epubcheck profiles with epubcheck-xproc

Gerrit Imsieke
le-tex publishing services GmbH

Frankfurt Book Fair, Oct. 14, 2011

epubcheck

– Industry standard QA tool

– Java command line application

– Wrapper around standard Relax NG schemas, Schematron, etc.

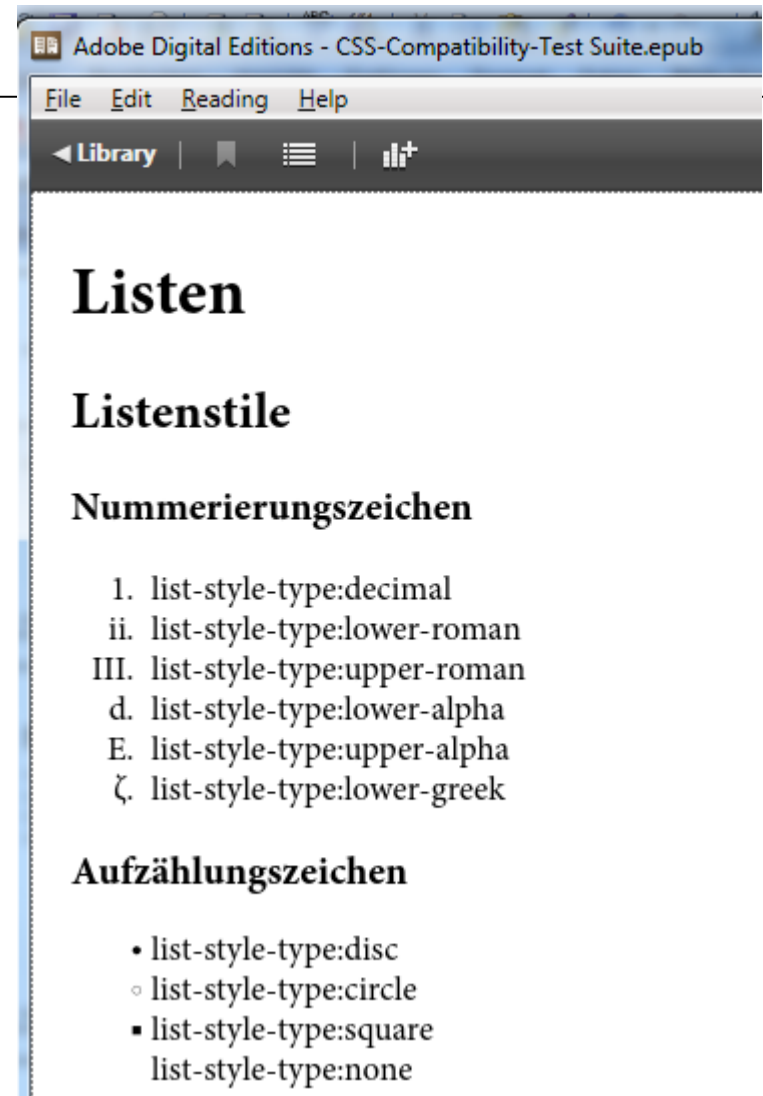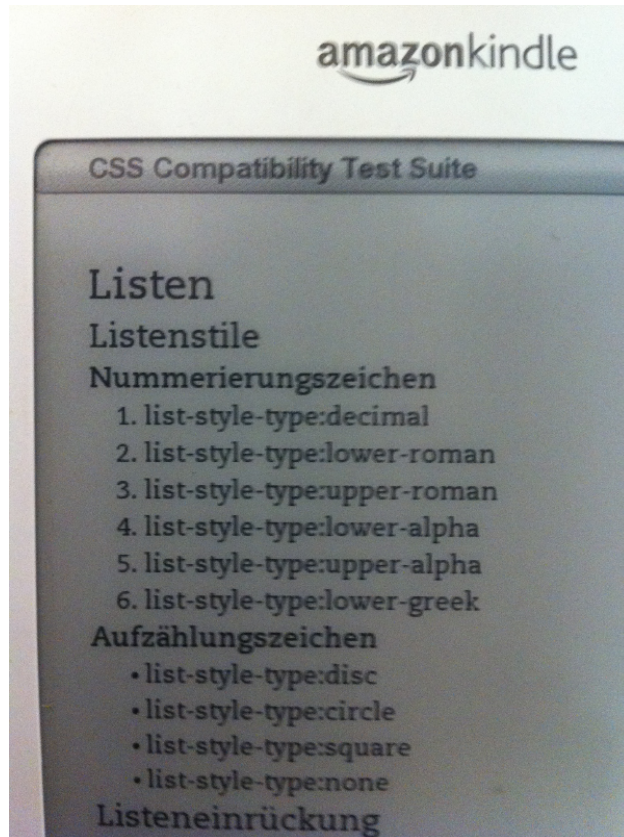– Graphical front end & localizations available

What's wrong with it?

