

An XML model of CSS3 as an X_YTeX-TeXML-HTML5 stylesheet language

S. Sankar, S. Mahalakshmi and L. Ganesh

TNQ Books and Journals
Chennai

Abstract

HTML5[1] and CSS3[2] are popular languages of choice for Web development. However, HTML and CSS are prone to errors and difficult to port, so we propose an XML version of CSS that can be used as a standard for creating stylesheets and templates across different platforms and pagination systems. X_YTeX[3] and TeXML[4] are some examples of XML that are close in spirit to TeX that can benefit from such an approach. Modern TeX systems like XeTeX and LuaTeX[5] use simplified fontspec macros to create stylesheets and templates. We use XSLT to create mappings from this XML-stylesheet language to fontspec-based TeX templates and also to CSS3. We also provide user-friendly interfaces for the creation of such an XML stylesheet.

Style pattern comparison with Open Office and CSS

Now a days, most of the modern applications have implemented an XML package format that includes an XML implementation of stylesheet: InDesign has its own IDML[6] (InDesign Markup Language) XML package format and MS Word has its own OOXML[7] format, which is another ISO standard format. As they say ironically, the nice thing about standards is that there are plenty of them to choose from. However, instead of creating one more non-standard format, we will be looking to see how we can operate closely with current standards. Below is a sample code derived from OpenOffice document format:

```
<style:style style:name="Heading_20_1"
  style:display-name="Heading 1"
  style:family="paragraph"
  style:parent-style-name="Heading"
  style:next-style-name="Text_20_body"
  style:default-outline-level="1"
  style:class="text">+
<style:font-face style:name="Adobe Caslon Pro Bold"
  svg:font-family="'Adobe Caslon Pro Bold'"
  style:font-family-generic="roman"
  style:font-pitch="variable" />
```

```

<style:text-properties fo:font-size="115%"
    fo:font-weight="bold" style:font-size-asian="115%"
    style:font-weight-asian="bold"
    style:font-size-complex="115%"
    style:font-weight-complex="bold" />
</style:style>
-<style:style style:name="Heading_20_2"
    style:display-name="Heading 2"
    style:family="paragraph"
    style:parent-style-name="Heading"
    style:next-style-name="Text_20_body"
    style:default-outline-level="2"
    style:class="text">
<style:font-face style:name="Arial"
    svg:font-family="Arial"
    style:font-family-generic="swiss"
    style:font-pitch="variable" />
<style:text-properties fo:font-size="14pt"
    fo:font-style="italic" fo:font-weight="bold"
    style:font-size-asian="14pt"
    style:font-style-asian="italic"
    style:font-weight-asian="bold"
    style:font-size-complex="14pt"
    style:font-style-complex="italic"
    style:font-weight-complex="bold" />
</style:style>

```

The equivalent CSS style is listed below:

```

heading1{
    font-family: Adobe Caslon Pro Bold;
    font-size:14pt;
    font-style: normal;
    font-variant: normal;
    font-weight: Bold;
    line-height: 16pt;
    text-align: left;
    color: black;
    background-color: none;
    text-decoration: none;
    text-transform: normal;
}

```

```

heading2{
    font-family: Arial;
    font-size:14pt;
    font-style: Italic;
    font-variant: normal;
    font-weight: Bold;
    line-height: 16pt;
    text-align: left;
    color: black;
    background-color: none;
    text-decoration: none;
    text-transform: normal;
}

```

When comparing the above two style coding standards, the Cascading Style Sheet (CSS) is a simple and straightforward formulation without complex namespaces, additional attributes and other details which are not mandatory to form a style.

Advantages of CSS style pattern:

CSS makes it very easy to change the style of a document. The useful feature of CSS is that the entire style and layout are abstracted out of the HTML, so the HTML has only the content and not the style aspects. Different stylesheets can be used for different media without even the user knowing it. Stylesheets can be made for different media like the printer and PDA[8]. CSS is considered a clean coding technique, which means search engines won't have to struggle to "read" its content.

Disadvantages of CSS style pattern:

While new additions to CSS3 provide a stronger, more robust feature-set for layout, CSS is still at heart a styling language (for fonts, colours, borders and other decoration), and not a layout language (for blocks with positions, sizes, margins, and so on). These limitations mean that creating fluid layouts generally requires hand-coding of CSS, and has held back the development of a standards-based WYSIWYG[9] editor.

Introducing the Cascading Style Sheet Markup Language (CSSML)

SASS[10] is a meta-language on top of CSS that is used to describe the style of a document cleanly and structurally, with more power than flat CSS. Sass provides a simpler, more elegant syntax for CSS and also implements various features that are useful for creating manageable stylesheets.

SASS (<http://sass-lang.com/>) is an extension of CSS3 and provides several useful features which can handle nested rules, common variables, etc. However, it is not an XML model and cannot be validated using Schema[11]/DTD[12].

We have introduced the Cascading Style Sheet Markup Language (CSSML), an XML version of CSS that can be used as a standard for creating stylesheets and templates across different platforms and pagination systems. It is also an extension of CSS3 to handle nested rules as in "SASS", and can be validated using Schema/DTD.

