EPUB 3

Steven R. Bagley

with Matthew R. B. Hardy

Introduction

- eBooks are everywhere
- Currently, most books are described using EPUB 2 (or Kindle `mobi`)
- Limitations to what can be expressed
- EPUB 3 aims to address these limitations

EPUB 3

- Aims to be standards compliant
- Replaced uncommon standards
- Normalizing to Web technologies
- SMIL, MathML, HTML5, CSS, etc. all used

EPUB 3

- Support for:
  - Right-to-left and vertical text
  - Mathematics (MathML)
  - Multimedia
  - Interactivity and Scripting
  - Fixed layout

Aims of the lecture

- To explore EPUB 3 and how it fits together
- See how to create exciting new types of eBooks
- Not going to go into depth of existing standards

Aims of the lecture

- Will paint the big structure
- But not going to go into detail on all the attributes on all the tags
- *Strongly* recommend you read:
  - EPUB Content Documents 3.0
  - EPUB Publications 3.0

http://www.idpf.org/epub/30/spec/epub30-contentdocs.html
http://www.idpf.org/epub/30/spec/epub30-publications.html

Viewing EPUB 3

- Erm...

Viewing EPUB 3

- Very limited support
- iPad supports some features (in a non-standard way)
- Adobe Digital Editions has limited supported (and associated eBook readers)
- Mainly RTL text...
- Readium, …

Structure of EPUB

- In many ways, mimics the structure of a physical book
  - Content (HTML, SVG,etc.)
  - Ordering (Package Document)
  - Binding (Open Container Format)

Open Container Format

- Based on *ZIP*
  - Flate compression for files
  - Standard zip structures
- Organization on top of ZIP
  - `mimetype` (uncompressed first file)
  - `META-INF` folder

Inside EPUB

Open Package File

- Package file is the entry point for the book
  - Book Metadata
  - Lists all files used in the book
  - Orders the content (twice)
- Can be compatible with EPUB 2

Open Package File

- Composed of three main sections
  - Metadata — `<metadata>...</metadata>`
  - Manifest — `<manifest>...</manifest>`
  - Spine — `<spine>...</spine>`
- Also, optionally…
  - Guide (optional) — `<guide>...</guide>`
  - Bindings (optional) — `<bindings>...</bindings>`

Each section is a separate XML element under <package>
Must be in order
Go look at an epub 3 OPF file

```
<package>

  <metadata>
  …
  </metadata>

  <manifest>
    <item>…</item>
  </manifest>

  <spine>
    <itemref>…</itemref>
  </spine>

</package>
```

Manifest

- Lists all the resources used by the eBook (directly and indirectly)
- Given a *unique-id* used within the package
- Provide a fallback for non-EPUB content documents (i.e not SVG, or XHTML)
- Defines the type of media it is…

Manifest

- EPUB 3 adds two new attributes:
  - `media-overlay`
    reference to the media-overlay document
  - `properties`
    used to specify certain properties about a specific resource

The properties means that the reader can make decisions about how to handle some content before opening (e.g. whether it contains scripted content)

```
<item id="p4"
      href="xhtml/4-oldage.xhtml"
      media-type="application/xhtml+xml"
      properties="scripted"/>
<item id="j1_320"
      href="images/1-childhood-320.jpg"
      media-type="image/jpeg"/>
<item id="prt"
      href="images/portrait.jpg"
      media-type="image/jpeg"
      properties="cover-image"/>
<item id="nav"
      href="xhtml/nav.xhtml"
      media-type="application/xhtml+xml"
      properties="nav"/>
```

Properties

- Used to alert the system about certain features being used, e.g.
  - Scripting
  - MathML
  - inline SVG
  - Pretty much all EPUB3 features
- `nav` property – defines the navigation document

