



“Using the EPUB 3 electronic book standard to identify areas lacking support in current eBook reading devices and software”

Honours Final Report

By  Davies  
BSc (Hons) Computing (Web Systems Development)  
Matric No: S1033376

Project Supervisor: Katrin Hartmann  
2<sup>nd</sup> Marker: Richard Foley

**Abstract:** eBooks have revolutionized the world of publishing in recent years, with the market for electronic books expanding rapidly. EPUB files are web-enabled files zipped up in a container and read as a single file, which allows people to read whenever they want to, no matter where they are. The recent EPUB 3.0 standard introduces capabilities for enhanced interactivity and accessibility in eBooks, allowing for opportunities to revolutionise the way people read and consume content. However, despite the standard allowing support for this, a considerable amount of eReading devices and software still do not provide adequate support. This project explores the interactive capabilities of the EPUB 3 and sets out to discover the extent to which developing fully-featured interactive and accessible books is plausible with the current software that exists for reading the files. A fully standards-compliant EPUB 3 eBook was produced, consisting of a number of sections. Each of the aforementioned sections corresponded to a feature within the EPUB 3 specification. A range of software was then tested for compatibility with the standard in order to determine compatibility issues and areas in which support is lacking, using the developed file. It was discovered upon analysis of the results of this testing that, much as industry experts and critics have suggested, support in the current time for EPUB 3 in its entirety is very much lacking, especially in dedicated eReading devices. Although it is expected that greater uptake will occur in the very near future, unfortunately – 3 years after release of the standard – there is still a long way to go before full support exists for EPUB 3.  
[Word Count: 268]

“Except where explicitly stated all work in this document is my own”

**Signed:**

**Date:** \_\_\_\_\_

# TABLE OF CONTENTS

---

Table of Figures.....	5
1. Introduction .....	6
<b>1.1. Project Background</b> .....	6
1.1.1. The History of the eBook.....	6
1.1.2. The Three Main Formats .....	6
1.1.3. The Problems with EPUB.....	8
<b>1.2. Project Outline and Research Question</b> .....	12
1.2.1. Project Outline .....	12
1.2.2. Research Question .....	13
1.2.3. Objectives.....	13
<b>1.3. Report Structure</b> .....	15
1.3.1. Literature Review .....	15
1.3.2. Design.....	15
1.3.3. Implementation.....	15
1.3.4. Evaluation.....	15
1.3.5. Conclusion.....	16
Chapter Summary.....	17
2. Literature Review .....	18
<b>2.1. EPUB Features</b> .....	18
2.1.1. EPUB Navigation Document .....	19
2.1.2. Package Document.....	19
2.1.3. HTML.....	20
2.1.4. CSS.....	20
<b>2.2. eReading Devices</b> .....	20
2.2.1. Dedicated eReaders .....	20
2.2.2. eInk.....	21
2.2.3. Computing Devices .....	22
<b>2.3. eReading Software</b> .....	23
2.3.1. Calibre .....	24
2.3.2. iBooks.....	25
2.3.3. Radium .....	25
2.3.4. AZARDI .....	25
<b>2.4. Literature Conclusion</b> .....	25

Chapter Summary.....	27
3. Design .....	28
<b>3.1. Planning</b> .....	28
<b>3.2. The Base Document</b> .....	29
<b>3.3. New Features in EPUB 3</b> .....	29
3.3.1. HTML5.....	30
3.3.2. EPUB Navigation Documents .....	30
3.3.3. MathML.....	30
3.3.4. Global Language Support .....	30
3.3.5. Scripting Support.....	31
3.3.6. Embedded Fonts .....	31
3.3.7. Media Overlays .....	31
Chapter Summary.....	32
4. Implementation .....	33
<b>4.1. EPUB Open Container Format</b> .....	33
4.1.1. Root Directory .....	33
4.1.2. META-INF Directory.....	34
4.1.3. EPUB Directory.....	35
<b>4.2. Issues</b> .....	37
Chapter Summary.....	39
5. Evaluation .....	40
<b>5.1. Testing</b> .....	40
5.1.1. Devices .....	40
5.1.2. Test Areas .....	41
5.1.3. AZARDI .....	42
5.1.4. iBooks.....	42
5.1.5. Kobo Touch.....	43
5.1.6. Radium .....	44
<b>5.2. Results</b> .....	44
5.2.1. E Ink.....	44
5.2.2. Computing Devices .....	46
Chapter Summary.....	47
6. Conclusion .....	48
<b>6.1. The Development</b> .....	48

<b>6.2. The Findings</b> .....	48
Glossary .....	50
References .....	51
Appendix 1: Code Listings.....	54
EPUB File.....	54
Root Directory .....	54
META-INF Directory .....	54
EPUB Directory .....	54
Appendix 2: Testing .....	56
Accessibility Checklist (IDPF) .....	56

## TABLE OF FIGURES

---

<a href="#">Figure 1: Reflowable Text Example</a> .....	7
<a href="#">Figure 2: Fixed Layout Example</a> .....	8
<a href="#">Figure 3: Bad PDF to EPUB Conversion Example</a> .....	9
<a href="#">Figure 4: Typical Root Directory</a> .....	30
<a href="#">Figure 5: Demonstration of Relative Path in <code>container.xml</code></a> .....	31
<a href="#">Figure 6: Device Display Information Table</a> .....	37
<a href="#">Figure 7: Test Items Table</a> .....	38

# 1. INTRODUCTION

---

This chapter introduces the project background, giving insights into the area in which the project is based and providing foundation knowledge relevant to the project. The project background is followed by the project outline and primary research question, which provide an introduction to the project and development itself, and introduce the main question to which the project aims to discover an answer. The last section of this chapter outlines the structure of the rest of the report.

## 1.1. PROJECT BACKGROUND

This section aims to give the reader a picture of the background of electronic books, covering the history of the eBook and introducing the EPUB format which will be central to this project. Also introduced are the recent developments and issues in the digital publishing industry which brought about the research question.

### 1.1.1. The History of the eBook

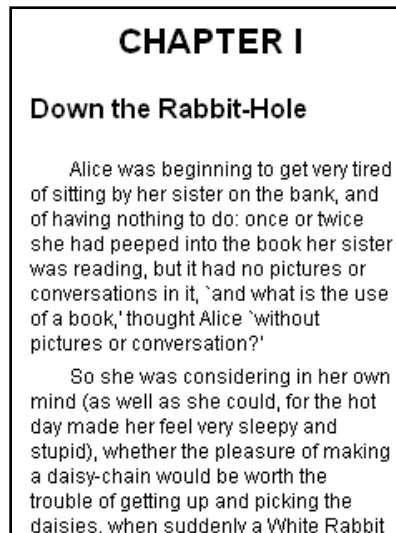
Electronic books, more commonly known as eBooks, have existed in one form or another for many years, with Project Gutenberg becoming the first digital library in 1971 (Lebert, 2008). As the internet spread throughout the 1990s, sharing of books digitally became increasingly easy and digital libraries expanded exponentially. These first digital books were simply text files in ASCII format, in order to allow for compatibility across devices. Over the years, a number of eBook formats were created, but support was generally limited to specific reading software and/or devices, and digital books weren't generally published in more than one of these formats (Lebert, 2009a). In the late 1990s, the Open eBook format was developed by the Open eBook Forum, based on subsets of the popular mark-up languages, XML and XHTML, with the added option to include images and design features specified in CSS stylesheets. From this format, two of the most popular and widely-supported (Lebert 2009b) formats today – EPUB and mobi – were created.

### 1.1.2. The Three Main Formats

Today, the majority of the eBook market is divided between the three most widely-supported formats: EPUB, Mobipocket and PDF (Coker, 2010). These three main formats will be introduced in the aforementioned order in the following three sections.

## ***EPUB***

EPUB is an open format which consists of a number of dynamic, web-enabled files, using defined subsets of XML and XHTML much like its predecessor – the Open eBook format. Because it consists of features commonly found in websites – such as HTML, CSS and image files – zipped up in a container, some have described it as “a website in a box” (Garrish & Gylling, 2013). Content of EPUB files is reflowable, meaning that the presentation of content can be altered in order to provide best fit on the display media. This allows for text, images and similar content to be resized at will. An example of reflowable text can be seen in Fig. 1, on the following page. This can be compared to the alternative fixed layout shown in Fig. 2, also found on the following page.



*Figure 1: Reflowable Text Example*

## ***Mobipocket***

The mobipocket format (referred to generally as just ‘mobi’), although similar in capability to EPUB, is not an open format. It, like the EPUB format, came about as an implementation of the IDPF’s Open eBook format. Amazon purchased Mobipocket, the company behind the mobi format in 2005, and the format became almost synonymous with Amazon’s range of Kindle devices. As of 2011, the mobi format was officially abandoned in favour of Amazon’s new KF8 format, which boasts greater interactivity in order to compete with the EPUB 3 standard. (Hoffelder, 2011) Despite the lack of continued support for the mobi format, it still retains its position as one of the most popular eBook formats existing at the current time.

## PDF

Adobe's PDF format is also popular for electronic books, with a lot of textbooks being offered digitally in PDF format due to the fact that it replicates print extremely well on larger screens. This is due to it being a fixed-layout format, with content existing in a fixed size in a static document. However, a lot of the time this does not transfer well onto the screens of dedicated eReading devices, as they tend to be a lot smaller than the page size, making words on the page small and difficult to read, often requiring the reader to zoom in. Depicted on the following page (Fig. 2), the display port (the boxed area on the left) is smaller than the fixed layout content (the greyed out area), and in order to view the rest of the content, the user is required to scroll.

