

Municodeプロジェクト

地方自治体の条例文書の 印刷用CSS

Eliot Kimber氏による
ケーススタディ

目次

1. プロジェクト紹介
2. フェーズ1: 概念証明 (Proof of Concept)
3. フェーズ2: 出版物の完全実装
4. フェーズ3: ルーズリーフ ソリューション
5. 処理パイプライン
6. 結論とまとめ

1. プロジェクト紹介

Municode プロジェクト

地方自治体の条例を出版する会社
米国の4,100以上の市区町村の規定条例を発行
3,000以上の出版物を管理(平均800ページ)
およそ80人で月5万ページのPDFを作成
現在XPPエンジンを使用
(SDL XML Professional Publisher)
印刷物とHTMLを両方配信
ソースはSGMLマークアップとXPP独自の組版
コードの組み合わせ

Municode プロジェクト

レガシーな文書をHTML5に変換する処理
HTMLをWeb配信に使っている
このHTMLはまた新CSSベースのシステムへの入
力ともなる

2014 – CIOがAntenna Houseに初めてコンタクト

Loose Leafはサポートしていないと返答

2016/12 – AH CSS Formatter評価版をダウンロード

2017/1 – Eliot Kimber氏をコンサルタントに採用

システムの交換の時期

- X P P (S D L Professional Publisher) が古く、維持が困難。
- X P P コードを扱う人を見つけるのが困難。
- X P P の年間保守料金が\$33,000と高い。
- 実際には自動化されたl o o s e leaf機能を使っているわけではない
- 標準仕様を使いたい。C S S が良いのではないか。
- X P P は将来の出版物には向いていない。

Loose leaf 要件

- 変更部分だけを地方自治体が提出、Municodeは変更したページのみアップデート。
- 変更したページを既存のページの間ポイントページとして追加する。
 - ページ付けは執筆者によって手作業で決定
 - XPPベースのシステムがポイントページを生成

2つのページレイアウトが必要

- レガシースタイル
 - 従来作成してきた歴史のある条例文書を反映
- 新しい未来型のスタイル
 - モダンなデザインを反映
- 出版物独自の組版ルールに対応しなければならない
 - XPPベースのシステムではインラインの組版コードで行った
 - 新しいシステムではCSSスタイル、Class やdata属性で行う

レガシー VS 新しい未来型

§ 1-74

ST. LUCIE COUNTY CODE

(c) If the code inspector has reason to believe a violation or the condition causing the violation presents a serious threat to the public health, safety, and welfare, or if the violation is irreparable or irreversible in nature, the code inspector shall make a reasonable effort to notify the violator and may immediately notify the enforcement board and request a hearing.

(d) If a repeat violation is found, the code inspector shall notify the violator but is not required to give the violator a reasonable time to correct the violation. The code inspector, upon notifying the violator of a repeat violation, shall notify the code enforcement board and request a hearing. The enforcement board, through its clerical staff, shall schedule a hearing and shall provide notice to said violator pursuant to section 1-80. The case may be presented to the enforcement board even if the repeat violation has been corrected prior to the board hearing, and the notice shall so state. If the repeat violation has been corrected, the enforcement board retains the right to schedule a hearing to determine costs and impose the payment of reasonable enforcement fees upon the repeat violator. The repeat violator may choose to waive his rights to this hearing and pay said costs as determined by the enforcement board.

(e) If the owner of property which is subject to an enforcement proceeding before an enforcement board, special magistrate, or court transfers ownership of such property between the time the initial pleading was served and the time of the hearing, such owner shall:

- (1) Disclose, in writing, the existence and the nature of the proceeding to the prospective transferee.
- (2) Deliver to the prospective transferee a copy of the pleadings, notices, and other materials relating to the code enforcement proceeding received by the transferor.
- (3) Disclose, in writing, to the prospective transferee that the new owner will be responsible for compliance with the applicable code and with orders issued in the code enforcement proceeding.
- (4) File a notice with the code enforcement official of the transfer of the property, with the identity and address of the new owner and copies of the disclosures made to the new owner, within five days after the date of the transfer.

