TEX 原稿から EPUB を作りたい

鹿野 桂一郎
k16.shikano@gmail.com
@golden_lucky

2015 年 11 月 7 日
TEX ユーザの集い 2015

EPUBの
なにがうれしいか

EPUB とは

- 電子書籍の標準フォーマットの規格
- 2007 年 9 月に IDPF（国際電子出版フォーラム）International Digital Publishing Forum が策定した
- 2011 年 10 月に現行のメジャーバージョンである EPUB3 が策定された

EPUB には HTML が必要

- EPUB ⊂ HTML を ZIP で固めたもの
  - 実際には、ZIP の中には HTML 以外にどんなメディアをいれてもいい
  - 規格上、テキストとして再生してもらえることになっているのは、HTML と SVG のみ
- レイアウトは CSS のみで指定する。JavaScript は基本的に使えない
- EPUB3 では HTML5 をサポートしているので、MathML が使えることになっている
  - TEX ユーザにとっては朗報ですね

EPUB といえばリフロー

- ユーザの手元で動的に作られる「ページ」
- 古き良き「ページ」というインターフェイスの再発明

EPUB といえばリフロー

- ユーザの手元で動的に作られる「ページ」
- 古き良き「ページ」というインターフェイスの再発明

- ページとその上の文字の物理的な制約が（主にユーザにとって）なくなる
  - 小さい画面、巨大な画面、丸い画面、どんな読書端末で読んでもいい

EPUB といえばリフロー

- ユーザの手元で動的に作られる「ページ」
- 古き良き「ページ」というインターフェイスの再発明

- ページとその上の文字の物理的な制約が（主にユーザにとって）なくなる
  - 小さい画面、巨大な画面、丸い画面、どんな読書端末で読んでもいい

月世界上陸

月世界の探険に於て、一番難所といわれるのは、無引力空間の通過だった。その空間は、丁度地球の引力と月の引力とが同じ強さのところであって、

海野十三『月世界探険記』より

EPUB といえばリフロー

- ユーザの手元で動的に作られる「ページ」
- 古き良き「ページ」というインターフェイスの再発明

- ページとその上の文字の物理的な制約が（主にユーザにとって）なくなる
  - 小さい画面、巨大な画面、丸い画面、どんな読書端末で読んでもいい

も　し
そこでまごまご
していたり、エンジ
ンが止(とま)ったりすると、
そこから先、月の方へ
ゆくこともできず、さ
りとて地球の方へ引
かえすことも出
来　ず

海野十三『月世界探険記』より

EPUB といえばリフロー

- ユーザの手元で動的に作られる「ページ」
- 古き良き「ページ」というインターフェイスの再発明

- ページとその上の文字の物理的な制約が（主にユーザにとって）なくなる
  - 小さい画面、巨大な画面、丸い画面、どんな読書端末で読んでもいい

宙ぶらりんになってしまって、ただもう餓死を待つより外しかたがないという恐ろしい空間帯だった。蜂谷艇長（はちやていちょう）の巧みな指揮が、幸いにエンジン

海野十三『月世界探険記』より

書籍に必要なのはコンテンツとメタ情報

- 大量のメタ情報が、リーダーで表示したり流通に乗せたりするために必要
  - 書名、書名の読み、書名の種類、同一書名で別バージョンの本を区別する情報、シリーズ中の順番、原書名
  - 著者名、著者名の別表記
  - 発行日、改訂日、重刷日
  - 出版社、製作所、コピーライト
  - ISBN とか DOI とか出版社独自の番号のような識別子
- EPUB3 では、作り手がメタ情報を破綻なく拡張もできる仕組み（<meta>要素）がある

文書の構造も一種のメタ情報

- 適切に意味付けされた構造は、アクセシビリティにとって重要
  - そもそも再生に支障をきたす
  - 読み上げとページめくりの同期とか、付加価値のある機能を提供するために必要
  - コンテンツの再利用性は高いほうがいい
- ナビゲーション（論理的な目次）がうまく機能するためには構造が必要

EPUB vs. PDF
≈ 構造とメタ情報 vs. 見た目

- **EPUB は、多くの読者にとって、見た目と手軽さでは PDF に劣る**
- **流通やアクセシビリティを考えると、書籍としてのメリットが出始めている**