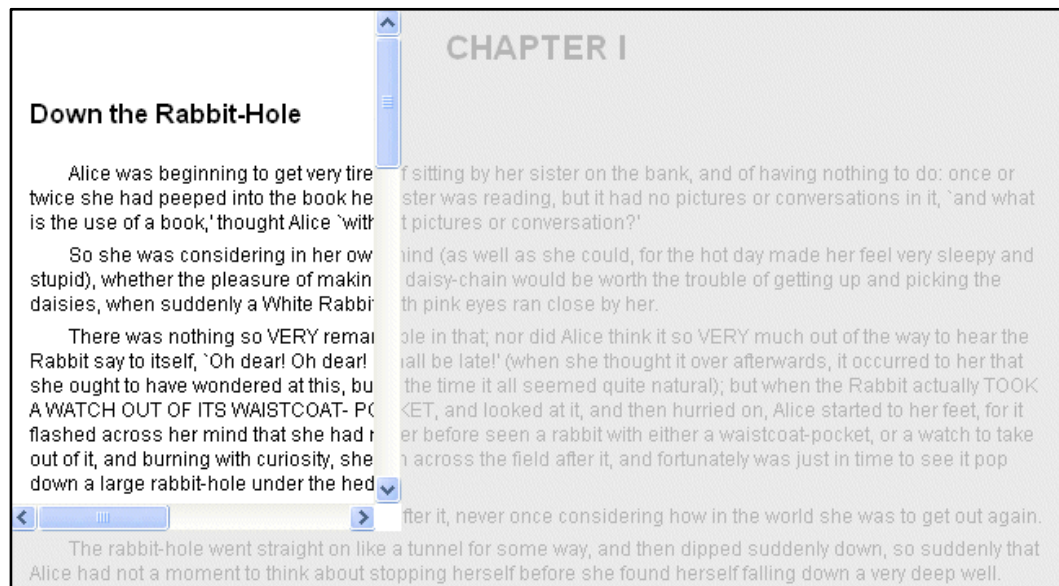


Figure 2: Fixed Layout Example

### 1.1.3. The Problems with EPUB

Despite the increasing popularity of eBooks in the format, there are a number of problems which commonly occur with EPUB files. The first thing to consider is the fact that EPUB files can be created by any person with a plain text editor, such as Notepad. While there are standards as defined by the International Digital Publishing Foundation (IDPF), it is not strictly regulated. The fact that they are EPUB files are built with web technologies allows for as much bad design and bad practice in their creation as in any website built with those same technologies. As well as this, the method of creation can also create issues. There are a number of tools which provide the capability to convert books into the EPUB format from



almost every other format imaginable. These issues will be covered in this section.

### ***Conversion Issues***

eBooks are available from many different sources in a wide variety of formats, and software exists to convert them into EPUB format with just the click of a button. This creates some of the most common, most complained-about issues with EPUB files. Conversion software often creates an end product with formatting errors littered throughout (Wikert, 2013a). These formatting errors can crop up as a result of various factors within conversion, outlined in the following example. PDF files are created for print, and as such they do not contain the information which assists software conversion of the file into a different format. This can result in special characters displaying incorrectly. Additionally, headers, footers, page numbers and the likes often end up as a part of the body text where they should not occur. It can also be very difficult for software to determine where paragraphs begin and end within PDF documents, relying on a more visual approach using features such as line breaks and indentation. (Gross, 2012) An example of such errors existing in eBooks after conversion from PDF can be seen in the Fig. 3, with page number (“2”) and heading (“Isaac Asimov”) being inserted into the flow of text, as well as each line being interpreted as a new paragraph and spaces being inserted wrongly in the middle of words.

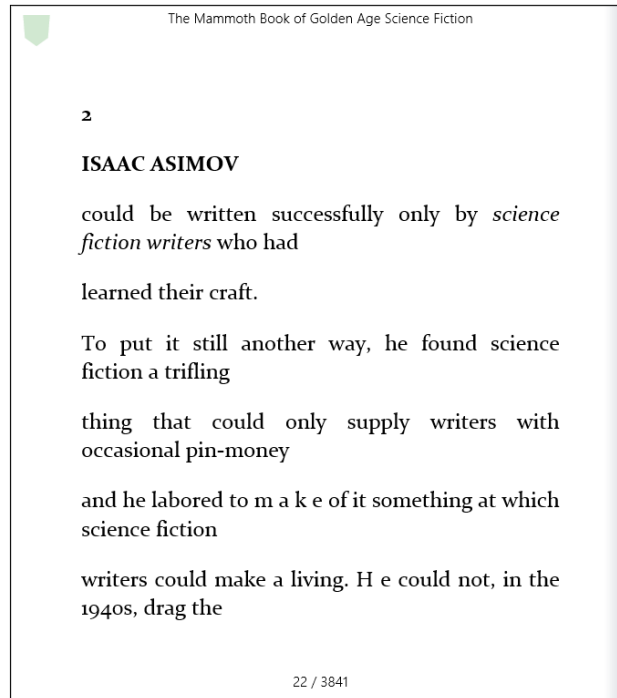


Figure 3: Bad PDF to EPUB Conversion Example

This is an issue for publishers and readers alike. Where publishers are concerned, having to add extra hours onto the development time of eBooks to account for the process of manual editing and cleaning of code is a costly addition to an already-costly process – both in terms of time, money and resources. For readers, the issue comes down to irritation at the quality of work they have purchased. Bad formatting can put readers off purchasing material from the same distributor, publishing house, or other involved party in the future, due to associating sub-par quality with a name or brand. It is also a concern for publishers. Publishers are also businesspeople, and profit is still an important part of their business, although it often goes hand-in-hand with satisfying the needs of the client. For the most part, they understand that satisfying the needs of readers results in greater sales and better business.

### ***Formatting Issues***

Formatting problems also occur in other situations. A study conducted by Chang & Yang (2013) highlighted issues regarding support for Asian script in the creation of eBooks in the EPUB format when using Chinese characters and vertical form. This is an issue which the EPUB 3 standard aims to address, but the support of reading software is also vital in order for full foreign language support to be achieved successfully.

Tools exist for validation of EPUB files, such as the International Digital Publishing Forum (IDPF)'s EPUBCheck, which has recently received an update by Barnes & Noble to further improve the content that publishers produce (Wireless News, 2013).

### ***Interactivity***

There has been a shift in recent times, with increased focus on Web 2.0 technologies (and even more recently Web 3.0) and interactivity. The introduction of HTML5 and CSS3 revolutionised the way websites are built and since then there has been greater focus on moving towards greater interactivity in eBooks. The fact that EPUB files consist of web-enabled files similar to those used in the construction of websites made this a logical step. The introduction of Javascript functionality into eBooks adds a great many possibilities to eBooks, giving authors the opportunity to insert quizzes to test learning within educational books (Tan, 2013), and to give users greater input and a more immersive experience while enjoying their digital media (Kasdorf, 2013). This project will focus on the EPUB format because of the recent EPUB 3 standard which was introduced by the IDPF (2011).

### ***Accessibility***

Although the hype surrounding the EPUB 3 format primarily focuses on the greater interactivity in files – around which questions have been raised (Habash, 2012) – there are a number of changes which make the format better all round for readers everywhere, and not just in eBooks with vastly interactive content. Accessibility is a huge issue, especially where reading and the web are concerned, and the EPUB 3 format introduces features which allow eBooks to be accessible to all people where they were previously lacking. Features such as text-to-speech make it easier for the print disabled to enjoy digital books. The introduction of more meta-data and clearer mark-up is also beneficial to the general user, as it is better structured (Garrish, 2012) (Enis, 2013).

### ***Industry Support***

Uptake of EPUB 3 has been slow, and it has been commented on by Hoffelder (2013) and others involved in the digital publishing industry that there is a catch-22 type situation within the industry which perpetuates this. It takes considerable investment for publishers to develop their content in the EPUB 3

format, and they are – quite understandably – reluctant to make the change until EPUB 3 features are more widely supported by eReading software, in order to make the change viable. Changes in software to support the EPUB 3 format are also slow to happen, however, because publishers are slow on the uptake of the format. Therefore, each part of the cycle is holding the other back in terms of progress for the development of the eBook industry as a whole. To combat this, the AAP have launched an initiative to try to speed up the uptake of EPUB 3 (Targeted News Service, 2013). This involves bringing together people from positions within all areas of the industry - such as publishers, retailers, developers and service providers – in an attempt to use the different perspectives to highlight priorities and best practices. This should see files in the EPUB 3 format become more widespread from 2014.

In summary, there has been an undeniable push for increased interactivity and accessibility in all sorts of digital publications in recent times. This trend in the digital publishing industry – coupled with the concern voiced by many that these features are simply not supported by the reading software which currently exists – leads to the research question of this project.

## **1.2. PROJECT OUTLINE AND RESEARCH QUESTION**

This section identifies the aim of the project and introduces the research question to which the undertaking of the project intends to provide an answer. It also presents both the primary and secondary research objectives and summarises their importance to the project as a whole.

### **1.2.1. Project Outline**

The aim of this project is to develop a fully standards-compliant EPUB 3 eBook which fully showcases the newly-introduced features of the standard and use it as a benchmark to test the capabilities of the main eReading applications and devices currently available to the general public. This project also aims to discover whether or not the newest features of the latest standard work in practice, as opposed to solely in theory.

The project will be of a Develop & Test type, with an EPUB 3 file as the main development. The proposed testing will be comparative in nature, comparing the capabilities of existing software to each other with regards to the developed file in order to come to a conclusion which can be utilized to gain greater insight into the project area and provide a potential answer to the research question. The Develop

& Test methodology is the most appropriate for this project as it involves comparison and assessment of the capabilities of technology.

### **1.2.2. Research Question**

Do current eReading applications provide enough support for fully standards-compliant, interactive, accessible eBooks developed using the EPUB 3 standard to make their development a viable practice for publishers?

### **1.2.3. Objectives**

This section will detail the objectives to be met through the course of the project, both through the review of literature and through development itself. It will also provide justifications for each.

#### **1.2.3.1. Literature Review Objectives**

- *Identify the new features supported in the EPUB 3 standard and how they can be appropriately applied as part of an eBook.*

The additional features which are new as part of version 3.0 of the EPUB standard are those which are likely to be currently unsupported by eReading software and devices, therefore it is important to identify them in particular. Due to their being new to the specification, it is also important to be able to understand how to properly develop these features.

- *Identify the constraints of devices used for eReading and the impact that they may have on support for newly-supported features of the EPUB format.*

Device constraints could have a significant impact on features which can or cannot be supported. Reviewing literature focusing on these issues will help to identify areas where support may lack before the development stage, creating hypotheses which can be tested.

- *Identify a range of eReading software over a range of devices.*

Testing is not possible on every piece of eReading software or each device that currently exists on the market. It is necessary to review the possible

options and make an informed selection of a range to use in the testing stage.

#### **1.2.3.2. Development Objectives**

- *Identify and become proficient in use of a development environment for electronic books in the EPUB format.*

Although becoming accustomed to a development environment itself is not an actual part of development, it is necessary to ensure that the latter part of the development time is focused on developing rather than still getting to know the environment.

- *Design an EPUB 3 eBook closely adhering to the IDPF specified standard.*

The design aspect of the eBook is important, as good design will prevent having to rework and rethink entire sections of the development at a later stage.

- *Develop an eBook according to the design and validate the file using the IDPF's EPUBCheck validation tool.*

This stage is where the coding of the files which make up the EPUB file is done, and is the core stage of development. Ensuring that the file is valid to the IDPF's standard is also integral to the project.

- *Test the eBook using the IDPF's Radium eReading software.*

Testing the eBook in a piece of software which claims the most support for the EPUB 3 standard allows the file to be read in optimum working conditions before then being used as a benchmark.

- *Test the functionality of each section of the developed eBook with all of the previously identified eReading software and/or devices.*

This is the stage in the development which will provide the results from which a conclusion can be drawn and used to gather an answer to the

research question for the benefit of industry professionals and the reading population alike.

