e-Readers and LaTeX

Alan Wetmore

Abstract

2011 has seen many e-readers arrive on store shelves; a new generation of "touch screen" devices that include the Nook Simple Touch, Kobo eReader Touch, and a higher resolution iRiver Story HD. They all have the capability of loading user created content, so the question arises: how well can they support my legacy documents? The answer just might be, surprisingly well. After we understand the capabilities and some of the limitations we will explore how we can re-purpose older documents and prepare new LaTeX documents for use with these e-readers.

1 Introduction

There are lots of new e-reader machines this year. I've been trying out three: a Nook, a Kobo, and an iRiver. When I was invited to give a talk at the conference I decided to tell you about some of things I discovered as I explored using these machines.

I have been interested in e-readers for a while, but they all seemed to make it nearly impossible to bring your own documents to them. Recently there have been advances on more direct support of e-readers published in the latest *TUGboat* by William Cheswick [1] and Hans Hagen [2] as well as papers at this conference by Boris Veytsman and Rishi. Based on all of this work I am looking forward to seeing some truly powerful capabilities arriving soon.

How we can use them: collect our legacy documents; carry our class notes; read a manual while working away from the computer or in the field; expand our markets, ...

1.1 A quick tour of the machines

Common elements are:

Size Height and width similar to a medium sized paperback book, but quite thin.

Screen a touch screen for selecting and navigating. About 6×8 inches (150×200 mm); either 600×800 or 768×1024 pixels. In 1991 these were fair to good resolutions for a PC or low-end workstation. There were a lot of 1024×768 X terminals out there.

Bezels surrounding the screen, giving us a way to hold the reader without obscuring the text as well as a convenient place for the buttons that navigate the menus and page through the document.

Buttons at least a power switch, often page turning buttons, sometimes a keyboard.

Syncing usually through a USB cable to a PC, sometimes via wireless to their proprietary "Bookstore". Usually there is also a program for our computer that manages our purchases on the device but the memory of the e-reader can also appear as an external disk with folders where we can copy our own files.

Memory expansion usually done through a removable microSD card (SD in the iRiver). We can copy files onto the card when the e-Reader is tethered via USB and the microSD card shows up as another drive, or the card can be removed from the reader and loaded and modified using a standard card reader.

"Bookstores" are a universal feature, everyone wants to "sell" you books, magazines, and anything else they can think of. Some bookstores make it easier than others to get the books you create on the "shelf".

Speed is one of the main things that affects our experience; how "snappy" is the menu navigation, how long does it take to process our document and display it so we can start reading, and how quickly and smoothly can we "turn the pages"? All three seem acceptable to me.

Library organization All of the readers organize your "Library" and let you sort the display by author or title. In addition there is usually a search function that includes author, title, and keywords from the metadata.

Navigation The readers all make it easy to page through a book using swipes on the touchscreen and or dedicated page turning buttons. In addition they have additional capabilities for jumping to particular pages.

Document formats While all of the readers can deal with quite a few formats we will concentrate on just two — ePub and PDF. ePub because it is usually the best supported format; it is what everyone but the Kindle sells in their store. PDF because we have large legacy document collections and produce them as a matter of course. ePub has extensive support in the readers for bookmarking and note taking to support our reading.

How can we make ePub? There are two ways; first by passing through an HTML intermediary, and second by processing a PDF file. In both cases we can use a tool called `calibre` to make the conversion to ePub.

1. LaTeX $\rightsquigarrow$ HTML $\rightsquigarrow$ ePub
   LaTeX2HTML (LaTeX) $\rightsquigarrow$ HTML
   `calibre` (HTML) $\rightsquigarrow$ ePub

I haven't explored this route very much, but it could be a feasible solution. The drawback is that the conversion to HTML is not particularly robust with respect to using arbitrary LaTeX package files to enhance our documents.

2. LaTeX ⇝ PDF ⇝ ePub
   PDFLaTeX (LaTeX) ⇝ HTML
   `calibre` (PDF) ⇝ ePub
   This approach is discussed further in section 3.3.

