

Creating an ebook for the performance archive

by Tom Wrobel

The APGRD holds a rich and diverse set of material relating to drama and performance, and has always looked to use its wealth of archival material to engage with people beyond the academy.

Digital initiatives within archives have generally focused upon either improving metadata and metadata availability (e.g. making catalogues and finding aids available online), or creating digital surrogates of archive material (e.g. scanning images, converting tape to digital audio).

However, the APGRD's archive catalogue is already available online, and for the most part is catalogued to the item level. In the second instance, mass digitization of material is expensive and time consuming, and copyright and ownership issues would prohibit the vast majority of the APGRD's collection from being made available online. The copyright issues encountered in the creation of the ebooks are a sample of what the archive would face in a broader digitization project.

It was decided that the best approach was to take a limited set of the material that the APGRD held, and for which copyright and permission might be practically sought, and to curate this into a digital exhibition or project. This has the dual advantage of being achievable as a self-contained project, as well as demonstrating the breadth and diversity of the materials within the archive.

1. Why an ebook?

An ebook offers a number of discrete advantages over the traditional method of digital exhibitions – i.e. the creation of a website. These can be summed up as: Explorability; Repeatability; and Sustainability.

Explorability

One of the biggest problems in developing an effective website is empowering the user to use and explore the material available. The nearly unlimited freedom of a web page presents its own challenge: because there is no predefined structure to the content, content creators constantly have to create and refine their own. Every exhibition has to create its own paradigm for describing its structure and how to navigate it, and then to communicate this successfully to the user.

The print book, on the other hand, comes with a built-in and well-understood paradigm: a book is organized into chapters; and to move through the content in a linear fashion the user simply turns the page. You don't have to explain to a reader what a book is, you just have to hand it to them.

Ebooks sit between these two worlds, and incorporate the best features of both. A reader has a book, with chapters and pages, to explore, and the next step is always available. There is also an index, glossary, and table of contents – all familiar ways of navigating content – and in addition there are powerful additional features like searches and tagged keywords. The use of hyperlinks within the ebook allows the user to explore the content in a non-linear fashion, as in a website.

The hyperlinks allow the ebook's creators to offer multiple routes through the material, suggesting different next steps at major engagement points in order to weave different narratives through the book. Ebooks also allow content creators to enable users to engage with material – to play audio or video clips, to enlarge images or image galleries, to 'pop up' and hide additional or explanatory material that is not part of the main narrative.

While the ebook does not offer the unlimited potential of a web page or an app, it offers the majority of their freedoms: hyperlinking, engagement with digital media, manipulation of text, combined with the (optional) prebuilt distribution system of an app store. It also provides the reader with a clear idea of what the digital product (the ebook) is, how to navigate and read it, and how to engage with its contents.

In short, by embracing the restrictions of working within an ebook format and structuring the content to fit its pattern, content creators are able to get the best of both worlds with few of their limitations. The ebook represents an exciting point between the traditional analogue and digital worlds.

Repeatability

In designing the ebook project, we tried hard to come up with a project that would act as a stepping-stone for other projects that might want to do something similar. This toolkit is part of that stepping-stone.

This meant that we adopted the principle of not using complex or specialist tools that would require long-term technical support or expertise.

While creating the ebooks, the team consisted of a project lead, an archivist/creator, a developer/creator, and a technical and content consultant: resources that would not be available once the project was over, and were not likely to be available to other teams. This meant, for example, that we did not want to create an app that would require ongoing support on multiple platforms. It also meant that we did not want to use custom-tools or handcrafted HTML/Javascript that would require maintenance.

Moreover, by creating an ebook to an external standard, we could be reasonably confident that the ebook and its many components would work on our audience's various devices. While the book was tested with a sample audience, we could be confident at an early stage that what we did would pass basic technical hurdles. A webpage that includes complex interactive content has to be tested on many platforms and browsers. A mobile app requires even more device testing, and might not even be approved for distribution if it failed certain criteria. By creating an ebook to a preset standard, we were able to bypass some of the necessity for user-functionality testing.

One of the great advantages of ebook technology is that there is purpose-built (or adapted) software to help authors to create new content. As a result, content can be created, and pages designed, using WYSIWYG software such as iBooks Author, much like creating a Powerpoint or Keynote presentation. Almost all of the content within the *Medea* ebook was created using tools that would be familiar to anyone who has used common creative software.

In addition, ebook creators can include pre-build interactive components, such as pop-up boxes or image galleries. These are known as 'widgets', and can be dropped into a page without requiring any coding or 'hand editing' of text files. This enabled the project to build some relatively advanced content, such as animated maps, without writing any code or using specialist software.

In addition, we decided that the toolchain would use free or relatively cheap commercial software. The only software costs were for Adobe Creative Suite, which is currently available to academic institutions for approximately £16 per month. Again, this would be well within the budget of most projects.

Sustainability

The third cornerstone of our approach was sustainability. We wanted to create a product, which would be accessible and enjoyable well beyond the lifetime of the project itself.

The normal approach to digital sustainability is to ensure that your content is as future-proof as possible by using open and well documented and used standards for files and file formats. It is also necessary to store these objects somewhere where they can be easily accessed in the future.