### **1.3. REPORT STRUCTURE**

This section gives the reader a brief outline of the chapters which make up the rest of the report, along with a summary of the contents to be found within each chapter.

#### **1.3.1. Literature Review**

The second chapter consists of the literature review which looks in more depth at areas which have been touched on in the Introduction chapter, within the Project Background section. The literature review is driven by objectives identified in the previous section of this chapter in order to focus in on areas of key importance which provide a foundation upon which the development stage will build. These areas of focus are: the features and make-up of the EPUB format itself, in its current third version; devices specifically created for the purpose of reading eBooks and their constraints; and software for reading eBooks on computing devices such as tablets, personal computers, and smartphones.

#### **1.3.2. Design**

The third chapter focuses on the design process carried out in the run-up to carrying out the actual development. The design of the EPUB file to be used in the later stages of development is one of the primary research objectives, and provides an introduction to the following Implementation chapter while documenting the completion of the associated objective.

#### **1.3.3. Implementation**

This chapter provides in-depth information about the development of the EPUB file which is integral to the project. It follows the development in a linear fashion and gives full justification for the decisions made in the development process.

#### **1.3.4. Evaluation**

The evaluation chapter is centred on the testing of the developed file and the insights which can be gained from the acquired results. This chapter is of prime

importance to the project as a whole and is the point at which answers are gained to questions which have formed the basis of the project.

### **1.3.5. Conclusion**

This final chapter essentially summarises the entire project in two sections. The first section discusses the main points of the development, as is discussed across the design and implementation chapters. The second section reviews the findings which were gathered as a result of the project being undertaken, and aims to concisely sum up the points made throughout the rest of the report.



## CHAPTER SUMMARY

---

The Introduction chapter summed up the surprisingly long history of the eBook, and introduced the three main formats which make up the majority of the market in the digital publishing world today. It went on to introduce some common problems that are found within eBooks, both during and after their development. This led on to the outline of the project, the objectives to be satisfied in the course of the project and the Primary Research Question:

*Do current eReading applications provide enough support for fully standards-compliant, interactive, accessible eBooks developed using the EPUB 3 standard to make their development a viable practice for publishers?*

The Introduction chapter also outlined the structure of the rest of the report, summarising the following sections which are, in order: Literature Review, Design, Implementation, Evaluation and Conclusion.

## 2. LITERATURE REVIEW

---

This review of literature aims to discuss areas identified as being of significant importance to the subject area, listed as objectives. Knowledge gained from respected sources forms a vital part of the project and the insights of others can be infinitely useful during the course of the project. The objectives of this literature review are as follows:

*Objective 1: Identify the new features supported in the EPUB 3 standard and understand how they can be appropriately applied as part of an eBook.*

Understanding the EPUB format, how it is formed and how each file within the eBook relates to the others is a vital first step in the project. A solid understanding of the way an EPUB file works will be highly beneficial in the development stages, and will help to cut down time required to acquire necessary knowledge during the development process. Reviewing the thoughts of others on the make-up of these files will also be highly beneficial in the process of designing and developing.

*Objective 2: Identify the constraints of devices used for reading eBooks and the impact that may have on support for newly-introduced features of the EPUB 3 specification.*

Device constraints are highly likely to be linked to the potential lack of support for the EPUB 3 standard which is the focus of this project. As such, this objective is of great importance to the project as a whole, and reviewing material focusing on these reported constraints and incompatibilities will allow for hypotheses to be made about the level of support for various EPUB features.

*Objective 3: Identify a range of eReading software over a range of devices.*

There is a wide range of eReading software available for computing devices, with more appearing as new technologies gain popularity. On iOS alone, there are more than 10 different eReading applications to choose from, and not all of the software available can realistically be tested within the scope of this project. Because of this, it is essential to identify a range to use within the testing stage.

### 2.1. EPUB FEATURES

The EPUB 3 specification supports a wider range of interactive features than the previous version (Scardilli, 2013), mirroring changes in the world of the web in general with the push for more interactive and immersive user experiences. These new additions include the following:

- Support for audio and video embedded in HTML5 <audio> and <video> elements

- Ability to include media overlays to provide audio narration in sync with text
- Support for interactivity with the use of JavaScript, enabling insertion of features such as quizzes and puzzles.
- Text-to-speech facilities
- Support for global languages, including right-to-left reading and vertical writing, as well as other typographical features with the use of CSS3
- Ability to embed fonts
- Improved meta-data and semantic tagging, providing greater accessibility
- New navigation documents making use of HTML5 to provide greater flexibility in the navigation system

Many eBooks are essentially just text - an electronic representation of novels, split into sections and subsections, such as parts and chapters. The market for these eBooks is voluminous (Chao et al., 2013), and EPUB 2 worked reasonably well for these sorts of eBooks. However, these are not the only types of publication in the market by any means, and publishers are constantly coming up with new ways of sharing content with consumers to provide them with a more interactive, immersive experience.

The information which follows is taken from the EPUB 3 Overview (IPDF, 2011).

#### **2.1.1. EPUB Navigation Document**

In the previous version of the EPUB specification, it was necessary to include an `.ncx` file to define navigation within the eBook. This has been superseded by a new navigation document, called the EPUB Navigation Document.

Being created as an HTML file with additional limitations, the EPUB Navigation Document is both human- and machine-readable. It enables the reading application to provide the user with a table of contents when requested and allows for navigation to a specific point in the eBook. It uses the `<nav>` navigation element with `<a>` links and `<li>` list items to mark navigation in much the same way as website navigation can be created in development of a website built using HTML.

#### **2.1.2. Package Document**

Each EPUB file includes a Package Document, which has the file extension `.opf`. This file contains metadata about the publication, lists the files

that make up the overall EPUB file and their types, and defines the reading order of the files.

Having a Package Document to enumerate all of the files which make up the EPUB file allows metadata to be applied to each of the constituent files in one place, instead of requiring definition in each file individually.

### **2.1.3. HTML**

HTML files make up the majority of EPUB files, containing the content of the book, marked up for display by the eReading software in the same way an HTML website is marked up for display by a web browser. Typically each section of a book exists in its own HTML file – a chapter per HTML document, for instance. These are numbered and the Navigation Document and Package Document use the information contained within them to determine which order the files should be displayed in.

### **2.1.4. CSS**

CSS documents dictate any style rules which should be followed within the eBook, similarly to the way in which they do in websites developed with HTML and CSS. CSS2.01 is fully supported, but only certain rules of CSS3 – those regarding text-to-speech and fonts – are supported in the EPUB format in its current version. There are also a number of rules specific to EPUB files, regarding word breaks, hyphenation and other text formatting rules.

## **2.2. EREADING DEVICES**

There are a wide range of devices used for reading eBooks on the market. Some of these are dedicated to reading eBooks, and others are general purpose but with applications to facilitate reading of eBooks. This section aims to cover both types of devices and areas pertaining to them.

### **2.2.1. Dedicated eReaders**

A huge market exists for eReading devices, which are portable and exist primarily to facilitate the reading of eBooks and newspapers. The market is dominated by devices from Amazon, Kobo and Barnes & Noble, although there are more than 40 dedicated devices on the market and/or being currently supported at this

present time. These devices boast either an eInk or LCD display. The eInk displays make up the majority of the market and have some advantages over the LCD alternative, which is discussed in greater detail in the next section.

As previously discussed in the project background, over the years many different file formats for eBooks have been created. All eReading devices have a range of file formats which they support, although depending on the company responsible for the device, they will have different ‘protected’ file formats which are used for books from that company’s bookstore. (Griffey, 2010). This causes fragmentation of the market, with Amazon being a particularly notable case with their refusal to support the widespread EPUB format in favour of their own format. It should be noted that all eReaders on the market currently – with the exception of Amazon’s Kindle range, one device by Samsung and one device by Sony – offer support for the EPUB format. The support for the Mobipocket format, bought out by Amazon in 2005 (Philadelphia Business Journal, 2005), is nowhere near as widely supported.

Companies have their own formats as a form of DRM (Digital Rights Management) which exists in an attempt to prevent breaches of copyright law. However, this can be an issue for users of a device who purchase a book from Amazon, for example, and then decide further down the line that they would prefer a model of eReader from a different company. There is then no way for the user to transfer those purchases onto that different device (Griffey, 2010).

### **2.2.2. eInk**

eInk (electrophoretic ink) is a technology best-known for its use in dedicated eReading devices, although it is used in a wider range of portable devices because of its properties. The process involves printing liquid containing white particles and dark dye onto a surface, which is connected to circuitry. The particles are then manipulated by electric fields in order to display an image on the surface. (Falk, 2000). Power is only consumed when the particles need to be manipulated, and therefore eInk devices tend to have far longer battery lives than other devices, boasting anywhere up to a month of use in a single charge. The displays are classed as ‘reflective displays’, meaning that instead of being backlit as LCD displays are, ambient light is reflected back at the eyes, much as it is with paper. This means that users can read in various lighting levels, and enjoy reading on a device without glare on the screen. In 2012 Barnes & Noble incorporated a backlight into their eInk device - the Nook with GlowLight (Wikert, 2012a), to allow users to read in dark environments.

One of the biggest reported problems with eBooks is eye fatigue after prolonged periods of reading (Gregory, 2008), and eInk displays are reported to reduce the strain on eyes, compared to the screens of computers (Garsia & Savova, 2012).

Despite the obvious advantages of eInk eReaders in some cases, the latest display, E Ink Carta (E Ink, 2013) is only capable of displaying 16 shades of grey, which is the same as reported as the most advanced technology allowed in 2010 (Griffey, 2010). The graphical limitations of this type of display in comparison to the LCD alternative are most definitely an issue with regards to support for EPUB 3.

A recent blog post by publisher and industry critic, Joe Wikert (2013b), made an observation about the way in which developers seem fixated with emulating print, rather than innovating and giving users a new way to read. This could be applied to eInk devices. For displaying text similarly to the way it would be displayed in most paper books - with the exception of glossy-paged image-heavy publications - eInk does the job well, and the revenue from such products is considerable. These sorts of devices are not nearly so suitable, at this point in time, for interactivity at the same level as would be possible on a device such as a tablet, smartphone or personal computer.

This can be seen by looking at the sorts of interactive applications currently available on eInk devices. Games for Kindle eInk eReaders are sold by Amazon through the Kindle Store, and are commonly games which require only basic interaction, such as card games, simple puzzles and quizzes. This is because the refresh rate of the screens on eInk devices tends to be much slower than a tablet computer, and because of this, rendering videos or graphics is not particularly viable (Griffey, 2010). In Wikert's (2012b) article about the development of colour eInk technology, he also highlights the issue of the refresh rate rendering slowly.

