated for each criterion. Effectiveness can be measured by the percentage of users completing a task, or the percentage of tasks a user completes successfully. Efficiency concerns the time required by a subject to complete a task, or the number of tasks completed per unit of time. Satisfaction refers to participants' opinions in terms of pleasantness, sense of engagement, ease of use, ease of learning, preference and usefulness.

Based on those criteria, four sets of experiments were designed to test whether users of Realistic Books would ...

Browse

H1.1 ... find answers, through browsing, faster than with other document formats

H1.2 ... make fewer browsing task errors than with other formats

H1.3 ... report higher browsing satisfaction than with other formats

Search

H2.1 ... find answers, through searching, faster than with other document formats

H2.2 ... make fewer searching task errors than with other formats

H2.3 ... report higher searching satisfaction than with other formats

H2.4 ... use the most effective search tools to answer questions, regardless of the document formats, i.e., back-of-the-book index for index questions and search tools for full-text search questions

H2.5 ... find answers faster when semantic links are present

H2.6 ... make fewer searching task errors when semantic links are present

H2.7 ... report higher searching satisfaction when semantic links are present

Personalise

H3.1 ... report higher personal satisfaction than with other document formats

Construct Realistic Books

H4.1 ... learn to create a book faster using the Bookmaker's Workbench than with an HTML editor

H4.2 ... create a book faster with the Workbench than with an HTML editor

H4.3 ... make fewer task errors with the Workbench than with an HTML editor

H4.4 ... report higher satisfaction with the Workbench than with an HTML editor

Because the data analysis that must follow data collection for process outcomes can be time consuming, most evaluations of electronic book applications concentrate on outcome measures. However, in long documents where navigation becomes a major issue in online reading, process measures can inform researchers about the difficulties that arise from reading with each electronic book application and the strategies that participants utilise while completing a task (Dillon, 1996). Table 6.4 summarises the criteria used in this research.

Data collection methods

To study the hypotheses above, a task based evaluation scheme was developed for a comparative user centred evaluation between Realistic Books and equivalent books in HTML, PDF and printed forms. For each task, the time taken for completing the task and the success score were recorded. To ensure that they felt no pressure and worked at their own pace, subjects did not know that the time they took to answer each question was being recorded.

Participants' views and subjective opinions on their experience were gathered through the use of questionnaires. They had to choose which formats were relevant and useful for them, easier to navigate or annotate, easier to locate information,

Criteria	Description
Outcome measures	
Efficiency	Time spent by the reader in answering each question
Effectiveness	Number of questions answered correctly
User's experience	A subjective assessment of usefulness, engagement, likeability, ease of use and preference
Process measures	The tools and strategies that participants use to complete the tasks

Table 6.4: Evaluation criteria used to evaluate a document representation

more pleasant and engaging to use, and finally which format they preferred overall. They were allowed to choose more than one format. They were then asked to explain their choices and state which features they liked and disliked, as well as suggest improvements.

Behaviour observation evaluation is suitable for investigating usability design issues, examining the interaction between users and a document representation, and recording the frequency of each tool used. There are various techniques for collecting process data: direct observation, video recording, software logging, think aloud protocol and eye movement record (Benyon et al., 1990). With the exception of software logging, all are obtrusive methods and may affect the user's reading activity and performance.

Although eye movement records have helped researchers identify the effect of a system on reading behaviour, measurement tools are expensive and force users to remain immobile throughout the experiment (Dillon, 1992). They were not used in this study.

Instead, subjects were encouraged to voice their thoughts and feelings as they worked. The functions that they used during the task were logged. During the experiment, the evaluator made notes of the user's actions. Because no elaborate equipment is used, the experiment's requirements are inexpensive, physically non-intrusive and relatively natural.

6.3 Browsing

Users often scan pages casually in the hope of finding something of interest. They rely on navigation mechanisms such as hyperlinks, document structure and navigation buttons while browsing visually through the text. To quantitatively evaluate the browsing experience in Realistic Books against conventional formats (HTML and PDF), Spool et al.'s (1997) scavenger hunt approach was adopted. A follow-up comparison with physical books was also conducted. These experiments were designed to evaluate the three browsing hypotheses listed in Section 6.2 and to demonstrate the appropriateness of using the book metaphor for presenting electronic documents.

Scavenger hunts involve browsing through test material in search of specific information without using *find* commands (where available) (Spool et al., 1997). The goals are to determine the time users need to find answers and the number of questions answered correctly. Because all questions are derived directly from the test material, it requires little time and expertise to assess the answers. Coupled with behaviour observation evaluation, this scheme can provide information about how easy it is for participants to find information in documents and the factors that help or hinder them during the task.

6.3.1 Materials

The same text may be read in different ways (Levy, 1997; Marshall et al., 1999; McKnight, 1996; Schilit et al., 1999). The Bible, for example, may be read as literature or as sacred text, leisurely or contemplatively. A novel does not make sense unless it is read from beginning to end. Most manuals and scientific reports also follow a linear thread from introduction to conclusion. Although the order of chapters and sections in a textbook reflect a carefully designed learning path, the information does not need to be accessed linearly: each chapter or section is often self-contained.

Readers determine their reading strategy according to their motivation, e.g., reading for professional or personal reasons, for information or pleasure, out of interest or out of need; the text content and structure, e.g., general or specific, technical or non-technical, textual or graphical, linearly or non-linearly; and their reading conditions or circumstances e.g., night or day, at home, beach, car or office, in a hurry or slowly (Dillon and McKnight, 1990). They check their progress and revise their strategies as they complete their tasks.

To investigate whether a user's reading performance is affected by type of the document, in the first study participants were asked to read three simple books: a handbook (*Raising Ducks: How to Begin*), a history book (*History of the United States Told in One Syllable Words*) and a children's book (*Alice's Adventures in Wonderland*). Each book contains about 100 pages (ranging from 80–140) and is divided into 8 to 18 chapters (some of which have subsections). For the follow-up comparison with printed books, subjects were asked to read two biographies from the Internet Archive: *Pavlova A Biography* (184 pages) and *Princess Mary: A Biography* (254 pages). Both have 10 chapters.

Each book is presented in three electronic formats: HTML, PDF, and Realistic Books. In each case it begins with a hyperlinked table of contents, and there is no back-of-the-book index. Documents in all formats (except HTML) were paginated in exactly the same way. To avoid delays in loading or refreshing, all documents were served off-line, having been fully loaded before the start of each experiment.

The HTML books were presented according to the Wikipedia article presentation style. To provide a visible cue to differentiate the main text from the table of contents, the former had a white background and the later a green one. Readers could click on section names and return to the table of contents by scrolling to the top or pressing the browser's *back* button. The standard Adobe Reader application was used to view PDF documents; it opens in thumbnail view. The beginning of each section was bookmarked, so the bookmark view provides an ever-present table of contents adjacent to the page currently being read.

Adobe Reader supports backtracking, allowing users to return to previously visited pages. For example, if users were on Page 1 and clicked a hyperlink that took them to Page 5, pressing the *previous view* button would return them to Page 1. However, *previous* and *next view* buttons are, by default, not visible in the toolbar: readers access them from a menu. In the pilot study, it transpired that if they were added to the toolbar, readers would confuse them with the navigation buttons. Users also thought they would not find them useful because these backtracking functions literally store all previously visited pages, i.e., if a user clicked the *next page* button on Page 5 to go to Page 6, pressing the *previous view* button would take him back to Page 5. Instead, participants wanted the *previous view* button to store only pages they navigated to from at least three pages away. Consequently, in this experiment, these buttons were not added.

For the Realistic Books format, users could utilise all the navigational tools except the thumbnail list described in Section 4.2. The physical copies for the follow-up study were created by printing the books in colour, double-sided, and stapling down the left-hand side. Participants were provided with Post-It bookmarks to facilitate easy access to marked pages.

6.3.2 Tasks

Browsing performance was evaluated according to the usability criteria and procedures described in Section 6.2. For each book, participants answered six multiple-choice questions seeking information that could be found in the document:

1. What kind of book is it?
2. Which of the following would make the best title for the book?
3. Who do you think this book was written for?
4. From the following list of topics, write down the section numbers that explain the topic.
5. Put a check next to each topic discussed in the book.
6. Put a check next to each statement that reflects the underlying beliefs or point of view of the book.

They were allowed to view the book while answering these questions. The first three were aimed at determining whether the participant understood what the book was about, and could be answered merely from the front matter and introductory section. The remaining three required users to search through the text, and were designed to assess critical reading activity. Users were unlikely to know the answers already. They often needed to browse more than one section to find the information, and draw a conclusion by relating several pieces of supporting evidence.

6.3.3 Navigation studies

24 university students aged 20 to 40 from various disciplines were recruited for the experiment. None had used the Realistic Books system. Because participants found these tasks quite demanding, the evaluation was divided into two: Realistic Books against the HTML and PDF formats, and Realistic Books against physical books. The main concern in both was to compare within-document navigation.

Realistic Books vs. HTML and PDF In the first study, participants were asked to read the three books, and answer the questions for each one. The history book represents a non-fiction work or school textbook, where readers need to concentrate and read carefully. The children's book represents the kind of fiction that people read for leisure. The handbook is factual; it demands more attention than a children's book but less than a textbook.

Participants read the books in the same order. They were exposed to each of the three formats, a different one for each book. For example, a participant might read book 1 in HTML format, book 2 in PDF format, and book 3 in Realistic Books format. The six possible orderings were allocated evenly between the participants: eight subjects read each book using the same document representation.

Realistic Books vs. physical book Only 16 of the original 24 subjects were recruited for this study, which took place shortly afterwards. They were asked to read and answer the questions for the two biographies, again in the same order. There are two possible orderings of format, and as in the first study these were allocated evenly: eight participants for each combination of book and document format.

6.3.4 Results

As noted earlier, reading performance can be evaluated in terms of outcome and process measures, as described in Table 6.4.

Efficiency

Table 6.5 shows the average time taken by participants to answer each question, expressed as mean ± standard deviation. The average ranged from just over one minute to five minutes.

There was great variation between questions. In the first user study (Table 6.5(a)), times per question ranged from three seconds to two minutes for the first three questions, and from one to fourteen minutes for the remainder. Most participants (67%) located information faster when reading the children's book.

Material	HTML	PDF	Realistic Books
Children's book	129 ± 31	104 ± 42	**68 ± 46**
Handbook	278 ± 173	201 ± 119	**89 ± 63**
History book	305 ± 150	221 ± 141	**134 ± 91**

(a) First user study

Material	Realistic Books	Physical Book
Anna Pavlova	283 ± 13	**281 ± 15**
Princess Mary	**264 ± 19**	279 ± 15

(b) Second user study

Table 6.5: Average time (in seconds) to answer each question

It is recorded that for this book they referred back to the table of contents at most twice to answer each question, whereas for the other two they referred back at least three times. When asked, they explained that they could recall approximately the location of the information relevant to the questions on the children's book because they had already read or heard the story. The other books were new, and subjects were unfamiliar with their structure and content. When asked to comment on their speed of finding information in the history book versus the handbook, they stated that the history book contained more information and so they spent longer seeking the answers.

In the second user study (Table 6.5(b)), times per question ranged from one second to three minutes for the first three questions and from one to sixteen minutes for the remainder. There was little variation in the time between participants. This might be caused by the practice effect, in which subjects had become more fluent at navigating the Realistic Books format since the first experiment (Kinnear and Gray, 2006).

Users read the second book faster than the first because it was better structured. They commented that since its contents were ordered by time, they could guess the location of relevant passages based on the time of the event. In contrast, the first book was ordered by chapter author, and participants had to check more than one chapter to gain complete information about a particular event.

H1.1 (speed) was partially upheld because Realistic Books participants answered questions significantly faster than PDF and HTML, but not significantly faster than physical books.

A one-way analysis of variance of the data in Table 6.5(a) showed that the differences due to document format were statistically significant at the five percent level. Further t-test analysis between the Realistic Books and its closest competitor (PDF) showed that the difference was significant at the ten percent level for the history book, and at the five percent level for the other two. A one-way analysis of variance in Table 6.5(b) showed no significant difference for either book at the five percent level (although for the second the difference was significant at the ten percent level).

Effectiveness

Table 6.6 gives the average number of questions that participants answered correctly. Most (58%) were not able to answer the last two questions correctly, which required readers to locate information throughout the text and draw appropriate conclusions. Some (33%) did not attempt to find all the relevant text before answering the question: they drew conclusions from partial information. As with the efficiency measure, performance with the history book was noticeably inferior to the other two books in the first study, and performance with Anna Pavlova was inferior to that with Princess Mary in the second.

H1.2 (accuracy) was not upheld. A one-way analysis of variance found no significant differences due to document format. From Tables 6.5 and 6.6, it can be concluded that while document format does not affect a user's level of comprehension, it does affect the time that readers require to process text.

Reading experience

During the familiarisation stage, before participants received the list of tasks, it was observed that most subjects (54%) were unaware of PDF's bookmark view, or did not know how to switch between thumbnail and bookmark views. They were also unaware of PDF's backtracking functions. With Realistic Books, although all participants realised that they could click on the bookmark tab to go to that page,

Material	HTML	PDF	Realistic Books
Children's book	**90 ± 10**	84 ± 9	86 ± 8
Handbook	85 ± 14	**86 ± 19**	81 ± 21
History book	74 ± 13	75 ± 12	**78 ± 13**

(a) First user study

Material	Realistic Books	Physical Book
Anna Pavlova	75 ± 14	**77 ± 13**
Princess Mary	**79 ± 14**	78 ± 18

(b) Second user study

Table 6.6: Average number of correct answers (in %)

Criteria	HTML	PDF	Realistic Books
Useful	0	17	**83**
Easy to navigate	0	17	**83**
Easy to locate information	0	0	**100**
Pleasant and engaging	0	0	**100**
Preferred	0	17	**83**

(a) First user study

Criteria	Realistic Books	Physical Book
Useful	44	**56**
Easy to navigate	**56**	44
Easy to locate information	50	50
Pleasant and engaging	50	50
Preferred	50	50

(b) Second user study

Table 6.7: Format preferences (in %)

most (58%) did not realise they could go to a page by clicking on its edge. They were then explicitly shown how to use these functions. Overall, subjects found the Realistic Books format to be natural and needed no more time to familiarise themselves with it than the other formats.

Once participants had completed all tasks, they were asked which formats were most relevant and useful, easy to navigate and to locate information, pleasant and engaging, natural and intuitive. Table 6.7 summarises their opinions. Note that users were allowed to choose more than one format.

Almost all readers (80%) commented that while HTML suffices for short documents, they prefer PDF and Realistic Books for longer ones. Pagination breaks the flow and helps them remember the location of information by relating it to its physical position on the page. They can then concentrate on locating the relevant information, rather than navigating. All participants judged Realistic Books to be the format that is most engaging, natural and intuitive. However, 17% of subjects preferred PDF because of their previous experience with it.

The second user study found little difference between Realistic Books and physical ones. Subjects said that both gave a similar experience, but the former had the advantage of hyperlinks, allowing them to jump more quickly between pages. Also, they did not need to look for pages that start a section; they could go straight there. In the physical book, they used the fingers of their left-hand to mark the table of contents and sought the appropriate page with their right-hand; they found this cumbersome.

H1.3 (satisfaction) was partially upheld. Because participants found Realistic Books to be as useful, engaging and easy to use as physical books, they preferred them over HTML and PDF.

Browsing strategies

There are many possible ways of navigating within the various document formats, and Table 6.8 shows the functions used by subjects during the user studies. Scroll bar means that users moved the scroll bar while reading. Select page means they positioned the HTML or PDF scroll bar to a certain location, clicked on a PDF thumbnail, or selected the edge of a Realistic Books or physical book page.

Functions	HTML	PDF	Realistic Books
Scroll bar	100	100	—
Arrow key	100	100	100
Previous and Next button	13	21	17
Type page number	—	0	0
Select page	0	0	75
Hyperlink	100	42	100
Bookmark	—	100	88
Page turning	—	—	33

(a) First user study

Functions	Realistic Books	Physical Book
Scroll bar	—	—
Arrow key	100	—
Previous and Next button	19	—
Type page number	0	—
Select page	88	100
Hyperlink	63	—
Bookmark	100	0
Page turning	13	100

(b) Second user study

Table 6.8: Navigation tools usage (in %)

Hyperlink means they clicked a hyperlink in the table of contents page, while bookmark refers to clicking in the PDF bookmark view or a bookmark tab in the Realistic Books. The other functions are self-explanatory. Subjects did not change any settings, such as font size, zoom factor, document view, bookmark style and page turning speed.

Before starting to seek the answer to a question, almost all participants (92%) took a brief overview of the document. In HTML or PDF formats they would scroll through the end. With the Realistic Books or the physical book, they would flip through the pages.

Participants answered the questions through extensive use of the table of contents and navigation within the document. They would look through the table of contents and guess which section contained the information they sought. They would then go to that section and scan through it to check on particular facts, expression or spellings. If they could not find the answer they would return to the table of contents and seek other sections that could be relevant. Even when they had already found the relevant passage, 54% of subjects would still check for other sections that were more relevant. This process involves returning to the table of contents to seek relevant sections and comparing the previously found passage with the new text. Having found the most relevant section, they would return to it and re-read the text to confirm or clarify their understanding.

No participant added their own bookmarks to help them return to a page in the book. They did not use Post-It bookmarks to mark the start of each section to facilitate access in the future while reading the physical book. When asked, participants said that they were too lazy do this, even though they admitted that it might help them find answers faster.

HTML Almost all participants (92%) scrolled through to the end of the document before starting to read the HTML books. They did so because they found it hard to assess its length just by looking at the scroll bar—they could of course tell that documents were long, but had little idea how long they were. They also found it hard to determine where they were in the document. Subjects felt lost in the flow of information. Some did not notice when they moved from one section to the next.

Because there is no notion of page, it is hard for users to recall the location of a piece of information. Instead, they used images surrounding the text to help recall locations. Because the history book contained no illustrations, participants commented that they were frustrated: the text was seemingly endless. In the handbook, illustrations follow each paragraph, and readers often used these, in conjunction with the table of contents, to determine which sections were relevant. They did this by scrolling rapidly, seeking appropriate images.