Definition of CSSML

The Cascading Style Sheet Markup Language (CSSML) contains the style format details in a well-structured XML format.

The style names used in CSSML are similar to CSS, and the only difference is that the style names are defined as XML tag elements.

The below example explains the difference between CSS and CSSML style coding:

CSS format: font-family: Adobe Caslon Pro Bold;

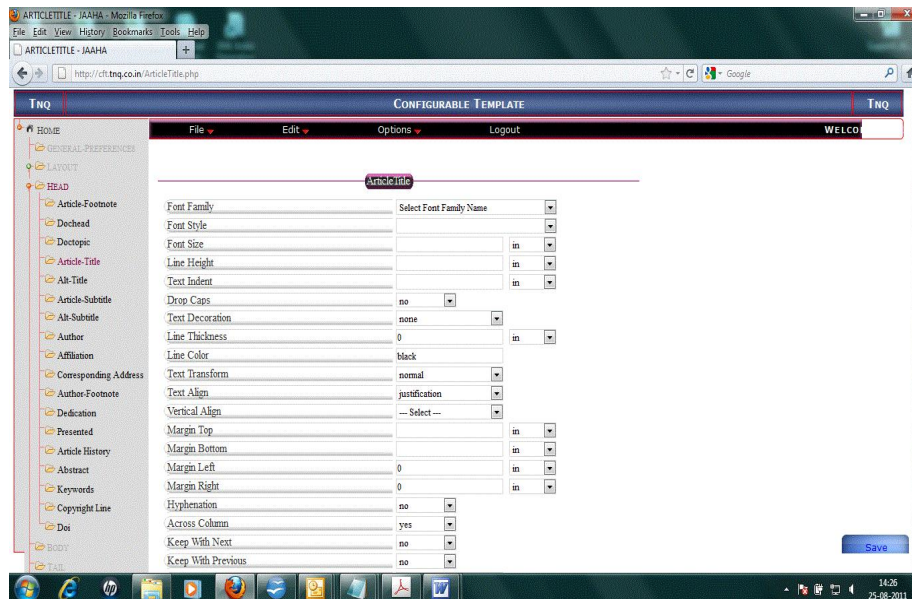
CSSML format: <font-family> Adobe Caslon Pro Bold</font-name>

The main advantage of the CSSML tag pattern is that we can validate the CSSML document using XML Schema or XML DTD which is not possible in CSS. We can write our own XML Schema/DTD to validate the CSSML document as follows:

- Elements and attributes that must/may be included, and are permitted in the structure
- The structure as specified by a regular expression syntax
- How character data is to be interpreted, e.g. as a number, a date etc.

However, creating the CSSML document is not as simple as creating a CSS. The CSSML needs XML tagging for all the data, and the user needs to wrap all the details with appropriate XML elements. To avoid such difficulties, we have provided a User Interface to create CSSML automatically with appropriate XML elements.

User Interface to get style specification details:



Sample CSSML coding:

```
<section1>
  <style>
    <font-family>Immono10-regular</font-name>
    <font-url>
      http://www.ctan.org/tex-archive/
      fonts/lm/fonts/opentype/public/lm/
      Immono10-regular
    </font-url>
    <font-size unit="pt">11</font-size>
    <font-style>normal</font-style>
    <font-variant>normal</font-variant>
    <font-weight>Bold</font-weight>
    <font-face>Immono10.OTF</font-name>
    <line_height unit="pt">13.2</line_height>
  </style>
  <section-label id="B12">
    <text-indent unit="pt">6</text-indent>
    <rule-color>black</rule-color>
    <text-transform>normal</text-transform>
    <vertical-align>bottom</vertical-align>
  </section-label>
  <section-title1 id="B13">
    <text-indent unit="in">0</text-indent>
    <text-align>justify</text-align>
    <word-break>hyphenate</word-break>
    <column-span>all</column-span>
    <vertical-align>bottom</vertical-align>
  </section-title1>
</section1>
```

CSSML to CSS3 conversion

CSSML provides a more elegant syntax for CSS and implements various features that are useful for creating stylesheets and \LaTeX templates. CSSML allows us to use formatting, nested rules, inline imports, etc., all with CSS compatibility. XSLT[13] is used to transform the CSSML XML to CSS format or \LaTeX class file with the use of fontspec package.

Formatting:

<pre><paragraph> <style> <font-name>Times</font-name> <font-size unit="pt">11</font-size> <line_height unit="pt">13</line_height> <text-indent unit="pt">11</text-indent> </style> </paragraph></pre>	<pre>paragraph { font-family: Times; font-size: 11pt; line-height: 13pt; text-indent: 11pt }</pre>
---	--

Nesting:

```
<section1>
  <style>
    <font-family>Times</font-family>
    <font-size unit="pt">11</font-size>
    <font-weight>Bold</font-weight>
    <line_height unit="pt">12</line_height>
  </style>
  <section-label id="B12">
    <text-indent unit="pt">6</text-indent>
  </section-label>
  <section-title id="B13">
    <text-align>left</text-align>
  </section-title>
</section1>
```

```
section1
{
  font-family: Times;
  font-size: 11pt;
  line-height: 12pt;
  font-weight: bold;
}
section1>section-label
{
  text-indent: 6pt;
}
section1>section-title
{
  text-align: left;
}
```

XPath is used to select nodes instead of CSS selectors. A good XPath to CSS mapping is given below:

CSS Selectors	XPath Pattern
h1 p	h1//p (Matches any p element that is a descendant of an h1 element)
h1 > p	h1/p (Matches any p element that is a child of an element h1)
p:first-child	*[1]/self::p (Matches element p when p is the first child of its parent)
h1 + h2	h1/following-sibling::*[1]/self::h2 (Matches any h1 element immediately preceded by an element h2.)