Spine

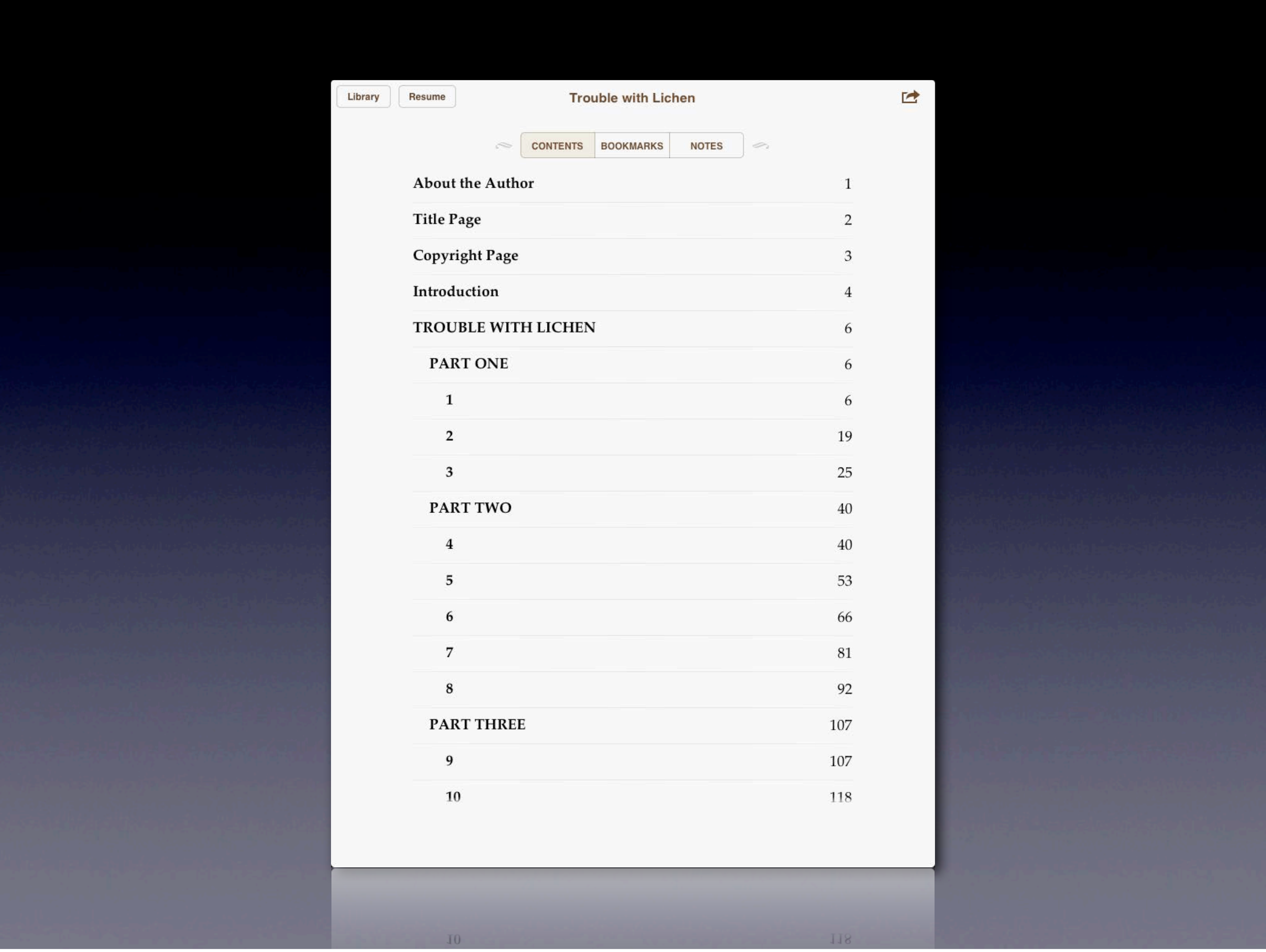
- The Spine acts like the spine of a book
- It orders the various content files
- One after another…
- The `<itemref>` tag used to reference `<items>` defined in the manifest by their `@id`

```
<itemref idref="title" />
<itemref idref="p1" />
<itemref idref="p2" />
<itemref idref="p3" />
<itemref idref="p4" />
```