Even when they recalled the approximate location of a piece of information in relation to the scroll bar position, readers said they found it difficult to return to that location because a small change in the scroll bar position completely altered the screen content. While reading a section, they preferred to scroll through the text by clicking the arrow buttons rather than by moving the scroll bar, to ensure that they did not miss any text.

Most participants (75%) did not utilise the browser's *previous* or *next* buttons to return to the table of contents; instead they used the scroll bar to return to the top. Consequently, they complained about the time taken to scroll back to the beginning of HTML documents to revisit the table of contents (a problem that would have been solved by providing explicit back links).

PDF Although information about document length and the current page number is clearly visible in the Adobe PDF Reader's toolbar, most subjects (67%) did not notice this. Pagination did help break the flow of information, but it was insufficient: almost all readers (83%) still felt lost in the documents. They did not know where they were, and sometimes did not even notice when they moved to the next section.

Subjects found it difficult to return to a passage because they were hazy about its approximate whereabouts, or forgot (or never noticed) the page number. When they did remember the location of the information, readers used the scrolling mechanism to go to that page. They rarely used the thumbnail view or typed a page number. Participants never utilised the backtracking functions to help them return to previously visited pages. When asked, they said that they forgot that these functions existed.

As with HTML, images served as valuable landmarks. To answer questions about heavily illustrated documents (like the handbook), subjects would first scroll rapidly to locate an appropriate image before consulting the table of contents. While reading, nearly all users (92%) used the up and down keys to scroll through the text to ensure that they read all the information.

They felt less frustrated with PDF than with HTML because the bookmark view enabled them to consult the table of contents from anywhere in the book. Hence, few subjects returned to the table of contents page to find a section. Some did, because they forgot about the bookmark view—they had closed it to enlarge the reading area.

The bookmark view and hyperlink capability available in both PDF and Realistic Books formats meant that participants did not feel the need to type in page numbers—although the page number and the total number of pages should still be displayed because it helps readers understand where they are in the document. Most readers (79%) did not click the PDF and Realistic Books *next* and *previous* buttons; instead they used the arrow keys to move between consecutive pages.

Realistic Books Participants said they always knew where they were in Realistic Books. The page edges helped them estimate the length of the document and their current position in it. The text did not seem to stretch indefinitely because the next section tab showed where the current section ended. They used the spatial reference of passages in the book to help them remember the location of information. They also utilised the last visited bookmark tab to orient their current location in the book and return to the previously visited page.

Users turned pages using the mouse (instead of the arrow keys) when reading a section of a book, particularly when they were comparing the contents of adjacent pages. Some participants preferred to use page-turning gestures throughout. Although only a few did so, all participants remarked that they liked the page turning animation because it made the document more engaging and helped them recall approximately how many pages they had read, which facilitated returning to particular passages because they could guess the number of pages to turn back. Moreover, they said that if they were reading in their leisure time they would make more use of page turning gestures.

In general, participants sought answers in Realistic Books in similar ways to the HTML and PDF formats. They guessed where to find the information by consulting the table of contents or the bookmark tabs on the page edges. However, they often randomly accessed pages of Realistic Books by clicking on their edges. They would open to the middle, or to the first or last quarter. When asked, they said they did this because the books were rather long and they did not really want to read them page by page, so they randomly accessed different parts of the book in the hope that they would land close to the passage they were seeking. They would guess the approximate location by using spatial information. Participants using HTML or PDF never randomly accessed any part of the book.

With the children's book, participants said that they were able to use their knowledge of the story to guess the location of the relevant passage. For the history and biography books, events are organised in time order. This means they could guess where the information was located. They did not need to consult the contents page so often because they used the pagination, and bookmark tabs, to guess the location. When participants realised that each image in the handbook corresponded to the adjacent paragraph, they began to use the flipping function in conjunction with the table of contents to find the relevant section. Some participants criticised the speed of page flipping: they wanted a faster way to flip through the book when scanning or skimming the material. They would like to be able to flick through pages quickly, as with a physical book. Others suggested a PDF-style thumbnail display to give a quick preview of the pages.

While reading Anna Pavlova's biography, subjects commented that they would have liked to have been able to compare any two pages side by side. This is because for this book they needed to combine information from various sections to get a complete picture of an event. They found it quite cumbersome to keep jumping between pairs of pages.

Physical books For physical books, participants used similar navigation strategies to Realistic Books. Also, they used one hand as a starting point for quickly flipping through pages. Physical books have the advantage that several non-contiguous pages can be viewed consecutively in short order: readers used their fingers to mark relevant pages and flip back and forth between them. For an unstructured work like Anna Pavlova where one had to read pages from several sections, this ability to compare any two pages was very helpful. This might be why physical books were found to be slightly more efficient than Realistic Books for this work (though the difference was far from significant statistically).

The disadvantage of physical books was the absence of hyperlinks. Participants commented that without hyperlinks they had to first find where a section started before seeking a relevant passage in it. They continually guessed the location of the page and randomly accessed it. In contrast, with Realistic Books participants guessed the location of a page based on nearby section tabs. They commented that this allowed them to make better educated guesses about which pages contain relevant information than with physical books. This was especially helpful with a well-structured book.

6.4 Searching

Subject indexes are an important access tool for physical books, whose utility frequently depends on the quality of the index. Indexing is a profession with its own societies, journals, standards and guidelines for index construction. As large volumes of text migrate to electronic formats, it is easy to assume that such indexes become superfluous.

Most electronic document readers have a text search function. Readers type what they seek into the *search* box and are taken to the next closest occurrence, with matching strings highlighted and buttons to move to the next and previous occurrences. However, literal string matching has limitations. Readers who do not know exactly what they seek or how it might be expressed may issue search queries that match many irrelevant passages—or no passages at all. Advanced search functions, which could be used to narrow down the search, often make people anxious because they lack experience with them.

To investigate the search hypotheses **H2.1** to **H2.7** mentioned in Section 6.2, the evaluation was divided into two: Realistic Books against HTML, PDF and printed books; and Realistic Books with automatic aliasing against the ones without. The first study was designed to examine the effect of a document representation on people's searching behaviour and performance. The follow-up study was conducted to determine the importance of automatically generated definition and semantic links.

Participants were asked to read a university level text called *Data Mining: Practical Machine Learning Tools and Techniques*. This book was used because it had a professionally produced subject index whose terms accurately represented the contents. To reduce the search space, only the first three chapters were used. The table of contents and back-of-the-book index were modified to remove all references to other parts of the book. This reduced the page count to 94. Chapters were divided into five to ten sections, some with subsections.

6.4.1 Realistic Books vs. HTML, PDF and physical books

People should be able to easily find their way around a document and access information sought without spending too much time looking for it. This section examines the importance of document representation, back-of-the-book indexes and full text search in retrieving a passage from a document. 32 high school and university students aged 15–40 from a variety of disciplines were recruited for this study. 16 of them had encountered Realistic Books at most twice before, and only two of them had read the book as part of their university studies.

Material

The modified book was presented in three electronic formats: HTML, PDF and Realistic Books. They were served off-line and had been fully loaded before the start of the experiment. To eliminate layout effects, the original book design and layout was retained. All three formats were paginated in exactly the same way as the physical book. Running headers contained the page number and section name. Table of Contents and back-of-the-book index were hyperlinked. By clicking a main term in the indexes, the book reader application (be it for HTML, PDF or Realistic Books) opens the document at the appropriate page; clicking a variant term follows a hyperlink to its main entry in the index.

As mentioned previously (Section 5.3.3), Back-of-the-book indexes contain bridging words that do not necessarily use the exact words of the text. This property caused some difficulty in preparing the electronic index page. When a user selected a term that is explicitly mentioned in the document, it was easy to take the user to the page containing the term, and automatically highlight all occurrences of the term in the page. But when the subject selected one of the bridging words, it was hard to automatically mark all the sentences or paragraphs corresponding to the term. To ensure uniformity in the way in which people access a term from the back-of-the-book index, clicking an index term only took users to the page containing the term. Subjects had to spend additional time and effort searching or browsing the page to find the location of the term in the page.

The browsing study, described in Section 6.3.4, found that participants preferred to read long articles in paginated style because it helped break up the text and provided orientation cues when navigating the document. Consequently, the HTML version of this book is paginated, and each page is numbered. Users can return to the table of contents by scrolling to the top, pressing the Web browser's *back* button or clicking the *home* link that is located at the top of each page. They can perform a full text search by typing their search queries into Firefox's *find* box.

Most participants (71%) in the browsing study said that table of contents in the PDF's bookmark tab helped them find information faster, so in this study the PDF book opens in bookmark view. Adobe Reader provides two full-text search functions that can be used in this experiment: *find* and *search*. The former scans the alphabetic listing of every word in the document. It is faster and more versatile than *find*.

To ensure that automatic word aliasing did not affect the result of this study, the *include synonyms* search option in Realistic Books was turned off. Again, the thumbnail list view was not included in the Realistic Books.

Because only the first three chapters were relevant to the questions asked, the modified subject index pages were printed and attached to the book. Participants were allowed to bookmark any page.

Tasks

This evaluation followed the study design, criteria and procedures explained in Section 6.2. There were four sets of tasks, which participants undertook in the same order. Each task asked four questions that involved seeking information in the book. A different format was used for each task, and participants were exposed to the formats in different orders. For example, one did Task 1 in HTML format, Task 2 in PDF format, Task 3 in Realistic Books format and Task 4 in physical book format. The 24 possible orderings were allocated evenly: eight participants performed each task using the same document representation.

Studies have shown that participants find information more efficiently and effectively with back-of-the-book index than with a full-text search tool (Barker et al., 1994; Landoni and Gibb, 2000; Summerfield and Mandel, 1999). However, back-of-the-book indexes are rarely found in electronic documents. To determine whether users will choose the most appropriate search tool for each question, each task posed two full-text search questions and two subject index questions in an order that changed from one task to another. Answers to the full-text search questions could be found by entering the search terms in the *find* box, but those terms are generally not available in the back-of-the-book index and in any case it would be less efficient to use it. Answers to the subject index questions could be found by looking up the terms in the back-of-the-book index. However, these terms either did not appear explicitly in the book's main text, or readers had to be able to understand their subheadings in the index to answer the question. It would take longer to answer these questions using full-text search. Thus participants should use search tools or the back-of-the-book index at least once while searching with each document format.

To provide a balanced assessment of the utility of search, the questions had varying degrees of difficulty: one factual and one analytical question for each question type. The former was a straightforward question to find a specific piece of information in the book; it could be answered by direct look-up using the appropriate search tool. The latter needed multiple searches as well as some browsing: participants had to identify relevant passages in different parts of the document to determine the answer. For example, these were the four questions used in Task 1:

1. Who created the CART algorithm? (Index, Factual)
2. Write the page numbers that explain a technique used to analyse people's buying patterns in a grocery store. (Index, Analytical)
3. In statistics, there are four types of attribute values. Choose two attribute values used by the data mining system. (Search, Analytical)
4. Write the page numbers that explain a situation where prior knowledge can be used to cut down a rule set. (Search, Factual)

	HTML			PDF			Realistic Books			Physical Book		
Task 1	975	±	1810	1139	±	2309	**677**	**±**	**1044**	1197	±	2265
Task 2	288	±	31	289	±	32	**257**	**±**	**36**	310	±	27
Task 3	281	±	31	284	±	32	**251**	**±**	**34**	302	±	27
Task 4	277	±	31	281	±	31	**244**	**±**	**33**	298	±	25

Table 6.9: Average time (in seconds) to answer each question

6.4.2 Results

The task-based approach defined above provided data on outcome and process measures listed in Table 6.4

Efficiency

Table 6.9 shows the time spent in answering each question. Averages ranged from just over 4 minutes to almost 20 minutes. There was a large variation between questions. For Task 1, times ranged from 12 seconds to 9 minutes for factual questions and from 3 minutes to just over an hour for analytical ones. For the remaining tasks they ranged from 24 seconds to 3 minutes for factual ones and from 2 to 12 minutes for the others.

When questioned about the reason they had taken a longer time to answer questions in the first task, participants said that they needed to familiarise themselves with the book and the type of questions being asked. Nearly all of them (94%) had not read the book and did not know which sections were relevant. They browsed through the books first before answering the questions. As each task proceeded they gained a better understanding of the content and structure of the book. The time needed to answer each question decreased for each format as the experiment proceeded.

A one-way analysis of variance showed that differences due to format were statistically significant at the one percent level for all tasks except the first. For the last three tasks, the differences between physical books and the PDF and HTML formats were significant at the ten percent level. The improvement from

	HTML			PDF			Realistic Books			Physical Book		
Task 1	79	±	17	77	±	22	**81**	**±**	**19**	75	±	20
Task 2	81	±	17	79	±	19	**83**	**±**	**18**	77	±	18
Task 3	77	±	25	75	±	25	**81**	**±**	**19**	79	±	26
Task 4	83	±	28	81	±	19	**87**	**±**	**17**	77	±	25

Table 6.10: Average number of correct answers (in %)

Realistic Books to physical books was statistically significant at the one percent level, and its improvement over PDF and HTML was statistically significant at the five percent level. There was no significant difference between PDF and HTML for any task.
H2.1 (speed): Subjects produced answers quicker with Realistic Books.

Effectiveness

Table 6.10 shows the average number of questions that were answered correctly in the four tasks. A one-way analysis of variance found no significant differences between document formats in the number of task errors made. As shown in Table 6.12, participants relied on full-text search to find answers in the electronic document formats. Because the subject index questions were designed to be difficult to answer without using the back-of-the-book index, participants generally got them wrong. With the physical book, most participants (66%) failed on the full-text search questions.
H2.2 (accuracy): Subjects' accuracy in finding answers was not influenced by the format of the document. Therefore, this hypothesis was not upheld

Search experience

Having completed all tasks, participants were asked which formats were most useful, easy to navigate and locate information, pleasant and engaging, and preferred overall. They were allowed to choose more than one format. Table 6.11 summarises their opinions.

Criteria	HTML	PDF	Realistic Books	Physical Book
Useful	5	25	65	5
Easy to navigate	0	30	70	0
Easy to locate information	0	9	82	9
Pleasant and engaging	0	0	100	0
Preferred	0	10	70	20

Table 6.11: Format preferences (in %)

The formats were judged useful for different types of documents and activities. HTML was preferred for short documents. Physical books were best for reading activities, but not for information seeking. For searching, PDF or Realistic Books were preferred. Participants thought that the Realistic Books format combined a good reading environment with a good searching experience. It was rated the most useful, engaging and easy to use of all. PDF was the next preferred format. **H2.3 (satisfaction):** Participants preferred to search in Realistic Books than in the other formats.

Subjects liked HTML and PDF because in continuous scroll mode they could see more than two pages at once. However, they found it hard to understand the structure of the document. They easily became disoriented, not knowing where they were in the document. Because they found it difficult to return to a specific location, participants rarely noticed the back-of-the-book index.

All participants preferred Realistic Books for searching. They could see all matching pages at once. Bookmark tabs and page edges made it easier to move between search results without losing orientation. Mousing over a tab showed the matched term's context. Unlike PDF and HTML which took readers to the closest point of occurrence while they were entering their query, Realistic Books waits until the *enter* key was pressed. This allowed users to perform a search on the currently visible pages without changing their view, which participants found useful.

Readers appreciated the ability to compare different pages of a physical book using their fingers as bookmarks. They liked how they could handle the book and read it anywhere they want. However, nearly all participants did not like

Functions	HTML	PDF	Realistic Books	Physical Book
Table of Contents	100	100	100	100
Back-of-the-book index	19	41	69	100
Find function				
Find text book	100	100	100	—
Previous/Next button	100	100	—	—
Find options	19	0	0	—
Find bookmark	—	—	100	—
Basic Search	—	3	—	—
Advanced Search				
Search text box	—	6	—	—
Search result list	—	6	—	—
Search options	—	6	—	—

Table 6.12: Searching tools usage (in %)

searching for information in physical books. They missed the *find* function of electronic document viewers, which located matching fragments in the main text and the back-of-the-book index. They also missed hyperlinks.

Participants' strategies

Table 6.12 shows what search functions participants used during the experiments. Find bookmark refers to clicking on a tab that marks pages containing the search terms.

Before seeking the answer to the first question in each task, nearly all participants (94%) took a brief overview of the document, and went to the table of contents page. They answered the questions through extensive use of the table of contents, navigation within the document and, if available, the *find* function. They utilised the same navigation strategies and had similar navigation experiences as recorded in the browsing study described in Section 6.3.4, and so they will not be discussed again here.

H2.4 (appropriate choice of search tool) was only partially upheld because all participants relied on full text search to find answers in all electronic formats. Back-of-the-book index use was considered as a secondary strategy only in Realistic Books.

When searching for relevant information in the HTML or PDF books, nearly all users neglected the back-of-the-book index, even though they had been told about it before beginning their tasks and it was listed in the table of contents. They only used the index when a search term appeared in it and nowhere else. These terms were usually synonyms or bridging words for the actual terminology in the text. Readers then clicked on the link associated with that term and searched for the answer in that page.

HTML After subjects gained an overview of the document's content and length, they typed their query terms into the *find* box. A few participants selected the *highlight all* option to help them scan the matches. If they judged the query to be good, they stepped or scrolled through all matching fragments to determine whether any of them seemed to contain the answer. Having exhausted all search terms they could think of, subjects would go to the table of contents and guess which section contained the information they sought. They then went to that section and scanned it to check particular facts, expressions or spellings.

Although the documents in HTML format had been paginated and each page began with a header, almost all participants (84%) did not notice when they had moved from one page to the next. They did not use the running headers as landmarks to help them to return to previously viewed information; instead they used figures and tables. They felt lost in the flow of information and found it hard to navigate around the document.

PDF Subjects used similar strategies for the PDF format. They scrolled through the document instead of clicking the *previous* and *next* buttons. Consequently, the disadvantages of HTML applied here too.