2 PDF metadata

Metadata in PDF files is used by the e-readers to fill out author information in the list of documents available. You can use a PDF manipulator such as Acrobat or you can include the information directly using the `hyperref` package when you create your PDF file with PDFLaTeX.

```
\usepackage[%
   bookmarks=true% style guide
  ,pdfauthor={Alan Wetmore}%
  ,pdfcreator={pdfLaTeX article.cls}%
  ,pdfkeywords={e-Readers,TUG2011}%
  ,pdfsubject={e-Readers}%
  ,pdftitle={e-Readers and LaTeX}%
]{hyperref}
```

3 The most important features

3.1 Standards compliance

It turns out that even though PDF and ePub are reasonably well defined "standards", e-readers are not particularly consistent with how they consume and display these files. When we feed these devices the same files, they can produce substantially different displays and they expose different navigation and viewing options for us to use when we explore and consume our documents.

3.2 Legacy PDF files

Many of us have large collections of PDF files that we have accumulated over the years. The older ones were of course generated without regard to reading on anything other than paper, A4 or letter-size for the most part, or on our computer screens. Many of our gizmos and gadgets now ship with nothing more than an abbreviated "Quick Start Guide"; sometimes with a CD in the package with a longer manual, sometimes with or without a hint that there is a longer manual hidden somewhere on a manufacturer's web site that we can download. In section 4, we'll explore a recently published textbook to see how well we can read it on the various e-readers.

3.3 Converting PDF to ePub with calibre

One popular and powerful tool for converting between the various e-book formats is `calibre` [5]. This is a free and open source e-book library management application; it is much more than conversion software. In addition to organizing documents on your computer, it interfaces with your reader device when you plug it in, and then copies your documents to and fro.

`calibre` doesn't extract a lot of structure information from PDF files, and so doesn't generate particularly good ePub files as a result. My experiments suggest that this is not yet a very fruitful path; perhaps the `poppler` library is now ready to be used to improve this.

Alternatively, LaTeX2HTML might well be a better starting point for generating ePub format documents.

3.4 Generating new PDF files

We will want to see what options we have for typesetting new PDF files to be used with our e-readers. In section 6 we'll take a look at what a couple of very simple choices with some standard packages can do for us.

4 Exploring a textbook

We will be using the text *Principles of Uncertainty* [3] by Joseph P. Kadane of Carnegie Mellon University. This is a large (499 pages) textbook of a traditional size; it was produced by our friend Heidi Sestrich using LaTeX. It is also available as a PDF file from the author's web site.

4.1 The original document

The original document was typeset on US letter (8.5 × 11 inch) size paper with margins and crop marks for the final production size of approximately 6 by 9 inches. It was typeset with pdfTeX using LaTeX and `hyperref`. There is plenty of moderately high level mathematical notation involved to exercise the e-reader's rendering engines. All of the readers successfully load the book and allow us to read it, with varying degrees of flexibility and robustness. I'll be concentrating on Chapter 4 of the book; the demonstrations are viewable on the `river-valley.tv` web site with the conference videos.

4.2 Nook

As for PDF files, loading and viewing a PDF file looks great at first; the full page is displayed on the screen with mathematics intact. However, when you zoom in for a closer look, things quickly break down. The zoom control uses the same seven font size buttons

as the ePub viewer, but — instead of zooming — the Nook adjusts the font size and reflows the document. Plain text doesn't fare too badly, but mathematics is very corrupted, with the layout destroyed and many missing symbols.

During the conference Ross Moore supplied some "tagged pdf" files with demonstrations of mathematics; when loaded onto the Nook all the mathematics in these survived the zooming tests. This might be one route to PDF files which are usable on all of the devices.

4.3 Kobo

When viewing PDF files, a double tap to the screen zooms the image to 200%; there are also seven zoom levels available through a slider control. When the image is zoomed, you can drag it around the screen to change the viewport, with an overview widget that shows which part of the page you are on. Unfortunately there is no simple page advance mechanism in zoomed mode, e.g., tapping at the margins doesn't advance the page. You must drag the image to expose either a left or right margin or activate the more complete navigation widget, or double tap again to return to full page mode and then tap at the margin.