For the ebook project, all of the digital objects which comprise the ebooks were either created and managed in open formats, or were converted into them for an archive copy and a final version. All of the digital objects will be placed in the ORA research archive (<http://ora.ox.ac.uk>) and available online in perpetuity.

This approach, however, does not fully satisfy the description of a sustainable resource. It works on the level of individual files and digital objects, not on preserving the

resource itself. The ebook may contain text, images, videos, etc., but it is more than the sum of its parts. We wanted an approach to digital sustainability that went beyond preserving the atomic units of content and looked to preserve the work as a whole.

The conventional approach to this with digital content such as exhibitions is to publish the work online using open standards, and to guarantee to maintain the relevant web site for as long as possible. However, this is not a practical or realistic guarantee beyond a relatively short (five to ten year) timeframe: websites must be funded, maintained, and managed, otherwise they fall prey to hackers or are simply switched off.

When coming up with the decision to use ebook technology, we did consider creating an app instead. Ebooks and apps have certain things in common: both generally rely on the same underlying technologies for book content (HTML, Javascript, and CSS); and both are generally downloaded from online marketplaces onto mobile devices.

Apps and ebooks share a major distinction over a conventional webpage, in that they can be distributed using an online marketplace. Online marketplaces, such as Apples iTunes and App Store or the Google Store, enable mobile users to discover and download apps and ebooks (and music, etc.), and act (to an extent) as a gateway which protects the user from potentially harmful content. Moreover, distribution via a marketplaces means that the content does not need to depend on a website being available: if a website is down for any reason, the content is unavailable, while an app remains present on a user's mobile device.

There are distinct differences between an ebook and an app, however. An ebook is content within a digital file that must be read by specific software (a 'reader'), while an app is a stand-alone piece of software. While the ebook creator controls the content, the structure of that content is laid down by the standard the ebook reader expects. A parallel would be a music file such as an MP3 that has to be played using an audio player.

An app does not require a reader, or rather it is book and reader together. Because the creator controls both the reader and the content, there are fewer constraints on what the content creator can do. App creators such as Touch Press (www.touchpress.com) have been able to create successful ebook-like products, such as their digitally enriched version of *The Waste Land*.

Because of this additional independence, apps also require regular maintenance. They are developed for specific versions of mobile operating systems, and once those systems are updated (or become obsolete), the app itself must be updated to meet the standards of the new platform. Like a web page, if an app is not regularly updated, it will become obsolete within a relatively short period of time, perhaps as little as three years.

Ebooks also do not require a marketplace: they can be transferred like normal files (even sent by email), and offered for download on a web site. With the rare exception of 'jailbroken' devices, most users will only download apps from a marketplace, and

generally from one of a few marketplaces. Ebooks can benefit from a marketplace, but do not require one. It is important to note that in order for an ebook to be downloadable, Digital Rights Management (DRM) encryption must be disabled. DRM systems only work within online marketplace ecosystems, who manage authentication and distribution. The ebooks produced by the APGRD were developed and distributed without DRM.

By using ebooks, and ebook standards, the project has taken a different approach. Instead of seeking to preserve one copy of the content on a website, the ebook means that there are many copies of the content distributed throughout the world. And because the ebook works to an existing standard, it sits alongside millions of other ebooks that have been published and purchased. Instead of being dependent upon a single website being maintained and funded, the project relies instead on a base of millions of readers and publications: if an ebook reader exists in twenty years capable of reading an EPUB3 or an .ibooks file, then *Medea, a Performance History* will be as readable as it is today.

2. Creating the ebook

[NB Each section is followed by a series of bullet points with what we consider to be best practice]

Technical guidelines:

a) Software and hardware used

In order to make sure that the project was ‘repeatable’, we decided at an early point to use widely available software and equipment where possible. For editing the digital objects, the Adobe Creative suite was used: Photoshop (for image manipulation), Audition (for audio files), and Premiere Pro (for video editing). It is worth noting that there are free software alternatives for all three pieces of software: Gimp, Audacity, and OpenShot.

The ebooks were put together using Apple’s iBooks Author (<http://www.apple.com/ibooks-author/>). The software proved remarkably easy to use, and is free of charge (so long as you have an Apple computer). Pages are created much like individual Keynote slides (the two applications share much of the same interface). Moreover, with small changes (on which see testing, below), the working output from iBooks Author matched the finished product extremely closely.

At the time the project began, Adobe InDesign was still a more complex alternative, and having chosen our principal working format (iBooks), we decided not to use it until we needed to create the EPUB3 version of the book.

We were fortunate to have access to a Panasonic AG-AC160A video camera provided by the Classics Centre. The camera was also equipped with wireless lapel microphones that

we used for almost all of our audio recording. We used a Canon EOS camera and a tripod for all of our photography.

While this is a comparatively rich set of equipment to have available to use, it was also better than necessary. The most important technical items were a good tripod for photographing objects and a microphone for audio recording: we found that readers would forgive poor video quality, but poor audio quality made certain digital objects nearly unusable. In fact, we had to re-record several of our audio objects - the performed quotations in multiple languages – after realizing, once the objects were in place within the ebook, that the audio quality was not sufficient.