- It will not check for Kindle compatibility
  - To do this, it will have to **parse & check CSS**
- It will flag errors when acceptable (even desired) Apple-specific features are used in EPUB2:
  - Video, audio, scripting, audio overlay
- It doesn't tell you which glyphs won't be rendered on the target platforms
- Wouldn't it be nice if it checked business rules, too?
  - Comparing metadata to publishers' ERP systems
  - Enforcing style guides
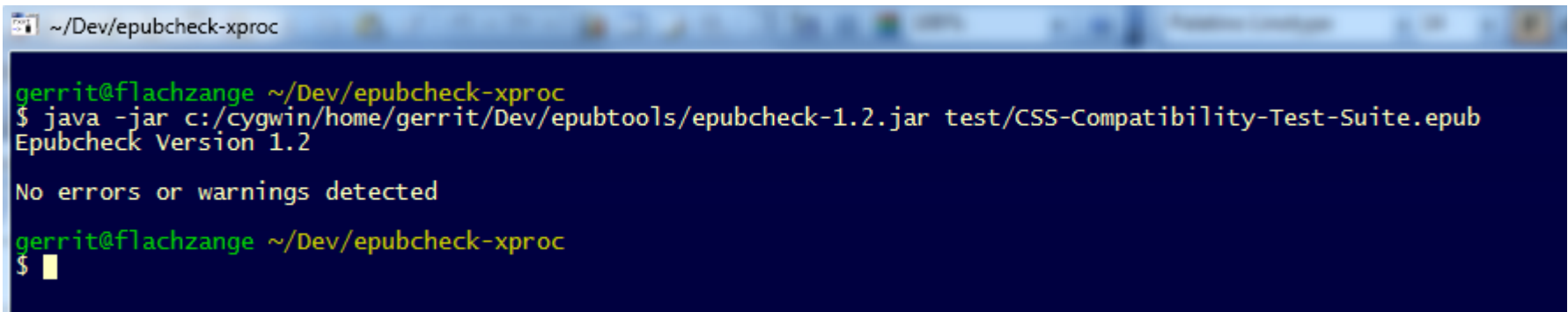- It's poorly extensible

Missing: Kindle compatibility check

– Kindle/Mobipocket format: lousy HTML, CSS, and font support

– Amazon accepts EPUB files and converts them

– You never know whether it will be rendered as intended

Kindle compatibility

epubcheck 1.2: no errors

epubcheck-xproc: errors & warnings related to unsupported CSS
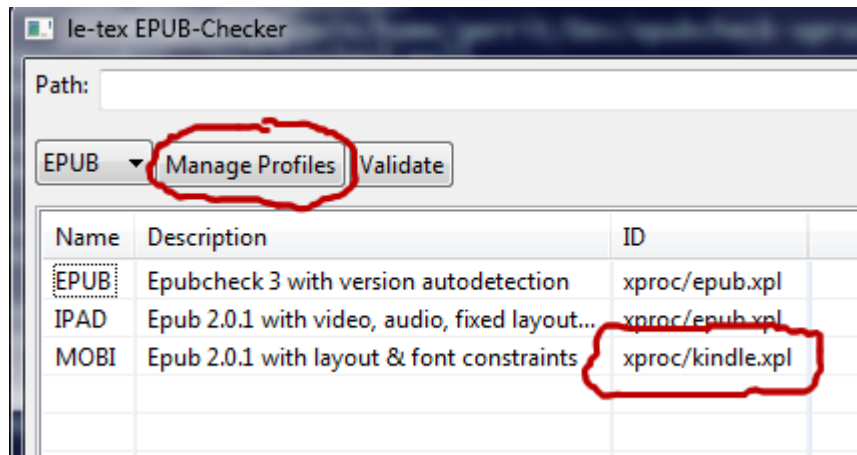
epubcheck-xproc profiles

```
kindle.zip
  manifest.xml
  xproc/kindle.xpl
  sch/kindle.sch
  l10n/kindle_de.xml
```