A failure to make the disclosures as provided in subsections (e)(1), (2), and (3) of this section before the transfer creates a rebuttable presumption of fraud. If the property is transferred before the hearing, the proceeding shall not be dismissed, but the new owner shall be provided a reasonable period of time to correct the violation before the hearing is held.

(Code 1982, § 1-2-22; Ord. No. 87-19, pt. A, 6-2-1987; Ord. No. 87-71, pt. A, 10-13-1987; Ord. No. 89-63, pt. A, 9-26-1989; Ord. No. 03-11, pt. A, 8-19-2003)

State law reference(s)—Similar provisions, F.S. 162.06.

Sec. 1-75. Calling and conduct of hearings.

(a) Upon the request of the code inspector, or at such other times as may be necessary, the chairperson of the enforcement board may call hearings of the enforcement board. Hearings

Chapter 8 Multi-Line Page Headers Example

Article II. Alternate text for "Authorities, Boards and Commissions"

Division 1. Alternate text for "Airport Authority"

Sec. 3. Alternate text for "Membership."

luctus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ut sodales est, ut gravida turpis. Aenean vestibulum nunc eu enim vestibulum, vel lobortis mi vulputate. Praesent cursus pharetra elit, in consequat magna auctor a. Quisque sodales nisl non malesuada posuere. In volutpat turpis et risus scelerisque, pellentesque ullamcorper tellus tincidunt. Pellentesque gravida tristique dui, vel rhoncus dolor sodales ac.

Sec. 4. Meetings.

Sed congue lectus orci, ut faucibus tortor pulvinar non. Proin eleifend massa vel nibh aliquet vulputate. Nullam ut imperdiet metus, a fringilla ex. Vivamus imperdiet, arcu commodo eleifend sollicitudin, velit enim tristique lectus, et ultricies dolor sem a ante. Curabitur a sodales lectus. Sed faucibus, neque quis semper dapibus, est ligula ultrices nulla, sed blandit orci justo bibendum libero. Maecenas sit amet justo justo. Nam nisl purus, eleifend vitae tempor molestie, mollis sed massa. Aliquam pulvinar consequat orci, ut malesuada augue scelerisque eget. Quisque molestie felis in ornare faucibus. Vestibulum ut massa at odio malesuada porta vitae at sapien. Phasellus tempor condimentum urna, vel ultrices odio mattis eu.

Aliquam in massa ac orci pellentesque hendrerit ac mollis ligula. In a dui quis velit bibendum venenatis sit amet sed augue. Cras sagittis luctus pellentesque. Integer vel malesuada nulla. Morbi elementum, nulla bibendum condimentum pretium, elit mi rhoncus neque, vel commodo nunc metus eget est. Etiam scelerisque, ipsum sed tempus blandit, lorem mauris fermentum elit, non gravida nibh sapien dictum sem. Proin eros purus, sodales id enim pharetra, rhoncus facilisis arcu. Nunc eget tortor vel leo fringilla aliquet non tincidunt nisi. Vestibulum lectus lorem, elementum aliquam consequat non, malesuada malesuada est. Morbi in porta nisi. Ut ornare posuere aliquet. Etiam in lacinia nisi.

Sec. 5. Vacancies in office.