Although all participants were familiar with PDF and had used it to read scientific papers, most (75%) were unaware of PDF's *search* tool and a few (22%) found it difficult to step through the search results. When stepping through matching fragments one by one, they often mistakenly pressed the *previous* and *next* navigation buttons instead of the corresponding *find* buttons. They would like a *highlight all* option and then skim through all the matched fragments. This may be why they found information faster using HTML.

When the *find* tool returned no results, few participants resorted to the *search* tool. Those with a computer science background tried to modify the search options to help them locate the relevant information. However, all others were confused by the terminology used in the advanced search tool and decided not to use it—for example, some asked what *stemming* meant. Subjects felt that the differences between the *find* and *search* tools were slight, and so they preferred to use *find*.

Realistic Books In general subjects' first search strategy was the *find* function. They entered their search terms into the *find* box, without pressing *enter*, to see whether any bookmark tabs popped up. If so, and no words were highlighted in the currently-visible double-page spread, they would press *enter* to go to the closest hit. Slightly more than half (56%) scanned the table of contents before entering their query terms. When they found suitable matches, they moused over the tab for information about the page number, the context of the matched text, and to determine whether it was at the beginning of a chapter. They looked at the section titles and tried to decide which matching pages belonged to a section that they thought contained the relevant information. Others consulted the table of contents, guessed which section might be relevant, and visited it before entering search terms. This told them whether the terms appeared in that page or in nearby ones. Their logic was that terms that appear far away from the current page were probably in a different section.

Unlike in HTML and PDF, participants scanned the Realistic Books' back-of-the-book index for more appropriate search terms when full-text search failed to find the answer, or when they could not think of any more suitable search terms. They associated the term's page references with the chapter titles on the tabs.

Because subjects found the Realistic Books natural and felt that they always knew where they were in the document, they navigated around more freely than with the other formats. The act of turning each page helped them recall approximately how many pages they had read. Once they were familiar with the document structure, most used the spatial position of passages in the book to help them remember where information was. They were not afraid to access pages randomly. For example, one participant who had not found the relevant information but had a hunch about which section might contain it clicked a random page in the middle because he recalled that the section he wanted to view was located in the middle of the book.

Physical book Participants relied on the subject indexes to answer each question. They would place their fingers to mark the table of contents and back-of-the-book index pages. They tried to guess relevant sections by looking at the table of contents. To find appropriate index terms in the back-of-the-book index they looked at the associated page numbers and chose terms that appeared within the relevant sections. Having no hyperlinks, readers had to first find the target page before seeking a relevant passage in it. If they still failed to find the answer, they carefully read sections of the book that they thought might contain relevant information.

Back-of-the-book indexes Most readers (75%) found the back-of-the-book index easy to use. They liked its lists of synonyms and bridging words. However, they did not like the cross-reference format. When a variant term was cross-referenced in the index, a further look-up for its main entry was needed before the list of associated pages could be determined. These pages explain the topic the term refers to, but might not contain the term itself. Almost all participants (91%) understood this when what was specified was a page range, but when the back-of-the-book index pointed to a single page they almost all assumed that the term would appear explicitly on that page. When they found it on the previous or subsequent page instead, they complained that the index was incorrect.

Participants found the hyperlinked version of the back-of-the-book index useful because it took them straight to the desired location in the text. However, they expressed regret at having to scan the text for the term, because it was not highlighted in the HTML, PDF and Realistic Books formats.

Find function Participants were not satisfied with simple string matching: they wanted a smarter one-box search function. They wanted to be able to do an

AND search on a single paragraph in HTML or on a single page in the paginated formats—e.g., find all pages containing the terms *attribute* and *model tree*. They were frustrated that the system did not detect typing errors and suggest the correct spelling. Participants wanted the search program to know as many synonyms as possible, to avoid having to consult the back-of-the-book index.

6.4.3 Plain vs. semantically linked text

While reading an article, users need to determine whether it is of interest to them, focus on particular parts, and locate information (Berkenkotter and Huckin, 1993; Crane, 2002; Dillon et al., 1989; Goldsborough, 2000; Murray, 2002; Poynter Institute, 2000). An electronic document application should provide a way for users to comfortably navigate through the document, quickly spot and understand keywords, and rapidly locate a topic and all its related pages. As described in Section 5.3, Realistic Books supported those activities through navigational, definition and semantic links, respectively. Section 6.3.4 showed that navigational links help readers scan and browse articles. This experiment aims to evaluate the utility of text with and without definition and semantic links.

Material

24 high school and university students aged 15–40 from a variety of disciplines were recruited for this study. None had used Realistic Books or read the *Data Mining* book. They were asked to read the modified *Data Mining* book in Realistic Books format that was used in the first search experiment, so the user interface would be identical.

To ensure that subjects could not scan for better query terms in the back-of-the-book index, the index pages were removed. There were three hypertext versions of the book: with only navigational links (Text N), with only navigational and semantic links (Text N+S), and with all three automatically generated links (Text N+S+D).

Tasks

This experiment's overall design, procedures and evaluation criteria were as described in Section 6.2. There were three sets of tasks, which participants performed in the same order. A different hypertext version was used for each task, and the three hypertext versions were presented in different orders to each participant. Complete 3×3 Latin squares, with eight replicates, were applied to control order effects. For example, one subject did Task 1 with Text N, Task 2 with Text N+S and Task 3 with Text N+S+D.

On each task, subjects answered three multiple-choice questions. Although none of the terms used in the questions appeared explicitly in the book, a superficial understanding of the text would be sufficient to guess the appropriate query term. As they read each version, participants answered the following questions:

1. Match the following terms with the specified list of definitions.

2. From the following list of topics, write the page numbers that explain the topic.

3. Put a check next to each topic discussed in the book.

To provide a balanced assessment, the questions had varying degrees of difficulty. The first was a general question designed to be easy to answer correctly, i.e., the term mentioned in the question was in the synonym list. The last two were aimed to assess an analytical reading activity, and required users to identify all the pages that mentioned a specified topic. Moreover, the term used in the question might not be in the synonym list.

6.4.4 Results

As in the previous user studies, this experiment utilises the criteria specified in Table 6.4 to measure the performance of each text representation.

	Text N	Text N+S	Text N+S+D
Task 1	379 ± 30	340 ± 28	**330 ± 28**
Task 2	375 ± 26	335 ± 26	**324 ± 26**
Task 3	373 ± 26	333 ± 26	**323 ± 26**

Table 6.13: Average time (in seconds) to answer each question

Efficiency

Table 6.13 shows the time spent in answering each question, expressed as mean ± standard deviation. Averages ranged from just over 5 minutes to 6 minutes. There was a large variation between questions, ranging from 12 seconds to 1 minute for general questions and from 1 to 3 minutes for the others.

H2.5 (speed): A one-way analysis of variance showed that differences due to the hypertext version were statistically significant at the one percent level for all tasks. The improvement from text with semantic links (Text N+S and Text N+S+D) to text without was statistically significant at the one percent level. There was no significant difference between the two semantically linked texts.

As in the first search evaluation study, as each task proceeded participants gained a better understanding of the document structure and the time needed to answer each question decreased. Most participants completed all tasks faster with semantically linked text because the *find* function in plain text does not search for all of the query term's synonyms, so they had to read the document carefully to find the answer. Furthermore, subjects reading text with definition links tended to find answers faster because they could quickly spot keywords.

Effectiveness

Table 6.14 gives the average number of questions that participants answered correctly in the user study. Half were not able to answer the last question correctly, which required subjects to draw appropriate conclusions from the information found in multiple locations. A few (12%) failed to find all the relevant text.

Overall, participants made fewer errors when answering with semantically

	Text N	Text N+S	Text N+S+D
Task 1	69 ± 24	71 ± 23	75 ± 24
Task 2	71 ± 25	73 ± 25	75 ± 25
Task 3	71 ± 23	75 ± 24	77 ± 23

Table 6.14: Average number of correct answers (in %)

linked text. This might be due to their ability to utilise the *find* function to locate every page that mentions a term or its synonyms, whereas in the plain text version they had to manually locate the relevant pages. Students reading books with definition links also obtained better scores. They remarked that by looking at the term description, they could understand the text better and formulate a more suitable query term.

H2.6 (accuracy): A one-way analysis of variance found no significant differences due to the hypertext version. From Tables 6.13 and 6.14, it can be concluded that while the hypertext version does not affect a user's level of comprehension, it does affect the time that readers require to process the text.

Searching experience

Having completed all tasks, participants were asked which hypertext version was most useful, easy to navigate and locate information, pleasant and engaging, and preferred overall. They were allowed to choose more than one version. Table 6.15 summarises their opinions.

When participants were asked to find related pages in the book, they would enter the term in the *find* function. They would then browse through each search bookmark tab along the page edges and read its mouseover text to decide whether the target page was relevant. Because they generally did not need a long time to determine whether their search term was correct, 60% of students spent their extra time consulting the table of contents to ensure that they had found all the relevant pages. A few (17%) added a bookmark to mark pages they considered relevant. The last visited bookmark tab was utilised to help subjects orient their current location in the book and jump around its pages.

Criteria	Text N	Text N+S	Text N+S+D
Useful	5	35	60
Easy to navigate	33	33	33
Easy to locate information	0	38	62
Pleasant and engaging	22	31	47
Preferred	0	29	71

Table 6.15: Format preferences (in %)

H2.7 (satisfaction): All participants found all three hypertext versions to be easy to navigate, and pleasant and engaging. However, they preferred semantically linked text. They said that the *find* function was more flexible and error tolerant because it automatically locates all terms related to the query term, and in some cases it handles misspelling. They could also obtain a better understanding of the document structure because it showed them which pages were related to one another and why.

When they could not think of a suitable query term, participants searching in the plain text version would look at the table of contents and guess which section might be relevant. They would then scan through it to find the relevant passages and obtain better query terms. Subjects reading the book with definition links utilise the brief description displayed when they moused over a keyword to help them obtain better search terms. 71% of students preferred the text with definition links because they felt that they could scan passages more quickly and understand the content better than the other hypertext versions.

Although users could click on a definition link to obtain a detailed explanation of the term in Wikipedia, only a few (13%) did so. The subjects said that the short explanation given by these links was enough for them. For those who did follow the links, they remarked that once they went to Wikipedia, they were tempted to browse to other unrelated material and needed some time to focus their attention back to their main task.

6.5 Personalising

As noted earlier, people often personalise documents to heighten their engagement with the material. To personalise an HTML page, they have to install an annotation plug-in in their Web browser. Because these plug-ins support similar annotation facilities to those in PDF, it is assumed that readers' personalisation behaviour and performance with them resembles their behaviour and performance with PDF. This study focused solely on personalisation for personal purposes.

To gather information about the different user patterns of personalisation for the task of active reading and to evaluate the hypothesis **H3.1** described in Section 6.2, an observational laboratory study with an artificial but realistic task was conducted. 24 Computer Science undergraduate students aged 20 to 29 were recruited for this experiment. None had used the Realistic Books system.

6.5.1 Material

Subjects were asked to read Chapters 6, 7 and 11 of a university level text called *Search User Interfaces*. These chapters were chosen because they were small (10 to 12 pages) and did not require background knowledge of the material to understand the text. There was no back-of-the-book indexes. The table of contents was modified by removing all references to other parts of the book. The modified book contained 38 pages, and each chapter was divided into 3–4 sections.

The book was presented in PDF and Realistic Books form and as a physical book. As in the previous experiments, the book's original design, layout and pagination was preserved in the digital forms. The document was served off-line and fully loaded before the start of each study. With PDF and Realistic Books, the beginning of each section in the book was bookmarked. The physical copy of the book was created by printing it out in colour, double-sided, and stapling down the left-hand side. Participants were provided with Post-It notes, bookmarks, pencils, pens and highlighters in different colours and sizes.

6.5.2 Tasks

This experiment was designed according to the procedure and criteria described in Section 6.2. Participants personalised the three chapters in the same order. They were exposed to each of three formats, a different one for each chapter. For example, a participant might personalise Chapter 6 in PDF format, Chapter 7 in Realistic Books, and Chapter 11 in the physical book. The six possible orderings were allocated evenly between the participants: eight subjects personalised each chapter using the same document representation. They were allowed to make any type of personalisation, e.g., scribbles, highlights, comments or bookmarks.

Participants were asked to undertake a series of tasks that explored all the basic personalisation functions provided by each document format:

1. Mark at least two passages in each subsection of the chapter

2. For each of those marked passages put, in a note, the reason for marking them

3. Bookmark each page containing a figure

4. At the end of the chapter, add a note containing at least a sentence or three keywords that best describe the chapter.

6.5.3 Results

Unlike previous experiments, this study focused on the readers' personalisation experience and strategies, not their outcome performance. There was no significant difference between each format in terms of the average number of personalisation created: physical book, 29.6 ($\sigma = 4.1$); Realistic Books, 29.5 ($\sigma = 3.4$); and PDF, 27.5 ($\sigma = 2.9$). There was also no significant difference in terms of summary quality.

During the familiarisation stage, all participants were able to explore and learn by themselves how to annotate, highlight and bookmark an electronic document. In general, they felt that all the available personalisation functionalities were

Criteria	Physical Book	PDF	Realistic Books
Useful	29	23	**46**
Easy to navigate	**42**	25	33
Easy to annotate	42	12	**46**
Pleasant and engaging	38	20	**42**
Preferred	38	12	**50**

Table 6.16: Format preferences (in %)

useful and easy to use. However, subjects preferred personalising Realistic Books over PDF and physical books.

It was observed that people tried to transfer their behaviour from paper to the computer. They utilised at most two annotation tools. Highlighting and underlining were preferred to note-taking. After they marked a passage, most participants (67%) immediately wrote their opinion of it; others waited until they had finished reading the whole chapter. Subjects only looked through their annotations when choosing keywords that best described the chapter. None wanted to publish their personalisation without further refinement.

With the electronic formats, nearly all participants (88%) bookmarked pages that contained a figure while reading the chapter. Others chose to go through the thumbnail view and bookmarked those pages first. Similarly, readers utilised the thumbnail view instead of the annotation list to help them navigate from one annotated page to the next. Although participants liked the option of changing the note from expanded to compressed form, they stated that they would rarely do so because they liked to see what they had written in order to help them locate interesting passages when revisiting the page.

As shown in Table 6.16, participants found it easier to annotate and navigate Realistic Books and physical books than PDF. Even though they only needed to read 10–12 pages in each format, they still had similar navigation experiences to those observed in the browsing experiment discussed in Section 6.3.4. This section focuses on their personalisation strategies and experiences.

Subjects felt that Realistic and physical books provided a more seamless transition from reading to personalising than PDF, where they had to frequently switch between many tools. Many (58%) complained about the fact that they did not know whether a page in PDF was bookmarked without consulting the bookmark view.

Users preferred Realistic Books over the physical copy because they could move, hide, edit and search through the annotations. They felt that they could write as much as they wanted without worrying about the free space on the page. A few (21%) said that they preferred to type than write their annotations because sometimes, when they revisit the note, they cannot understand their own handwriting.

38% of participants preferred the physical book because they were more accustomed to it and felt more comfortable with it, especially with the fluidity of scribbling. Also, they could read a document while not at a workstation.
H3.1 (satisfaction) was upheld because participants preferred to personalise Realistic Books more than the other document representations.

Physical book In the printed copy, 42% of subjects highlighted the important passages. The others preferred to underline with their pen. When asked, 75% of them said that they had a pen in their hand already, so it was quicker to underline than to highlight. The others preferred to underline because highlighter is more likely to bleed through to the other side of the page and they felt that the text looked tidier and more readable.

Participants wrote their comments in the margins near the marked passages, or in the free text space preceding the section headings. They did not want to write on a Post-It note because it covered the original text in the document. They always wanted to see the original text—providing context to their annotations. They said that they would only use a Post-It note if they wanted to write more than the margin permits, or if they wanted to remove it sometime in the future.

In this study, 22 out of 24 participants only used one pen or highlighter colour, and one underline style. Only one reader employed a colour-coded personalisation scheme: red for statements that he did not agree with, orange for definitions, and blue for important pages. The others utilised different line styles to indicate the importance of the underlined text: dashed for interesting passages and solid for text they want to remember. In PDF and Realistic Books, no subjects modified the annotation colour—they all commented that they were too lazy to go through the settings menu to change it.

PDF Although all participants had used PDF extensively and knew about the available annotation and bookmark tools, only 21% had used them. They commented that they preferred to read a printed copy of the documents, and did not personalise their electronic texts. Adobe Reader supports many annotation functions. However, almost all subjects (88%) were confused as to what each of the icons represented. Even when they knew which button to press, they felt that the annotation task distracted them from the main task because every time they wanted to highlight or add comments, they had to click the appropriate button first. Furthermore, they noticed that every time they selected an annotation tool that they had not used for a while, it responded slowly to the user's action.

Users could annotate a PDF article by selecting a *note* tool, creating a text box, or double-clicking on highlighted text. With the first two methods, the note was not associated with any text or object on the page, and could be placed anywhere in the document. With the last method, it was automatically linked to the highlighted text, and removing the highlighting automatically deleted the note. However, during the user study, nearly all participants (92%) performed a single instead of a double-click. As a result, they resorted to the *note* tool or the text box, and complained that they had to manually delete the appropriate note whenever they removed the associated highlighting.

When a user creates a note using either the first or the third method, PDF places a note icon inside the page and the associated text area outside. When the user mouses over either of them, PDF displays a line that connects them. 62% of participants preferred the line to be visible at all times so that they could immediately determine the passage referred to.

Most readers (75%) thought that selecting a passage would automatically highlight it. They found it cumbersome that they had to click the *highlight* button first before selecting the text to highlight.

Participants chose to highlight the selected text rather than underlining it. When asked, some (38%) said that they did not like the appearance of the underlined text. Others were confused with the icon, and tried to use other tools, such as line and pen, to underline the passages; they found it hard to draw a line with the mouse. Realistic Books' readers also preferred highlighting, instead of underlining text.

Realistic Books Nearly all participants (83%) found Realistic Books' personalisation tools to be intuitive. They were able to tell immediately which pages had been bookmarked. They also liked how they could just select a passage to highlight it, and that single-clicking the highlighted text automatically created a Post-It note associated with that text. Although they found the *scribble* tool useful, they remarked that it was cumbersome to draw with a mouse and they would only use it if Realistic Books was displayed on a touch screen.

