On the use of TEX as an authoring language for HTML5

S.K. Venkatesan

Abstract

The TEX syntax has been fairly successful at marking up a variety of scientific and technical literature, making it an ideal authoring syntax. The brevity of the TEX syntax makes it difficult to create overlapping structures, which in the case of HTML has made life so difficult for XML purists. We discuss S-expressions, the TEX syntax and how it can help reduce the nightmare that HTML5 markup is going to create. Apart from this we implement a new syntax for marking up semantic information (microdata) in TEX.

1 Introduction

The brevity of TEX syntax has made it fairly successful at marking up a variety of scientific and technical literature. On the one hand, modern markup languages such as (X)HTML and XML have verbose syntax which is not only difficult to author but also produces non-treelike structures such as overlapping structures that need to be checked for well-formedness. On the other hand, TEX and its macros are difficult to parse and validate, compared to XML with a DTD or schema. Many XML versions of TEX have been proposed such as TEXML [3] and XLATEX [5] that are intrinsically close to (LA)TEX. The main advantage of such a system is that one can introduce a validator using a DTD or schema to check the syntax before passing it to the TEX engine.

However, XML syntax is difficult to author and in fact is prone to producing overlapping structures that need to be avoided for it to be well-formed, and as a result these XML versions have not become popular for authoring. In this article, we propose something that is quite the reverse, i.e., TEX as an authoring syntax for both XML and HTML.

2 TEX, S-expressions and XML

Let us look at the following TEX code:

```
\title[lang=en]{Title of
  a \textit{plain} article}
```

The same code in a Lisp-like S-expression would be:

```
(title (@ (lang="en")) ("Title of a ")
  (italic "plain") ("article"))
```

or if one would like to treat elements and attributes in the same way:

```
(title (@lang="en") ("Title of a ")
  (italic "plain") ("article"))
```

The difference between the above two S-expressions is that the former introduces a deliberate asymmetry between attributes and elements, whereas the latter treats attributes on a par with elements. However, both S-expressions can be considered as an improvement on XML as they allow further nesting within attributes. The corresponding XML code would be:

```
<title lang="en">Title of
  a <italic>plain</italic> article</title>
```

In both TEX and XML syntax, further nesting of structures is not possible within attributes, which makes TEX ideal for authoring XML or HTML5.

There are further similarities between the TEX and SGML/HTML syntaxes. Attribute minimization used in HTML, like not quoting attribute values, is very much practiced in TEX syntax, more as a rule rather than the exception; e.g.,

```
\includegraphics[width=2cm]{myimage.gif}
```

Unlike SGML/HTML, TEX typically uses a comma as the separator between attributes, instead of the word-space used in SGML/HTML. TEX also uses complete skipping of attribute values, similar to the commonly used HTML code: `<option selected>`. Quite like TEX, HTML also has the practise of shrinking multiple spaces to a single space. All of these similarities make it clear that authoring HTML in TEX would be an ideal proposition.

3 Overlapping markup in HTML

Since HTML is marked up by humans, there tend to be many situations with overlapping elements or other eccentric markup which do not confirm to a well-formed SGML or XML syntax. Consider the HTML markup:

```
<p>Text with <i>unique <b>and</i>
strong formatting</b> issues</title>
```

A utility like HTML Tidy [6] or TagSoup [1] can convert this into well-formed markup such as:

```
<p>Text with <i>unique </i><b><i>and</i>
strong formatting</b> issues</title>
```

However, it is not always clear what should be done with such a non-standard markup. The HTML5 specification defines clearly how such a non-standard markup should be interpreted [7] but the HTML implementations in browsers currently deal with it differently from each other.

W3C has been insisting for some time that the next generation of markup should be XML-compliant like XHTML+MathML+SVG profiles, with other intricacies such as namespaces. However, more than 99% of HTML pages in the wild are invalid, according to the HTML4 DTD or schema. This being the case, W3C gave up on the idea of an XML solution and moved on to HTML5 with added elements and features, such as MathML, SVG and video, audio and additional microdata elements.

Given the experience with HTML4, it can be safely predicted that the more features one adds to HTML, the greater the scope for non-standard markup such as overlaps and entanglement that can create a great deal of difficulty for browsers and users.

We will consider here, e.g., Microsoft's interpretation of MathML in HTML5. Microsoft has been pushing for certain agenda in MathML3 (although I must say with great relief that much of it has not been accepted by the MathML committee). Based on their own experience with OML, a subset of OOXML markup, they would like to add formatting features in MathML such as bold, italic and paragraph elements inside MathML. Consider the following markup:

```
<math><b><mi>r</mi></b>=<mfenced><mi>x</mi>
  <mi>y</mi></mfenced></math>
```

the corresponding pure MathML coding would be:

```
<math><mi mathvariant="bold-italic">r</mi>
  <mo>=</mo><mfenced><mi>x</mi>
  <mi>y</mi></mfenced></math>
```

Mixing elements from different namespaces is one of the side effects one can expect in HTML5. It is not clear if MathML elements could be included within SVG elements or vice versa. One can expect such new non-standard markups to be created that will be quite difficult for browsers to handle.

New elements such as <section> have been introduced, so one can expect more confusion:

```
<section><h2>Section title</h2>
  <section><h1>Another section title</h1>
  </section>
</section>
```

The intended meaning of <h1> or <h2> is not clear from the above markup, and you could say either 'I mean what I say' or 'I say what I mean', with our own impressionistic interpretations.

In this article we do not want to convey the impression that everything about HTML5 is out of the wild west; rather, it is a rich arena that needs to be authored carefully, because there are so many pitfalls. In fact, HTML5 introduces new features like MathML, SVG, video and audio features that are essential for further enrichment of basic content [4]. The important reason for using a TeX-like system is that it doesn't allow one to see the output if there are errors in the code and one can only produce well-formed code.