The ebook's text was shared between the team members using Microsoft Word documents with comments enabled. Each chapter had its own file, and this was held by the editorial team. The status of the digital objects was managed using a colour-coded mind map, which displayed the status of each object: planned, in progress, ready for the ebook, and copyright-cleared. This allowed us to keep a visual check on the progress of every digital object within the ebook. For the purposes of tracking, each version of a digital object – for example a thumbnail and a full-screen version of an image – was tracked separately. A spreadsheet would be an alternative for projects, which do not want to use a mind map.

- Use readily available software. If there are problems with it, it is much easier to find someone who knows how to help, or to find help online.
- Make sure that you cover the basics when you record or capture an object: images must show the whole object and not be blurry; audio must be clear and without external noises or bangs. Check what you've captured before you finish: it's easier to do another take than to arrange another recording session.
- You don't need to pay for expensive software or hardware. Although if you are using iBooks Author, then you will need an Apple computer.

b) Choosing an ebooks format

The project chose to use Apple's .iBooks format as its principal output format. This decision was not without concerns. For example, the ecosystem is entirely under the control of one corporation, and the file format used (.iBooks) can only be used on Apple devices.

This is why the project settled on two output formats, the iBooks format and the open EPUB3 format. The creation of the EPUB3 format is discussed in detail by Chris Jennings elsewhere in the toolkit. However, it is worth noting that while EPUB3 can do almost everything that the iBooks format can do, it could not do everything at the time the ebooks were being created (the animated maps, for example).

The reasons for the adoption of the iBooks format as the primary format over standard

EPUB3 were fourfold. First, Apple’s own iBooks Author software is extremely easy to use, and reliable. It also allows for the creation of more sophisticated ‘widgets’ such as the animated maps. This satisfies the ‘creatability’ requirement, above. At the time when the project began, Adobe InDesign was not nearly as capable at producing reliable EPUB3 files. Finally, one format had to be selected as the primary format for workflow purposes, and it made sense that the ‘fullest’ format (i.e. the one with the most content within it) was the one chosen.

Second, the .ibooks format is, while a proprietary format, based on open standards. While Apple is the only company able to implement it (currently), the standard is a mixture of XML/HTML, EPUB3, CSS, and Javascript. All of the technologies it uses and relies upon are open and well understood. The files within an iBook are themselves humanly readable. If, in the future, the Apple Corporation no longer existed, the ebook could still be recreated from these open standards.

Third, when the project began, the reader software capable of reading an advanced EPUB3 file was Apple’s own iBooks. No other reader was close to its level of support for interactive content. There seemed to be little point in sacrificing the advantages of using the iBooks format if the only platform we could develop a truly interactive ebook for was the iBooks reader itself. Research also showed that the ebooks market for interactive/multimedia ebooks was dominated by Apple and its iBooks store. By choosing the .ibooks format we were not noticeably shrinking our potential audience as a result of a technical decision.

Finally, there was the need for a clear and consolidated workflow. Creating one ebook was a considerable amount of work, which required managing over 80 digital objects and content. It was important to have one, and only one, working version of the ebook. To maintain two working versions during the editorial process would have duplicated a vast amount of work unnecessarily. Our strong advice to any other project is to create alternate file formats only when the content has reached its final stage.

- Make sure to pick a format that is available to your audience
- Consider sustainability not just of the ebook itself but of its component parts
- Try to ensure that there is an alternative format available (even if it is just a PDF)
- Do not to lock yourself into one supplier or piece of software where possible.

c) Files, file formats and sizes

A multimedia ebook project, by definition, handles a number of different types of original object – such as an audio recording, videotape, poster, newspaper cutting – and a variety of processing methods will be used in order to effectively manage and curate the material.

The first step for each archive object was to create a digital surrogate: a digital file or

files that represented that object as accurately as possible. For print materials, this was a scan, or a photograph (for objects that could not be scanned). For video and audio materials, a digital copy was made, sometimes simply by duplicating an existing digital file. The digital surrogate would always be made in the highest practical resolution – for some videos from VHS, this was not always a very high resolution. This enabled us to be sure that what we had was as accurate and faithful as possible, and so that if the object needed to be reused in the future (when file sizes are likely to be less of a problem than they are today) recapture would hopefully not be necessary.

In practice, this did not take up a great deal of disk space. Even a high resolution scan (say, 6MB) is not a large file size in modern computing. The only significant file size issues the project ran into were with video files, and then only truly in one instance. File size of print, audio, and short video files is not truly an issue within a modern computing infrastructure.

Where file size did have an impact was on the size of the finished ebook. We had aimed to keep the ebook to within 512MB, simply to avoid the final ebook being too big. Happily the ebook was a slightly slimmer 469MB. However, in order to do this we did have to downsample some longer videos slightly, which we would have preferred to avoid. In general, we will await feedback on whether the ebook was too big, or whether the final file size did not present a problem.

The project also made a number of original recordings: the introductory video recording, the performed quotations in multiple languages, and video interviews. These recordings were treated in the same way as archive items: a digital surrogate was created, and the working files were created from that surrogate, so as not to risk the originals.

