Introducing Epub2Html - adding a plain HTML view to an EPUB

[This was originally posted on the jiscPub blog – if you have any comments please go there.]

Background

EPUB ebook files are useful if you have an application to read them, but not everyone does. We have been discussing this in the Scholarly HTML movement; to some of us EPUB looks like a good general purpose packaging format for scholarship. Not just for HTML (if you can make it XTHML, that is) but potentially for other stuff that makes up a research object, such as data files or provenance information. One of the big problems, though is that the format is still not that widely known; what is a researcher to do when they are given file ending in `.epub`? That question remains unresolved at the moment, but in this post I will talk about one small step to making EPUB potentially more useful in the general academic community.

This week, I was looking at the potential for EPUB support in repositories, which I will cover in my next post. An EPUB is full of HTML, but it's not something that is necessarily straightforward to display on the web. jiscPUB colleague Liza Daley's company has a thing called IbisReader that serves EPUB over the web and worked on BookWorm, parts of which are also available as open source.

What I wanted was a bit different – I wanted to be able to add something equivalent to a README file to an EPUB that let people read the content and web site or repository managers would be able to do something with it. So, I wrote a small tool intended as demonstrator only which:

- Generates a plain HTML table of contents.
- Adds an index.html page to the root of an EPUB (this is legit, it gets added to the manifest as well, but not the TOC) with a simple frame-based navigation system so if you can open the EPUB zip, you can browse it.
- Bundles in a lightweight JavaScript viewer. Initially I tried the Paquete system from USQ, but it turned out to have a few more issues than I had hoped. For this first release I have used a bit of Liza's code from a couple of years ago, epubjs with couple of modifications. Status? Works for me. [Update a day later, not so good for long docs – but the point on the jiscPUB project is to show the kind of thing that can be done; we can look for other toolkits or improve this one.]

Demo

So here's what it looks like in real life, warts and all.

I used the test file I was working on earlier in the week with embedded metadata.