EPUB vs. PDF
≈ 構造とメタ情報 vs. 見た目

- EPUB は、多くの読者にとって、見た目と手軽さでは PDF に劣る
- 流通やアクセシビリティを考えると、書籍としてのメリットが出始めている

- ただし PDF/UA（もしくはタグ付き PDF）のような規格もあるので「PDF はアクセシビリティだめ」というわけではない

TeXをHTMLに変換したい

TeX → HTML 変換の要件

- EPUB の仕様にかなった HTML になること
  - XML 版の HTML5 であること
  - 要素の id 属性や図に重複がないこと
- 本の構造にかなった HTML になること
  - 本の中の位置に見合った連番
  - 相互参照
- 手作業で HTML を編集しないで済むこと
  - TeX 原稿から完全に機械的に変換したい

数式について

- EPUB3 では MathML（Presentational）がサポートされているので、MathML に変換すればよい
- Kindle でも出したかったら、SVG の画像にしておくのがベストプラクティス
- MathJax のことは忘れましょう
  - JavaScript が動く EPUB3 リーダーはいまのところ一般的ではない
  - 動いても、ネットワークもしくは mathjax パッケージと数式用フォントの埋め込みが必要で、専用リーダーには非現実的

数式について

- EPUB3 では MathML（Presentational）がサポートされているので、MathML に変換すればよい
- Kindle でも出したかったら、SVG の画像にしておくのがベストプラクティス
- MathJax のことは忘れましょう
  - JavaScript が動く EPUB3 リーダーはいまのところ一般的ではない
  - 動いても、ネットワークもしくは mathjax パッケージと数式用フォントの埋め込みが必要で、専用リーダーには非現実的
- いずれにせよ、ツールでなんとかなる時代になっています

TeX から HTML を手に入れる方法まとめ

1. テキストフィルタ型

- TeX 原稿をテキストとしてパーズし、HTML として出力
- Pandoc、LaTeX2html など
- 気軽に使えるが、拡張性はほぼない

2. TeX エミュレート型

- TeX の処理（トークンを読み込んで箱を並べる）を模倣
- LaTeXml、plasTeX、HeVeA など
- 良好な結果がえられるが、独自の TeX マクロなどは自力で拡張が必要

3. DVI ウェア型

- LaTeX に DVI を作らせて、それを HTML にする
- TeX4ht など
- ほぼ無敵（pTeX を除く）だが、文書の構造が取れるわけではない

TeX から HTML を手に入れる方法まとめ

1. テキストフィルタ型

- TeX 原稿をテキストとしてパーズし、HTML として出力
- Pandoc、LaTeX2html など
- 気軽に使えるが、拡張性はほぼない

2. TeX エミュレート型

- TeX の処理（トークンを読み込んで箱を並べる）を模倣
- LaTeXml、plasTeX、HeVeA など
- 良好な結果がえられるが、独自の TeX マクロなどは自力で拡張が必要

3. DVI ウェア型

- LaTeX に DVI を作らせて、それを HTML にする
- TeX4ht など
- ほぼ無敵（pTeX を除く）だが、文書の構造が取れるわけではない

LaTeXml

- TeX の消化の仕組みを Perl で模倣し、出力ルーチンの代わりに DOM を組み立てて生の XML を吐き出すイメージ
- 吐き出した XML を XSLT で後処理して使う
- 独自の TeX マクロや LaTeX コマンド・環境に対する処理を、Perl のモジュールを書いて追加できる

```
DefMacro('\mybold{}','\textbf{#1}');
```

LaTeXml（つづき）

- 精力的に開発されているようで、昨年の時点で TikZ サポート率では TeX4ht を抜いたらしい
- 2014 年 5 月には、直接 EPUB を作ることも可能になっている

```
$ latexml --inputencoding=utf8 \
          --dest=test.xhtml test.tex
$ latexmlpost --format=epub test.xhtml
```

TeX4ht

- プロ向けにはよく使われているらしい
- 基本的な仕組みは、
  1. HTML 構造用のヒントを\special で埋め込んだ特別な dvi を作るパッケージを読み込む

```
\documentclass{jsbook}
...
\usepackage[xhtml,mathml,charset=utf-8]{tex4ht}
...
\begin{document}
```

  2. 生成された特殊な dvi を、tex4ht というコマンドで処理すると、HTML ができる
  3. さらに t4ht というコマンドで処理することで CSS を作る
- tex4ht コマンドは、platex で処理された dvi（に指定されている日本語用の jfm）を読めない！