CSSML to \TeX (FONTSPEC) conversion

We are using “fontspec” package for font definition (fd)[14]. The “fontspec” package allows users of $X_{\text{e}}\mathbb{L}\text{A}\mathbb{T}\text{E}\mathbb{X}$ or $\text{Lua}\mathbb{T}\text{E}\mathbb{X}$ to load OpenType fonts in a $\mathbb{L}\text{A}\mathbb{T}\text{E}\mathbb{X}$ document. No font installation is necessary, and font features can be selected and used as desired throughout the document.

$X_{\text{e}}\mathbb{L}\text{A}\mathbb{T}\text{E}\mathbb{X}$ and $\text{Lua}\mathbb{T}\text{E}\mathbb{X}$ also allow fonts to be loaded by file name instead of font name. When you have a very large collection of fonts, you will sometimes not wish to have them all installed in your system’s font directories. In this case, it is more convenient to load them from a different location on your disk.

Font Declaration:

```
<font-group>
  <font-family>Times</font-family>
  <font-style-1>Times CG</font-style-1>
  <font-style-2>Times-Bold</font-style-2>
  <font-style-3>Times-Italic</font-style-3>
  <font-style-4>Times-BoldItalic</font-style-4>
  <font-style-5>Times-BoldSC</font-style-5>
</font-group>
```

```
\fontspec
[%
BoldFont= Times-Bold.otf ,
ItalicFont= Times-Italic.otf ,
BoldItalicFont = Times-BoldItalic.otf,
SmallCaps=Times-BoldSC.oft,
]{Times.otf}
```

Paragraph Style:

```
<paragraph>
  <style>
    <font-name>Times</font-name>
    <font-size unit="pt">11</font-size>
    <line_height unit="pt">13</line_height>
    <text-indent unit="pt">11</text-indent>
  </style>
</paragraph>
```

```
\def\normalsize{%
\fontsize{11}{13}
\fontspec{Times}
\paraindent=11\p@
}
```

Section Heading Style:

```
<section1>
  <style test="section1">
    <font-family>Times</font-family>
    <font-size unit="pt">11</font-size>
    <line-height unit="pt">13</line-height>
  </style>
  <section-label>
    <style test="section1">
      <font-variant>Bold</font-variant>
    </style>
  </section-label>
  <section-title>
    <style test="section1">
      <text-align>center</text-align>
      <margin-top unit="pt">12</margin-top>
      <margin-bottom unit="pt">6</margin-bottom>
    </style>
  </section-title>
</section1>
```

```
\newcommand\section{\@startsection%
{section}%
{1}%
{0}%
{-12\p@ \@plus -2\p@ \@minus -2\p@}
{6\p@}%
{\centering\fontsize{11}{13}%
\selectfont\bfseries}%
}
```

References

- [1] <http://www.w3.org/TR/html5/> - A vocabulary and associated APIs for HTML and XHTML.
- [2] <http://www.css3.info/> - Everything you need to know about CSS3.
- [3] <http://www.tug.org/mailman/listinfo/xelatex> - A DTD/Schema which is very close.
- [4] <http://getfo.org/texml/> - An XML syntax for \TeX .
- [5] <http://www.luatex.org/> - An extended version of pdf \TeX using Lua as an embedded scripting language.
- [6] http://blogs.adobe.com/indesignsdk/2009/03/idml_file_types.html - IDML for representing InDesign content.
- [7] http://en.wikipedia.org/wiki/Office_Open_XML - Open Office XML definition.

- [8] <http://www.webopedia.com/TERM/P/PDA.html> - What is PDA? A word definition from the Webopedia.
- [9] <http://www.webopedia.com/TERM/W/WYSIWYG.html> - What is WYSIWYG? – A word definition from the Webopedia.
- [10] <http://sass-lang.com/> - Syntactically Awesome Stylesheets.
- [11] <http://www.w3.org/XML/Schema> - XML Schemas express shared vocabularies and allow machines to carry out rules made by people.
- [12] http://en.wikipedia.org/wiki/Document_Type_Definition - DTD is a set of markup declarations that define a document type for SGML-family markup languages (SGML, XML, HTML).
- [13] http://www.w3schools.com/xsl/xsl_intro.asp - XSLT is a language for transforming XML documents into XHTML documents or to other XML documents.
- [14] <http://www.ctan.org/tex-archive/macros/xetex/latex/fontspec/> - The fontspec package provides an automatic and unified interface for loading fonts in \LaTeX .