Even files referenced by content documents should be defined in here

Navigation

- `<spine>` provides the ordering of content
- But it does not help with user navigation
- EPUB allows the creator to specify many navigation methods for the user, such as the Table of Contents
- This *must* be present…



Sample TOC from iBooks on an iPad

Navigation

- Previously EPUB2 used *Navigation Center eXtended* (`ncx`) to define the table of contents
- EPUB3 replaces this with a simpler arrangement using HTML5 `<nav>` element
- Navigation file is also human readable
- Referenced by a `nav` property on the relevant `<item>`
- `ncx` can still be present, but must be ignored by EPUB3 compliant readers

ncx was taken from Daisy, again EPUB3 is trying to standardise on a few key core tech.

EPUB3 <nav>

- EPUB places some restriction on the navigation document
- The <nav> must have an <ol> as its direct child
- A single optional heading may precede it
- Each <li> must contain either a link (<a>) or a

EPUB3 `<nav>`

- Can have multiple navigation types
  - Table of contents, figures etc.
- The `epub:type` attribute is used to distinguish the type. e.g. the table of contents `<nav>` must have an `epub:type="toc"`

```
<nav epub:type="toc" id="toc">
  <h1>Table of contents</h1>
  <ol>
    <li>
      <a href="chap1.xhtml">Chapter 1</a>
      <ol>
        <li>
          <a href="chap1.xhtml#sec-1.1">Chapter 1.1</a>
          <ol hidden="">
            <li>
              <a href="chap1.xhtml#sec-1.1.1">Section 1.1.1</a>
            </li>
            <li>
              <a href="chap1.xhtml#sec-1.1.2">Section 1.1.2</a>
            </li>
          </ol>
        </li>
        <li>
          <a href="chap1.xhtml#sec-1.2">Chapter 1.2</a>
        </li>
      </ol>
    </li>
    <li>
      <a href="chap2.xhtml">Chapter 2</a>
    </li>
  </ol>
</nav>
```

Sample nav file…

Content

- EPUB2 supported HTML and DTBook
  - With SVG inserts, and XML islands
- EPUB3 drops support for XML islands and DTBook
  - Promotes SVG to being a top-level content

Content Documents

- Directly includes:
  - (X)HTML 5 with CSS 2.1, Select CSS 3 & Embedded MathML
  - SVG
- Additions, restrictions and requirements on content formats
- Can include other markup inside of HTML 5 and SVG
- Fallback mechanisms required

HTML 5 — Semantic Inflection

- Semantic inflection by attaching additional meaning to elements
- Richer semantics than just HTML 5
- Extended vocabulary
- `epub:type` attribute

`epub:type` attribute

- Specified on any element
- Cannot change the role (e.g. “p” -> “list-item” disallowed)
- Used to identify subtypes (e.g. “section” is a “chapter”)
- EPUB 3 Structural Semantics Vocabulary

Footnotes

- EPUB3 also uses **`epub:type`** attribute to provide support for pop-up footnotes
- Place a link to the footnote where it is called out using an **`<a>`**
- Use an **`<aside>`** element to define the footnote
- Decorate both using **`epub:type`** to inform the reading system that this is a footnote

```
<p>
   In that
   year<a href="#ft2f" epub:type="noteref">2</a>
   there were 67 mills engaged in the manufacture of
   cotton goods …
</p>

<aside id="ft2f" epub:type="footnote">
   <p>
      2 The manufacturing statistics for 1900 which
      follow are not those given in the Twelfth
      Census, but are taken from the
      <em>Census of Manufactures</em> …
   </p>
</aside>
```

Footnotes
Can also handle rearnotes, etc.