### **2.2.3. Computing Devices**

Dedicated eReading devices make up a considerable portion of the market, but the EPUB 3 standard embraces a shift away from these devices and towards increased readership using devices such as tablet computers and smartphones (Scardilli, 2013). These computing devices offer more in-depth, interactive features, the ability to display stunning graphics and play video and audio content, as well as easily access hyperlinks within documents to other location. They have processing power which allows developers the freedom to innovate and create a reading experience unlike anything that has been possible in previous times. To a certain degree, greater interactivity in eBooks delivered to tablets and desktop

applications has already started to take off, especially in the world of children's storybooks, where interactivity is an important part of the learning experience. There have also been numerous interactive eBooks released in the iBooks store as promotional items to accompany film releases. These books incorporate images, audio and video to create the feel of an enhanced version of a glossy visual encyclopedia.

While the trend towards using more powerful, varied computing devices to read increasingly advanced eBooks allows a greater amount of creative freedom for the author and/or developer, it also complicates things to a greater degree than ever before. While aspect ratios for the displays of most dedicated e-reading devices tend to be uniform, this varies a great deal more across other computing devices on which people may choose to read. The immense variety makes it difficult to cater for each and every device in certain areas of development where the display's aspect ratio makes a difference, such as cover images, full page images and fixed-page layouts. Fixed-page layout eBooks have been adopted more in recent times for specific developments, such as interactive books to be sold specifically in the iBooks store to owners of iPads. It is in the iBooks store also where the greater adoption of EPUB 3 interactive features can be seen to be effective.

### **2.3. EREADING SOFTWARE**

There is a wide range of eReading software available for the computing devices mentioned in the previous section, and these offer various different features. In this section, there will be a summary of these different devices. As this paper deals with the EPUB format and Amazon's lack of support for the EPUB format has been documented in previous sections, the Kindle app for desktop, tablets and smartphones will be disregarded in this project, despite its popularity. A major point to be considered when selecting appropriate software to test is the fact that many of the popular pieces of software for eReading do not in fact claim to be able to support the EPUB 3 standard. In this case, testing each of these pieces of software for compatibility with the EPUB 3 standard would be a redundant pursuit, as the developers of the software never intended for that piece of software to support a file from such a specification. Therefore, the following software listed all claim to support EPUB 3 documents in some capacity.

### 2.3.1. Calibre

Calibre is not exactly a piece of eReading software, as such. It is an eBook library organiser, file converter and eBook reader all in one for desktop. Versions exist for Windows, Mac and Linux, with a portable version also available for removable storage. Its features make it one of the most useful tools for anyone who maintains an eBook library, boasting the following:

- The ability to manage an entire eBook library, with advanced sorting and search features, and the ability to edit all meta-data through a simple interface.
- Conversion from 24 different input file formats into 17 output formats, with all of the most widely supported formats included in both input and output lists. It should be noted that Calibre allows conversion from PDF but strongly encourages users not to convert from PDF into another format unless absolutely necessary (Calibre, 2014), due to numerous issues, including, but not limited to:
  - PDF is a fixed format, as opposed to the ‘reflowable’ formats, such as mobi and EPUB. Because of this, the conversion software struggles to identify beginnings and ends of paragraphs
  - Headers and footers on each page are added to the body of the text, so the author’s name and page number would crop up within the narrative of a novel, for example.
  - Tables, links, vector images and special characters fail to display properly.
- Support for the syncing of DRM-free eBooks to eReading devices
- The ability to take RSS feeds and convert them into eBook formats for reading on devices.
- An eBook viewer with the ability to alter the look and feel of the book using CSS and embedded fonts.
- A content server to enable access to an eBook library from anywhere.

As well as being able to boast all of these features, Calibre has the ability to display eBooks in a wide variety of formats. While it has not been claimed by the developer that there is any form of EPUB 3 support, it is compatible with HTML5 features due to its webkit use. Therefore, it is a point of interest to discover how much of the EPUB 3 standard is supported thanks to this, and how much is still lacking.



### **2.3.2. iBooks**

Apple's eBook reading software was available only on iOS until recently, when it was released as a free desktop app as part of OS X Mavericks. iBooks is one of the few pieces of software which can boast support for EPUB 3 features (even at the current point in time – almost 3 years since the EPUB 3 specification was released), although complete support is still a vision for the future. The fact that Apple have committed to the support of the EPUB 3 format so soon in its life is reassuring, and use of the interactive features has become more prevalent in its iBooks store in recent times (Apple, 2014).

Despite Apple's commitment to the support of EPUB 3, development of interactive eBooks to be sold in the iBooks store use a version of the EPUB 3 standard that Apple have altered for their own requirements.

### **2.3.3. Radium**

Radium is currently being developed by the IDPF, and it currently exists in the form of a browser extension for the popular Google Chrome web browser. Because it is developed by the IDPF, who are responsible for the EPUB 3 format specification, it supports the new features in the third version of the EPUB standard better than any other software currently available to the public. It, like the EPUB format, is open-source. This is hugely beneficial for the uptake of EPUB 3, as developers will be able to take the basic working model and create the next generation of eReading software.

### **2.3.4. AZARDI**

AZARDI is an EPUB 3 compliant eBook reading program, and boasts both desktop and browser-based options. Both desktop and web applications are free from the official website, with a desktop version available for Windows, Mac and Linux operating systems.

## **2.4. LITERATURE CONCLUSION**

The world of eBooks and eReading is vast, even when only EPUB files are taken into account. Massive libraries of books are accessible on just about every mainstream computing device, and the market share for eBooks are still growing. The new features of the EPUB format are important for the world of digital publishing, and not only

because they allow audio, video and interactivity. They also are an improved version of the last release with a lot of consideration towards the importance of meta-data and semantics, giving meaning to text, and putting greater emphasis on accessibility for all people, regardless of age, race, gender and disabilities. While interactivity could revolutionise the world of publishing, so can bringing the joy of books to more people.

After reviewing literature in the subject area, it can be hypothesised that dedicated eReading devices with eInk displays are not yet fully capable of supporting the more interactive features of the EPUB 3 specification. The refresh rates and graphical capabilities allow for crisp display of text at high contrast, but little beyond that. In order to provide a set of results from a device with an eInk display, a recent model of the Kobo Touch by Kobo Inc. shall be used in the testing stage of this project.

## CHAPTER SUMMARY

---

The Literature Review chapter was an extensive review of literature in the project area in order to satisfy the secondary research objectives. This involved a summary of the features of the EPUB format, as written in the IDPF's specification, going into detail about the construction of the average eBook in the aforementioned format. Following on from a discussion of EPUB's main features was an analysis of the different types of devices which exist for the purpose of eReading, covering dedicated eInk reading devices, smartphones, tablet computers and personal computers. After discussion of devices used for reading, was a section investigating a few main pieces of eReading software which boast support for the EPUB 3 standard.

## 3. DESIGN

---

This chapter focuses on the design of the EPUB document to be created as the development part of this Develop & Test project. The first section will look at considerations made when planning the development, before any development could take place. The second section will discuss what has been termed the ‘base document’, which is essentially the main skeleton of a well-formed and valid EPUB 3 document before any real content has been added to it.

### 3.1. PLANNING

Before the beginning of the development, it was essential to make a number of decisions about the form which the development would take. As the EPUB specification is written, at the current point in time it is recommended to create EPUB 3 documents with back-compatibility built in for readers which still only support the previous standard. It would make sense therefore that in the development of a commercial eBook, full compatibility with the previous EPUB standard should be strived for, in order to ensure that the eBook is accessible to as many people as possible over as wide a range of software and devices as possible. For this project, however, the aim is not financial gain, or to create an eBook which displays well in as many pieces of software as possible – it is to develop to a standard in order to highlight flaws and areas of incompatibility in currently existing software.

Within the EPUB 3 specification, there are two types of features. They are those which are *required* to be supported by EPUB 3-compliant reading software, and those which are classed as *optional*. The amount of features within the specification is vast, and to attempt to create a file which tests each and every one of these would be unreasonable within the time allocated for this project. As such, the majority of the focus will be on those aspects for which it is deemed to be essential that an eReader can support.

Additionally, while all features of the EPUB 3 standard could be tested given enough time, many of these were supported in the previous EPUB 2.01 standard, and are therefore assumed to receive at least a reasonable amount of support from existing software. As such, the main features to be tested as a part of this project will be those which have been newly introduced to the format since the previous version.

### 3.2. THE BASE DOCUMENT

There are aspects of EPUB documents which remain practically the same regardless of content. These should be developed first, before the main content is inserted. These components are the MIME type document, the container.xml file, and the package.opf file. Each EPUB file has its directories laid out very similarly to others of the same type, with the root directory looking identical in each.

A more in-depth description of the directories is given within the following Implementation chapter, but a basic illustration is as follows, with directories emboldened for emphasis:

- **mimetype** (MIME type file)
- **META-INF** (Directory)
  - container.xml (Container file)
- **EPUB** (Directory)
  - **audio** (Directory)
  - **video** (Directory)
  - **styles** (Directory)
  - **images** (Directory)
  - **fonts** (Directory)
  - package.opf (Package file)
  - cover.html (Front cover)
  - toc.html (Navigation Document)
  - ch01.html (Typical main content document)
  - ch02.html (Typical main content document)

### 3.3. NEW FEATURES IN EPUB 3

As it has been decided that there will be a focus on the newer features of the EPUB standard in the testing of software carried out in this project, is essential to pinpoint these features to direct the development in a productive direction, so that time is not wasted testing features which have been supported by eReading software and devices for a considerable length of time already. The IDPF provide clear documentation detailing the

changes from EPUB 2.01 to the new standard of EPUB 3, the major points of which are detailed in the following subsections.

### **3.3.1. HTML5**

Although not exactly a feature, as such, it is a key area to mention that the HTML files which make up the EPUB document use an XML serialisation of the more recent HTML5, rather than the previous XHTML. This means that updated elements in HTML5 are supported, and that powerful new features of HTML5 can be applied when designing eBooks, rather than just websites. A particularly important example of this is inclusion and support for the `<audio>` and `<video>` elements. Hand-in-hand with this exists the new inclusion of trigger elements, which exist to provide a way in which to control the playback of multimedia embedded within the eBook.

### **3.3.2. EPUB Navigation Documents**

Navigation documents have been included in the EPUB 3 standard in place of the .ncx file which used to define the contents of an EPUB file in version 2.01. The new navigation document uses HTML like the majority of the other files within the EPUB container. The new navigation documents make use of HTML5's `<nav>` element, and the document has the advantage of being both human- and computer-readable. It is essential to include a navigation document within the development, due to its required status within the EPUB 3 standard.

### **3.3.3. MathML**

MathML is completely new to the EPUB 3 standard, unlike some of the other features listed within this chapter. MathML is a mark-up language used specifically for marking up equations and mathematical terms in order to display mathematical notation more accurately within web-enabled files. While this may seem of little consequence to those who aren't mathematically minded, it is of great importance to the mathematical and scientific communities, which have taken to publishing works and papers in eBook formats in recent years.