4 TeX as an input format for HTML5

In this section we would like introduce LaTeX environment for authoring HTML5. Many of these features have been introduced before, say, e.g., in XLaTeX and other concepts.

4.1 Main structural elements of the document

HTML5 has introduced new content elements that bring it closer to the standard LaTeX classes. We propose the following TeX macros.

No.	HTML	LaTeX	Description

1	<article>#1
	</article>
		\begin{article}
		#1
		\end{article}
			article

			headings:
2	<h1>#1</h1>	\Ha{#1}	— level one
3	<h2>#1</h2>	\Hb{#1}	— level two
4	<h3>#1</h3>	\Hc{#1}	— level three
4	<h4>#1</h4>	\Hd{#1}	— level four
5	<p>#1</p>	\p{#1}	paragraph
6	<span>#1</span>	\s{#1}	text span

4.2 Simple formatting elements

We propose the following TeX macros for HTML formatting elements:

No.	HTML	LaTeX	Description
1	<b>#1</b>	\B{#1}	bold
2	<i>#1</i>	\I{#1}	italic
3	<b><i>#1</i></b>	\BI{#1}	bold-italic
4	<tt>#1</tt>	\M{#1}	text
5	<sup>#1</sup>	\sp{#1}	superscript
6	<sub>#1</sub>	\sb{#1}	subscript

4.3 MathML elements

We propose the following TeX macros for MathML formatting elements:

No.	MathML	LaTeX	Description
1	<mrow>#1</mrow>	{#1}	grouping
2	<mi>#1</mi>	{#1}	variables
3	<mo>#1</mo>	{#1}	operators
4	<mn>#1</mn>	{#1}	numbers
5	<mtext>#1</mtext>	\mbox{#1}	monospace
6	<mfrac>#1#2</mfrac>	\frac{#1}{#2}	fraction
7	<msup>#1#2</msup>	{#1}^{#2}	superscript
8	<msub>#1#2</msub>	{#1}_{#2}	subscript
9	<mover>#1#2</mover>	{#1}^{#2}	over
10	<munder>#1#2</munder>	{#1}_{#2}	under

No.	SVG	LaTeX	Description
1	`<circle cx="#1" cy="#2" r="#3" style="stroke:#4; stroke-width:#5;fill:#6;"/>`	`\circle[x=#1,y=#2,r=#3 s=#4,sw=#5,f=#6]`	circle
2	`<ellipse cx="#1" cy="#2" rx="#3" ry="#4" style="stroke:#5; stroke-width:#6;fill:#7;"/>`	`\ellipse[x=#1,y=#2,rx=#3, ry=#4,s=#5,sw=#6,f=#7]`	ellipse
3	`<rect x="#1" y="#2" width="#3" height="#4" style="stroke:#5; stroke-width:#6;fill:#7;"/>`	`\rect[x=#1,y=#2,w=#3, h=#4,s=#5,sw=#6,f=#7]`	rectangle

Table 1: Proposed TeX macros for SVG formatting elements.

No.	Microdata	LaTeX	Description
1	itemscope	`\s[is=on]`	top element that indicates descendants are in scope
2	itemtype	`\s[it=http:// data-vocabulary.org/Person]`	property URL
3	itemid	`\s[iid=p0312]`	unique ID of the person
4	itemprop	`\s[ip=name]`	name of the person
5	itemref	`\s[ir=http:// www.ctan.org/pub/article]`	reference URL

Table 2: Proposed TeX macros for HTML5 microdata.

4.4 SVG elements

We propose the TeX macros in table 1 for SVG formatting elements. These can be implemented using LaTeX graphics packages such as TikZ [2].

4.5 Microdata attributes

Since microdata (semantic) attributes can be added to any of the basic HTML elements, we need to be able to add attributes to any of the HTML5 TeX macros as well. Table 2 shows how these microdata attributes for `<span>` element are indicated using TeX macro `\s` defined in §4.1.

5 MuLTiFlow

We have created a WYSIWYG editor for authoring HTML5, released under the GPL v3 license. It can be installed either as a Firefox addon or as a standalone program. The project is hosted at `http://sourceforge.net/projects/multiflow` and is also available through the Firefox addon network. At present this editor uses HTML5 and UTN28 markup for authoring complex equations, but it will use the proposed TeX syntax for authoring HTML5 from version 1.1 onwards.

References

[1] John Cowan, TagSoup: A SAX parser in Java for nasty, ugly HTML, `http://home.ccil.org/~cowan/tagsoup`.

[2] Andrew Mertz and William Slough, Graphics with PGF and TikZ, *TUGboat* 28:1 (2007), 91–99, `http://tug.org/TUGboat/tb28-1/tb88mertz.pdf`.

[3] Oleg Parashchenko, TeXML: Resurrecting TeX in the XML world, *TUGboat* 28:1 (2007), 5–10, `http://tug.org/TUGboat/tb28-1/tb88parashchenko.pdf`.

[4] Mark Pilgrim, HTML5: Up and Running, Dive into the Future of Web Development, O'Reilly Media, 2010.

[5] John Plaice and Yannis Haralambous, XLaTeX, a DTD/schema which is very close to LaTeX, *TUGboat* 24:3 (2003), 369–376, `http://tug.org/TUGboat/tb24-3/haralambous.pdf`, `http://omega.enstb.org/xlatex`.

[6] Dave Raggett, HTML Tidy, `http://tidy.sourceforge.net`.

[7] W3C, HTML5 Working draft, `http://www.w3.org/TR/html5/introduction.html#syntax-errors`.

◇ S.K. Venkatesan
TNQ Books and Journals
Chennai, India
`skvenkat (at) tnq dot co dot in`