In Realistic Books, clicking anywhere on a page automatically turns the page. Although participants liked this behaviour while navigating, 70% forgot about it when annotating. They thought that they needed to click on the page first to create a Post-It note, and felt annoyed when the page suddenly turned by itself.

Before typing their comments, some (58%) forgot to move their mouse cursor to the location where they wanted to place the note and instead the note was created over the page text, covering what they were commenting on. Subjects remarked that only the note icon needed to be on the page: the note text area could be outside the margin.

6.6 Constructing Realistic Books

In order to gain experience with how users perceive the Bookmaker's Workbench, a small evaluation study was performed. The final outcome of the book creation process is a file called *book.htm* that contains all the relevant metadata and defines the book's content in terms of the file names of each page image. An alternative, more primitive, way of creating books is to use an HTML text editor to construct this file directly, and shell scripts exist to convert other documents such as PDF files to page images for the book as well.

This evaluation study compared user experiences with the Bookmaker's Workbench with this primitive method to investigate the last four hypotheses listed in Section 6.2. Adobe Dreamweaver was used as the HTML editor. It supports

HTML tag auto-completion and provides a visual editing environment. 16 high school and university students aged 15–40 from a variety of disciplines were recruited to create two books using each method. Participants had encountered Realistic Books at most twice before. None had previously used the Workbench, but all knew how to create a web-page using Dreamweaver.

6.6.1 Tasks

This experiment was designed according to the procedure and criteria described in Section 6.2. Each participant was first trained to use both methods to create a book. After gaining confidence, they performed two tasks, which they undertook in the same order. Each task asked participants to fill in missing information in a 50-page book—e.g., insert a bookmark to the page that mentions a specified term, add an image that best describes this text—and to modify its appearance—e.g., change the book's size to make it more readable. This involved a total of 15 sub-tasks.

Participants used the Bookmaker's Workbench for one task and the HTML editor for the other, but in different orders: Task 1 using the Workbench and Task 2 with the text editor, or vice versa. The 16 possible orderings were allocated evenly between participants, so that eight created each book using the same tool.

6.6.2 Results

The usability of each method is assessed in terms of the outcome and process measures defined in Table 6.4. In contrast to the previous experiments, this study includes *learnability*—the time spent by participants learning each tool—to the outcome measures criteria. This experiment is something of a straw-man comparison, although the Workbench might be frustratingly slow because it frequently repaginates in order to show a faithful preview of the book. Also, the learnability of each system is informally assessed only, because the experimenter gave a verbal explanation and demonstration of the two systems. Consequently, this study focuses on qualitative comparisons and user comments rather than quantitative results.

Learnability

Quantitatively, the learnability of a system was measured by the time it took subjects to build a book, and the number of questions asked during the learning process. It is assumed that a system is easier to learn and understand if subjects spend less time and ask fewer questions during the training process. Subjects were also asked at the end of the study whether they found the system easy to learn.

The time spent creating a book using the HTML editor ranged from 18 to 45 minutes, with an average of 37 minutes. The average number of questions asked was 15. With the Workbench, the time taken ranged from 10 to 40 minutes, with an average of 28 minutes. On average, about 9 questions were asked during the training process. A t-test analysis between the two systems on both criteria—the time spent and the number of questions asked—showed that the difference between the systems was statistically significant at the five percent level.

Regardless of their computer experience, participants found the Workbench to be natural and needed less explanation about its usage than the HTML editor. They were able to quickly discover and learn the tools required to complete a task by themselves without consulting the user manual or the researcher. When asked, all participants commented that the icons chosen to represent the tools was clear and they could guess the function of each button in the toolbar. With the HTML editor, participants relied heavily on the user manual to know what tags were appropriate for completing a task.

H4.1 (easy to learn): Participants considered the Workbench easier to learn.

Efficiency

After just one training session lasting less than 50 minutes on each system, most subjects were able to create a book with minimal requirements (i.e., front and back covers, table of contents and content comprising both text and graphics) independently, without any help from the researcher, in about 20 minutes less than during training. This could be interpreted as a positive indication that subjects easily understood how to use the program.

The average times spent per sub-task ranged from 10 seconds to 9 minutes with the HTML editor, and from 5 seconds to 5 minutes with the Workbench. There was little variation in the time between participants. A t-test analysis between the two systems showed that the difference was significant at the five percent level.

Most participants completed all sub-tasks faster using the Workbench than with the HTML editor. It was observed that all participants spent most of their time in the HTML editor making sure that the book template file was a valid XHTML file. For example, to change the format of a specified piece of text in the book, participants needed to ensure that they had inserted the appropriate closing and opening tags in the correct places in the document. In fact, every time users made any kind of change to the book template file, however small, they would straight away try to visualise the resulting Realistic Books to check that the book template file was still well-formed.

Subjects did not want to run the conversion script to automatically ensure that the template file is valid, because the script took quite some time to finish. They only used it when they had made extensive changes.

The Workbench gives an instant preview of the book every time designers make any changes, and ensures that the template file is well-formed. Participants only needed to concentrate on completing the task. They spent longer creating a book using the Workbench than the HTML editor only because they performed extra measures that were not required to complete any of the sub-tasks to heighten the aesthetics of finished book with a variety of information type (i.e., graphics, audio and video).

H4.2 (speed): Participants consistently created Realistic Books faster with the Workbench than with an HTML editor.

Effectiveness

With the Workbench, participants could clearly see the correct section level and position in the page. However, a few forgot to press the *add section* button in the designer tools area when they were adding a new section. They typed the title straight into the page's main text area, so the section was not added to the automatically generated table of contents and the page was not bookmarked as a section.

Because the section number was not visible when the book template was viewed with the HTML editor, when participants were asked to insert a new section at the specified location most either specified an incorrect section level or inserted the section tag at the wrong place.

H4.3 (accuracy): All participants made fewer errors when creating books with the Workbench than with the HTML editor. The average number of sub-tasks that were completed successfully was 67% using the HTML editor and 87% using the Workbench. A t-test analysis between the two systems showed that the difference was significant at the five percent level.

User experience

Having completed all tasks, participants were asked which tools were most useful, easy to use, easy to remember, pleasant and engaging, and preferred overall.

Most subjects (88%) commented that while the HTML editor suffices for short documents without any sections or animation, they preferred the Workbench for long and more complex books. With the Workbench, they did not need to know the tags required to specify and control the appearance and the contents of the book. They could rapidly start doing their work instead of spending time ensuring that they had entered the right tags or that the template file was a well-formed XHTML file.

With the Workbench, they felt that they would still know how to use it after a time. They do not have to re-learn it all over again. Because it provides a preview of the final product, they felt more confident in exploring each tool that was available. They could see the results of the changes they made, instantly. With the HTML editor, participants felt that they always needed to consult the user manual to create a book.

H4.4 (satisfaction): All participants judged the Workbench to be the program that is most pleasant, engaging, natural and intuitive.

12% of participants said that they did not have any preference about which program to used. They felt that once familiar with the tags, they would not need to rely on the book preview to make any changes to the appearance and contents of the book, nor the list of reader tools available to the book readers.

Most subjects criticised the size of the Workbench's book space area: they wanted more space for the book. They suggested that there should be a button that allows users to minimise or hide the repository area and the design toolbar area. They also wanted a more explicit way to add additional synonym terms to the synonym table.

Participants said that when learning how to use the tool, automatic repagination after each change was useful because they could see the effect straightaway. However, for long documents, repagination takes some time to finish. Participants suggested that there should be an option to turn off automatic repagination and let the designer decide when repagination should be performed. For example, when trying different page and cover sizes for the book, designers did not want the text to be repaginated. They only wanted to repaginate once they were satisfied with the chosen page and cover size.

Users also suggested that there should be a list of possible front covers that designers could choose for their books, as well as a colour panel that allows them to select a colour instead of entering a value.

6.7 Summary

This chapter has investigated user performance and strategy while browsing, searching and personalising Realistic Books, HTML, PDF and physical books. It has shown that subjects completed tasks significantly faster, and preferred Realistic Books over the other formats. Readers can navigate and annotate Realistic Books as easily as printed books, while retaining the advantages of an electronic environment, such as editing, searching and accessing multimedia. In line with earlier studies, there was no significant difference between each format in terms of accuracy and user comprehension level (Cushman, 1986; Dillon et al., 1988; Kak, 1981; Muter et al., 1982). Both Realistic Books and the Bookmaker's Workbench were easy to learn and use.

While reading electronic text, people tend to employ the same tools and strategies as the ones that they utilise in paper books. This experiment has shown the validity of simulating documents using the book metaphor. Page turning and page edges help subjects browse around the material. The personalisation tools provided by Realistic Books provide a relatively seamless integration between reading and annotating. The automatically generated definition and semantic links help users understand the manuscript better and find related information quicker.

Some tools, such as history mechanisms and a settings tab, were rarely used by participants. There are two possible reasons: they were not required by the tasks assigned, and/or they were not considered useful.

The outcomes explained and outlined in this chapter have provided guidelines for producing electronic books, which will be discussed in the final chapter of this thesis.

7

Conclusions

The use of books to store knowledge has a long tradition—it is arguably the most mature user interface devised for presenting information. Section 2.2 described how over centuries each element of the book—shape, binding, page size, text layout and document structure—were developed to assist readers accomplish their tasks. People have also acquired a variety of strategies and manipulation techniques, such as using one finger as a bookmark or riffling through pages while browsing (Kerr, 1986; Severinson et al., 1996). Comparing a medieval book with a modern one shows that the same principles are still at work. With these conventions, readers only need to learn it once and they can use it for the rest of their life.

Current electronic document systems offer great added value over paper books. For instance, authors can revise outdated information quickly, minimise publishing costs, and incorporate hyperlinks and multimedia. Furthermore, a reader can customise how the text is presented and locate specific words or phrases through full-text search. However, current electronic document systems fail to provide adequate cues about where readers are in a document, and cannot support interactions that are quickly achieved and non-disruptive to ongoing reading activities (Conklin, 1987; Dillon, 1992; Edwards and Hardman, 1989; McDonald and Stevenson, 1998; O'Hara and Sellen, 1997). These properties are exactly the features that their physical counterparts have evolved to support.

This thesis argues that

> *Modelling digital documents as physical books can significantly improve reading performance over conventional document representations.*

Opinion on the importance of emulating the appearance of the printed books in the electronic environment is divided (Section 1.2). Some researchers argue that it is unnecessary to paginate a document or maintain its traditional page layout (Kol and Schcolnik, 2000; McKnight et al., 1991; Nielsen, 1998a; Rowland et al., 1995; Shneiderman, 1998). With further training or practice, users will no longer rely on the book model to find information.

Other researchers argue that users are dissatisfied with current digital document systems due to the lack of familiar book metaphors they embed (Barker et al., 1994; Benest, 1990; Crestani and Ntioudis, 2001; Henke, 1998; Landoni et al., 2000). Their studies observed that all participants, regardless of computing experience, find the metaphor easy to use and understand. People can utilise their previous experience with paper documents to navigate and obtain information from the digital text.

Because most of these researchers did not compare the performance of their new system with conventional ones, there was no direct evidence to support their contention that electronic documents are best displayed as books. In order to investigate whether digital books can capitalise on readers' experience with physical documents to enhance electronic reading experience, this project developed a realistic book model and utilised it to answer the three research questions posed in Section 1.4. This chapter summarises the steps taken to simulate the book model, revisits the research questions, and, finally, describes possible extensions and improvements to the work described in this thesis.

7.1 Simulating realistic book model

For an electronic book model to be widely used, it must combine not only the advantage of physical books and the electronic environment, but also be lightweight, responsive, scalable and accessible within standard Web browsers. The main challenge of simulating such a model is finding a suitable technique to compute the appearance of the turned page, the stack of pages on either side of the book and the book's spine that looks sufficiently realistic but still meets all the specified criteria.

Book model

There are two types of simulation: geometric and physical. The former defines a set of geometric equations to model the appearance of a book when a user performs a particular action. For example, one equation for turning a rigid page made from stiff cardboard, and another for turning a flexible page made from ordinary paper. The latter takes into account the forces applied by the user and the physical properties of the book—how the document is bound, and the size, shape and type of paper used. Thus, it is more versatile. A single model can be utilised to handle various possible user actions on any kind of book.

This research first surveyed and implemented four geometric and four physical computer graphic techniques for modelling the act of page turning. The geometric models are shearing techniques, a peeling effect in two and three dimensions, and a wrapping method around a cylinder or cone. With the first method, the appearance of a turned page is represented by a parallelogram that was created by shearing the page in the x direction. The parallelogram narrows as the page turns, and widens again once past the halfway point. The peeling effect simulates the act of page turning by creating a sequence of successively larger dog-ears (Section 3.1). The last model wraps the turned page around a notional cylinder or cone and alter its axis and radius in a way that follows the user's turning path (Section 3.2).

The physical models are three types of mass-spring models and a finite element method. With the former, the page is divided into a rectangular grid of small particles connected by notional springs that mimic the material's elasticity (Section 3.3). It balances the internal forces exerted by each particle against external forces—gravity, the force exerted by the book's spine and the force that the reader exerts to turn the page—to determine the paper's deformation at any given time. The latter divides the page into rectangular patches (Section 3.4). The forces on each patch—including the external forces listed previously, and those exerted by neighbouring patches—produce stress resulting in strain or deformation. The relationship between stress and strain is given by a matrix for which the components depend on the paper's type.

Chapter 3 explained and evaluated all eight models in terms of visual fidelity, execution speed and space complexities. There is a trade-off between accuracy and the system overhead: the more realistic and flexible is the model, the less interactive and scalable the system is (Section 3.5). For instance, a two-dimensional peeling technique requires less than a millisecond to turn a page, while a physical finite element model requires almost an hour. To reduce the resources required to simulate the page turning with a physical model, the paper's deformation can be pre-computed (Chu et al., 2004). However, pre-computing must be performed for every possible book form and turning path. Therefore, physical models are less versatile and interactive than the geometric ones.

For an electronic book application to be usable in practice, interactivity and scalability is more important than visual realism (Card et al., 2006; O'Hara and Sellen, 1997). This research has developed a realistic digital book system that fulfils the criteria mentioned in the beginning of Section 7.1, by utilising the simple two-dimensional peeling and shearing page turning techniques, and the geometric book model presented in Section 4.1.2. Electronic documents are simulated as rectangular hard-bound or paperback books with stiff or flexible pages. The appearance of each element in this model is updated at the end of each complete page turn. The contents of each page are loaded on demand rather than all at once before reading can commence.

Although the appearance of the turned page is less realistic compared with the one produced by the other page turning models, the differing degrees of visual accuracy, as illustrated in Figures 3.19 and 3.20, cannot readily be observed when they are viewed only from the top and are not placed side by side. Similarly, to enhance text readability, flat parallelograms, instead of curved shapes, are utilised to represent the two top-most papers, the stack of page edges and the spine of the book. The usability evaluation conducted in this research found that the highly reactive nature of the system enhances visual realism (Sections 6.3.4 and 6.4.2).

Reading services

Section 2.1 showed that reading is not merely the act of taking a book in hand and looking at its contents. Readers may look back to re-read text, underline important passages or search for specific information. They utilise different strategies when reading for pleasure and enjoyment than when reading for information and learning. Sections 4.2 and 4.3 describe all the within-document navigation and personalisation reading tools available in Realistic Books.

To support various reading tasks, users of Realistic Books are not only provided with a responsive page turning mechanism, but they can also search, annotate, bookmark, access the book non-sequentially, magnify the display and print the document. Pages can have different versions (like an animated PowerPoint), and contain hyperlinks, sound and movies, along with player controls.

According to Shum (1990) a document's physical structure should be designed to reflect its conceptual structure by utilising landmarks as reference points to important entry points. In Realistic Books, bookmarks can be automatically added to provide entries to the table of contents, back-of-the-book index, and pages containing images, annotations, terms matching the search query, and the start of a chapter or a section. Readers can also determine their style and colour.

For immersive reading, a user's interaction with an electronic text should be as quick and effortless as on paper (Dillon, 1992; O'Hara and Sellen, 1997). In Realistic Books, all the interactions are designed to ensure that a minimal number of clicks is required to perform a task, and users rarely need to shift their focus away from the book. For instance, to insert a comment, users need only point at a location where they want to insert the comment and start typing there. To highlight a passage, users need only to select the text. They do not need to click any button or press any key to switch from annotation mode to highlighting or page turning modes. The interactions are smoothly integrated within the flow of a user's reading activities, as confirmed by the user studies (Sections 6.3.4, 6.4.2, 6.4.4, and 6.5.3).

Creating a book

Simple Realistic Books are straightforward to create from PDF or HTML files using a shell script—albeit with a rudimentary structure. To take advantage of advanced features that are not normally present in such source files, book producers must edit a configuration file manually or work directly in the Flash application. Both require specialist knowledge. To rectify this, an interactive Bookmaker's Workbench was developed. It is implemented within the Adobe Flash environment, making it accessible and usable within Web browsers. A usability study of the Workbench (Section 6.6.2) found that participants thought it easy to use, understand and learn.

The process of transforming digital documents into Realistic Books involves three steps: acquisition, automated document processing and customisation. First, designers specify all the source material that constitutes the book. The text is then automatically indexed, paginated and typeset according to the method presented in Section 5.4. Simple typographic cues are utilised to indicate the document's structure. Running headers and page numbers are added automatically. Finally, designers can further customise the resulting book. Section 5.5 described how users can interact with the Workbench to choose which reading tools should be available, specify how the book should be typeset, and modify the document's content and structure.

To promote the connection of ideas in a document, the Workbench can automatically generate a hyperlinked table of contents, list of figures and back-of-the-book index pages during the second stage of the book creation process, as explained in Section 5.3. By utilising open-source software, it can automatically identify key phrases in the document, cross-reference them to Wikipedia articles, and add the phrases' synonyms to the synonym list. For instance, when readers look up "George W. Bush," the Realistic Books application bookmarks every page containing the search term or its synonyms, such as "dubya." By clicking on these bookmarks, readers can non-sequentially browse through related pages in the book.