Duis vel cursus mauris, eget sollicitudin velit. Phasellus lorem nisl, porttitor et magna a, consectetur semper magna. Proin tincidunt mi sed turpis mollis, sit amet finibus purus hendrerit. In eget fermentum nibh. Cras nibh magna, elementum sed augue a, consectetur molestie nunc. Ut vitae libero turpis. Duis sed semper orci. Nulla facilisi. Mauris tristique magna tincidunt convallis aliquam. Etiam tortor libero, bibendum et tincidunt tempor, finibus quis enim. Donec pellentesque ac nibh consectetur auctor. Praesent pharetra massa libero, id semper urna mollis ac. Ut condimentum id tellus vitae tempus. Nulla ut mauris sed neque imperdiet varius. Nulla quis auctor nisi. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Nullam vehicula gravida justo, nec porttitor libero euismod quis. Fusce dictum erat justo, quis sollicitudin neque posuere et. Ut elit orci, porttitor vitae mollis vitae, sollicitudin vel lectus. Vestibulum maximus orci eu orci dignissim semper. Donec elementum sapien in ligula dictum, sit amet dapibus nibh feugiat. Quisque et commodo mauris. Proin velit velit, finibus quis neque et, tristique molestie mi. Aenean quis hendrerit lorem. Interdum et malesuada fames ac ante ipsum primis in faucibus. Fusce sit amet quam metus.

Vivamus egestas faucibus neque, eget lacinia nisi dignissim nec. Phasellus vel elit ac lectus tempus imperdiet sed nec mi. Pellentesque faucibus cursus lacus, non ullamcorper metus posuere sed. Donec fermentum enim ut massa bibendum, ac porttitor neque mattis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque habitant morbi tristique senectus et netus et

地方自治体条例文書の必要条件

- 複雑な表(長大な表も含む)
- 多段組み文書のインデックス
- 複数ページにまたがる脚注
- 複雑な欄外表題(ランニングタイトル)・レベル6～7の見出し対応
- インデックス用の“continued”タイトル
- 目次
- 異なるサイズの画像
- 深く入れ子になったリスト
- 回転ページ
- 折り込みページ
- 縦方向に中央揃えのコンテンツ 例えばタイトルページなど
- ページ数、画像数、など 組版文書内容を反映したサマリーページ

プロジェクトの全体目標

- 既存のレガシー・フォーマットを可能な限り複製することで、条例文書のページ付けを有効にする
- 新しい未来型のレイアウト設計を使用して条例文書のページ付けを有効にする
- ポイント・ページを入れた変更パッケージの生成を可能にする

プロジェクトは3つのフェーズから成る

フェーズ1. CSSとAntenna House Formatterが動作することを確認する概念の証明

フェーズ2. 完全な印刷出版物の制作

フェーズ3. ルーズリーフ変更パッケージプリント制作の導入

- 2017年6月でフェーズ1は完了、フェーズ2はほぼ完了。
- 2017年8月フェーズ3がスタート

2 . フェーズ1 : 概念実証 (Proof of Concept)

概念を実証する

- CSSレンダリングとAntenna House Formatterで、このプロジェクトはうまくいくのか
- 要件を満たせるのか
 - ページ付け
 - 複雑な表
 - 回転ページ
 - 特定フォントの使用
 - 欄外タイトル
 - ページ番号生成 など

概念実証で発見したこと

1. CSS仕様は不完全で不安定 だが...
2. Antenna House Formatterの拡張でほぼレイアウト要件を満たす
3. 既存のHTMLソースはCSSベースのページ付け用に生成しなくてはならない
4. CSSはXSL-FOより扱いが簡単
5. AH Formatterのエリアツリー出力はサマリーページ、ポイントページの生成を可能にする
6. CSSのデバッグはAH Formatterの課題

概念実証で証明できたこと

- Antenna House Formatterを使って、クライアントの課題の多いページ制作要件の実装は可能であった。
- 結果として、システムは、クライアント社内の人材と知識を活用して、クライアントによって維持が可能であるという確信を提供

3 . フェーズ2 : 出版物の 完全導入

出版物を完全に新システムに導入する

- このフェーズはほぼ完了
- 試行錯誤を経て、元のXPPシステムとほぼ正確なページの忠誠度を達成
- CSSで作成されたページのページ区切りは、XPPで作成されたページのページ区切りとほぼ同じ
- 完成度はクライアントの要求と期待を超えた