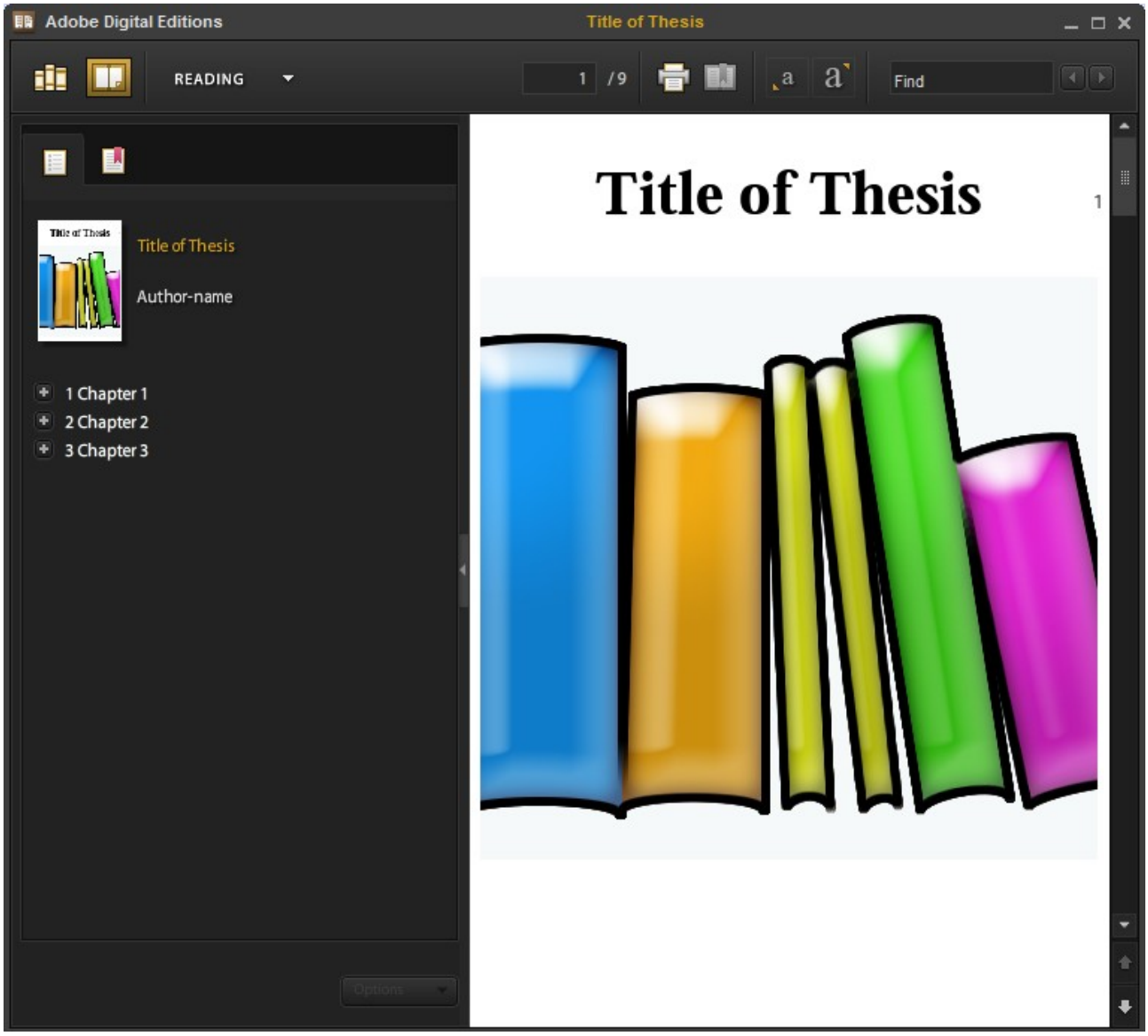

Illustration 1: Test epub from Edinburgh thesis template, with added metadata in Adobe Digital Editions

I ran the new code:

```
python epub2html.py  Edinburgh-ThesisSingleSided-plus-inline-metadata.epub
```

Which made a new file. (It does make epubckeck complain, but that's mostly to do with HTML attributes it doesn't like, not EPUB structural problems).

Edinburgh-ThesisSingleSided-plus-inline-metadata-html.epub

Now, if I unzip it there is an index.html, and some JavaScript from epubjs. In Firefox that looks like this.

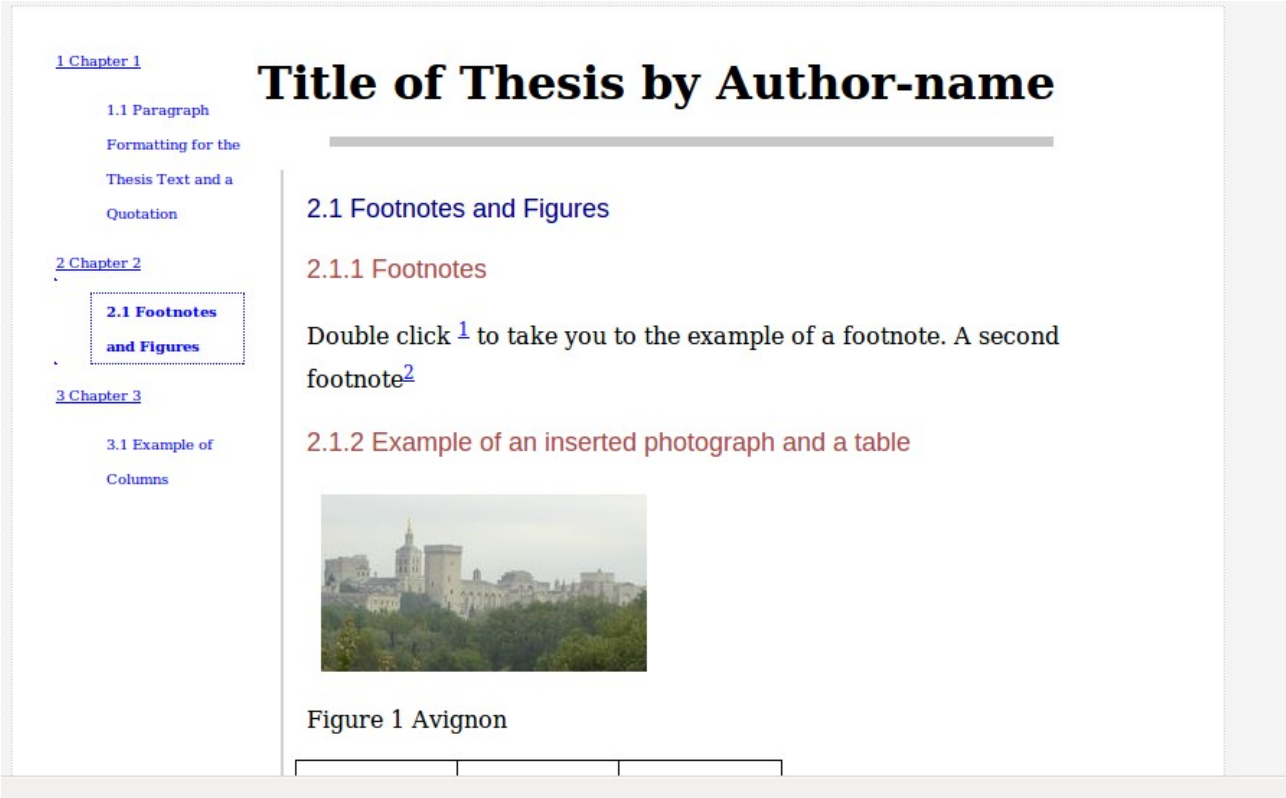

Illustration 2: HTML view of the EPUB being served from the file system, using epubjs for navigation

But, if the JavaScript is not working, then you can still see the content courtesy of the less than ideal inline frame:

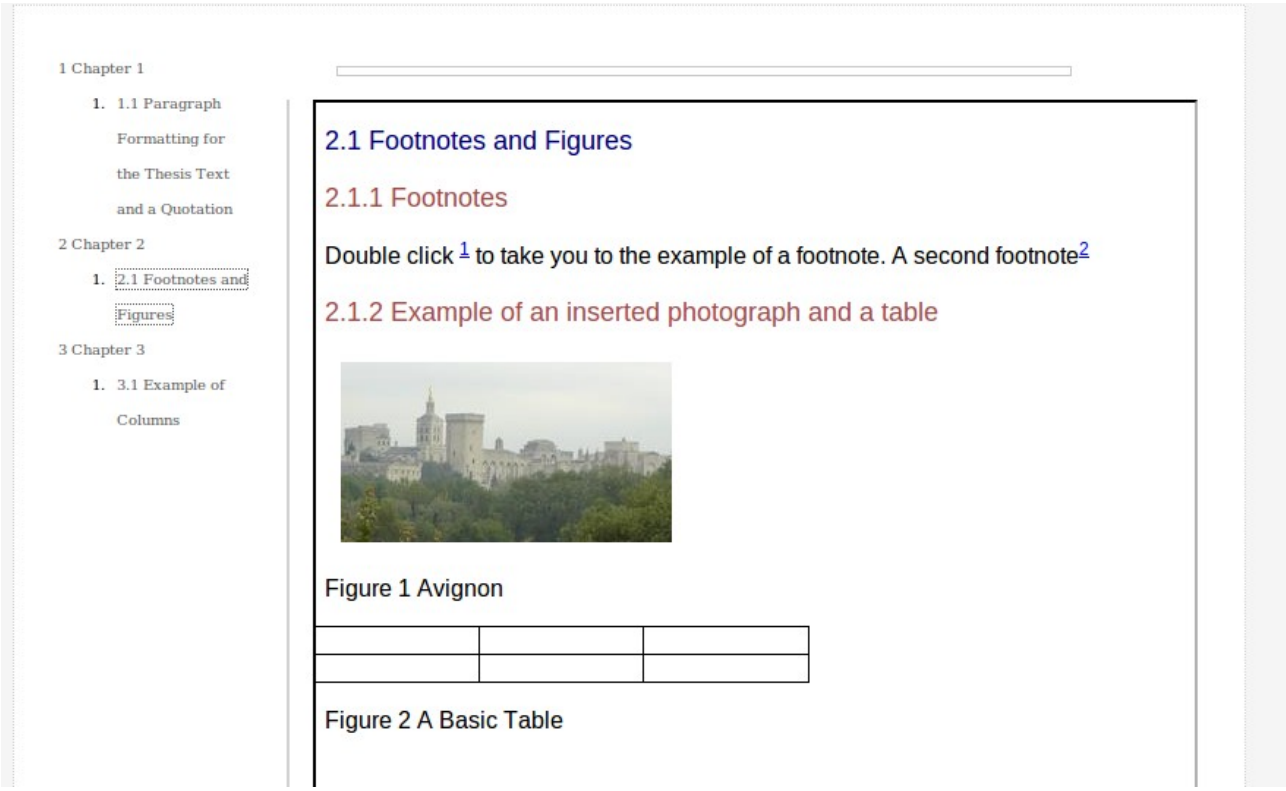

Illustration 3: Fall-back to plain HTML with no JavaScript, the index.html file has an inline frame for the EPUB content. Not elegant, but lets the content be seen.

Trying it out / the future

If you want to try this out, or help out you can get the tool from Google code.

```
svn co https://integrated-content-
environment.googlecode.com/svn/branches/temp-2011/epub2html
```

There are lots of things to do, like add command line options for output files, extracting the EPUB+HTML for immediate use (after safety checking it), choosing whether to bundle the JavaScript in the EPUB or linking to it via the web. Does anyone want this? Let us know.

One of the things I like about Paquete is that it generates # URLS for the different pages you view, making bookmarking chapters possible like this: http://demo.adfi.usq.edu.au/paquete/demo/#configuration.htm. I will explore whether this can be added to epubjs or whether it is worth pressing on with Paquete, which does have some more options like navigation buttons and a tree-widget for the table of contents.

Like I said, I did this as part of the notes I was putting together for how repositories might support EPUB, and maybe, finally, start serving real web content rather than exclusively PDF, more on that soon.

This approach might also help us add previews to web services so people can see their content in ereader-mode, something I know David Flanders the JISC manager on this project is keen on.

And finally something like this approach might be part of a tool-chain that could help people break up long documents into parts, packaged in EPUB and upload them to services like http://digress.it which want things broken up into parts.

[This was originally posted on the jiscPub blog – if you have any comments please go there.]

This post was written in OpenOffice.org, using templates and tools provided by the Integrated Content Environment project.