When reading text online, people can easily access other information on the Web. Although this feature can help them understand the material better, it also makes them more susceptible to distractions. Comprehension is a continuous process; interruptions can be harmful (Black et al., 1992; Dee-Lucas and Larkin, 1992; Wright, 1993). Once readers wander away from text, it requires some time to relocate themselves in it. They are less likely to shift into deep reading. To alleviate this problem, Realistic Books enables a user to hover over a key term to display its definition; clicking the term opens the corresponding Wikipedia article.

7.2 Evaluating the hypothesis

Three research questions were introduced in Section 1.4. This section summarises the answers and shows how they help support the main hypothesis.

Q1) Does reading experience and performance significantly improve when documents are simulated as books?

People rarely read books linearly; they skip about. They may search for information in either a familiar or novel destination, browse for new information, and explore the book to determine the quantity and coverage of information without regard to the details (Canter et al., 1985). They rely on various orientation cues provided by the document interface to know their current position, the size of the document, where to go next and how to go to other areas that are not in view. Just as when people navigate on the street, they look for any landmark that have a particular characteristic, a unique purpose or meaning, or is located in a central or prominent location.

Although a considerable part of the book's physical design is devoted to these orientation cues, many electronic document systems replace them. For instance, Nielsen (1998a) and Rowland et al. (1997) think it is a waste of screen real estate to present documents as double-page spreads or to display the stack of page edges on either side of the book. In some cases, these systems provide more powerful or appropriate features than paper books for meeting the readers' needs. For example, a user can easily find a term's definition through the embedded dictionary function and modify the font size to increase text readability.

To evaluate the utility of simulating the look and feel of a physical book in an electronic environment, a realistic book model is simulated (Chapter 4) and its performance is examined in a series of usability evaluations. These studies compared people's browsing, searching and annotation performance and behaviour in HTML, PDF, physical and simulated books. Sections 6.3 to 6.5 explain in detail the study's procedures and results. Performance is evaluated based on three criteria: *efficiency*, the time it took to complete a task, *effectiveness*, the number of tasks completed successfully, and *user experience*, the participant's subjective assessment on the ease of use, likeability, engagement and usefulness.

Although there are no significant differences between the formats in terms of effectiveness at all tasks, there are significant difference in terms of efficiency and user experience. As reported in Sections 6.3.4 and 6.4.2, participants completed the browsing and searching tasks significantly faster with the simulated books than the other document representations. They also preferred to browse, search and annotate (Section 6.5.3) the book models more than the other formats. Subjects thought that the Realistic Books format combined a good reading environment with a good searching experience. These results support the claim made in the hypothesis of this thesis.

Q2) Which features of paper and electronic books are essential to be incorporated into an electronic document application?

As discussed in Sections 2.3 and 2.4, current digital document systems replace or remove some of the orientation cues available in paper books. However, the results of the user studies performed in this research showed that removing these cues leads to participants feeling lost, disoriented and frustrated. Consequently, they require significantly longer to browse or search for specific information in HTML and PDF files than in the realistic book model.

To identify which printed and electronic document features are essential, this project recorded the frequency of each feature used during the studies. This section ranks the most commonly used orientation cues and interactions in the HTML, PDF, physical and simulated books based on the participants' subjective opinion on each feature, as explained in Sections 6.3.4, 6.4.2, 6.4.4 and 6.5.3.

1. ***Pagination*** helps break the flow of information in a long document. Participants get lost more easily in HTML documents than PDF, physical or

simulated books. Page size, running header and page number define the physical boundary of the text in a page. Although theoretically page numbers and running headers can be used as landmarks that inform users of their current position, not many people notice or remember this information, regardless of document format. Instead, they utilise any visual landmark surrounding the text, the appearance of the stack of page edges or the position of the scroll bar slider to return to an approximate location.

2. ***Double-page spread, full-page view, typographic cues, illustrations and annotations*** provide readers with a broader context of the article and the way the information is organised. As a result, unique text layout, illustrations, comments, scribbles, highlighted passages, bookmarks and section titles can be utilised as visual landmarks.

   With a partial-page view or a single-page spread, participants could not immediately tell whether a page has been typeset differently than the other pages. Without any annotation or illustration, they felt that the texts always look the same. Subjects commented that they would prefer to alter the font size to make the text more readable as opposed to modifying the document's zooming factor and not being able to view the content of the whole page.

   Because in HTML and PDF documents, bookmarks are usually not visibly displayed on a page, people could only utilise section titles, figures and annotations as their visual landmarks.

3. ***Stack of page edges*** on either side of the book suggest the size of the document and the reader's current position. In HTML and PDF, the thickness of the book is represented by the size of the scroll bar's slider and the total number of pages. However, participants did not consider them as a good replacement. They would tend to scroll to the end of the document to judge its length and utilise visual landmarks to know their exact position. They found it hard to return to a known location with a scroll bar because a slight change in the slider position can significantly change the part of the document shown.

4. ***Bookmarks*** make a page accessible from any other page. Because they are always visible on the book's page edges, they are utilised by users to indicate important pages, the size of a section and the users' relative position. However, in HTML and PDF, participants had to consult a bookmark list to know whether a page was bookmarked. They could not use bookmarks as a point of reference for backtracking because the list does not give any indication on how far the reader's current position is to a bookmark.

   Very few printed books, except for dictionaries, bibles and encyclopaedias, explicitly mark chapter openings. However, Realistic Books can automatically and dynamically bookmark chapter and section openings, table of contents, back-of-the-book index, pages with images or annotations, and all pages related to a specific topic. Taking advantage of the book's spatial layout, this feature makes those special pages accessible from any page and provides users with a visual indication of the document's logical structure.

5. ***Fluid interactions between browsing, searching and personalising*** help readers find information faster and have a more pleasant experience, compared with HTML and PDF. People often look back to re-read some text to get a better understanding of the passage they are currently reading. They also look ahead in the document to know the length of the current section and to decide whether it is worth continuing. When they are critically reading a document, subjects usually annotate it to mark important passages or insert their comments. These activities are all intertwined. Unlike in the physical and simulated books, these interactions are interruptive in HTML and PDF rather than unselfconscious and integrated into the flow of regular activities. Hence, people tend to shy away from them.

   As mentioned in Section 6.3.4, the interactive page turning simulation makes document more engaging, ensures that readers never lose contact with the text, and gives an indication of the users' direction of travel and helps people remember how many pages they have read. This in turn facilitates returning to known information because they remembered the number of pages to turn back or forward. Sections 6.5.3 and 6.6.2 reported that the act of riffling through pages can be preserved simply by displaying a list of the page thumbnails. People can browse through the list looking for pages with distinctive visual landmarks while still viewing the page they are currently reading, and find interesting pages by accident.

6. ***Hyperlinked table of contents and back-of-the-book index*** both function as a map that informs a reader about the depth of the information space, text organisation, and possible ways to reach specific information. Although participants utilised the table of contents heavily in all document formats, they did not consult the index page in HTML and PDF books. Moreover, the absence of various orientation cues in HTML and PDF formats means that, having looked at the index or the table of contents page, people cannot easily return to their original location and continue.

7. ***Key phrases*** in Realistic Books can be automatically identified and cross-referenced to Wikipedia articles, providing readers with another form of visual landmark and the ability to search over synonyms. Participants appreciate the fact that they can just hover over a key phrase to obtain its definition: they do not have to shift their focus away from the text to the glossary page or external dictionary function to obtain the term's definition. Section 6.4.4 showed that adding these features to Realistic Books can significantly improve users' reading experience and performance.

To summarise, the usability studies conducted in this research (Chapter 6) showed that the most essential features that need to be implemented in an electronic book application are hyperlinks, table of contents, back-of-the-book index, bookmarks, pagination, full-page view, typographic cues, stack of page edges, fluid interactions between various reading activities, and full text search function over terms and its synonyms. This results indicate that reading experience and performance in HTML and PDF documents can be enhanced by providing readers with a smooth integration between browsing, searching and personalising, and by automatically identifying and cross-referencing keywords to Wikipedia articles. The physical act of riffling pages, backtracking mechanism and the ability to change the document's magnification level are rarely used during the studies and therefore they are the least important features to be incorporated into an electronic book application.

Q3) Is it possible for a realistic book model to be routinely used in practice?

Realistic physically-based computer models of page-turning have been around for years. Although these systems make eye-catching demos, they are not used for actual reading. However, the usability evaluations conducted in this research have shown that realistic electronic books can capitalise on the wealth of experience that all readers have with paper books to provide a competitive electronic reading experience. This result suggests that the reason why those physical models are not more widely available in practical digital libraries is purely technological and not because of any usability issues.

To view a simulated realistic book, special software must be installed on the reader's computer. For instance, 3Book (Card et al., 2006) and 3D Book Visualizer (Chu et al., 2004) require the Java Runtime Environment (JRE), and Turning the Pages (British Library, 2006) utilises Adobe Shockwave Player. According to a Millburn Survey (2010), JRE and the Shockwave Player are installed on approximately 77% and 52% of Internet-enabled computers, respectively. This indicates that not only do users have to wait for the documents to be downloaded, but they may have to wait for the required software to be downloaded and installed. The more complex the software, the longer it takes to download.

Furthermore, with all of the above book models, the content of each page in a document is rendered and stored in the computer as a high-quality bitmap image. To ensure that the text is still readable at various screen resolutions, a different page image is generated for each screen size. As a result, the system requires a large amount of memory to store the page images. For example, 3Book needs 550 MB of disk space to store the contents of a 700 page A4 book (Card et al., 2006).

Users want to open and view electronic documents as quickly as opening physical books (McKnight and Dearnley, 2003). Studies have shown that Adobe Reader is a source of frustrations for many readers because of its slow start-up delay (Malama et al., 2005; Nicholas et al., 2008). People frequently access the HTML or plain text version of a PDF file despite knowing that it will be grossly inferior in appearance. For a realistic electronic book to be deployed in practice, it must look sufficiently natural, responsive in real time, scalable, incorporate all the essential features mentioned above, and not require complex or rarely installed applications for display. The realistic book model described in this thesis satisfies all of these demands.

Section 6.1 showed that Realistic Books application is highly responsive, scalable to handle large number of pages and can be build upon Adobe Flash. Flash was chosen because it is installed on 99% of Web browsers and worked across all platforms (Millburn Survey, 2010). Its file size (1.86 MB) is significantly smaller than the size of JRE (16 MB), Adobe Reader (26.1 MB) and Shockwave Player (4 MB). Users without Flash would only have to wait a maximum three minutes of in a 56K modem to install it. By utilising Adobe Flash, websites can offer Realistic Books to ordinary readers of Web documents for everyday use.

7.3 Future works

The focus of this research is to investigate whether modelling documents as books can increase a user's reading performance over reading with physical books, HTML or PDF, and if so, how significantly. A series of usability evaluations has shown that not only do people prefer to read with Realistic Books, they can also find information significantly faster without any loss of accuracy. Because physical books cannot present animation or multimedia, the focus of this thesis has been on non-interactive documents. In recent years, there has been much research on utilising interactive web-based systems as an alternative to face-to-face learning. The next step in taking the results of this research further will be to investigate whether representing audio books and interactive exercise books as Realistic Books can increase a user's learning performance as well as their reading performance.

Realistic Books have been designed so that they can be embedded into a digital library. The most immediate extension of this research is to provide a way for people to build digital libraries of books, and to investigate the usability, interface and digital rights management issues that arise. This involves creating a mechanism that allows users to automatically convert any of their conventional computer-readable documents into collections of Realistic Books using digital library software. Also, Realistic Books should be extended to work with multiple shared documents—an issue that is ignored in this thesis.

Popular book readers (like Amazon's Kindle) use relatively primitive page-turning mechanisms. Other devices simulate the effect of page-turning in a superficial way that is eye-catching but quickly loses its appeal. In contrast, Realistic Books simulates and extends the properties of books rather than pages. As described in the previous section, this is a crucial distinction that yields measurable advantages in terms of utility over other electronic displays and also over physical books. Further research should be conducted to investigate the application of this technology to electronic book readers and other hand-held devices. Some implementations of "virtual ink" (such as the E-Ink used by Kindle) are unsuitable for rapid interaction, but many other devices are, such as Apple's new iPad, as well as the more traditional iPod Touch. In addition, a usability evaluation should be conducted to examine the impact that various page-turning simulations with varying degrees of visual realism have on users' reading experience and performance.

For the last two thousand years, physical books have successfully supported people's reading activities. Although it has been predicted that paper books would no longer be a viable medium with the increasing number of texts available online, this thesis has proven that the book paradigm will be able to survive, and provides a vision of the appearance and functionality of the future book. Evaluation of the Realistic Books system has shown that a document representation that combines the look and feel of a book with the advantages of the electronic environment can significantly enhance people's online reading experience and performance.

References

Adler, A., Gujar, A., Harrison, B. L., O'Hara, K., and Sellen, A. (1998). A diary study of work-related reading: Design implications for digital reading devices. *Computer-human interaction*, pp. 241–248.

Adler, M. J. and van Doren, C. (1972). *How to read a book*. New York: Simon and Schuster.

Amazon (2009). Kindle. Available at http://www.amazon.com/Kindle-Wireless-Reading-Device-Display/dp/B00154JDAI.

Anderson, T. H. and Armbruster, B. B. (1984). Studying. *Handbook of reading research*, pp. 657–679.

Apple (2010). iBooks: A novel way to buy and read books. Available at http://www.apple.com/ipad/features/ibooks.html.

Baines, J. (1983). Literacy and ancient Egyptian society. *Man*, ***18***(3), 572–599.

Baldonado, M., Cousins, S., Gwizdka, J., and Paepcke, A. (2000). Notable: At the intersection of annotations and handheld technology. *Proceedings of HUC 2000*, pp. 100–113.

Barker, P., Richard, S., and Benest, I. (1994). Human computer interface design for electronic books. *Proceedings of the* 18^{th} *international online information meeting*, pp. 213–225.

Barnard, S. B. (1999). Libraries and e-Books: Opportunities and issues. Available at http://www.jmc.kent.edu/futureprint/1999spring/barnard.htm.

Barnes and Noble (2009). Nook. Available at http://www.barnesandnoble.com/nook/features/techspecs/?cds2Pid=30195.

Bates, M. (1986). Subject access in online catalogs: A design model. *Journal of the American society for information science*, ***37***, 357–376.

Bathe, K. J. (1995). *Finite element procedures.* New Jersey: Prentice-Hall.

Beaudouin-Lafon, M. (2001). Novel interaction techniques for overlapping windows. *UIST '01: Proceedings of the* 14^{th} *annual ACM symposium on user interface software and technology*, pp. 153–154.

Benest, I. D. (1990). A hypertext system with controlled hype. *Hypertext: State of the art*, pp. 52–63.

Bennett, A. (1994). *Writing home.* London: Faber.

Benyon, D., Davies, G., Keller, L., Preece, J., and Rogers, Y. (1990). *A guide to usability - usability now!* Milton Keynes: The Open University.

Berkenkotter, C. and Huckin, T. (1993). Rethinking genre from a sociocognitive perspective. *Written communication*, ***10***(4), 475–509.

Bernard, M., Baker, R., and Fernandez, M. (2002). Paging vs. scrolling: Looking for the best way to present search result. Available at http://psychology.wichita.edu/surl/usabilitynews/41/paging.htm.

Bertschi, N. (1520). *Gallican psalter with canticles.* Germany: The Library Company of Philadelphia.

Bhangal, S. (2004). The page turn effect in Flash MX. Available at http://www.oreillynet.com/pub/a/javascript/2004/09/03/flashhacks.html.

Birkerts, S. (1994). *The Gutenberg elegies: The fate of reading in an electronic age.* Boston, MA: Faber and Faber.

Black, A., Wright, P., Black, D., and Norman, K. (1992). Consulting on-line dictionary information while reading. *Hypermedia*, ***4***(3), 145–169.

Bouma, H. (1973). Visual interference in the parafoveal recognition of initial and final letters of words. *Vision research*, ***13***, 762–782.

Boyle, C. and Snell, J. (1988). Intelligent navigation for semi-structured hypertext documents. *Hypertext: State of the art*, pp. 34–45.

British Library (2006). Turning the Pages. Available at http://www.bl.uk/onlinegallery/ttp/ttpbooks.html.

Bucalem, M. L. and Bathe, K. J. (1993). Higher-order MITC general shell elements. *International journal of numerical methods in engineering*, ***36***, 3729–3754.

Buchanan, G. and Pearson, J. (2008). Improving placeholders in digital documents. *ECDL 2008*, pp. 1–12.

Burbules, N. C. (1998). Rhetorics of the Web: Hyperreading and critical literacy. *Page to screen: Taking literacy into the electronic era*, pp. 102–122.

Canter, D., Rivers, R., and Storrs, G. (1985). Characterizing user navigation through complex data structures. *Behaviour and information technology*, ***4***, 93–102.

Card, S. K., Robertson, G. G., and York, W. (1996). The WebBook and the WebForager: An information workspace for the World-Wide Web. *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 111–117.

Card, S. K., Hong, L., Mackinlay, J. D., and Chi, E. H. (2004a). 3Book: A scalable 3D virtual book. *Extended abstracts of the ACM international conference on human factors in computing systems (CHI 2004)*, pp. 1095–1098.

Card, S. K., Hong, L., Mackinlay, J. D., and Chi, E. H. (2004b). 3Book: A 3D electronic smart book. *Proceedings of the working conference on advanced visual interfaces*, pp. 303–307.

Card, S. K., Hong, L., and Chen, J. D. (2006). Turning pages of 3d electronic books. *2006 IEEE symposium on 3D user interfaces*, pp. 159–165.

Carroll, J. M. and Mack, R. L. (1985). Metaphor, computing systems and active learning. *International journal of man-machine studies*, ***22***, 39–57.

Chiesi, H. I., Spilich, G. J., and Voss, J. F. (1979). Acquisition of domain-related information in relation to high and low domain knowledge. *Journal of verbal learning and verbal behavior*, ***18***, 275–290.