The project started out with the aim of preserving each digital object in all its forms – first digital capture, edited intermediate files, final version in high resolution, final version in output resolution. This ideal proved impractical, partly because this required a lot of tracking and accounting of working files, and because some of the video files – for example the outsourced interviews – were so large that they needed to be downsampled before they could even be edited on our equipment.

We therefore sought to keep the original digital capture of an object – photograph, scan, original recording – no matter how high a resolution it was obtained in, and a highest-practical-resolution version of the final digital object.

In all cases the digital surrogate was either created or converted to an open standard file format. As a result of the ebook standards (both iBooks and EPUB3), the output files were all in an open standard as well: JPEG for images without any transparent parts to them; PNG for images with transparency; AAC for audio (with mp3 also captured), and H.264 for video.

This does not mean that working files were not in ‘closed’ formats, (the project’s working files were often Adobe Photoshop .psd files, for example), just that there should be a high-resolution version of the original and final versions of the digital object which are in an open format.

As an important part of keeping the overall file size of the finished book down, we were careful to optimize each image before producing the production version. Careful attention to the export settings in, for example, Adobe Photoshop, allowed us to reduce file sizes by as much as 90% for some images without affecting quality. We also cut video files where at all possible – shorter is better is a general rule in any case.

- Use ‘open’ formats to capture and archive digital objects
- Use whatever format is convenient for working versions of the content
- Capture in as high a resolution as you can: it’s very hard to create a higher-quality version of an object at a later stage, while a lower resolution version is easy to create
- Don’t worry too much about file sizes when capturing or archiving.
- Pay close attention to file sizes for digital objects in the ebook, as little things add up.

d) Workflow and publication

With so many digital objects being processed, which took a considerable amount of time, a streamlined technical workflow was vital in order to keep the project moving successfully.

One of the first and most important decisions we made was as a result of an early, now fixed, bug within InDesign’s EPUB3 output. The bug meant that default text styles were overriding the project’s own in certain circumstances. The fix for the bug was simple – the EPUB3 file had to be taken apart, one file edited, and then put back together. However, this had to happen every time the file was edited.

In technical development, there is an important principle that something should ideally be done only once. If you have to follow a complex series of manual steps in order to produce a new version of the ebook, then this adds a great deal of time and overhead to the project.

In a normal ebook or web development process, the overhead is managed by having a two-stage production process. In most scenarios, the content is completed, and then only when finished is it handed over to a technical team, who, like typesetters of old, turn the content into the finished product.

However, we found that it was very hard to create a multimedia/interactive ebook this way. The digital objects were so much the centerpiece of the book that it was only once

they were in place on the page, and the interactive components added, that it was clear how they would work and how the content needed to be structured.

In this respect, the WYSIWYG editing features of iBooks Author proved invaluable. They enabled members of the editorial team to make changes to the text content without involving members of the technical team, and enabled the technical team to mix prototype digital objects with text for review. It also meant that a new version of the ebook was only ever a save and export away. If hand coding and editing of XML/Javascript had been required, this workflow would not have been possible, or would have been much more difficult.

Moreover, the tracking of each digital object was vital to the technical management of the project. The process of creating the digital objects was driven by and interacted with the editorial process, but they were two separate workflows and were managed separately. All recordings with a single performer took place on one day, for example, and some of the basic technical processes such as copying the audio file to the server, removing excess white noise/external interruptions, and tagging the audio stream for editing, had to happen at one time, even though the recording contained material that would be used in different chapters. On a more practical level, VHS content could not be digitized until we had access to the relevant equipment.

- Make it as easy as possible to move between the working version of the ebook and an outputted version for testing and eventual publication
- It is easier to create one version of an ebook, and then recreate that ebook in another format, than it is to manage two parallel working versions
- The digital objects need to be tracked and managed separately from the editorial content.

e) Testing and review

Technical testing is one of the more important parts of any technical project. In the early stages of the ebook project, we created a technical prototype ebook. This ebook contained examples of all of the types of digital objects we wanted to create, the widgets we wanted to use, the styles and layout we were likely to use, and so on. This technical prototype was then tested on a variety of platforms and found to work successfully.

In addition, we obtained, or had privately, tablets and phones on which the technical prototype and other versions could be quickly tested. This gave us confidence that the final version would work effectively. It would not be wise to proceed without regularly exporting the working content into reader software in order to check that it works.

The main technical problems in testing were, in fact, due to the difficulties in distributing a prototype ebook. Although an ebook file can be downloaded like any other file and imported into a reader, this proved difficult for many of our test users. We did hit one serious bug, in that the old version of an ebook should be completely deleted before importing the new version: sometimes the file does not update, even though a success is reported. In part, this confirmed our decision to use an ecosystem like the iBooks store that allows for one-click downloading of ebooks for our finished ebook.

Technical testing did reveal some important design/technical problems. First, the 'click' interface (using a mouse to navigate the ebook) and the 'touch' interface (using your fingers on an ipad) produced subtly different results. In particular, clickable objects (controls) that were not too close together on screen proved to overlap in the touch interface, while some controls were unresponsive in one or the other interface. We avoided this issue by allowing for greater gaps around control objects such as images and pop up buttons, to avoid their overlapping.