Antenna House のサポート

- バグ、特定機能の追加をサポートに依頼
 - ほとんどが迅速に修正、次期バージョンで対応
- 唯一、回避策を使うケースがあった
 - 脚注番号に記号を使いたい
 - Antenna House エリアツリーの後処理を行う必要があった
 - それさえも難しいことではなかった
 - Antenna House がエリアツリーを提供しているという
事実は、回避策が可能であったことを意味する
- この機能はAntenna House Formatter次期バージョンで対応

脚注番号に記号を使う回避策の例

- C S S仕様では、記号は番号振りに定義できる。
Municodeは“*,†,‡”を脚注番号に使いたい。
- X S L Tを使ってエリアツリーの後処理で脚注番号を記号に変換
- エリアツリーは組版文書のXML表現なので、組版ページの
詳細をすべて含んでいる
- XMLなのでX S L Tを使って最終P D F生成の入力として使う
前に変換が可能

脚注番号に記号を使う回避策の例

- 脚注の元のエリアツリーマークアップは次の様に表す

```
<FootnoteArea ...>
  <BlockArea ...>
    <LineArea ...>
      <InlineArea ...>
        <TextArea ...
          text="1"
        /> ...
```
- エリアツリー文書でページにどの脚注が含まれているかわかる
- エリアツリー内で数字(text="1") を記号(text="*")に変えることが可能

脚注番号に記号を使う回避策の例

- XSLTを使う

```
<xsl:template match="at:FootnoteArea/at:BlockArea[1]/at:LineArea[1]/  
                    at:InlineArea[1]/at:TextArea[1]/@text">  
  <xsl:attribute name="{name(.)}"  
    select="at:getFootnoteSymbol(ancestor::at:FootnoteArea[1])"  
  />  
</xsl:template>
```

- at:getFootnoteSymbol() 関数が番号を記号に変えるローカル関数
- あともう少し複雑なテンプレートも使う
- 結果は。。。。

脚注番号に記号を使う回避策の例

Secs. 1-12—1-40. Reserved.

ARTICLE II. CODE ENFORCEMENT^{*}

Division 1. Generally

Secs. 1-41—1-70. Reserved.

Division 2. Code Enforcement Board[†]

at: getFootnoteSymbol() function

Sec. 1-71. Creation.

^{*} **State law reference(s)**—Code enforcement, F.S. ch. 16. blah blah blah blah blah blah blah blah
blah blah blah blah blah blah blah blah blah blah blah blah blah

[†] **State law reference(s)**—Code enforcement boards, F.S. § 162.01 et seq.

エリアツリーとは？

『XSL-FOの基礎』 第5章 page 43を参照

XSL-FO プロセサの組版処理ではFO からページ上のエリアという長方形領域を生成する。エリアにはその中の文字を表示したり、図形やイメージを配置するためのフォント情報、位置情報、間隔・空き情報などの情報が付属している。FO ツリーはエリアの木構造であるエリアツリーに変換される。段落のFO が複数のページに跨るような場合、一つのFO から複数のエリアが作られる。表のように階層化されているFO から作られるエリアは階層化する。エリアツリーにおけるエリアの順序は、原則としてFO ツリーの順序になる。フーノートや脚注などは例外で、FO ツリーの順序とは出力位置が変わる。XSL-FO プロセサはエリアツリーを画面に表示したり、印刷したり、PDF にする。エリアモデルはXSL-FO 仕様書に抽象的な概念として記述されており、仕様の一部として標準化されているわけではない。XSL-FO プロセサのベンダーは各社独自にエリアツリーを実装している。AH Formatter のエリアツリーに関する情報は次にある。

- <https://antennahouse.github.io/AreaTree/ja/>

課題と向き合う

- 最大の課題はページマスターを定義し管理する方法を理解すること
 - CSS仕様は現在、ページマスターを定義し適用するのに必要な機能と関連していない
 - CSSのモデルはXSL-FOのモデルとかなり異なる
- 欄外表題ヘッダフッタに動的コンテンツを配置するのにCSSメカニズムには限界がある。
 - HTMLに欄外表題を作る要素を追加する前処理で対応