Choi, K. (2010). Significance of trade. Available at http://www.econ.iastate.edu/classes/econ355/Choi/sig.htm.

Choi, K. J. and Ko, H. S. (2002). Stable but responsive cloth. *SIGGRAPH 2002 conference proceedings: ACM transactions on graphics*, ***21***(3), 604–611.

Chu, Y.-C., Witten, I. H., Lobb, R., and Bainbridge, D. (2003). How to turn the page. *JCDL '03: Proceedings of the* 3^{rd} *ACM/IEEE-CS joint conference on digital libraries*, pp. 186–188.

Chu, Y.-C., Bainbridge, D., Jones, M., and Witten, I. H. (2004). Realistic books: A bizarre homage to an obsolete medium? *JCDL '04: Proceedings of the* 4^{th} *ACM/IEEE-CS joint conference on digital libraries*, pp. 78–86.

Cipolla, C. M. (1969). *Literacy and development in the West*. Harmondsworth: Penguin.

Cockburn, A. and Greenberg, S. (1999). Issues of page representation and organization in Web browser's revisitation tools. *Australian journal of information systems*, ***7***, 120–127.

Conklin, J. (1987). Hypertext: An introduction and survey. *IEEE Computer*, pp. 17–41.

Cooper, W. (1969). Is inter-indexer consistency a hobgoblin? *American documentation*, ***20***, 268–278.

Coulmas, F. (1991). *Writing system of the world*. Oxford: Wiley-Blackwell.

Crane, G. (2002). Cultural heritage digital libraries: Needs and components. *Proceedings of* 6^{th} *European conference on research and advanced technology for digital libraries*, ***2458***, 626–637.

Crawford, W. (1998). Paper persists: Why physical library collections still matter. *Online magazine*, pp. 42–48.

Crestani, F. and Ntioudis, S. P. (2001). User centred evaluation of an automatically constructed Hyper-TextBook. *Journal of educational multimedia and hypermedia*, pp. 3–19.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on Wikipedia data. *Proceedings of conference on empirical methods in natural language processing.*

Cushman, W. H. (1986). Reading from microfiche, VDT and the printed page: Subjective fatigue and performance. *Human factors*, ***28***(1), 63–73.

Darnton, R. (1989). Towards a history of reading. *Wilson quarterly*, ***13***(4), 87–102.

Davidson, K. T., Shields, M. A., and Biscos, G. (1997). *Network, screen and page: The future of reading in a digital age.* Interquest: University of Virginia.

Davis, J. N. (1989). Facilitating effects of marginal glosses on foreign language reading. *The modern language journal*, ***73***(1), 41–48.

Dayton, D. (1998). Technical editing online: The quest for transparent technology. *Journal of technical writing and communication*, ***28***, 3–38.

Dee-Lucas, D. and Larkin, J. H. (1992). *Text representation with traditional text and hypertext.* Pittsburgh: Carnegie Mellon University.

Dennison, W. (1906). Syllabification in Latin inscriptions. *Classical philology I*, pp. 47–68.

Denoue, L. and Vignollet, L. (2000). An annotation tool for Web browsers and its applications to information retrieval. *Proceedings of RIAO 2000.*

di Cantino, G. (1200). *Liber largitorius*. Rome: Biblioteca Nazionale Centrale.

Dillon, A. (1992). Reading from paper versus screens: A critical review of the empirical literature. *Ergonomics*, ***35***(10), 1297–1326.

Dillon, A. (1996). TIMS: A framework for the design of usable electronic text. *Cognitive aspects of electronic text processing*, ***LVIII***, 99–120.

Dillon, A. and McKnight, C. (1990). Towards a classification of text types: A repertory grid approach. *International journal man-machine studies*, ***33***, 623–636.

Dillon, A., McKnight, C., and Richardson, J. (1988). Reading from paper versus reading from screens. *The computer journal*, ***31***(5), 457–464.

Dillon, A., Richardson, J., and McKnight, C. (1989). Human factors of journal usage and design of electronic texts. *Interacting with computers*, ***1***(2), 183–189.

Diringer, D. (1953). *The hand-produced book*. New York: Philosophical Library.

Dix, A., Finlay, J., Abowd, G. D., and Beale, R. (2003). *Human-computer interaction*. Harlow: Prentice Hall.

Drucker, J. (2006). The virtual codex from page space to e-space. Available at http://www.philobiblon.com/drucker/.

Dunn, J. (2006). Discovering writing. Available at http://www.touregypt.net/historicalessays/discwriting.htm.

Easley, M. F. (2008). Dead sea scrolls coming to N.C. museum of natural sciences. Available at http://www.enr.state.nc.us/upclose/pages/bulletinboard27.html.

Edwards, D. and Hardman, L. (1989). Lost in hyperspace: Cognitive mapping and navigation in a hypertext environment. *Hypertext: theory into practice*.

Egan, D., Remde, J., Gomez, L., Landauer, T., Eberhardt, J., and Lochbaum, C. (1989). Formative design-evaluation of SuperBook. *ACM transactions on information systems*, ***7***(1), 30–57.

Engelbart, D. C. and English, W. K. (1968). A research center for augmenting human intellect. *Proceedings of the 1986 fall joint computer conference*, pp. 171–184.

Eveland, W. P. and Dunwoody, S. (2001). User control and structural isomorphism or disorientation and cognitive load? Learning from the Web versus print. *Communication research*, ***28***(1), 48–78.

Fidler, R. (1998). Electronic books: A good idea waiting for the right technology. Available at http://www.jmc.kent.edu/futureprint/1998fall/fidler.htm.

Fillion, F. M. and Boyle, C. D. B. (1991). Important issues in hypertext documentation usability. *Proceedings of the* 9^{th} *ACM annual international conference on systems documentation.*

François I (1527). *Lettres de François* I^{er} *au Pape.* Paris.

Fuller, D. (2006). Honors scholars get a world lesson exploring books in human culture. Available at http://www.uc.edu/news/NR.asp?id=3707.

Furnas, G. (1985). Experience with an adaptive indexing scheme. *Proceedings of CHI'85 human factors in computer systems*, pp. 131–135.

Furnas, G., Landauer, T., Gomez, L., and Dumais, S. (1987). The vocabulary problem in human-system communication. *Communications of the ACM*, ***30*** (11), 984–971.

Gabrilovich, E. and Markovitch, S. (2005). Feature generation for text categorization using world knowledge. *Proceedings of the* 19^{th} *international joint conference on artificial intelligence.*

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *Proceedings of the* 20^{th} *international joint conference on artificial intelligence.*

Gaur, A. (1984). *A history of writing.* London: British Library.

Giles, J. (2005). Internet encyclopedias go head to head. *Nature*, ***438***, 900–901.

Goldsborough, R. (2000). Text demands respect on the Web. *Advertising age*, ***71***(32), 44.

Goldschmidt, E. P. (1943). *Medieval texts and their first appearance in print.* London: Bibliographical Society.

Golovchinsky, G., Price, M. N., and Schilit, B. N. (1999). From reading to retrieval: Freeform ink annotations as queries. *Proceedings of SIGIR'99*, pp. 19–25.

Gomez, L., C.C., L., and T.K., L. (1990). All the right words: Finding what you want as a function of richness of indexing vocabulary. *Journal of the American society for information science*, ***41***(8), 547–559.

Good, M., Whiteside, J., Wixon, D., and Jones, S. (1984). Building a user-derived interface. *Communications of the ACM*, ***27***, 1032–1043.

Gordon, S., Gustavel, J., Moore, J., and Hankey, J. (1988). The effects of hypertext on reader knowledge representation. *Proceedings of the human factors society*, pp. 296–300.

Gorman, M. and Crawford, W. (1995). *Future libraries: Dreams, madness and reality.* Chicago, IL: American Library Association.

Gotoda, H. (2000). Moving finite element for simulation creasing phenomena of nearly unstretchable sheet materials. *IEEE computer graphics and application.*

Gould, J. D. (1988). How to design usable systems. *Handbook of human-computer interaction.*

Gould, J. D., Alfaro, L., Barnes, V. M., Finn, R., Grischkowsky, N., and Minuto, A. (1987). Reading is slower from CRT displays than from paper: Attempts to isolate a single-variable explanation. *Human factors*, ***29***(3), 269–299.

Graham, J. (1999). The Reader's Helper: A personalized document reading environment. *Proceedings of CHI'99*, pp. 481–488.

Haines, H. E. (1965). *Living with books: The art of book selection.* New York: Columbia University Press.

Hammond, N. and Allison, L. (1987). The travel metaphor as design principles and training aid for navigating around complex system. *Proceedings of* 3^{rd} *of human-computer interaction*, pp. 75–90.

Hammond, N. and Allison, L. (1989). Extending hypertext for learning: An investigation of access and guidance tools. *Proceedings of people and computers V*, pp. 293–304.

Hansen, W. J. and Haas, C. (1988). Reading and writing with computers: A framework for explaining differences in performance. *Communication of ACM*, ***37***(9), 1080–1089.

Harris, W. V. (1989). *Ancient literacy.* Cambridge, MA: Harvard University Press.

Henke, H. (2002). Survey on electronic book features. Available at http://www.openbook.org/doc_library/surveys/features/downloadformats/ebook_survey.pdf.

Henke, H. (1998). Are electrons better than papyrus? (or can Adobe Acrobat Reader files replace hardcopy? *ACM SIGDOC conference on scaling the heights*, pp. 29–37.

Hill, W. C., Hollan, J. D., Wroblewski, D., and McCandless, T. (1992). Edit wear and read wear. *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 3–9.

International Digital Publishing Forum (2010). Wholesale electronic book sales. Available at http://www.idpf.org/doc_library/industrystats.htm.

Internet Archive (2007). BookReader. Available at http://www.OpenLibrary.org/dev/docs/bookreader.

ISO 9241–11 (1998). Ergonomic requirements for office work with visual display terminals (VDTs)—Part 11: Guidance on usability.

Janssen, W. C. (2005). ReadUp: A widget for reading. 9th *European conference on digital library*, pp. 230–241.

Just, M. A. and Carpenter, P. A. (1980). A theory of reading: From eye fixations to comprehension. *Psychological review*, ***87***(4), 329–354.

Kahan, J. ad Koivunen, M. R. (2001). Annotea: An open RDF infrastructure for shared Web annotations. *WWW10 international conference*, pp. 623–632.

Kak, A. V. (1981). Relationships between readability of printed and CRT-displayed text. *Proceedings of human factors society annual meeting*, pp. 137–140.

Kearsley, G. (1988). *Online help: Systems design and implementation*. Norwood, NJ: Ablex.

Kennedy, A. (1984). *The psychology of reading*. London: Methuen.

Kerr, S. T. (1986). Learning to use electronic text: An agenda for research on typography. *Information design journal*, ***4***(3), 206–211.

Kilgour, F. G. (1998). *The evolution of the book*. New York: Oxford University Press.

Kinnear, P. R. and Gray, C. D. (2006). *SPSS 14 made simple*. East Sussex: Psychology Press.

Kol, S. and Schcolnik, M. (2000). Enhancing screen reading strategies. *Calico journal*, ***18***(1), 10–14.

Krotzsch, M. and Ponzetto, S. P. (2006). Wikirelate! Computing semantic relatedness using Wikipedia. *Proceedings of the* 21st *national conference on artificial intelligence*.

Landoni, M. and Gibb, F. (2000). The role of visual rhetoric in the design and production of electronic books: the visual book. *The electronic library*, ***18*** (3), 190–201.

Landoni, M., Wilson, R., and Gibb, F. (2000). From the Visual Book to the WEB Book: The importance of good design. *Research and advanced technology for digital libraries*, pp. 305–314.

Landoni, M., Wilson, R., and Gibb, F. (2001). Looking for guidelines for the production of electronic textbooks. *Online information review*, ***25***(3), 181–195.

Landow, G. P. (1996). Twenty minutes into the future, or how are we moving beyond the book? *The future of the book*, pp. 209–238.

Levin, H. and Kaplan, E. L. (1970). Grammatical structure and reading. *Basic studies on reading*.

Levy, D. M. (1997). I read the news today oh boy: Reading and attention in the digital library. *Proceedings of DL' 97*, pp. 202–211.

Lewis, J. R. (1994). Sample sizes for usability studies: Additional considerations. *Human factors*, ***36***, 368–378.

Lichtheim, M. (1973). *Ancient Egyptian literature: Volume I: The old and middle kingdoms*. Los Angeles: University of California Press.

Liu, Z. (2005). Reading behavior in the digital environment. *Journal of documentation*, ***61***(6), 700–712.

Lovelace, E. A. and Southall, S. D. (1983). Memory for words in prose and their locations on the page. *Memory and cognition*, ***11***(5), 429–434.

Lowe, E. A. (1914). *The Beneventan script: A history of the south Italian miniscule*. Oxford.

Mair, V. H. (2002). *The Columbia history of Chinese literature*. New York: Columbia University Press.

Malama, C., Landoni, M., and Wilson, R. (2005). What readers want: A study of e-Fiction usability. Available at http://www.dlib.org/may05/wilson/05wilson.html.

Mandler, J. and Johnson, N. (1977). Remembrance of things parsed: Story structure and recall. *Cognitive physchology*, ***9***, 111–151.

Mangiaracina, S. and Maioli, C. (1995). Building hypermedia for learning: A framework based on the design of user interface. *Proceedings of the* 6^{th} *international conference on human computer interactions*, pp. 857–862.

Marchionini, G. and Shneiderman, B. (1988). Finding facts vs. browsing knowledge in hypertext systems. *IEEE computer*, ***21***, 70–80.

Marley, K. (1984). Inter-indexer consistency tests: A literature review and report of a test of consistency in indexing visual materials. *Library and information science research*, ***6***, 155–157.

Marshall, C. C. (1997). Annotation: from paper books to the digital library. *Proceedings of the* 2^{nd} *ACM international conference on digital libraries*, pp. 131–140.

Marshall, C. C. and Bly, S. (2005). Turning the page on navigation. *JCDL '05: Proceedings of the* 5^{th} *joint IEEE and ACM conference on digital libraries*, pp. 225–234.

Marshall, C. C. and Ruotolo, C. (2002). Reading-in-the small: A study of reading on small form factor devices. *JCDL '02: Proceedings of the* 2^{nd} *joint IEEE and ACM conference on digital libraries*, pp. 56–64.

Marshall, C. C., Price, M. N., Golovchinsky, G., and Schilit, B. N. (1999). Introducing a digital library reading appliance into a reading group. *Proceedings of ACM digital libraries '99*, pp. 77–84.

Martin, D. (1989). *An outline of book design.* London: Blueprint/The Publisher Association.

Martin, H. J. (1994). *The history and power of writing.* Chicago: University of Chicago Press.

Martinez, A. (2010). Amazon.com's Kindle fails first college test. Available at http://seattletimes.rwsource.com/html/businesstechnology/2011938870_kindle24.html.

McCusker, D. (1998). The design of Rocket eBook. Available at http://www.zippublishing.com/abstracts/rocket_ebook_ergonomics.htm.

McDonald, S. and Stevenson, R. J. (1998). Navigation in hyperspace: An evaluation of the effects of navigational tools and subject matter expertise on browsing and information retrieval in hypertext. *Interacting with computers*, ***10***, 129–142.

McEneaney, J. E. (2000). Learning on the Web: A content literacy perspective. *Reading online.*

McKeown, M. G., Beck, I. L., Sinatra, G. M., and Loxterman, J. A. (1992). The contribution of prior knowledge and coherent text to comprehension. *Reading research quarterly*, ***27***, 79–93.

McKnight, C. (1997). Electronic journals: What do users think of them? *Proceedings of the international symposium on research, development and practice in digital libraries.*

McKnight, C. (1996). What makes a good hypertext? *Cognitive aspects of electronic text processing*, ***LVIII***, 213–238.

McKnight, C. and Dearnley, J. (2003). Electronic book use in a public library. *Journal of librarianship and information science*, ***35***(4), 235–242.

McKnight, C., Dillon, A., and Richardson, J. (1989). A comparison of linear and hypertext formats in information retrieval. *Proceedings of hypertext II.*

McKnight, C., Dillon, A., and Richardson, J. (1991). *Hypertext in context.* Cambridge: Cambridge University Press.

McNamara, D. S., Kintsch, E., Songer, N. B., and Kintsch, W. (1996). Are good texts always better? Text coherence, background knowledge and levels of understanding in learning from text. *Cognition and instruction*, ***14***, 1–43.

Michalowski, P. (1993). Writing and literacy in early states: A Mesopotamianist perspective. *Literacy: Interdisciplinary conversations*, pp. 49–70.

Mihalcea, R. and Csmoai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. *Proceedings of ACM conference on information and knowledge management*, pp. 233–242.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into texts. *Proceedings of the conference on empirical methods in natural language processing*.

Millburn Survey (2010). Flash player penetration. Available at http://www.adobe.com/products/player_census/flashplayer.

Mills, C. B. and Weldon, L. J. (1986). Reading text from computer screens. *ACM computing surveys*, ***19***, 329–357.

Milne, D. and Witten, I. (2008). Learning to link with Wikipedia. *Proceedings of ACM conference on information and knowledge management*, pp. 509–518.

Mindlin, R. D. (1951). Influence of rotary inertia and shear on flexural motion of isotropic elastic plates. *Journal of applied mechanics*, ***18***, 31–38.

Mitchell, D. C. (1982). *The process of reading: A cognitive analysis of fluent reading and learning to read*. New York: Chichester.

Mogel, L. (1996). *Making it in book publishing*. New York: Arco Pub.

Monk, A. (1990). Getting to known locations in hypertext. *Hypertext: state of the art*, pp. 20–27.

Morrison, S. (1973). *A tally of types*. Cambridge: Cambridge University of Press.

Murata, A. (1996). Empirical evaluation of performance model of pointing accuracy and speed with a PC mouse. *International journal of human computer interaction*, ***8***(4), 457–469.

Murray, T. (2002). Metalinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. *International journal of artificial intelligence in education*, ***13***(1).

Muter, P. (1996). Interface design and optimization of reading of continuous text. *Cognitive aspects of electronic text processing*, ***LVIII***, 161–180.