Speech Synthesis

- Speech Synthesis Markup Language (SSML)
  - Utilizes the attributes of the phoneme element
- `ssml:ph` attribute
  - Used to provide a phonetic equivalent to some textual content
    - Cannot be nested within another element with an `ssml:ph` attribute
- `ssml:alphabet`
  - EPUB specified International Phonetic Alphabet (IPA) as required
  - Can identify others, but a reader may not be able to process them

Content Switching

- `epub:switch` element
  - Provides alternative content (fallbacks)
    - Ordered set of choices, from which a reader should choose the first it understands
- Can be used to provide backwards compatibility for EPUB 2 with MathML content

Content Switching

- `epub:case` element
  - Identifies a choice within an epub:switch element
  - Utilizes the required-schema attribute to identify the language
  - Contains the content to be rendered (e.g. `ChemML`)
- `epub:default`
  - Must provide content in a base EPUB 3 type (e.g. XHTML 5, SVG or MathML)
  - Used if the reader cannot process any of the cases
  - Must always be provide for compatibility

```
<epub:switch id="cmlSwitch">

   <epub:case required-namespace="http://www.xml-cml.org/schema">
      <cml xmlns="http://www.xml-cml.org/schema">
         <molecule id="sulfuric-acid">
            <formula id="f1" concise="H 2 S 1 O 4"/>
         </molecule>
      </cml>
   </epub:case>

      <epub:default>
         <p>H<sub>2</sub>SO<sub>4</sub></p>
      </epub:default>

</epub:switch>
```

Using an epub:switch

```
<epub:switch id="mathmlSwitch">

   <epub:case required-namespace="http://www.w3.org/1998/Math/
MathML">
      <math xmlns="http://www.w3.org/1998/Math/MathML">
         <mrow>
            <mn>2</mn>
            <mo> &#x2061;<!--INVISIBLE TIMES--></mo>
            <mi>x</mi>
         </mrow>
         <mrow>
            <mo>+</mo>
            <mi>y</mi>
            <mo>-</mo>
            <mi>z</mi>
         </mrow>
      </math>
   </epub:case>

   <epub:default>
      <p>2x + y - z</p>
   </epub:default>

</epub:switch>
```

Providing mathml fallback for EPUB2.1 documents
Problem -- iPad displays both :)

Triggers

- `epub:trigger` element
  - Declarative mechanism for process actions
  - Primarily defined for media control and content visibility
    - Identifies a target (`ref`) for the action, a source for the event and an event
- `action` attribute
  - *show* and *hide* values for content
  - *play*, *pause*, *resume*, *mute* and *unmute* for media control

Triggers

- `event` and `observer`
  - When an event occurs on an identified observer, the action shall occur on the target
    - For example, a click on an item of content should pause playback
    - The valid event types are those defined by W3C XML Events

```
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:epub="http://www.idpf.org/2007/ops"
  xmlns:ev="http://www.w3.org/2001/xml-events">
  <head>
    <epub:trigger ev:observer="pause" ev:event="click" action="pause"
ref="test"/>
    <epub:trigger ev:observer="resume" ev:event="click" action="resume"
ref="test"/>
    <epub:trigger ev:observer="mute" ev:event="click" action="mute"
ref="test"/>
    <epub:trigger ev:observer="mute" ev:event="click" action="show"
ref="muted"/>
    <epub:trigger ev:observer="unmute" ev:event="click" action="unmute"
ref="test"/>
    <epub:trigger ev:observer="unmute" ev:event="click" action="hide"
ref="muted"/>
  </head>
  <body>
    <video id="test" src="birds.mp4" width="320" height="240"/>
    <p>
      <span class="button" id="resume">Play/Resume</span>
      <span class="button" id="pause">Pause</span>
      <span class="button" id="mute">Mute</span>
      <span class="button" id="unmute">Unmute</span>
      <span id="muted">MUTED</span>
    </p>
  </body>
</html>
```

epub:triggers

Content Restrictions

- MathML has two flavours
  - *Content* – Semantic form
  - *Presentation* – Display form
- EPUB 3 requires *Presentation* MathML
  - Highly prevalent
  - Easier to process
  - Content MathML as an alternative semantic representation
- Disallows MathML 3.0 deprecated elements