ページマスタの例

```
@page portrait:right {
```

```
    size: 8.5in 11in;  
    margin-left: 7.5pc;  
    margin-right: 7.5pc;  
    counter-increment: myPage;  
    counter-reset: footnote;
```

```
@footnote {
```

```
    width: 100%;  
    border-top: 0.5pt solid black;  
    margin-top: 0.5in;  
    border-length: 0.5in;  
}
```

```
@top-left {
```

```
    content: '¥00a0';  
    margin-top: 0.25in;  
    margin-bottom: 2pc;  
}
```

```
@top-center {
```

```
    content: string(chapterTitle);  
    vertical-align: bottom;  
    margin-top: 0.25in;  
    margin-bottom: 2pc;  
    text-transform: uppercase;  
}
```

```
@top-right {
```

```
    content: element(sectionNumber last);  
    vertical-align: bottom;  
    text-align: right;  
    margin-top: 0.25in;  
    margin-bottom: 2pc;  
}
```

```
@bottom-center {
```

```
    content: string(prefolio, first) counter(myPage)  
    string(postfolio );  
    margin-top: 1pc;  
    vertical-align: top;  
    font-family: "New Century Schoolbook", serif;  
    font-size: 10pt;  
}
```

```
@bottom-right {
```

```
    content: '¥00a0';  
    margin-top: 1pc;  
    vertical-align: top;  
    font-family: "New Century Schoolbook", serif;  
    font-size: 10pt;  
}
```

```
}
```



ページマスターの例

- ページマスターは要素と関連している

```
section.chapter {
  break-before: right;
  page: portrait-first portrait;
}
```
- CSSは、ページシーケンスマスターを定義するためのXSL-FOほどのレベルを持っていないが、ほとんどのドキュメントで十分。
- Municodeのドキュメントでは、要素の@class値を使用して特定のページマスタを要求できる
- <div>要素の@classを使用して、セクションの任意のポイントに回転ページや折り込みページ作成

ページマスターとは？

『XSL-FOの基礎』 第3章 第4章 page 23 ~ を参照

ページマスターはページの雛形を設定する。ページシーケンス(3.3 ページシーケンスを参照)は子孫としてテキスト・表・画像などドキュメントの内容を含むFOである。XSL-FOプロセサによる組版では、ページシーケンスの内容をページマスターに流し込んで組版済みのページを生成する。ページマスターには 単純ページマスター(本章で解説する)と ページシーケンス・マスター(4.3 ページシーケンス・マスターを参照)がある。

続く。。。。

欄外表題に動的コンテンツの例

- セクションタイトルとセクションメタデータの一部を欄外のヘッダフッタに入れる必要があった
- CSSは要素をメインフローからマージン領域に移動するために`running()`関数を使用。
- メインフローとマージン領域の両方に要素を入れることができない。
- マージン領域を使うための新しい要素を作成する前処理をXSLT変換で行う

欄外表題に動的コンテンツの例

- 元のマークアップ

```
<section id="LEGACY_x8813D1AE308E" class="section" data-type="sect2">
<header>
  <h2 id="LEGACY_x8813D1AE30A0">
    Sec. 1-1.
  </h2>
  <p id="LEGACY_x8813D1AE30AA" data-type="subtitle">
    How Code designated and cited.
  </p>
</header>
```

- 前処理が終わった後のマークアップ

```
<section class="section" data-type="sect2" id="LEGACY_x8813D1AE308E">
<header>
  <sectionNumber> § 1-1</sectionNumber>
  <h2 id="LEGACY_x8813D1AE30A0">Sec. 1-1.</h2>
  <p data-type="subtitle" id="LEGACY_x8813D1AE30AA">How Code designated and cited.</p>
</header>
```