Muter, P. and Maurutto, P. (1991). Reading and skimming from computer screens and books: The paperless office revisited? *Behaviour and information technology*, ***10***(4), 257–266.

Muter, P., Latremouille, S. A., Treurniet, W. C., and Beam, P. (1982). Extended reading of continuous text on television screens. *Human factors*, ***24***(5), 501–508.

Narayanan, N. H. and Hegarty, M. (1998). On designing comprehensible interactive hypermedia manuals. *International journal of human-computer studies*, ***48***, 267–301.

Nelson, T. (1967). Getting it out of our system. *Information retrieval: A critical review.*

Newmark, N. M. (1959). A method of computation for structural dynamics. *ASCE journal of engineering mechanics division*, ***85***, 67–94.

Nicholas, D., Huntington, P., Jamali, H. R., Rowlands, I., Dobrowolski, T., and Tenopir, C. (2008). Viewing and reading behaviour in a virtual environment. *Aslib proceedings: New information perspectives*, ***60***(3), 185–198.

Niederhauser, D. S., Reynolds, R. E., Salmen, D. J., and Skolmoski, P. (2000). The influence of cognitive load on learning from hypertext. *Journal of educational computing research*, ***23***(3), 237–255.

Nielsen, J. (2000). Why you only need to test with 5 users. Available at http://www.useit.com/alertbox/20000319.html.

Nielsen, J. (1993). *Usability engineering.* London: Academic Press.

Nielsen, J. (1998a). Electronic books: A bad idea. Available at http://www.useit.com/alertbox/9890726.html.

Nielsen, J. (1998b). 2D is better than 3D. Available at http://www.useit.com/alertbox/981115.html.

Nielsen, J. and Molich, R. (1990). Heuristic evaluation of user interfaces. *Proceedings of CHI'90 conference on human factors in computing systems*, pp. 249–256.

Obendorf, H. (2003). Simplifying annotation support for real-world settings: A comparative study of active reading. *Proceedings of Hypertext'03*, pp. 120–121.

O'Hara, K. and Sellen, A. (1997). A comparison of reading paper and on-line documents. *Proceedings of SIGCHI conference on human factors in computing systems*, pp. 335–342.

Olsen, J. (1994). *Electronic journal literature: Implications for scholars.* London: Mecklermedia.

Olsen Jr., D. R., Taufer, T., and A., F. J. (2004). ScreenCrayons: Annotating anything. *UIST '04: Proceedings of the* 14^{th} *annual ACM symposium on user interface software and technology*, ***6***(2), 165–174.

O'Shell, S. (2006). Page flip. Available at http://www.actionscripts.org/showMovie.php?id=459.

Ovsiannikov, I. A., Michael, A. A., and McNeil, T. H. (1999). Annotation technology. *International journal of human-computer studies*, pp. 329–362.

Pakdel, A. (2006). Dokeos user interface guidelines. Available at http://www.dokeos.com/duig.php.

Paris, S. G., Wasik, B. A., and Turner, J. C. (1991). The development of strategic readers. *Handbook of reading research*, ***2***, 609–640.

Parkes, M. B. (1991). *Scribes, scripts and readers.* London.

Parkes, M. B. (1993). *Pause and effect: An introduction to the history of punctuation in the West.* Los Angeles: University of California Press.

Parkes, M. B. (1976). The influence of the concepts of ordinatio and compilatio on the development of the book. *Medieval learning and literature: Essays presented to Richard William Hunt*, pp. 115–141.

Parsons, C. G. (2001). *The efficiency and preference implications of scrolling versus paging when information seeking in long text passages.* PhD. thesis, Department of Educational Technology, University of Northern Colorado, Greely, Colorado.

Piolat, A., Roussey, J. Y., and Thunin, O. (1998). Effects of screen presentation on text reading and revising. *International journal of human computer studies*, ***47***, 565–589.

Plenkers, H. (1906). *Untersuchungen zur Überlieferungsgeschichte der ältesten lateinischen Mönchsregeln.* Munich.

Poynter Institute (2000). EyeTracking online news. Available at http://www.poynterextra.org/et/i.htm.

Pratesi, A. (1946). Sulla datazione del Virgilio Mediceo. *Rendiconti dell' accademia Naz. dei Lincei*, ***8***, 396–411.

Provot, X. (1995). Deformation constraints in a mass-spring model to describe rigid cloth behaviour. *Proceedings of graphic interface'95*, pp. 147–154.

Quesenbery, W. (2001). What does usability mean: Looking beyond 'ease of use'. *Proceedings of the* 48^{th} *annual conference of society for technical communication.*

Reed, R. (1972). *Ancient skins, parchments and leathers.* London: Seminar Press.

Reinitzer, H. (1983). *Biblia deutsch: Luthers Bibelubersetzung und ibre tradition.* Hamburg: Herzog August Bibliothek.

Reissner, E. (1945). The effect of transverse shear deformation on the bending of elastic plates. *Journal of applied mechanics*, ***67***, A69–A77.

Reynolds, L. D., Marshall, P. K., and Mynors, R. A. B. (1983). *Texts and transmissions: A survey of the Latin classics*. Oxford: Clarendon Press.

Roberts, C. H. (1954). The codex. *Proceedings of the British academy*, ***XL***, 169–204.

Roberts, C. H. and Skeat, T. C. (1983). *The birth of the codex*. London.

Roscheisen, M., Mogensen, C., and Winograd, T. (1995). Beyond browsing: Shared comments, soaps, trails, and online communities. *Computer networks and ISDN systems*, ***27***(6), 739–749.

Rothkopf, E. S. (1971). Incidental memory for location of information in text. *Journal of verbal learning and verbal behavior*, ***10***, 608–613.

Rouet, J.-F., Levonen, J. J., Dillon, A., and Spiro, R. J. (1996). An introduction to hypertext and cognition. *Hypertext and cognition*, pp. 3–8.

Roush, W. and Schaul, J. (2000). 3,500 sound off in Rocket eBook survey. Available at http://www.ebooknet.com/story.jsp?id=3591.

Roux, G. (1992). *Ancient Iraq*, 3rd edition. London: Penguin Books.

Rowland, F., McKnight, C., and Meadows, J. (1995). *Project ELVYN: an experiment in electronic journal delivery*. London: Bower Saur.

Rowland, F., Bell, I., and Falconer, C. (1997). Human and economic factors affecting the acceptance of electronic journals by readers. *Canadian journal of communication*, ***22***(3), 61–75.

Rubin, J. (1994). *Handbook of usability testing: How to plan, design and conduct effective tests*. New York: John Wiley and Sons.

Rubinstein, R. (1988). *Digital typography: An introduction to type and composition for computer system design*. Reading: Addison-Wesley Publishing Company.

Rutherford, W. G. (1905). *A chapter in the history of annotation*. London: Macmillan and Co.

Ryan, C. and Henselmeier, S. (2000). Usability testing at Macmillan. *Keywords*, ***8***, 188–202.

Saenger, P. (1996). The impact of the early printed page on the history of reading. *Bulletin de bibliophile*, (2), 237–301.

Saenger, P. (1990). The separation of words and the order of words: The genesis of Medieval reading. *Scrittura e Civiltà*, pp. 49–74.

Saenger, P. and Heinlen, M. (1991). Incunable description and its implication for the analysis of fifteenth-century reading habits. *Printing the written word: The social history of books, circa 1450–1520*, pp. 225–258.

Sandys, J. E. (1919). *Latin epigraphy: An introduction to the study of Latin inscriptions*. New York: G. P. Putnam's Sons.

Schilit, B. N., Golovchinsky, G., and Price, M. N. (1998). Beyond paper: Supporting active reading with free form digital ink annotations. *Proceedings of CHI'98*, pp. 249–256.

Schilit, B. N., Price, M. N., Golovchinsky, G., Tanaka, K., and Marshall, C. C. (1999). As we may read: The reading appliance revolution. *IEEE computer*, ***32***(1), 65–73.

Scholes, R. (1989). *Protocols of reading*. New Haven: Yale University Press.

Schumacher, G. and Waller, R. (1985). Testing design alternatives: A comparison of procedures. *Designing usable texts*.

Schwarz, E., Beldie, I. P., and Pastoor, S. (1983). A comparison of paging and scrolling for changing screen contents by inexperienced users. *Human factors*, ***24***, 279–282.

Sellen, A. and Harper, C. (2001). *The myth of the paperless office*. Cambridge, MA: MIT Press.

Sellen, A. and Harper, R. (1997). Paper as an analytic resource in the design of new technologies. *Proceedings of CHI'97*, pp. 319–326.

Severinson, E. K., Fatton, A., and Romberger, S. (1996). The paper model for computer-based writing. *Cognitive aspects of electronic text processing*, ***58***, 137–160.

Shackel, B. (1991). Usability—Context, framework, definition, design and evaluation. *Human factors for informatics usability*, pp. 21–38.

Shneiderman, B. (1989). Reflections on authoring, editing and managing hypertext. *The society of text: Hypertext, hypermedia and the social construction of information.*

Shneiderman, B. (1998). *Designing the user interface: Strategies for effective human-computer interactions.* Menlo Park, CA: Addison-Wesley.

Shum, S. (1990). Read and virtual spaces: Mapping from spatial cognition to hypertext. *Hypermedia*, ***2***(2), 133–158.

Singh, S. (1998). *Fermat's enigma.* Toronto: Viking.

Sony (2009). Sony Reader digital books. Available at http://www.sonystyle.com/webapp/wcs/stores/servlet/CategoryDisplay?catalogId=10551&storeId=10151&langId=-1&categoryId=8198552921644523779&N=4294954529&Name=All+Reader+Digital+Book.

Spool, J., Scanlon, T., Schroeder, W., Snyder, C., and DeAngelo, T. (1997). *Web site usability: A designer's guide.* North Andover: User Interface Engineering.

Strassmann, P. A. (1985). *Information payoff.* New York, NY: The Free Press.

Suetonius (2000). *Lives of the Caesars.* USA: Oxford University Press.

Summerfield, M. and Mandel, C. A. (1999). On-line books at Columbia: early findings on use, satisfaction and effect. *Technology and scholarly communication*, pp. 282–239.

Sutherland, J. (1993). Modes of production. *The Times literary supplement.*

Thomas, A. G. (1975). *Great books and books collectors.* New York: G. P. Putnam's Sons.

Tufte, E. R. (1990). *Envisioning information.* Cheshire, Connecticut: Graphics Press.

Ullman, B. L. (1960). *The origin and development of humanistic script.* Rome: Edizioni di Storia e letteratura.

Virzi, R. A. (1990). Streamlining of the design process: Running fewer subjects. *Proceedings of the* 34^{th} *annual meeting of human factors society*, pp. 291–294.

Walsh, W. S. (1892). *A handy-book of literary curiosities.* Philadelphia: J.B. Lippincott Company.

Wan, X., Yang, J., and J., X. (2007). Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. *Proceedings of the* 45^{th} *annual meeting of the ACL.*

Wang, P., Hu, J., Zeng, H.-J., Chen, L., and Chen, Z. (2007). Improving text classification by using encyclopedia knowledge. *Proceedings of the* 7^{th} *IEEE international conference on data mining*, pp. 332–341.

Watt, A. and Policarpo, F. (1998). *The computer image.* New York: ACM Press/Addison Wesley.

Waycott, J. and Kukulska-Hulme, A. (2003). Students' experiences with PDAs for reading course material. *Personal and ubiquitous computing*, ***7***(1), 30–43.

Wearden, S. (1998). Electronic books: A study of potential futures and their perceived value. Available at http://www.jmc.kent.edu/futureprint/1998fall/wearden.htm.

Weiberg, B. H. (2000). Book indexes in France: Medieval specimens and modern practices. *The indexer*, ***22***(1), 2–13.

Weinberg, B. (1996). Complexity in indexing systems, abandonment and failure: Implications for organizing the Internet. Available at http://www.asis.org/annual-96/ElectronicProceedings/weinberg.html.

Williams, G. (2000). A few things that absolutely must change in the e-Book industry. Available at http://www.ebooknet.com/story.jsp?id=2232.

Wilson, R., Landoni, M., and Gibb, F. (2003). The WEB Book experiments in electronic textbook design. *Journal of documentation*, ***59***(4), 454–477.

Wingo, E. O. (1972). Latin punctuation in the classical age. *Janua linguarum seris practica*, ***133***, 164–166.

Wright, P. (1993). To jump or not to jump: Strategy selection while reading electronic texts. *Hypertext: A psychological perspective.*

Yee, K. P. (2002). CritLink: Advanced hyperlinks enable public annotation on the Web. *ACM conference on computer-supported cooperative work*, Available at http://zesty.cs/pubs/cscw-2002-crit.pdf.

Zeichick, A. (1999). Third voice: Extending the interpersonal communications potential of the World Wide Web. Available at http://www.thirdvoice.com/about/WhitePaperPage1.htm.

Zhou, B., Luo, P., Xiong, Y., and Liu, W. (2009). Wikipedia-graph based key concept extraction towards news analysis. *2009 IEEE conference on commerce and enterprise computing*, pp. 121–128.

A

Finite element method calculation details

This appendix contains technical details of the finite element method calculations that were omitted from Section 3.4 of the main text. The critical steps concern calculating the strain, obtaining the deformation matrix, and turning the forces into a system of equations that can be integrated to determine the spatial evolution of every finite element at successive time steps.

List of symbols

x, y, z	Global coordinate system
ξ, η, ζ	Local coordinate system
$\hat{x}, \hat{y}, \hat{z}$	Normal-vector coordinate system
$\boldsymbol{p}$	Point in global coordinate
$\boldsymbol{p}_i$	Reference point i in global coordinate
$\boldsymbol{\pi}$	Point in local coordinate
$\boldsymbol{\pi}_i$	Reference point i in local coordinate
$\boldsymbol{n}$	Vector normal to the shell
$\boldsymbol{n}_i$	Vector normal to the shell at reference point i
$\boldsymbol{d}$	Displacement vector
$\boldsymbol{q}_i$	Displacement vector for reference point i
$w_i(\xi, \eta)$	Interpolation function for reference point i
t	Element's thickness
E	Young's modulus
ν	Poisson's ratio
K	Shear correction factor

A.1 Calculating the strain

The strain is determined by obtaining the derivatives of the displacement $\boldsymbol{d}$ with respect to the ξ, η and ζ axes and then multiplying these by the inverse of the Jacobian matrix.

The Jacobian calculation is defined in equation (3.28), and can be calculated by taking derivatives of point $\boldsymbol{p} = (x, y, z)$ from equation (3.23) with respect to the ξ, η and ζ axes:

$$\begin{aligned}
\frac{\partial \boldsymbol{p}}{\partial \xi} &= \sum_{i=1}^{8} \frac{\partial w_i(\xi, \eta)}{\partial \xi} (\boldsymbol{p}_i + \zeta \frac{t_i}{2} \boldsymbol{n}_i) \\
\frac{\partial \boldsymbol{p}}{\partial \eta} &= \sum_{i=1}^{8} \frac{\partial w_i(\xi, \eta)}{\partial \eta} (\boldsymbol{p}_i + \zeta \frac{t_i}{2} \boldsymbol{n}_i)
\end{aligned}$$

$$\frac{\partial \boldsymbol{p}}{\partial \zeta} = \sum_{i=1}^{8} w_i(\xi,\eta)\frac{t_i}{2}\boldsymbol{n}_i \ .$$

This yields the value of the Jacobian matrix $\boldsymbol{J}$.

The strain is the derivative of the displacement $\boldsymbol{d}$ with respect to the global coordinates x, y and z. The displacement is given in terms of the local ξ, η and ζ coordinate system by equation (3.25), and its derivatives with respect to those coordinates are:

$$\begin{aligned}
\frac{\partial \boldsymbol{d}}{\partial \xi} &= \sum_{i=1}^{8} \frac{\partial w_i(\xi,\eta)}{\partial \xi}\left\{\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} + \zeta\frac{t_i}{2}(\beta_i \boldsymbol{e}_i^{\hat{x}} - \alpha_i \boldsymbol{e}_i^{\hat{y}})\right\} \\
\frac{\partial \boldsymbol{d}}{\partial \eta} &= \sum_{i=1}^{8} \frac{\partial w_i(\xi,\eta)}{\partial \eta}\left\{\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} + \zeta\frac{t_i}{2}(\beta_i \boldsymbol{e}_i^{\hat{x}} - \alpha_i \boldsymbol{e}_i^{\hat{y}})\right\} \\
\frac{\partial \boldsymbol{d}}{\partial \zeta} &= \sum_{i=1}^{8} w_i(\xi,\eta)\frac{t_i}{2}(\beta_i \boldsymbol{e}_i^{\hat{x}} - \alpha_i \boldsymbol{e}_i^{\hat{y}}) \ .
\end{aligned}$$

The strain at time h is obtained by multiplying the above derivatives by $\boldsymbol{J}^{-1}$ according to the chain rule, as given in equation (3.27):

$$\begin{aligned}
\boldsymbol{\epsilon} &= \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix} \boldsymbol{d} \\
&= \sum_{i=1}^{8} \left\{ \begin{bmatrix} a_i & 0 & 0 \\ 0 & b_i & 0 \\ 0 & 0 & c_i \\ b_i & a_i & 0 \\ c_i & 0 & a_i \\ 0 & c_i & b_i \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} + \begin{bmatrix} f_i & 0 & 0 \\ 0 & g_i & 0 \\ 0 & 0 & h_i \\ g_i & f_i & 0 \\ h_i & 0 & f_i \\ 0 & h_i & g_i \end{bmatrix} (\beta_i \boldsymbol{e}_i^{\hat{x}} - \alpha_i \boldsymbol{e}_i^{\hat{y}}) \right\}
\end{aligned}$$

$$\boldsymbol{\epsilon} = \sum_{i=1}^{8} \boldsymbol{B}_i \boldsymbol{q}_i \ .$$

The strain-displacement matrix for $\boldsymbol{p}_i$ is

$$\boldsymbol{B}_i = \begin{bmatrix} a_i & 0 & 0 & -f_i l_i^{\hat{y}} & f_i l_i^{\hat{x}} \\ 0 & b_i & 0 & -g_i m_i^{\hat{y}} & g_i m_i^{\hat{x}} \\ 0 & 0 & c_i & -h_i n_i^{\hat{y}} & h_i n_i^{\hat{x}} \\ b_i & a_i & 0 & -g_i l_i^{\hat{y}} - f_i m_i^{\hat{y}} & g_i l_i^{\hat{x}} + f_i m_i^{\hat{x}} \\ c_i & 0 & a_i & -f_i n_i^{\hat{y}} - h_i l_i^{\hat{y}} & f_i n_i^{\hat{x}} + h_i l_i^{\hat{x}} \\ 0 & c_i & b_i & -h_i m_i^{\hat{y}} - g_i n_i^{\hat{y}} & h_i m_i^{\hat{x}} + g_i n_i^{\hat{x}} \end{bmatrix} ,$$

where

$$\begin{aligned} a_i &= J_{11}^{-1} \frac{\partial w_i(\xi,\eta)}{\partial \xi} + J_{12}^{-1} \frac{\partial w_i(\xi,\eta)}{\partial \eta} \\ b_i &= J_{21}^{-1} \frac{\partial w_i(\xi,\eta)}{\partial \xi} + J_{22}^{-1} \frac{\partial w_i(\xi,\eta)}{\partial \eta} \\ c_i &= J_{31}^{-1} \frac{\partial w_i(\xi,\eta)}{\partial \xi} + J_{32}^{-1} \frac{\partial w_i(\xi,\eta)}{\partial \eta} \\ f_i &= \frac{t_i}{2}(a_i \zeta + J_{13}^{-1} w_i(\xi,\eta)) \\ g_i &= \frac{t_i}{2}(b_i \zeta + J_{23}^{-1} w_i(\xi,\eta)) \\ h_i &= \frac{t_i}{2}(c_i \zeta + J_{33}^{-1} w_i(\xi,\eta)) \ . \end{aligned}$$

Here, J_{ij}^{-1} refers to the component of $\boldsymbol{J}^{-1}$ at row i, column j. The derivatives of the interpolation function $w_i(\xi,\eta)$ with respect to the ξ and η axis are given in Table A.1.