Importantly, technical feedback also allowed us to notice that text that was acceptably small on a desktop screen was too small on a tablet. This was also the case for some links, which were too small to easily click on. We changed this by ensuring a minimum font size throughout, and by enlarging all of our controls.

- Have the devices that you want your product to work on readily available so you can test things quickly
- Create a technical prototype so that you can be confident you won't hit any major technical snags in later development
- Test the book with a small focus group who will give you genuine and critical feedback.

3. Future considerations

Ebooks, limitations and the future

Ebooks do come with restrictions and limitations. They contain only a limited set of the things that can be done with a web page. For example, content *must* be organised in chapters or pages – there are few opportunities to break away from the paradigm where appropriate. It might have been nice, for example, to have had a 'home' page for readers to return to, but that was not possible. In addition, features such as tables of contents are generally automatically generated, and have set formats, and it was not possible to incorporate working hyperlinks within glossary text.

One missing component was that it is not possible to build in social media applications, such as Facebook or Twitter. It might have been valuable to allow users to upload their own encounters with Medea, or submit comments or suggestions, but we could not do this.

In addition, all content must be present on the reader's device – the content creator cannot assume that the user will have internet access. This is also a benefit in certain situations, as it allows an ebook to be used in places where no internet connection is available.

Moreover, unlike a web page, there are severe limitations on what can be done with the content. Client side code, normally Javascript, can be used to animate and enrich a web page in a near-infinite number of ways. Reader software restricts the use of Javascript for security and practicality purposes, but this does curtail the use of some features: we could not have reproduced all of the features of Touch Press' *The Waste Land* within our ebook, for example.

In addition, the software available now offers new opportunities to creators. One new feature that iBooks Author has developed since the end of the project is to import data from web sources into widgets – this could have been used to link the productions database and archive catalogue into the ebook, and would have been a valuable addition to the content.

What we might have done differently from a technical perspective

Although we did manage to incorporate two testing periods into the development of the *Medea* ebook, we would probably have benefited from closer engagement with a test audience. Creating such a new and different publication for the first time was made harder by the fact that we quickly developed a familiarity and understanding of the ebook that few others had. At times, we may have found it difficult to keep our audience's response, and especially their first reactions, in mind. A more rigorous approach to user testing would be a good possible corrective.

In addition, the finished ebook is probably still too big: even in 2016, 500MB is a lot of space to ask from a mobile device user whose tablet may only have 16 or 32GB of space. Such space limitations will obviously ease in future, and for various reasons – mostly the limitations of human eyesight – it is not likely that optimum video resolutions will grow much bigger for mobile devices, 1080p is the maximum foreseeable resolution for a tablet. However, future ebooks should probably aim to demand less of a resource commitment from the user.

In addition, the desire for 'creatability', above all, and the need to ensure that the ebook content would work across multiple devices may have led us to be a conservative with our development of custom widgets. The interactive maps were a strong addition to the ebook, and in hindsight we could have been a little more ambitious with similar content. Greater familiarity with the ebook production process should allow future projects to be more adventurous.

The most important technical change since starting the project is, in fact, the proliferation of non-Apple based tablet computers. In addition, iBooks Author now allows the user to create an EPUB3 compatible ebook, although only if the ebook is created that way from the template up. It is likely that the balance between iBooks and EPUB3 as the target format would be a difficult decision if we had to make it today.

Creating the ePub3 version of the *Medea* ebook

by Chris Jenkins

What is here?

This set of pages explains various aspects of the tools used to convert the iBooks Author version of the APGRD project: *Medea, A Performance History*. This was finally published on the Apple book store in May 2016.

It was a requirement that a more easily accessible version was created using the ePub3 standard. This documents the process.

Theoretical Considerations

iBooks Author (from Apple) is a wonderful piece of software for creating ebooks with lots of marvellous interactive features.

There are some very good reasons for using this software to add rich multimedia and interactivity to educational content.

- It's free. Apple provide this software as a download from their website and it runs perfectly well on most recent MACs under MacOS.
- It is (fairly) easy to learn, and, there are several books and online resources (some free ebooks as well) available to help learn.
- Proofing, transferring and distribution can all be done from within the software itself.
- There are lots of templates for the beginner to use, to easily get started.
- iBooks Author can be used by an author or editor to construct a prototype, that can then be further developed and enhanced.

The disadvantages to using iBooks Author

- The product that is exported from iBooks Author (.iBook) can only be viewed with an Apple device (MacOS or iOS)
- By default, the ebooks produced can only be sold through Apple's own iBookstore although they can be **freely** distributed by other means.
- The content cannot be exported in any form other than the raw text and PDF.
- Although the content can also be exported as PDF, this is not at print resolution.
- Although there is an ePub template available, this is only in the reflowable form.
- You cannot open an existing ePub file for editing.
- Using iBooks Author to create ebooks from existing print focussed digital assets is very difficult but, with some effort, possible (see the third section).

The main disadvantage is that there will be a large potential audience (particularly in the education sector) who will not be able to benefit from the iBook created with iBooks Author since they do not own MACs or iPads. This is a shame, because it is perfectly possible to create an ebook with very similar qualities that will perform well on 'non-Apple' devices, such as a PC under Microsoft Windows, some tablet devices *as well as* the MAC and iPad. This is known as the ePub format.