TeX4ht を日本語で使う方法、その1

- **pTeX を使わなければいい**
- **TeX4ebook というパッケージの機能を利用して fontspec を使う**

```
\documentclass{book} % jbook/jsbook は NG
\usepackage{alternative4ht}
\altusepackage{fontspec}
\altusepackage{xeCJK}
\altusepackage{xunicode}
\setCJKmainfont{IPAMincho}
...
\begin{document}
```

- **実行には LuaLaTeX が必要（-l オプション）**

```
$ make4ht -l book.tex
```

TEX4ht を日本語で使う方法、その 2

- 1 つめの方法だと pTEX のプリミティブが封じられる
- platex でコンパイルした dvi を tex4ht に読ませる手段はないか？
- tex4ht が読めない pTEX 由来の dvi 命令 set2 #N を、set_char_#n 命令に変換できないか？
- 実は jTEX は set_char_#n 命令だけで日本語の文字を印字している！

pTEXのdviをjTEXのdviに変換

- 必要なもの：
  - dvi2dvi
  - dvi2dvi が使う仮想フォント（Debian なら dvi2ps-fontdata-a2n）
  - jtex 用の (dgj|dmj)*.tfm（Debian なら jtex-base）
  - tex4ht が使う、(dgj|dmj)*.tfm から HTML 用の文字への対応表（このチートを開発した行木孝夫先生がむかし作ったものが W32TEX に同梱されている）
- 実行手順：

```
$ platex book.dvi
$ dvi2dvi -F a2n -S book.dvi > book-ntt.dvi
$ tex4ht -i~/texmf/tex4ht/ht-fonts/ja/dnp \
         -cunihtf -utf8 book-ntt.dvi
```

- （ひょっとして W32TEX ではこれを自動でやってくれる？）

HTMLをEPUBにする

CalibreかPandocでHTML → EPUB

- 電子書籍管理アプリケーションの **Calibre** には、各種フォーマットの変換コマンド ebook-convert が用意されている

```
$ ebook-convert --extra-css=book.css \
                book.html book.epub
```

- ただし、**Calibre** は **EPUB2** しか生成できないので、いまは **Pandoc** を使うのがよさそう

```
$ pandoc -t epub3 -o book.epub \
         --epub-stylesheet=book.css \
         book.html
```

- 実際にはこんなオプションでは済まない！

自力でやるのも現実的

- HTML をかき集めてメタ情報を用意すればいいので、難しくはない
- 実際に自作して使っている
  - **qnda**（https://github.com/k16shikano/qnda）
  - LaTeX が自動生成する情報（\ref とか連番）を生成するのが面倒
  - それでも、Calibre や Pandoc のオプションを調べるより楽だと思う（個人差があります）
- いずれにせよ epubcheck を忘れずに

```
$ java -jar epubcheck.jar book.epub
```

TEXユーザがEPUBを気にする必要はあるのか？

TEX はいかにもリフローと相性が悪そう

- **EPUB は、デバイスに非依存で表示を保証することを考えていないメディア**
- **DVI は、デバイスに非依存で表示を保証することを目指していたメディア**

LaTeX で書かないという選択（まとめのようなもの）

- EPUB が必要なら、LaTeX をソースにすることをあきらめる
  - PDF を作るつもりで書いた原稿を EPUB にしたい、というのは倒錯では？
- EPUB にしやすい LaTeX のサブセットを作る？
  - TeX っぽいシンタックスで EPUB を作りたい、というのは、やっぱり倒錯なような
- タグ付き PDF や固定レイアウトでいいのかもしれない

参考資料

- **EPUB について**
  - **IDPF** http://idpf.org/epub
  - **Garrish & Gylling “EPUB 3 Best Practices”, O’Reilly, 2013**
  - **epubcheck** https://github.com/idpf/epubcheck
- **PDF/UA について**
  - **“PDA/UA in a Nutshell”** http://www.pdfa.org/wp-content/uploads/2013/08/PDFUA-in-a-Nutshell-PDFUA.pdf

参考資料（つづき）

- 変換ツール
  - **Pandoc** http://pandoc.org/
  - **LaTeXml** http://dlmf.nist.gov/LaTeXML/ (https://github.com/brucemiller/LaTeXML)
  - **plasTeX** http://tiarno.github.io/plastex/
  - **HeVeA** http://hevea.inria.fr/
  - **TeX4ht** https://www.tug.org/tex4ht/
  - **TeX4ebook** https://www.ctan.org/pkg/tex4ebook (https://github.com/michal-h21/tex4ebook)
  - **ebook-convert（Calibre）** http://manual.calibre-ebook.com/cli/ebook-convert.html