### **3.3.4. Global Language Support**

While we are used to content being displayed in English, and other languages which use a similar character set with a reading direction of left-to-right, support

has previously been lacking for languages which use vastly different script and are to be displayed either vertically or right-to-left. The populations of people who read and write in languages such as Japanese, Chinese and Arabic make up a huge percentage of the world's population, and therefore it is of great importance to include tests for this within the developed EPUB file.

#### **3.3.5. Scripting Support**

Scripting support allows the eBook to include features which use Javascript to provide previously unsupported features, which include but are not limited to geolocation, DOM manipulation, forms and use of the canvas element to dynamically insert images.

#### **3.3.6. Embedded Fonts**

It is possible to embed font files within an eBook with the EPUB 3 standard. This allows greater use of impressive and custom typography within eBooks, allowing design to play a much greater part in the production of books. It must be emphasised that it is still possible, however, for readers to override these custom fonts with their own personal settings as they have defined within their chosen eBook reading software.

#### **3.3.7. Media Overlays**

Media overlays are possibly one of the biggest features of EPUB 3, and take accessibility standards to a new level by allowing an audiobook to be combined with the eBook of the same text. The user can then opt to listen to the book being narrated as they follow along with the words on the screen of their eReading software.

## CHAPTER SUMMARY

---

The relatively concise Design chapter discusses the planning process which took place before the main development of the EPUB file central to the project began. It discusses issues which had a bearing on the direction which the development ended up taking, such as the mandatory formatting of the core eBook file, and the main new features in EPUB 3 which were most likely to have dubious levels of support by existing software, even when EPUB 3 support is claimed by the software's creator.

The discussed features include the mathematics mark-up language, MathML, global language support, support for Javascript features such as geolocation, trigger elements for the control of multimedia, ability to include embedded fonts and the inclusion of media overlays, which allow an audio track to play in sync with the reading of the text.



## 4. IMPLEMENTATION

---

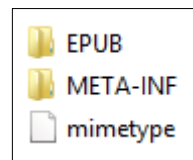
This chapter will detail the components that make up the development, and in-depth information about the development process.

### 4.1. EPUB OPEN CONTAINER FORMAT

The EPUB Open Container Format (OCF) specification defines the way in which files should be encapsulated within the EPUB format, including the directories and files which should reside in each of them.

#### 4.1.1. Root Directory

The root folder of the EPUB file contains a file which defines the MIME type, as well as two directories – the META-INF directory and the EPUB directory. These shall be discussed in the following sections. The composition of the root directory is shown below (Fig. 4)



*Figure 4: Typical Root Directory*

#### ***MIME Type Document***

The ‘mimetype’ file exists within the root directory of the EPUB file system. It does not have a file extension and consists of only one line of text. This communicates to the reader how the file is formatted – known as its MIME type. For an EPUB file, this is as follows:

```
application/epub+zip
```

This file is the same in each and every EPUB document. It is required to be the first file in every EPUB container and must only contain the above string. The document is whitespace sensitive, so the author must make sure that no whitespace occurs around the string.

#### 4.1.2. META-INF Directory

The META-INF folder resides in the root folder and contains an XML file named `container.xml`, as well as optionally containing any additional files which may be required for added features such as Digital Rights Management (DRM).

##### *Container.xml*

The `container.xml` file exists in order to point the reading software to the `package.opf` file which resides within the EPUB directory as an entry point. This file, like the `mimetype` file, is small in size and tends to be fairly consistent across EPUB documents, with differences only occurring where the EPUB directory is named differently.

A point to be noted about the construction of this file is that the section of the code which references the `.opf` file in the EPUB directory uses relative referencing, as is common in XML and HTML, with one difference. The relative path to be used should be relative to the root directory, rather than to the META-INF directory in which the `container.xml` file exists (Fig. 5).

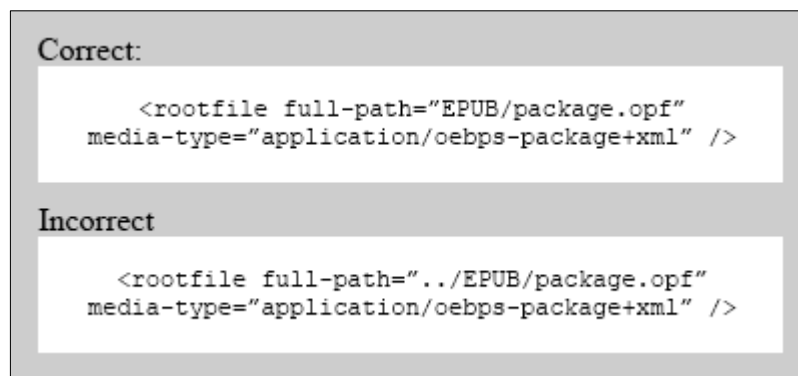


Figure 5: Demonstration of Relative Path in `container.xml`

##### *Metadata.xml*

The `metadata.xml` file is classed in the OCF 3.0 specification as an optional file which exists to store container-level metadata. It is not absolutely necessary to include this file to have a well-formed and valid EPUB document, and it has not been included due to the fact that there is no outline for it as part of the EPUB 3.0 specification. If the software being used to read the EPUB files does not recognise the root element of this document, under the OCF specification it should be able to ignore the file altogether.

### ***Optional Files***

Additional files, as listed in the IDPF's OCF specification, are as following:

- *signatures.xml* – to store digital signatures
- *encryption.xml* – to store information about encryption of any files within the EPUB document. The root document, mimetype file and the files contained within the META-INF folder must not be encrypted.
- *rights.xml* – to store information pertaining to DRM

#### **4.1.3. EPUB Directory**

The main directory in an EPUB file system is the EPUB folder – although, unlike the META-INF directory, this folder can have a different name, so long as it is properly referenced from within other documents. It is, however, standard practice to name the folder either 'EPUB' or sometimes 'OPS'. Inside this folder are a collection of files and folders which make up the content of the eBook to be displayed to the reader. They are organised in a number of sub-folders to collect similar content together – audio, video, HTML, CSS, images and so on – and although these files are not organised in the order that they are to occur inside the eBook, the toc.html document exists to define the order of the content.

#### ***Package.opf***

The package file defines globally applicable metadata and is identified by its .opf extension, as it is not required in the EPUB 3.0 specification that it must be called 'package.opf' in all publications. As well as metadata it contains the <manifest> element which includes an itemised list of every file of which the eBook consists, and the <spine> element which defines the order in which the files should be read.

##### **4.1.3.1. Images Directory**

The Images directory resides within the EPUB directory and contains all image resources to be used in the eBook. These include:

- The cover image for the book (which is discussed in greater detail below).

- The same image in .png, .jpg and .gif formats, in order to test the ability of the software to display all three essential image formats.

### ***Cover.jpg***

The aspect ratio and resolution of a cover image for an eBook can vary greatly, depending on the device on which the file is read. Different bookstores give different requirements for a cover image, including specific resolutions, file formats and file size.

Some important decisions have been made about the development of this file. These are as follows, with justification:

- *Aspect ratio of 3:4* – This was the most sensible choice as an aspect ratio for the cover image due to the fact that it is the aspect ratio of the majority of dedicated eReading devices, and also that of the Apple iPad (Stokes, 2014). The aspect ratio matters most on devices such as tablet computers, smartphones and dedicated reading devices due to the fact that applications for reading on devices such as laptop and desktop computers tend to be windowed, allowing size and aspect ratio to vary.
- *1400 pixels x 1867 pixels* – As mentioned by Stokes (2014) in her discussion of aspect ratios and formatting sizes, in order for a cover image to be accepted for sale by Apple’s iBookstore, the cover image must have a minimum width of 1400 pixels. Using that minimum as the base value for the width of the image and the chosen aspect ratio of 3:4, the height of 1867 pixels was calculated.
- *.jpg format* – The decision for file format came down to a choice between .png and .jpg. In terms of quality of the output, .png is far greater, however .jpg is far more common to encounter as a file format. Choosing to use a .jpg file meant choosing a slight loss of quality, but a far greater chance of compatibility with every device and piece of software.

### ***4.1.3.2. Audio Directory***

The Audio directory resides within the EPUB directory and contains any and all audio resources to be included and used within the book. An open source sound file was downloaded and included in both MP3 and AAC formats, in order to test the HTML5 <audio> feature.

#### **4.1.3.3. Video Directory**

This directory also resides within the EPUB directory and contains video resources for use within the book. Similarly to the audio files, an open source video file was downloaded and included in MP4 format to test the HTML5 `<video>` feature. It was important that the video chosen was only a small file, lasting only 21 seconds, in order to ensure that the addition of the file to the document would add sufficient content to test the functionality while ensuring that the size of the EPUB document did not grow to unwieldy sizes and cause the loading of the document to become slow.

#### **4.1.3.4. Styles Directory**

This directory contains any CSS files included to define style rules which should apply to the document. It should be noted that eReading software often allows readers to personalize their reading experience by selecting their own preferred font, font size, background colour and more, depending on the software.

#### **4.1.3.5. Fonts Directory**

As with the styles, images, audio and video directories, the fonts directory exists to hold any font files which are to be embedded within the document. It is worth noting that any files included within an EPUB file can be extracted by any individual. This means that third-party fonts which have been licensed for commercial use could be taken for free by the individuals who own a copy of the eBook. It is therefore recommended that, to avoid problems with copyright infringement, any fonts which are not open source should be obfuscated, with the appropriate accompanying xml file inside the META-INF directory.

#### **4.1.3.6. HTML Documents**

The HTML documents which make up the chapters of the EPUB document all follow the pattern of a typical HTML document like any used in development of simple web pages. There is a *cover.html* file which consists solely of the cover image. Also within the EPUB directory is the important *toc.html* file, which

## **4.2. ISSUES**

Contents in EPUB 2.01 are defined in an .ncx file. This file does not exist under the EPUB 3.0 standard and, as such, opening EPUB 3.0 files in Sigil throws up an error declaring the EPUB file improperly formed, and an .ncx file is automatically created.

The contents pane in the Sigil development environment does not work properly as a result of this. This may indicate an area of concern with regards to compatibility in software, and raises the following questions:

*Will contents of the EPUB file display correctly with the new contents document and no .ncx file?*

*Should an .ncx file be included with an EPUB 3 document for sake of back-compatibility?*

The answers to these questions have been discovered during the course of the project, and have determined that an EPUB file will indeed display in eReading software with no .ncx file, even when the software does not boast support for the EPUB 3 standard. However, for good practice it has been recommended by the IDPF that the .ncx file should indeed be included in EPUB 3 documents for commercial use in order to maximise compatibility between devices and software.