SVG

- SVG 1.1
  - Fully conformant document
  - Can be included at the top-level or included in XHTML
  - Should also meet the SVG Accessibility guidelines
  - SVG Animations are disallowed in EPUB 3.0
  - SVG Foreign Objects must contain XHTML valid to the rules of EPUB 3

Scripting

- Scripting is allowed in EPUB 3
  - Container-constrained or spine-level scripts
- Container-constrained
  - Must not modify page layout properties (e.g. size)
  - Must not modify the DOM of the container
- Spine-level
  - If rendered without scripting, document must not alter its basic structure
  - Accessibility considerations
- Should follow WAI-ARIA and WCAG 2.0 guidelines for web accessibility
- Minimal or no scripting is ideal

CSS

- Cascading Style Sheets (CSS) 2.1
  - Complete inclusion except…
  - The `position` property should not use the `fixed` value
  - The `direction` and `unicode-bidi` properties must not be used (use XHTML equivalents)
- CSS 2.0 `list-style-type`
  - `cjk-ideographic`, `hebrew`, `hiragana`, `hiragana-iroha`, `katakana`, `katakana-iroha`
- CSS 3.0 Speech
  - Included using `-epub-` prefix for specific elements `cue`, `pause`, `rest`, `speak`, `speak-as`, `voice-family` (e.g. `-epub-cue`)

CSS Fonts

- CSS Fonts Level 3
  - EPUB 3.0 allows the embedding of fonts using the `@font-face` rule
  - Web Open Font Format (WOFF)
  - OpenType

CSS Text

- CSS Text Level 3
  - Utilizing the **`-epub-`** prefix for many
    - `hyphens`, `line-break`, `text-align-last`, `text-emphasis`, `text-emphasis-color`, `text-emphasis-style` and `word-break`
  - Directly including others such as `text-transform`

CSS Media Queries

- Inclusion of the `@media` and `@import` rules

Miscellaneous

- CSS Namespaces
  - Inclusion of `@namespace` rule
- CSS Multi-Column Layout
  - Entire Multi-Column Layout rules included
    - Exception of column-span property
- Ruby Positioning
  - Addition of the `ruby-position` property (with `-epub-` prefix)
- Display Property Values
  - `oeb-page-head` and `oeb-page-foot` added to the display properties

Media Overlays

- Or ‘read aloud’
- Allows you to synchronize the display of parts of the document to an audio file
- SMIL document with some restrictions

MO SMIL restrictions

- Designed specifically to support read aloud
- Limited to a `<seq>` of audio and text in `<par>`
- Cut down schema specified
- Can specify a CSS style to style the highlighting of content

EPUB 3 Viewing

- Little EPUB3 support
- Only where needed for a specific job
- iPad supports some features

At least in Sept. 2012,

iPad EPUB

- iBooks contains some support for EPUB 3, but incomplete…
- MathML supported, but not `<epub_switch>`
- Worse, some features are only supported when added to *EPUB2* documents…
- Or with restrictions…

Coping with Fragmentation

- iPad a major player for advanced eBooks
- As is Kindle Fire
- But neither support EPUB 3 directly
- Don’t cope with this by coding for each device
- Code pure EPUB3 and convert

EPUB3 and XSLT

- EPUB3 is pure XML and so a prime candidate for XSLT transformation
- Very easy to write scripts that can convert ‘pure’ EPUB3 into iPad or Kindle variants
- Generate NCX from `nav` document
- Can even build everything using XSLT