What is ePub?

The first version of the ePub standard was released by the International Digital Publishing Forum (IDPF) in 2007. The IDPF is a consortium of organisations and individuals. The standard is open and anyone can create software that will read and display the ePub.

The specification was originally in use as version 2 (ePub2) and in this release books could **only** be reflowable with text not formatted as is for print but more like a webpage; flowing to fit the size of the device or window.

In 2014 the ePub3 version was released and now includes the ability for content to be arranged as 'fixed' or precise layout. This new format also can include 'overlays' to deliver javascript, multimedia and animation.

In the latest version of the format ePub3, we have 2 available modes: the re-flowable and the fixed-layout forms. They are different in the amount of display control that they leave to the user/reader. Within a re-flowable ebook, the user may change the font size, the typeface and (on some devices), display features such as leading, alignment, justification and hyphenation. On the other hand, the fixed-layout form removes any such flexibility, and the page designer can arrange text and image on each page and be sure that this is how it will remain. The fixed-layout form is the *friend to the graphic designer*; the re-flowable form provides *more accessibility for the user*.

So, in summary, the ePub can either be used to deliver *re-flowable* or *fixed-layout* content.

For the fixed-format, the ePub that we create *must* be created to the ePub3 standard. Let's look at what an ePub file actually is.

HTML, XML and CSS

The ePub file is actually a package of files. Although an ebook is one file (example.ePub) it can be 'unpacked' by renaming as a zip (example.zip).

Each page in the ebook is an HTML file (actually **XHTML**). This is where the fixed-layout and re-flowable ebook diverge; in a re-flowable ebook, each chapter (or section) is a single HTML file. There are a number of other files inside the ePub package, but essentially, the ePub package is like a *website in a box* with an interactive table of

contents that provides navigation and a section of the content manifest that provides the reading order.

How does this differ from the Apple iBooks format?

Apple's iBook (sometimes referred to as 'Multi-touch book') is similar to the ePub and is also a package of files wrapped-up in a ZIP file. The similarity ends there though, because there are many proprietary features that cannot be constructed in any way other than through Apple's iBooks Author.

Can we make a more broadly available ePub version from the original Apple iBooks Author construct?

There is not a simple answer to this, but in the next section we will describe how we built the ePub version from the original iBook.

Practical Considerations

For the APGRD project it was decided that using iBooks Author offered the easiest way to overcome the main objectives; building an interactive information resource from diverse types of assets in the archive.

The benefits and limitations of this software were soon understood by the project team, and it was perfectly possible to use the tool as a means to communicate the various options as the items in the archive were digitised.

The Multi-touch iBook for Apple

After various iterations this version was completed and delivered to the Apple book store and users were able to download for their Apple devices.

The iBooks version did not include a 'scrolling' view to give users the ability to change font size.¹

Since it was always in the plan to build a more accessible version for use on non-Apple devices, a strategy was then developed to make this conversion from the final iBook built with iBooks Author.

From iBooks to fixed-layout ePub

As explained above, there are **no export** options in iBooks Author that can provide easy conversion to the ePub format.

¹ This would have meant keeping the text in a continuous stream (more on this in section 3).

The iBooks Author application creates 2 file types:

- the .iba or iBooks Author file; saved as the work is developed and can be loaded back to iBooks Author for further development
- the .ibook file is created when the ebook is exported from iBooks Author; this is the publishable file

This .ibooks file is considered to be the final (signed off) copy. The one that gets distributed. It is this file that needs to be used to extract the assets for the ePub version.

The .ibook file is a package (much like the ePub) and it is possible to extract the content from this package by unpacking the file.

Technical and toolset Requirements

Before we go through the steps in the workflow, here is a description of the set up and software requirements:

Software

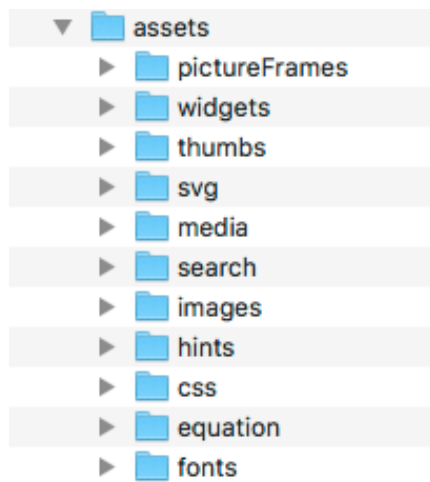
1. InDesign CC 2015. This was the best version for creating a fixed-layout ePub3.
2. PhotoShop. For some image editing.
3. iBooks Author
4. ImageOptim. This can reduced the size of the graphic images files with control over the quality.
5. Sound Normaliser. This takes sound from all media and sets at the the same level.
6. Miro Video Converter. To convert various formats; audio and video.
7. BBedit. Preferred editor for changes to ePub components.

Hardware

2. A MAC with Two Monitors. We need to have iBooks open at the same time as InDesign so we need a lot of screen 'real estate' to be able to see the final version of the iBook as well as the InDesign pages and the assets folder.