## CHAPTER SUMMARY

---

The Implementation chapter discusses the actual construction of the EPUB file used within this project for testing the supported features within eReading software. It looks in greater detail than any other chapter of this report at the content documents from a more technical perspective, examining how the different files work together and connect to each other to create a working EPUB 3 file.

Also discussed in the Implementation chapter are some issues which came to light during the construction of the EPUB document, and proposed solutions to these issues.

## 5. EVALUATION

---

This chapter is divided into two sections – testing and results. The testing section will deal with the process of testing that has been carried out and a brief outline of the results acquired from the tests. The results section will then take the results acquired during testing and look at them in more detail, aiming to draw useful conclusions from them for use by developers of eBooks in the EPUB format.

### 5.1. TESTING

This section will discuss in-depth the testing of the developed eBook. The first section will deal with the devices and software which will be used to carry out testing of compatibility between software and the standard which is claimed to be supported.

Each test conducted will be run by opening the developed EPUB file in a designated piece of software, denoted by the section heading in which the description of the testing appears. For each feature built into the EPUB document, compatibility can be either classed as a pass or a fail, and will be human-evaluated. The pieces of eReading software used within the testing process are listed within this section in alphabetical order for ease of location.

#### 5.1.1. Devices

The devices used as part of this project are as follows. Listed are the technical specifications of their displays pertaining to the development. The information given is about the displays rather than other information such as processing power due to the fact that so far as the devices are concerned, the display is more relevant to the information being gathered as part of the project than the processing power of each device.

Device	Operating System	Display Size	Display Resolution	Display Type	Contrast Ratio
iPad (3 <sup>rd</sup> Gen)	iOS 7	9.7"	2048 x 1536	LCD	8 0 0 : 1 (Typical)
iPhone 4	iOS 5	3.5"	960 x 640	LCD	8 0 0 : 1 (Typical)
K o b o Touch	Kobo 3.2.0	6"	800 x 600	E Ink Pearl	10:1 (Min)



Laptop	Windows 8.1	15.6"	1366 x 768	LCD	Variable
--------	-------------	-------	------------	-----	----------

*Figure 6: Device Display Information Table*

### 5.1.2. Test Areas

Each piece of software is tested in the same areas. These are as follows:

Area of Test	Nature of Test	Importance
Audio	Testing of both MP3 and AAC playback	Required
Video	Testing of MP4 playback	Optional
Trigger Elements	Testing whether EPUB trigger elements can be used to control multimedia	Required
MathML	Testing of ability to display mathematical notation instead of linear text	Required
Global Language Support	Testing of ability to display foreign language scripts, such as Arabic and Hebrew. Testing of right-to-left reading direction.	Required
Styling	Testing of changes to the display of text.	Required
CSS	Testing of CSS rules, such as ability to include background image, the ability to display content in multiple columns, the ability to style lists in a number of ways	Required
Scripting	Testing of ability to include scripting features which enable things such as geolocation, forms and the canvas element in particular.	Mostly optional
Media Overlays	Testing of ability to include an audio track which is synchronised with the text	Required

Embedded Fonts	Testing of the ability to include fonts which are not found on computers as standard.	Required
----------------	---	----------

*Figure 7: Test Items Table*

### 5.1.3. AZARDI

The AZARDI desktop application was run on the laptop device listed in Fig. 6. AZARDI fared well in the test of the newly-introduced HTML5 `<audio>` element, with files embedded in both MP3 and AAC displaying and playing properly. The testing of the `<video>` element went similarly well with the MP4 video, with the file displaying smoothly. The area where AZARDI seems to be lacking in the multimedia section is the functioning of EPUB trigger elements, with no functionality at all in that area.

MathML was the next area to be tested, and AZARDI gave a surprisingly good performance, rendering mathematical symbols well, rather than displaying the equation in a linear fashion. It also rendered foreign language scripts well, with both Arabic and Hebrew displaying as they should.

Some styling of text works as it should, with the font size, weight, style and decoration displaying as they should. Attributes which determine additional spacing between words, and additional spacing between letters, however, do not change the way the text is displayed in any way. Positioning seems to work well, with tests of differing line and character alignments displaying as they should. The software fails to display a background image defined in a CSS rule, but other CSS rules seem to work fairly well.

AZARDI does reasonably with the scripting test, offering partial support for geolocation, and full support for use of forms and the canvas element. Unfortunately, AZARDI falls flat when it comes to the support of Media Overlays, with them completely lacking support. Embedded fonts are partially supported, working only in conjunction with HTML documents and not with SVG documents.

### 5.1.4. iBooks

iBooks was tested using an iPhone 4 running iOS5. iBooks was effective at displaying both the HTML5 `<audio>` and `<video>` elements. MP3 and AAC playback worked well and the MP4 video displayed as expected. An interesting

observation to make is the fact that initiating playback of a video opens that file and displays it full-screen, rather than in-line with the other content of the eBook. This is as standard when opening videos within iOS and causes no real problems in the reading of the book. Trigger elements used to start and stop playback of multimedia within the eBook also works within iBooks, a factor which sets it apart from the other selected software tested within this project thus far.

In the testing of MathML, it was found that while there is *some* support for displaying equations with proper mathematical notation, there is still a relatively large gap in support, with a number of symbols displaying as the □ character. Foreign scripts support, like the MathML support, was hit-and-miss. Some characters displayed as expected while others displayed the same white square character as when symbols within the MathML were missing.

iBooks is particularly strong when it comes to rendering styles within the text, with each of the CSS text transformation tests displaying correctly, including the testing of additional spacing between characters and between words. This, again, puts iBooks just ahead of AZARDI in the capabilities with manipulating text.

Where scripting is concerned, iBooks offers a decent range of support. It does not, for example, support mouse events, but that can be fully expected in a program which operates on a range of devices which do not require mouse input. It does, however, support both touch and keyboard events. Geolocation is supported, but it is still not particularly viable for inclusion in interactive stories, as it displays location with co-ordinates as opposed to supporting the display of place names.

#### **5.1.5. Kobo Touch**

The Kobo Touch with its Kobo operating system does not perform anywhere near as well as its competitors which use more powerful computing devices. This is due to its lack of technological capability. It is unable to play any audio whatsoever, and the extremely slow refresh rate of the screen means that video playback is virtually impossible. Because of the incompatibility with multimedia playback, the trigger elements can be disregarded. Also unsupported because of the inability to play sounds is the media overlay feature.

Despite the lack of support for multimedia, there is support for MathML and the writing of equations. This is a positive result for the scientific and mathematical

community, allowing equations to be published within EPUB files which can be read on even the most basic of portable reading devices.

Text altering and varied positioning is supported, but when it comes to RTL global language support, it is still lacking, with support not currently provided.

Where scripting is concerned, there is very basic inclusion of some important elements, such as form buttons and canvas elements. In anything more advanced than that, support is also currently lacking, with features like geolocation unsupported. The varied support is also evident in the ability to embed fonts, with only certain types of fonts displayed properly.

#### **5.1.6. Radium**

Radium is determined by [epubtest.org](http://epubtest.org) to be the most EPUB 3 standard compliant eReading software that exists at present. It comes in the form of a browser extension for Google Chrome, and allows users to add their eBooks to a library which is stored online.

Radium boasts impressive support for audio and video, including support for subtitles and captioning which is lacking in all of the other tested software. The trigger elements all work well, with one exception. However, this still outperforms the rest.

Despite the impressive start, the complete lack of support for MathML is disappointing, given the support from the rest of the tested software, proving to be an anomaly.

Text transformation brings Radium back onto the previous good form with which it started, with all of the tested transformations displaying as expected. The rest of the tests are completed with seeming ease, with scripting, language support, embedded fonts and media overlays displayed fully as expected.

## **5.2. RESULTS**

This section will present and break down the results acquired through the testing discussed in the previous section of this chapter, presenting them in a digestible form along with conclusions that may be drawn from the findings.

### **5.2.1. E Ink**

eInk devices are wonderful at replicating the feel of published texts such as novels and short stories, where content is generally high contrast and black and white.

As figures show (cite), there is a huge demand for this, and the devices are successful at providing the experience users expect. They are not so well equipped to handle more rich and interactive content, and there are a number of reasons why they fall flat with respect to the areas which have been enhanced in the latest version of the EPUB standard.

#### ***5.2.1.1. Images***

eInk devices can display images with their text, which allows image content which may have been part of the physical book to be replicated in the eBook version. However, as with black and white printed content in novels, images in eBooks which are to be displayed on eInk screens need to have a certain level of contrast in order for them to display well. Colours which look vastly different on full colour screens may be practically indistinguishable when displayed in shades of grey, meaning that some of the content may be lost in translation.

#### ***5.2.1.2. Audio***

eInk devices lack capability to play sound in the way that other computing devices built for multimedia do. This means that any eBook with an audio track for the benefit of the visually impaired will be unable to be fully utilised.

#### ***5.2.1.3. Interactive Content***

eInk devices have a much slower refresh rate than other computing devices, which is not a problem when the action is choosing an option from a menu or turning a page of a book – actions which only need to be carried out once every 30-60 seconds depending on the speed at which the user of the device is reading. One of the benefits of eInk devices is that power is only consumed when the display changes. Anything which might require refreshing more often, such as once every second or less, becomes painfully noticeable as soon as the user of the device wants to interact many times in a short space of time, such as choosing an answer in a quiz and then submitting. Where the user may be used to these things being able to be done instantly on another device, the two second wait may seem unreasonable and clunky. Features such as drawing are supported on recent eInk devices such as the Kobo Touch, but compared to drawing on tablet devices, the wait that the user is required to endure before they see the result of their input makes the application feel clunky and unresponsive.

### **5.2.2. Computing Devices**

It is clear from the results acquired during testing that support for different features within software developed for general computing devices varies widely. Some pieces of software offer full support for certain features which are completely unsupported in similar software by a different developer. While support tends to seem to linger above the 50% mark for all devices at the current time, it is clear that many improvements could be made.

## CHAPTER SUMMARY

---

The Evaluation chapter deals with the main part of the project which produces the results from which conclusions can be gleaned. It is split into two sections. The first section covers the testing process, listing the devices and the areas which were required to be tested to check for compatibility with the identified features of the EPUB 3 standard. There is a subsection for each piece of software tested using the developed file, with evaluation of the performance in each area of testing.

The second section summarises the results of the testing, taking into account the strengths and weaknesses of each of the pieces of software, and drawing a conclusion about the general state of support for features in current times.

## 6. CONCLUSION

---

This section will serve to provide an overall summary of the development which has taken place for the purpose of finding answers to the primary research question, and to satisfy the objectives identified in the planning stages of the project. It will also sum up the findings acquired when testing the development and what these findings mean for stakeholders – in this case, those within the digital publishing industry (be it professional publishers or self-publishers), retailers of eBooks in the EPUB format and members of the general public who read books electronically as well as, or instead of, physically.