セクション番号を含む<sectionNumber>要素が追加された。このセクション番号は、<h2>要素のテキストから作成される。

欄外表題に動的コンテンツの例

- CSSでは、positionプロパティと共にrunning()関数を使用して要素変数を作成。
header > sectionNumber {
position: running(sectionNumber);
}
- ページルール定義では、セクション番号を表示する必要がある各ページに、次のようなマージン領域定義がある。

```
@page portrait:right {  
  ...  
  @top-right {  
    content: element(sectionNumber last);  
    vertical-align: bottom;  
    text-align: right;  
    margin-top: 0.25in;  
    margin-bottom: 2pc;  
  }  
}
```

- このページでは、sectionNumber変数の最後の値が、右上のページ余白領域の内容として使用される

課題と向き合う

- CSSに組み込まれたパラメータ化メカニズムがないので、スタイルシートを維持するのが難しい
 - CSSの最近の仕様でパラメータ化に対応を知る
The specification CSS Custom Properties for Cascading Variables (<https://www.w3.org/TR/css-variables-1/>)
 - Antenna Houseに対応リクエストを出した
 - LessのようなCSSをパラメータ化する言語を使うか？
 - それに伴うフラストレーションの方が大きい
 - 手作業で対応した。対応ボリュームとスタイルシートがそれほど大きくなかったので問題は深刻ではなかった

4 . フェーズ3 : ルーズリーフ ソリューション

ルーズリーフでページ変更

- フェーズ3はこれからスタート
- エリアツリーを使って、エディターによるマークアップで対応できる自信がある。
- ソースXMLでエディターが次を定義する
 - 変更の始まりと終わり時点
 - 変更の後に始まるページ番号
 - ポイントページ番号の組み立て方の詳細
- この情報で変更の範囲の一連のページを決定し、新しいページ番号を計算することが可能になる
- システムは最終変更したパッケージを生成するエリアツリーを生成する
- 将来的に手作業部分を自動化するため、ソースコンテンツの要素をページ番号とマッピングすることを計画中

5. 処理パイプライン

XMLからPDFへのパイプライン

1. ソースXHTMLを、CSSが必要とする機能を追加した「拡張」XHTMLに変換
 - 目次、欄外表題ヘッダフッタの要素、および追加のclass属性値
(たとえば、要素の組版が続く兄弟要素の有無に基く場合、
CSSセレクトでテストできない条件の場合)
 - この変換は比較的単純なXHTMLからXHTMLへの変換であり、
複雑なXSLT処理は含まれない。
2. 最初のAntenna House Formatterパスを実行して、初期エリアツリーを生成
3. 初期エリアツリーを更新したエリアツリーに変換
4. 更新したエリアツリーに対して2番目のAntenna House Formatterパスを実行して、最終ページをPDFとして生成

6 . 結論とまとめ

結論とまとめ

- 1 . XHTML+CSS+Antenna Houseで自動ページ化システムが実現する事をこのプロジェクトが証明した。
- 2 . CSSは開発の観点から見てXSL-FOより簡単。長い間XSL-FOの開発者であったがXML-to-Printのアプリ開発をCSSで使う方が好き。
- 3 . XHTMLを入力とすることがCSS開発をより簡単にする。
ソースがDITA, Docbook, JATSであってもそれをXHTMLにまず変換
XML言語はHTML変換のパイプラインをすでに持っている
このアプローチでCSS開発者はXMLを詳しく理解しなくて済む
XSLT変換は必要だがそれは同等のXML-to-XSL-FO変換より楽
開発とメンテナンスコストを削減できる

しかしながら

- 4．2017年7月時点でCSSのページ化仕様は、開発途上で正確さを欠き、完成にいたるところではない。
- 5．しかし、Antenna Houseの拡張機能と組み合わせれば、ほとんどの出版物の要求仕様を十分に満たす。
- 6．CSSの使いやすさ、しいてはCSS開発者の集めやすさのほうが、未完成なCSSページ化仕様がもたらすリスクを負うよりはるかに勝る。
- 7．Antenna Houseは常にCSS実装に関して迅速なバグ対応機能追加リクエスト対応を行う。決して裏切らない。

プロジェクトはまだ続く

プロジェクトはまだ途中である。

今年後半まで続くが、うまく終わることができるかと確信している。