4.4 iRiver

The original test document didn't display at all well, appearing as broken pages with overlong lines. However, the version cropped to the text box appeared correctly with mathematics intact. The iRiver has a zoom control for PDF files that sort of works; alternatively, one of the buttons brings up a rotate menu allowing the image to rotate 90° to landscape mode that expands the width of the image to fill the wider screen by applying a one-third magnification. For a well-crafted document this will usually be enough zoom to make out the details of sub- and super-scripts.

5 Idiosyncrasies of the devices

5.1 Nook

One interesting thing about the Nook is the simple ability to load and use your own images for screensavers. I've loaded mine with a collection of iconic TUG images and meeting posters.

5.2 Kobo

After copying files to the SD card memory, when you eject the memory card from the computer, the Kobo spends some time "processing" the files. It is likely that this includes scanning for meta-data and rebuilding the list of books.

5.3 iRiver

The iRiver doesn't have a touch screen, instead it has lots of buttons arranged mostly as a `QWERTY` keyboard plus a few more navigation and control keys. It also uses full size SD cards instead of the microSD format that the other devices use.

6 Generating new PDF files

As promised, we will now explore what happens as we process some text with a couple of options. The first thing we will do is use the `12pt` option to get larger text as the starting point on the device. This should reduce the need for magnifying text that caused problems for the Nook, while maintaining maximum compatibility with LaTeX packages and options.

First we will use the `geometry` package to choose our "paper" size.

6.1 Letter paper size

Using the `geometry` package to see where things go, we start with the standard 8.5 by 11 inch `papersize`. We are also using the `margin=1in` option. This results in 95 character lines and oversize margins that waste considerable screen space.

6.2 The screen option

Adding the `screen` option from the `geometry` package generates output for a 4×3 screen ratio, which matches our screen rotated 90°. It uses a 225 mm by 180 mm `papersize`. This results in 100 character lines.

6.3 An epaper option

Setting the `papersize` to 100 mm by 125 mm results in 54 character lines. This is the final choice for the `epaper` option. The following line can be added to the internal database in the package file to add our new epaper option.

```
\@namedef{Gm@epaper}#1{%
     \Gm@setsize{#1}(100,125){mm}}%
       %for e-readers
```

And for a final consideration, we'll increase the margins to 2mm. Until the `geometry` package has been updated with a suitable `epaper` option we can use the following in our preamble.

```
\usepackage[%
    papersize={100mm,125mm},%
    margin=2mm,%
    includeheadfoot%
    ]{geometry}
```

7 The future

We can expect more generations of these devices in the next few years. Will they have better PDF engines? We can hope. We can also hope that software updates to the current machines improve the PDF engines. Maybe they will make use of internal PDF links so our table of contents and cross references will work.

Will MathML be accepted into a future revision of ePub? Probably, but will it actually be supported by the readers? That is a much less certain outcome.

Will searching within a document be improved? The current capabilities are simply too slow for searching through reference materials, and without live links in the index, that doesn't really offer much of an option.

While e-readers seem to be popular for best sellers and light reading, they don't as yet replace real textbooks or the more robust PDF capabilities of real computers or powerful tablets. But perhaps they are more suited to consuming smaller chunks of material on the order of individual lessons from the Khan Academy [4].

References

[1] Cheswick, William: *iTEX — Document formatting in an ereader world*, *TUGboat* **32**:2, 2011, 158–162. http://tug.org/TUGboat/tb32-2/tb101cheswick.pdf.

[2] Hagen, Hans: *E-books: Old Wine in New Bottles*, *TUGboat* **32**:2, 2011, 152–158. http://tug.org/TUGboat/tb32-2/tb101hagen.pdf.

[3] Kadane, Joseph P.: *Principles of Uncertainty*, CRC Press, 2011. http://uncertainty.stat.cmu.edu.

[4] *Khan Academy*. http://www.khanacademy.org.

[5] Schember, John: *Calibre Quick Start Guide*. http://calibre-ebook.com/about.

◇ Alan Wetmore
US Army Research Laboratory
alan dot wetmore (at) gmail dot com