A.2 Calculating the deformation matrix

As explained in Section 3.4.8, the deformation matrix $\widehat{\boldsymbol{D}}$ quantifies the relationship between stress and strain at point $\boldsymbol{p}$ in the normal-vector coordinate system. The transformation matrix $\boldsymbol{Q}$ is used to map $\widehat{\boldsymbol{D}}$ to the global coordinate system.

Table A.1: Interpolation function $w_i(\xi, \eta)$

i	$\frac{\partial w_i(\xi,\eta)}{\partial \xi}$	$\frac{\partial w_i(\xi,\eta)}{\partial \eta}$
1	$\frac{1}{4}(2\xi+\eta)(1-\eta)$	$\frac{1}{4}(1-\xi)(2\eta+\xi)$
2	$\frac{1}{4}(2\xi-\eta)(1-\eta)$	$\frac{1}{4}(1+\xi)(2\eta-\xi)$
3	$\frac{1}{4}(2\xi+\eta)(1+\eta)$	$\frac{1}{4}(1+\xi)(2\eta+\xi)$
4	$\frac{1}{4}(2\xi-\eta)(1+\eta)$	$\frac{1}{4}(1-\xi)(2\eta-\xi)$
5	$-\xi(1-\eta)$	$-\frac{1}{2}(1-\xi^2)$
6	$\frac{1}{2}(1-\eta^2)$	$-\eta(1+\xi)$
7	$-\xi(1+\eta)$	$-\frac{1}{2}(1-\xi^2)$
8	$\frac{1}{2}(1-\eta^2)$	$-\eta(1-\xi)$

The elements of $\boldsymbol{Q}$ are obtained from the direction cosines of the $\hat{x}, \hat{y}, \hat{z}$ coordinate axes at point $\boldsymbol{p}$ measured in the x, y, z coordinate system. These are calculated as follows. The $\hat{x}$ axis is in a direction perpendicular to the ξ axis at $\boldsymbol{p}$, namely $\boldsymbol{e}^{\hat{x}} = \frac{J_1}{|J_1|}$ where the vector J_k is the k^{th} row of the Jacobian matrix. The $\hat{z}$ axis is in a direction normal to the shell's surface at $\boldsymbol{p}$, and can be obtained by taking the cross product of two different vectors tangential to this surface. It can therefore be expressed as $\boldsymbol{e}^{\hat{z}} = \frac{J_1 \times J_2}{|J_1 \times J_2|}$. Once the directions of the $\hat{x}$ and $\hat{z}$ axes are established, the $\hat{y}$ axis is chosen to be perpendicular to them both; the unit vector along this axis is $\boldsymbol{e}^{\hat{y}} = \boldsymbol{e}^{\hat{x}} \times \boldsymbol{e}^{\hat{z}}$.

The components of the matrix $\boldsymbol{T}$ are the direction cosines of the normal-vector coordinate system at point $\boldsymbol{p}$, measured in the x, y, z coordinate system:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{e}^{\hat{x}} \\ \boldsymbol{e}^{\hat{y}} \\ \boldsymbol{e}^{\hat{z}} \end{bmatrix} = \begin{bmatrix} l^{\hat{x}} & m^{\hat{x}} & n^{\hat{x}} \\ l^{\hat{y}} & m^{\hat{y}} & n^{\hat{y}} \\ l^{\hat{z}} & m^{\hat{z}} & n^{\hat{z}} \end{bmatrix}.$$

The matrices $\widehat{\boldsymbol{D}}, \boldsymbol{D}$, and $\boldsymbol{Q}$ are all of the form

$$\begin{bmatrix} A_{1111} & A_{1122} & A_{1133} & A_{1112} & A_{1113} & A_{1123} \\ A_{2211} & A_{2222} & A_{2233} & A_{2212} & A_{2213} & A_{2223} \\ A_{3311} & A_{3322} & A_{3333} & A_{3312} & A_{3313} & A_{3323} \\ A_{1211} & A_{1222} & A_{1233} & A_{1212} & A_{1213} & A_{1223} \\ A_{1311} & A_{1322} & A_{1333} & A_{1312} & A_{1313} & A_{1323} \\ A_{2311} & A_{2322} & A_{2133} & A_{2312} & A_{2313} & A_{2323} \end{bmatrix} .$$

The suffixes have a slightly different interpretation in each of the three cases. An element in the local deformation matrix $\widehat{\boldsymbol{D}}$ is $\widehat{D}_{rstu}$, where $r, s, t, u \in \{1, 2, 3\}$. The numbers $1, 2$ and 3 signify the $\hat{x}, \hat{y}$ and $\hat{z}$ axes respectively. The global deformation matrix $\boldsymbol{D}$ has elements D_{ijkl}, with $i, j, k, l \in \{1, 2, 3\}$—and here the numbers $1, 2$ and 3 signify the x, y and z axes. The relationship between $\widehat{D}$ and D is given by the fourth-order tensor transformation

$$D_{ijkl} = \sum_{r=1}^{3} \sum_{s=1}^{3} \sum_{t=1}^{3} \sum_{u=1}^{3} T_{ir} T_{js} \widehat{D}_{rstu} T_{kt} T_{lu} \ ,$$

where T_{ab} refers to the component of $\boldsymbol{T}$ at row a and column b. This transformation can be written in matrix form:

$$\boldsymbol{D} = \boldsymbol{Q}^T \widehat{\boldsymbol{D}} \boldsymbol{Q} \ .$$

Finally, $\boldsymbol{Q}$ is an orthogonal matrix with elements Q_{mnop}, where $m, n \in \{1, 2, 3\}$ signify the axes of the global coordinate system and $o, p \in \{1, 2, 3\}$ the axes of the normal-vector coordinate system:

$$Q_{mnop} \quad = \quad \begin{cases} T_{mo} T_{np} + T_{mp} T_{no} & : \quad o \neq p \\ T_{mo} T_{np} & : \quad o = p \end{cases} .$$

A.3 Effect of forces

The element's internal energy in Section 3.4.9 is

$$\begin{aligned}
\delta E_K + \delta E_P = & \int_V (\boldsymbol{\delta d})^T (\boldsymbol{F}_I + \boldsymbol{F}_D)\, dV + \int_V (\boldsymbol{\delta \epsilon})^T \boldsymbol{\sigma}\, dV \\
= & \int_V (\boldsymbol{w \delta q})^T (\rho \boldsymbol{w} \ddot{\boldsymbol{q}}' + \kappa \boldsymbol{w} \dot{\boldsymbol{q}}')\, dV + \\
& \int_V (\boldsymbol{B \delta q})^T (\boldsymbol{D B q}' + \boldsymbol{\sigma}_0)\, dV \\
= & (\boldsymbol{\delta q})^T (\boldsymbol{M} \ddot{\boldsymbol{q}}' + \boldsymbol{C} \dot{\boldsymbol{q}}' + \boldsymbol{K q}' + \boldsymbol{S}_0)\ ,
\end{aligned}$$

where

$$\boldsymbol{M} = \int_V \rho \boldsymbol{w}^T \boldsymbol{w}\, dV \tag{A.1}$$

$$\boldsymbol{C} = \int_V \kappa \boldsymbol{w}^T \boldsymbol{w}\, dV \tag{A.2}$$

$$\boldsymbol{K} = \int_V \boldsymbol{B}^T \boldsymbol{D B}\, dV \tag{A.3}$$

$$\boldsymbol{S}_0 = \int_V \boldsymbol{B}^T \boldsymbol{\sigma}_0\, dV\ . \tag{A.4}$$

As noted in Equation (3.31), conservation of energy implies that this internal energy equates to the element's external energy, which is

$$\begin{aligned}
\delta E_T &= \int_V (\boldsymbol{\delta d})^T \boldsymbol{F}_B\, dV + \int_A (\boldsymbol{\delta d}^A)^T \boldsymbol{F}_S\, dA + (\boldsymbol{\delta q})^T \boldsymbol{R}_N \\
&= \int_V (\boldsymbol{w \delta q})^T \boldsymbol{F}_B\, dV + \int_A (\boldsymbol{w}^A \boldsymbol{\delta q})^T \boldsymbol{F}_S\, dA + (\boldsymbol{\delta q})^T \boldsymbol{R}_N \\
&= (\boldsymbol{\delta q})^T (\boldsymbol{R}_b + \boldsymbol{R}_s + \boldsymbol{R}_N)\ ,
\end{aligned}$$

where

$$\boldsymbol{R}_B = \int_V \boldsymbol{w}^T \boldsymbol{F}_B\, dV \tag{A.5}$$

$$\boldsymbol{R}_S = \int_A (\boldsymbol{w}^A)^T \boldsymbol{F}_S\, dA\ . \tag{A.6}$$

Here, $\boldsymbol{w}^A$ is obtained from the matrix $\boldsymbol{w}$ by substituting the appropriate ξ and η values.

To solve for the values of $\boldsymbol{M}, \boldsymbol{C}, \boldsymbol{K}, \boldsymbol{R}_B, \boldsymbol{R}_S$ and $\boldsymbol{S}_0$, three-point Gaussian quadrature integration is used:

$$
\begin{aligned}
\int_V f(V)\, dV &= \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} f(\xi, \eta, \zeta) |\boldsymbol{J}|\, d\xi\, d\eta\, d\zeta \\
\int_A f(A)\, dA &= \int_{-1}^{1} \int_{-1}^{1} f(\xi, \eta) |\boldsymbol{J}|\, d\xi\, d\eta \ ,
\end{aligned}
$$

where $|\boldsymbol{J}|$ is the determinant of the Jacobian matrix. In terms of the grid size, the computational complexity of $f(V)$ and $f(A)$ is $O(n^2)$. The complexity of the finite element method after the integration is $O(n^4)$.

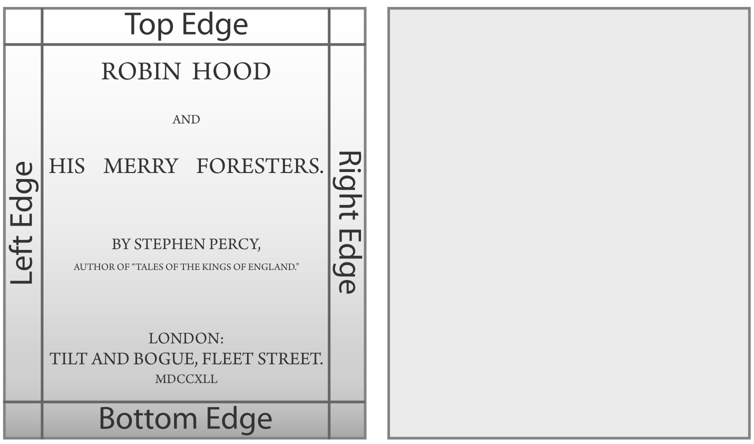
B

Calculating average colour

This appendix explains the algorithm used by the Realistic Books application to calculate the default background colour for the covers and pages of a book. Given an arbitrary image, the algorithm, shown in Figure B.1, calculates the average colour found on the outer edge of the image. The average colour is chosen instead of the majority colour to handle an image with a gradation of colour as its background. For instance, there is no majority colour found in Figure B.2(a). However, its average colour is light grey, and so the background colour for Figure B.2(a) is Figure B.2(b).

```
function getAverageColour (img):
    h ⇐ the height of img
    w ⇐ the width of img
    imgEdge[0] ⇐ all pixels on img's left edge from (0, 0) to (6, h)
    imgEdge[1] ⇐ all pixels on img's right edge from (w − 6, 0) to (w, h)
    imgEdge[2] ⇐ all pixels on img's top edge from (0, 0) to (w, 6)
    imgEdge[3] ⇐ all pixels on img's bottom edge from (0, h − 6) to (w, h)
    count := red := green := blue := 0
    for each edge in imgEdge
        for each pixel in edge
            red := red + pixel.red
            green := green + pixel.green
            blue := blue + pixel.blue
            count++
    avgColour := new Color(red/count, green/count, blue/count)
    return avgColour
```

Figure B.1: Average colour on the edges of an image



(a) (b)

Figure B.2: Example of (a) a page image and (b) its average colour

C

Calculating the width and colour of bookmarks

Realistic Books can automatically add bookmarks on pages that contain the beginnings of a chapter or a section, illustrations, annotations or search terms, and on the last page visited. To distinguish each type of bookmark, the algorithms specified in Figures C.1 and C.2 define the colour and width for each bookmark's type.

If the book designer does not specify the value for $BaseBookmarkColour$, the base colour of the bookmark is the combination of the $CoverColour$ and the $PageColour$ in a $4:1$ ratio. This ratio was chosen to ensure that the chosen $BaseBookmarkColour$ is still distinguishable from $CoverColour$ and $PageColour$. The colour of each bookmark's type is then obtained by changing the order of the red, green and blue values of $BaseBookmarkColour$.

For all types of bookmark, the default width is half the bookmark's height. However, the width of bookmarks that snap to the beginning of each chapter or section correspond to the section's level.

```
function getBookmarkColourList (BaseBookmarkColour):
⋮ colourList := [BaseBookmarkColour]
⋮ if CoverColour ≠ white
⋮ ⋮ colourList[0].red := (0.8 * CoverColour.red) + (0.2 * PageColour.red)
⋮ ⋮ colourList[0].green := (0.8 * CoverColour.green) + (0.2 * PageColour.green)
⋮ ⋮ colourList[0].blue := (0.8 * CoverColour.blue) + (0.2 * PageColour.blue))
⋮ shadeNum := 1
⋮ ## colour gradation for Bookmark_section
⋮ while colourList.length < 5
⋮ ⋮ lightColour ⇐ a colour that is a shadeNum lighter than colourList[0]
⋮ ⋮ darkColour ⇐ a colour that is a shadeNum darker than colourList[0]
⋮ ⋮ if lightColour ≠ white
⋮ ⋮ ⋮ colourList.push(lightColour)
⋮ ⋮ if colourList.length < 5 and darkColour ≠ black
⋮ ⋮ ⋮ colourList.push(darkColour)
⋮ ⋮ shadeNum++
⋮ ## colourList[0] and colourList[4] ⇐ colours for Bookmark_section
⋮ colourList.sortDarkColourToLight()
⋮ ## colourList[5] and colourList[6] ⇐ colours for Bookmark_search and Bookmark_clip
⋮ core := [colourList[2].red, colourList[2].green, colourList[2].blue]
⋮ core.sortMaxValueToMin()
⋮ if colourList[2].red > (colourList[2].green and colourList[2].blue)
⋮ ⋮ colourList.push(new Color(core[2], core[0], core[1]))
⋮ ⋮ colourList.push(new Color(core[2], core[1], core[0]))
⋮ else if colourList[2].green > (colourList[2].red and colourList[2].blue)
⋮ ⋮ colourList.push(new Color(core[1], core[2], core[0]))
⋮ ⋮ colourList.push(new Color(core[0], core[2], core[1]))
⋮ else
⋮ ⋮ colourList.push(new Color(core[0], core[1], core[2]))
⋮ ⋮ colourList.push(new Color(core[1], core[0], core[2]))
⋮ return colourList
```

Figure C.1: Calculate the list of possible colour for a bookmark

```
function getWidthForBookmark (tab):
⋮ if tab.type = section
⋮ ⋮ return tab.height ∗ 0.1 ∗ (8 − tab.section_level)
⋮ else
⋮ ⋮ return tab.height ∗ 0.5

function getColourForBookmark (tab):
⋮ baseColour :=new Color(100, 151, 202)
⋮ if BaseBookmarkColour ≠ null
⋮ ⋮ baseColour := BaseBookmarkColour
⋮ colourList := getBookmarkColourList(baseColour)
⋮ if tab.type = user
⋮ ⋮ return tab.colour
⋮ else if tab.type = section
⋮ ⋮ return colourList[tab.section_level − 1]
⋮ else if tab.type = (picture or note)
⋮ ⋮ return colourList[2]
⋮ else if tab.type = search
⋮ ⋮ return colourList[5]
⋮ else if tab.type = clip
⋮ ⋮ return colourList[6]
```

Figure C.2: Width and colour of a bookmark