Workflow

1. Unpack the .ibook file by renaming as .zip and using an archive extraction tool.
2. Drive through the files to locate the assets folder. This contains the images folder and the media folder amongst many other things:



assets folder

3. The images folder contains all images that are needed to build a new version but also the thumbnail images and various icons and interface graphics that the iBooks version uses. We can delete all thumbnails and icons. We can tidy this folder up by removing all unwanted images.
4. The media folder contains all audio and video files. We will need to convert the audio to .mp3 files and the video to .mp4 so that InDesign can use these files.

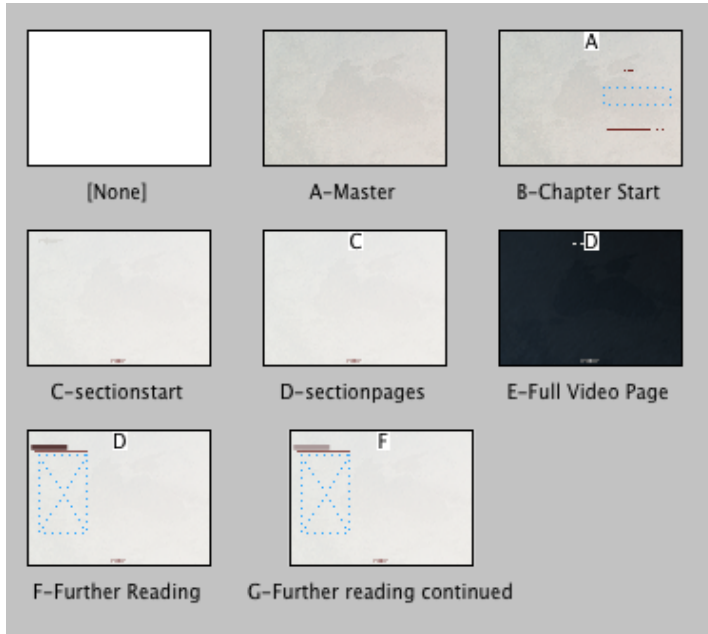
InDesign now needs to be opened and a template created. The InDesign page size is based on the format for landscape iPad. For this project, 1024 x 768 pixels was chosen although future projects might now target the 'retina' version at 2048 x 1536 pixels. Styles to equate with those in the iBooks Author version were created, and fonts added to the author system as required.

Body text changed for easier reading experience. Feedback from the users of the published iBooks Author version suggested that a sans-serif font would be better so Avenir Next was chosen since this already exists on Apple iOS systems and can be embedded in the ePub version.

Two Screens are better than one. The best option for this kind of work is a 2 screen display, so that InDesign can be on the main display and the iBook version on the other. This helped in matching each page content.

Text should flow. One problem with the iBooks Author (from a conversion perspective) is that the text on each page was placed in separated boxes; the headings in one, then 2 or more for the body text and a further set for the captions. It was the intention in capturing this text that all text (apart from the captions) would be in one single flowing text. It was also determined for consistency to set a spacing style for the headings so that the body text would always follow on at the same space. This spacing was a little bit random in the iBooks Author version.

Master Pages and grid. The master pages including page numbers and heading decoration (dagger!) were also set up for the template. There are several master page options. Grid lines were used to set up the consistent positioning of elements on the page.



An InDesign Book. The chapters were kept as separate files and combined in the InDesign book panel.

From iBooks Author to InDesign

The text

The only way to get the text across was to copy and paste; with the two-screen system described above, the text is copied within iBooks Author and then pasted into the text box on the equivalent page in InDesign.

Text enlargement

Where there was the need for an enlarged block of text (translation or alternative version) then these were created as separate objects on a hidden layer and then revealed on click over a button.

The glossary

It was not possible to get an equivalent feature to work in the ePUB version. I did experiment with various options, but popover text is not supported on most platforms so this was not pursued to a satisfactory conclusion.

The images

The images that decorate the page (backgrounds, daggers, underlines) were collected in a devoted folder. For the single information images, they needed to be located in the assets folder and then placed in the InDesign page. The size of the images was always larger, since in iBooks Author images can be made to enlarge. Once the image was placed and reduced down, then a second copy was placed on another layer set up for the purpose. This second copy was enlarged to fit the complete page and attached to an object style that delivered a very large semi-transparent outline (to cover the whole page). The first (smaller) copy of the image was then made into an interactive button scripted to reveal the larger version. *Note:* This is a somewhat simplified description of the process.

Image sequences.

iBooks Author has a 'Gallery' widget feature that allows a sequence of images to be constructed. These images are represented by thumbnails and can be enlarged individually and the sequence navigated full screen. There is no exact equivalent in InDesign, so the solution was to display the images larger on the page and construct a 'multi-state' object, with interaction on each image to move to the next in the sequence. Captions were added to the state for each image and displayed nearby.

Audio

The MP3 files were placed on the page and then hidden behind images or text blocks. Graphic buttons were created for play and pause and these were then linked to the MP3 on that page.

Video

The MP4 video was placed on the page and the poster image selected from one frame. Video can be played full screen as well as within the page. Many videos are delivered on a devoted page in the ebook, but there are some where 2 videos are on the same page; in this situation javascript had to be added to prevent the 2 videos being played at the same time.