### 6.1. THE DEVELOPMENT

The development in this project was an EPUB 3 standards-compliant eBook, which was created to the standards defined by the IDPF in their specification. The eBook itself consisted of a number of directories and a range of files within them. Some of these files were configuration documents which defined important information allowing for the document to be run by the appropriate range of software across a range of devices. The *mimetype* file within the root directory told applications which type of file was being run. The *container.xml* file was contained within the META-INF directory (used to hold all xml files which allows for definition of important metadata, encryption information, digital rights management and more) and communicated to the reader the path from the root directory to the entry point of the eBook, which was the *package.opf* file. This file then acts as a gateway to the rest of the document, which consists of HTML files, CSS stylesheets, images, audio files, video files and embedded font files.

### 6.2. THE FINDINGS

Some of the most definitive findings are those concerning dedicated eReading devices with eInk displays. While they are very well suited to displaying novels in a form that very closely replicates novels in paper form, they are not well suited to carrying out tasks which are more intensive. The good thing is that insofar as replicating novels is concerned, eInk devices have little need to display more graphically intense content, or to include seamless multimedia content. The refresh rates of the screens which exist at this current point in time are too low to properly display animations, yet that is not a huge problem considering that the main content consumed with eInk devices tends to be no more than just formatted text. There is a market for these devices in the same way there is still a market for novels in paper form, and it is doubtful whether divergence in the types of eBook available will take anything away from that.

There are, however, areas in which these eInk devices could yet improve to meet the EPUB 3 standard. Improved metadata and accessibility features would be beneficial for the visually impaired, the print disabled and other groups which rely on accessible



features to allow them to enjoy content as much as their able counterparts. This aspect is of great importance as it does not just provide entertainment – it provides those less able with a way to enjoy content they may not have been able to access otherwise, and should be the next big focus of those who create eInk eReaders.

Inclusion of media overlays with sound could be a great step towards improving accessibility where eInk devices are concerned. However, despite the fact that this may be ideal, this raises the question – *Would the hardware support those features without putting strain on the device or diminishing features which are classed as advantages of eInk devices, such as the battery life?*

Insofar as software for general computing devices is concerned, support is generally fairly good, but lacking in varying areas. It would seem that support depends on the features which the software developers are most keen on including, with some providing greater MathML support but falling flat when it comes to accessibility features. This is the area which requires greater support, and hopefully in the next few years that will begin to come through in new versions of eReading software. That is, of course, if publishers and software developers alike agree that progress needs to be made, and the step towards greater interactivity and accessibility can then be taken. Until then, support will largely depend on trial and error. The best software for one feature may be a terrible choice for another desired feature.

## GLOSSARY

---

<i>Reflowable</i>	Content which can change its presentation to fit within differently sized content areas.
<i>Fixed-layout</i>	Content which has its presentation set in a content area of a specific size.
<i>DRM</i>	Digital Rights Management. Technologies which can be used to control the way a file is used after it has been sold to a customer.
<i>GUI</i>	Graphical User Interface.
<i>IDPF</i>	International Digital Publishing Foundation. The body behind the EPUB specification.
<i>RTL</i>	Right-to-left. Used primarily when talking about reading direction.
<i>DOM</i>	Document Object Model.
<i>SVG</i>	Scalable Vector Graphics. Documents, commonly images which are created dynamically, allowing for resizing without loss of quality.

## REFERENCES

---

- Apple (2014) *iBooks Textbooks For iPad* [Online] Available from: <https://www.apple.com/education/ipad/ibooks-textbooks/> [Accessed: 5<sup>th</sup> February 2014]
- Calibre (2014) *Convert PDF Documents* [Online] Available from: <http://manual.calibre-ebook.com/conversion.html#pdfconversion> [Accessed: 3<sup>rd</sup> February 2014]
- Chang, C. & Yang, H. (2013) A Study of EPUB 3.0 Authoring Tools Availability in Vertical Form Function, *Journal of Software Engineering and Applications*, vol. 6, no. 9, pp. 482-488.
- Chao, C., Fuxman, L. & Elifoglu, I. H. (2013) Electronic Books Impact Global Environment – An Empirical Study Focus on User Perspectives, *Journal of Management and Strategy*, vol. 4, no. 2, pp. 52.
- Coker, M. (2010) *Most Popular eBook Formats Revealed* [Online] 10<sup>th</sup> February 2010. Available from: <http://blog.smashwords.com/2010/02/most-popular-ebook-formats-revealed.html> [Accessed: 9<sup>th</sup> November 2013]
- E Ink (2013) *E Ink Carta Imaging Film* [Internet], Billerica, E Ink. Available at: [http://www.eink.com/sell\\_sheets/carta\\_spec\\_sheet.pdf](http://www.eink.com/sell_sheets/carta_spec_sheet.pdf) [Accessed: 26<sup>th</sup> January 2014]
- Enis, M. (2013) Accessibility Upgrade, *Library Journal*, vol. 138, no. 7, pp. 31-n/a.
- Falk, H. (2000) E-Ink and E-paper, *The Electronic Library*, vol. 18, no. 5, pp. 366.
- Garrish, M. & Gylling, M. (2013) *EPUB 3 Best Practices*. Sebastopol: O'Reilly Media
- Garrish, M. (2012) *Accessible EPUB 3*. Sebastopol: O'Reilly Media
- Garsia, M. & Savova, M. (2012) McGill Library Makes E-books Portable: E-reader Loan Service in a Canadian Academic Library, *Portal: Libraries and the Academy*, vol. 12, no 2, pp. 205-222.
- Gregory, C. (2008) 'But I Want a Real Book': An Investigation of Undergraduates' Usage and Attitudes Toward Electronic Books, *Reference and User Services Quarterly*, vol. 47, no. 3, pp. 266-273
- Griffey, J. (2010) Electronic Book Readers, *Library Technology Reports*, vol. 46, no. 3, pp. 7-19,2.
- Gross, M. (2012) *Understanding Content Conversion: Unfortunately There's No 'Easy' Button* [Online] 17<sup>th</sup> January 2012. Available from: <http://www.thedynamicpublisher.com/2012/01/17/>

[understanding-content-conversion-unfortunately-there%E2%80%99s-no-%E2%80%98easy%E2%80%99-button/](#) [Accessed: 5<sup>th</sup> March 2014]

Habash, G. (2012) Preparing for the Future, *Publishers Weekly*, vol. 259, no. 51, pp. 25-n/a

Hoffelder, N. (2013) *Bill McCoy is Wrong – Epub3 Isn't Ready* [Online] 28<sup>th</sup> May 2013. Available from: <http://www.the-digital-reader.com/2013/05/28/bill-mccoy-is-wrong-epub3-isnt-ready/> [Accessed 4<sup>th</sup> February 2014]

Hoffelder, N. (2011) *RIP: Mobipocket 2000-2011* [Online] 2<sup>nd</sup> November 2011. Available from: <http://www.the-digital-reader.com/2011/11/02/rip-mobipocket-2000-2011/> [Accessed 5<sup>th</sup> March 2014]

IDPF. (2014) *EPUB 3 Accessibility Guidelines* [Online] Available from: <http://www.idpf.org/accessibility/guidelines/content/qa/qa-checklist.php> [Accessed 20<sup>th</sup> March 2014]

IDPF. (2011) *EPUB 3 Overview* [Online] Available from: <http://www.idpf.org/epub/30/spec/epub30-overview.html> [Accessed 13<sup>th</sup> November 2013]

Kasdorf, B. (2013) Are You Getting the Most Out of EPUB 3?, *Book Business*, vol. 16, no. 3, pp. 22-24,26,32.

Lebert, M. (2009a) *A Short History of EBooks* [Online] Toronto: NEF, University of Toronto. Available from: <http://www.gutenberg.org/ebooks/29801> [Accessed: 5<sup>th</sup> November 2013].

Lebert, M. (2009b) *Technology and Books for All* [Online] Toronto: NEF, University of Toronto. Available from: <http://www.gutenberg.org/ebooks/27098> [Accessed: 5<sup>th</sup> November 2013].

Lebert, M. (2008) *Project Gutenberg (1971-2008)* [Online] Toronto: NEF, University of Toronto. Available from: <http://www.gutenberg.org/ebooks/27045> [Accessed: 5<sup>th</sup> November 2013].

*Philadelphia Business Journal*, (2005). Franklin sells interest in company, retires shares. 31<sup>st</sup> March 2005, Philadelphia.

Scardilli, B. (2013) Moving Toward an Ebook Standard, *Information Today*, vol. 30, no.11, pp.12-13.

Stokes, E. (2014) *eReader and Print Book Formatting Aspect Ratios* [Online] 2014. Available from: <http://erikastokes.com/kindle-and-ebook-formatting/ebook-display-and-pod-book-aspect-ratios.php> [Accessed: 10<sup>th</sup> March 2014]

Tan, T. (2013) In a Constant State of Flux and Frenzy: Digital Solutions in India 2013, *Publishers Weekly*, vol. 260, no. 15, pp. 4-n/a

*Targeted News Service*, (2013) AAP EPUB 3 Implementation Project. 24<sup>th</sup> July, Washington, D.C.

Wikert, J. (2013a). *High-quality PDF-to-EPUB conversion* [Online] 14<sup>th</sup> January 2013. Available from: <http://toc.oreilly.com/2013/01/high-quality-pdf-to-epub-conversion.html> [Accessed: 6<sup>th</sup> November 2013]

Wikert, J. (2013b). *Our fixation on containers* [Online] 16<sup>th</sup> December 2013. Available from: <http://www.olivesoftware.com/3/post/2013/12/our-fixation-on-containers-by-joe-wikert.html> [Accessed: 1<sup>st</sup> February 2014]

Wikert, J. (2012a) *B&N's Nook with GlowLight: Why I Still Use an eInk eReader* [Online] 7<sup>th</sup> May 2012. Available from: [http://jwikert.typepad.com/the\\_average\\_joe/2012/05/bns-nook-with-glowlight-why-i-still-use-an-eink-ereader.html](http://jwikert.typepad.com/the_average_joe/2012/05/bns-nook-with-glowlight-why-i-still-use-an-eink-ereader.html) [Accessed: 1<sup>st</sup> February 2014]

Wikert, J. (2012b) *Is Color eInk a Game-Changer?* [Online] 21<sup>st</sup> February 2012. Available from: [http://jwikert.typepad.com/the\\_average\\_joe/2012/02/is-color-eink-a-game-changer.html](http://jwikert.typepad.com/the_average_joe/2012/02/is-color-eink-a-game-changer.html) [Accessed: 1<sup>st</sup> February 2014]

*Wireless News*, (2013) Barnes & Noble College Updates the Open Source EPUBCheck Tool. 26<sup>th</sup> October, Coventry