XProc

```
 2 ▽ <p:pipeline xmlns:p="http://www.w3.org/ns/xproc"
 3       xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 4       xmlns:css="http://www.w3.org/1996/css"
 5       xmlns:html="http://www.w3.org/1999/xhtml"
 6       xmlns:epub="http://www.idpf.org/2007/ops"
 7       xmlns:c="http://www.w3.org/ns/xproc-step"
 8       version="1.0">
 9
10       <p:option name="epubdir" />
11       <p:option name="lang" select="'en'"/>
12
13       <p:import href="epub.xpl" />
14       <p:import href="basic.xpl" />
15
16 ▽    <epub:load-if-exists name="l10n">
17         <p:with-option name="href" select="concat('../l10n/kindle
18       </epub:load-if-exists>
19
20 ▽    <epub:basic name="basic">
21         <p:with-option name="epubdir" select="$epubdir" />
22       </epub:basic>
23       <p:sink/>
```

```
25 ▽    <epub:css-expanded-spinecontent>
26         <p:with-option name="epubdir" select="$epubdir" />
27 ▽      <p:with-option name="opfdir" select="//opfdir">
28           <p:pipe step="basic" port="opfdir"/>
29         </p:with-option>
30 ▽      <p:input port="opf">
31           <p:pipe step="basic" port="opfout" />
32         </p:input>
33       </epub:css-expanded-spinecontent>
34
35 ▽    <epub:schematron-spinehtml name="schematron-spinehtml">
36         <p:with-option name="schematron" select="'../sch/kindle.sch
37 ▽      <p:input port="l10n">
38           <p:pipe step="l10n" port="result" />
39         </p:input>
40       </epub:schematron-spinehtml>
41       <p:sink/>
42
43 ▽    <epub:htmlreport name="htmlreport">
44 ▽      <p:input port="source">
45           <p:pipe step="basic" port="html-report-fragments" />
46           <p:pipe step="schematron-spinehtml" port="result" />
47         </p:input>
48       </epub:htmlreport>
49
50     </p:pipeline>
```

basic epubcheck

**CSS parsing, Kindle compatibility
check (Schematron), HTML report**

```
 2 ▽ s:schema
 3     xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 4     xmlns:html="http://www.w3.org/1999/xhtml"
 5     xmlns:s="http://purl.oclc.org/dsdl/schematron"
 6     >
 7
 8     <s:ns uri="http://www.w3.org/1996/css" prefix="css"/>
 9
10 ▽  <s:pattern id="css-position">
11 ▽    <s:rule context="*[@css:position]">
12 ▽      <s:report test="@css:position = 'fixed'">
13          <s:span class="severity">ERR</s:span>
14          <s:span class="msgid">SCH_Kindle_0016</s:span>
15          position:fixed not supported
16        </s:report>
17 ▽      <s:report test="@css:position = 'absolute'">
18          <s:span class="severity">ERR</s:span>
19          <s:span class="msgid">SCH_Kindle_0017</s:span>
20          position:absolute not supported
21        </s:report>
22      </s:rule>
23    </s:pattern>
24
25 ▽  <!-- *
26          * font rules
27          * -->
28 ▽  <s:pattern id="css-font">
29 ▽    <s:rule context="*[@css:font-family]">
30 ▽      <s:report test="@css:font-family">
31          <s:span class="severity">WRN</s:span>
32          <s:span class="msgid">SCH_Kindle_0001</s:span>
33          font-family not supported (<s:span class="info"><xsl:value-of select="@css:font-family"/></s:span>)
34        </s:report>
35      </s:rule>
36 ▽    <s:rule context="*[@css:font-size][matches(., '^[0-9.]+em$')]">
37 ▽      <s:report test="replace(@css:font-size, 'em$', '') cast as xs:double gt 2">
38          <s:span class="severity">WRN</s:span>
39          <s:span class="msgid">SCH_Kindle_0002</s:span>
40          Font sizes larger than 2em are without effect (<s:span class="info"><xsl:value-of select="@css:font-size"/></s:span>)
41        </s:report>
42      </s:rule>
43 ▽    <s:rule context="*[@css:font-size][matches(., '^[0-9.]+px$')]">
```

Schematron

`kindle.sch`

**17 Rules so far,
about 50 rules needed**

Comparison

	epubcheck	**epubcheck-xproc**
Architecture	Java, Relax NG, NVDL, Schematron	Xproc, Relax NG, NVDL, Schematron, XSLT2
Profiles	hard-coded	configurable
Extensibility	subclass, compile	upload new schemas, rules, transformations, localizations as Zip
Image checks	no	not yet
CSS parser	no	yes
Maturity	high	low (pre-alpha)
License	BSD	BSD

Future

– More checks (everything that epubcheck does)
– Read image resolution / color space information
– Device/font specific character support lists & checks
– iBooks fixed layout / audio overlay profile
– Metadata check against a Klopotek XML export
– Dealing with all kinds of zipped XML formats
  – IDML checker & EPUB converter
  – .docx → IDML converter
  – …

Thank you

Code & binaries:

https://github.com/gimsieke/epubcheck-xproc

@gimsieke