Special Features

There are some special features in the iBooks Author version that were built with Apple Keynote (this a great way to add animated sequences into iBooks Author); there is no direct equivalent for this in the ePub version but sequences were constructed as multi-state objects and then navigation was provided by adding buttons to the page.

Table of Contents

The iBooks Author version uses a different structure for the table of contents with page thumbnails displayed at the bottom of the page as well as the drop-down menu on the MAC version. The ePub uses a simpler approach with a single drop-down menu or a

display of thumbnails for each page along the bottom on the MAC or over the whole screen on the iPad.

The final file size

The final edited version with embedded media and fonts was 439MB. By further compressing the images the file was reduced to 377MB. The iBooks Author version is 469MB.

The ePub on different platforms

The finished ePub validates to the ePub3 standard and is tested to work well with Adobe Digital Editions version 4.5 and above. It also works well with the Radium plugin for the Google Chrome browser. Theoretically, the ePub3 should work with Google Playbooks, but the finished file is too large (377MB) for Playbooks, which restricts to 50MB.

Some experiments were conducted to build a version without video, **but** this still did not get the file size down enough.

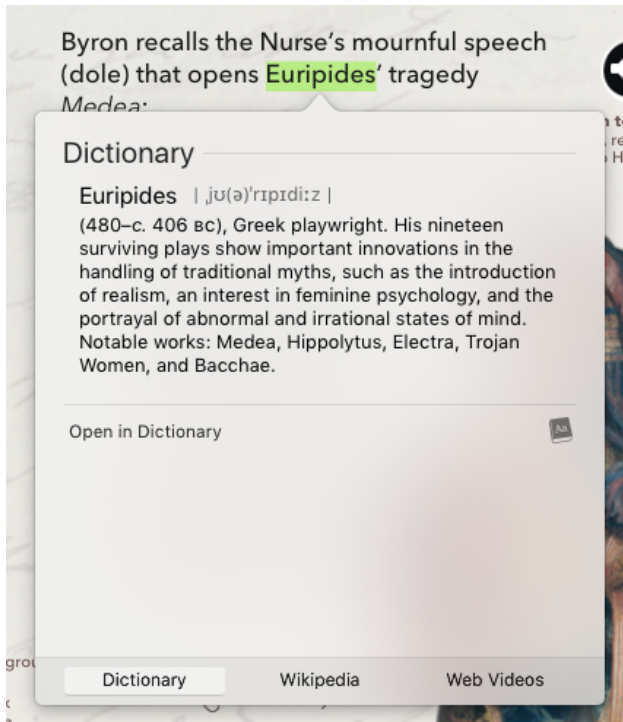
A version for the Kindle Fire was also attempted, but all of the javascript would have meant a complete rebuild.

Limitations and the future

Building an ePub version from the iBooks Author version was always going to be a struggle, since Apple do not provide easy access to the file format, and there are only limited export features. The description provided in the previous section shows that 'copy-paste' was the only way the work towards a near equivalent version. The ePub version does behave differently when it comes to the interactivity, but otherwise all of the content is the same, with some limitations.

The Limitations of the ePub3 version

The authors and editors (as predominantly academic writers) using iBooks Author found the Glossary feature very valuable as a way to explain certain words that may be unknown to the young reader. Getting the glossary to work in the ePub version was attempted, but eventually this option was removed from the ePub since the popups could not be made to go full screen. It is always possible to use the in-built feature of some eReader software, to give a popup on right-click over a word:



the in- built dictionary

The future

From InDesign to iBooks Author

I do believe that there is a better way to achieve both the iBooks Author version and the –more standards compliant– ePub3 version and that is to work in the opposite workflow direction and here are the steps:

1. Write the text using any preferred text system. It may be better, with multiple authors, to use an online versioning system like GitHub or BitBucket as a means to develop a system where texts can be 'signed-off' as approved for incorporation into the whole.
2. Simultaneous to this would be the digitising of images with metadata added to include the caption and copyright information.
3. Videos and audios will be sourced according to the notes made in the text above. These assets can be edited and held ready for delivery within the repository.
4. Once the text is approved it can then be placed into the template created in InDesign. Images can be placed, text-flowed and media components added and embedded.

This then forms the basis of the ePub3 version, although there will be more work needed to finalise this version. What we need now is to get this text (and images) into iBooks Author.

XML / IDML

InDesign has an underlying data format called IDML (InDesign Markup Language). This is effectively XML by another name.

iBooks Author can import IDML files into a chapter. This then forms the basis of the iBooks Author version. The important point here is that although there is still more work to do in both versions (adding gallery widgets, multi-state objects etc), the main text is available in both.

Accessibility could be better

The re-flowable ePub is a type that would be very difficult to use for this type of product, since images and text need to be kept in close proximity. This is a shame, because the re-flowable format does offer the user the ability to increase the size of the font for easier reading. This is not offered in the fixed-layout ePub. However, the iBooks Author toolset does offer a 'scrolling' view option that can be enabled *as long as the text is provided in a threaded stream*.

If the workflow method suggested above is used then the text will be threaded and the iBook can also have the scrollable view, selectable by the user and affording them the possibility to enlarge the text.

Chris Jennings @pageboy December 2016