## APPENDIX 1: CODE LISTINGS

---

### EPUB FILE

#### Root Directory

##### *mimetype*

application/epub+zip

#### META-INF Directory

##### *container.xml*

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<container
xmlns="urn:oasis:names:tc:opendocument:xmlns:container"
version="1.0">
  <rootfiles>
    <rootfile full-path="EPUB/package.opf" media-
type="application/oebps-package+xml"/>
  </rootfiles>
</container>
```

#### EPUB Directory

##### *package.opf*

```
<?xml version="1.0" encoding="UTF-8"?>
<package xmlns="http://www.idpf.org/2007/opf"
version="3.0"
xml:lang="en"
unique-identifier="pub-id">

  <dc:language>en</dc:language>
  <dc:title>EPUB 3</dc:title>
  <dc:creator id="creator">Amanda-Jayne Davies</
dc:creator>
  <meta refines="#creator"
property="role"
scheme="marc:relators">aut</meta>
  <meta
property="dcterms:modified">2014-01-01T00:00:00Z</meta>
  <dc:publisher>Gummybuns</dc:publisher>
  <dc:date>2014-01-01</dc:date>
  <meta property="dcterms:dateCopyrighted">2014-01-01</
meta>
</package>
```

```

    <manifest>
      <item id="cover" href="cover.xhtml" media-
type="application/xhtml+xml" />
      ...
      <item id="cover-image" properties="cover-image"
href="Images/contrast.jpg" media-type="image/jpeg" />
    </manifest>
    <spine page-progression-direction="ltr">
      <itemref idref="cover" linear="yes" />
      ...
      <itemref idref="ch012" linear="yes" />
    </spine>
</package>

```

## APPENDIX 2: TESTING

---

### ACCESSIBILITY CHECKLIST (IDPF)

The following Accessibility Checklist is as given by the IDPF (2014), and was used in development to ensure that all aspects of the development met the IDPF's accessibility standards. Items followed by [R] are classed as required, and those followed by [O] are classed as optional within the EPUB 3 Specification.

#### Semantics

##### The `epub:type` attribute

- Semantics from the EPUB Structural Semantics Vocabulary added to identify structural sections and significant content structures

##### Logical Reading Order

- Secondary content is identifiable from logical reading order by inclusion in `aside` and `figure` tags
- The order of the markup in content files matches the primary narrative
- The publication is readable from beginning to end without scripting or styles applied

##### Separation of Style

- CSS properties not defined using the `style` attribute [R]

### XHTML Content Documents

#### Language

- The `lang` and `xml:lang` attributes are set on all root `html` elements
- The `lang` and `xml:lang` attributes are set on all instances of foreign language prose

#### Emphasis and Bolding

- `em`, `strong`, `i` and `b` tags used as per their definitions in HTML5 [R]
- CSS styling used for presentational bolding and italics [R]

#### Links

- Link text clearly indicates destination, or meaningful alternative is provided in a `title` attribute
- A secondary means of identifying linked text has been provided [R]



## Sections

- All structurally-significant content grouped in **section** elements [R]

## Headings

- Numbered headings accurately reflect the document hierarchy
- Each sectioning element has only a single heading
- Headings have not been used in **blockquote** and **figure** elements [R]

## Lists

- Sets of related items marked up using **ol** or **ul** elements, as appropriate
- Glossaries, dictionaries, faqs and similar name/value association lists marked up using **dl** elements

## Tables

- All table heading cells identified using **th** elements
- The **thead** element used to group multi-row headings
- The **scope** attribute added when heading cell applicability is not clear [R]
- The **headers** attribute added to cells when their heading is ambiguous or contained in more than one cell [R]
- A summary of the table layout and content provided for all complex tables [R]
- Captions provided, as applicable [R]
- Tables not used for presentational or fixed layout of content

## Audio

- Native controls enabled by default
- To assist when audio may not be audible, one or more of the following provided: [R]
  - Transcripts of audio content provided
  - Sign-language interpretation provided via an associated video element
- Navigation by scene provided, when applicable [O]

## Video

- Native controls enabled by default
- To assist when audio may not be audible, one or more of the following provided: [R]
  - Captions for all video content containing audio content
  - Subtitles for all video content containing spoken language
  - Sign-language interpretation track for all video content

- Transcripts of video content provided
- To assist when video may not be visible: [R]
  - Text description tracks provided
- Navigation by scene provided, when applicable [O]

### Figures

- Images, charts, code samples and other secondary content encapsulated in **figure** tags
- The **figcaption** element used to identify figure captions

### Images

- Text alternatives provided for all significant images
- Descriptions provided for all significant images [R]
- **alt** attribute left empty for all decorative images
- The ARIA **role** attribute included with the property **presentation** for all decorative images [R]

### Image Maps

- **alt** attribute used to provide meaningful labels to image map areas
- Server-side image maps not used [R]

### Inline Frames

- A meaningful title added to all significant **iframe** elements
- A link to the associated content document included as a fallback for all significant **iframe** elements [R]
- Relative sizing units used for all **iframe** elements [R]
- Scrolling not disabled by default on **iframe** elements [R]

### Notes

- Notes within the primary narrative identified using the **aside** element with an appropriate **epub:type** value
- Groups of notes identified by an appropriate **epub:type** value
- Note references identified using the **epub:type** attribute with the value **noteref**

### Annotations

- Annotations within the primary narrative identified using the **aside** element with an **epub:type** attribute with the value **annotation**
- Inline annotations identified using the **ruby** element [R]
- Annotation references identified using the **epub:type** attribute with the value **annoref**

### Context Breaks

- Changes in context identified using **hr** elements [R]

### Page Numbers

- The print source identified in a **dc:source** element in the package document **metadata** section
- Print page break locations included when the EPUB has a print equivalent [R]
- The navigation document includes a **page-list nav** to facilitate page navigation [R]

### **MathML**

#### Descriptions

- MathML markup included for all math content
- Simple MathML content described using the **alttext** attribute [R]
- Complex MathML content described using an embedded **annotation-xml** element [R]

### **SVG Content Documents and Images**

#### Language

- The **xml:lang** attribute set on all root **svg** elements
- The **xml:lang** attribute set on all instances of foreign language text within each SVG image

#### Titles and Descriptions

- All significant images include a title (**title** element)
- All complex images include a description (**desc** element)
- All components within an SVG image include a **title** and **desc**
- All presentational SVG images marked using the ARIA **role** attribute [R]

#### Components

- Component definitions re-used whenever possible to simplify overall comprehension [R]

#### Text Content

- All embedded text content contained in **text** elements
- Sufficient contrast maintained between embedded text and background image
- Custom fonts used in place of creating text as images

#### Linking

- All embedded links include an **xlink:title** attribute explaining the target location
- Linked components are visually distinguishable from unlinked components [R]

#### Styling

- CSS rules defined for **title** and **desc** elements to enable rendering
- Visual styling of components separated into CSS style sheets to facilitate reader customization [R]

### Interactivity

- Scripting not required to render or understand SVG images
- ARIA roles, states and properties used to enhance the accessibility of scripted content
- Text equivalents are updated whenever dynamic content changes
- Device-independent events used for all interactive actions

## **EPUB Style Sheets**

### Colors

- Semantic markup used whenever colors convey information visually
- Sufficient contrast levels maintained between the text and any background colors and images (as defined in WCAG 2.0)
- Text not positioned over pictures and other images with noisy backgrounds [R]
- Red/green combinations have been avoided [R]
- Red/black combinations have been avoided [R]
- Brightly colored hue combinations have been avoided [R]

### Backgrounds

- Background images not used to convey information
- Sufficient contrast levels maintained between the text and any background colors and images (as defined in WCAG 2.0)

### Hidden Content

- Content critical to the logical flow of a publication not initialized in a hidden state
- HTML5 and ARIA techniques used to include optional text and descriptions
- The state of the **aria-hidden** attribute maintained as the visibility of content is toggled

## **Fixed Layout**

### XHTML

- Logical reading order maintained in markup
- Semantic markup applied to content to facilitate comprehension of layout

## Image

- Publication does not contain empty documents that employ the **background-image** property to display images

## **Navigation**

### Table of Contents

- The **toc nav** includes the full structure of the publication
- The **hidden** attribute used to disable visual rendering of unwanted levels [R]

### Landmarks

- Links to all significant sections of the publication added [R]

### Page List

- The package document metadata includes a **dc:source** element identifying the print edition

### List of Illustrations

- Links to all illustrations and figures in the publication provided [R]

### List of Tables

- Links to all tables in the publication provided [R]

## **Media Overlays**

### Highlighting

- Sufficient contrast maintained between active text and the highlighting color

### Lists

- Each list in the markup represented by a **seq** element with its **epub:type** attribute set to **list** [R]
- Each list item in the markup represented either by a **seq** or **par** element with its **epub:type** attribute set to **list-item** [R]
- All related **dt** and **dd** elements in a definition list are grouped together in a **seq** element with its **epub:type** attribute set to **list-item** [R]

### Tables

- Each table in the markup represented by a **seq** element with its **epub:type** attribute set to **table** [R]

- Each table row in the markup represented by a **seq** element with its **epub:type** attribute set to **table-row** [R]
- Each table cell in the markup represented either by a **seq** or **par** element with its **epub:type** attribute set to **table-cell** [R]

## **Text-to-Speech**

### PLS Lexicons

- The lexicon defines all unique proper names, nouns and other complex words [R]
- The lexicon includes the default pronunciation for any heteronyms [R]

### SSML

- Pronunciations defined to disambiguate heteronyms [R]
- Pronunciations defined for ambiguous number forms [R]
- Pronunciations defined for jargon, slang, slurred and other non-standard speech forms [R]

### CSS3 Speech

- All abbreviations, acronyms, numbers and other constructs that should be spelled out have the **spell-out** property defined [R]

## **Scripted Interactivity**

### Progressive Enhancement

- Scripting not required in order to traverse the complete logical reading order of the publication
- Scripting not required in order to make content visible
- Readers can access all content regardless of location, preference settings, etc. [R]

### Content Validity

- Scripting not used to manipulate the HTML DOM in such a way that it produces invalid markup

### WAI-ARIA & Custom Controls

- All custom controls are reachable in a device-independent manner
- All custom controls fully implement ARIA roles, states and properties, as appropriate

### Forms

- Labels and controls linked via markup

- All custom controls are reachable in a device-independent manner
- Form controls laid out in logical order or **tabindex** attribute used to give order
- All custom controls made accessible via use of ARIA roles, states and properties
- The **aria-invalid** attribute set when fields contain invalid data

### Live Regions

- All live regions identified using ARIA roles and properties
- Assertiveness of all announcements matches their importance to the reading experience

### Canvas

- **figure** and **figcaption** used to add context when **canvas** represents a static image
- **canvas** only used to enhance graph, chart and similar data representations
- ARIA roles, states and properties used to make interactive **canvas** elements accessible
- Alternate representations and fallbacks included when accessible scripting not possible