Writing Structured and Semantics-Oriented Documents: TeX vs XML*

Jean-Michel HUFFLEN
LIFC (FRE CNRS 2661)
University of Franche-Comté
16, route de Gray
25030 BESANÇON CEDEX
FRANCE
hufflen@lifc.univ-fcomte.fr
http://lifc.univ-fcomte.fr/~hufflen

Abstract
Using xml-like syntax for documents gives them a tree structure, inducing a notion of structured document. Defining domain-dependent tags introduces a notion of semantics-oriented writing. These two points result in a new view about document production. In fact, they have already existed within TeX, but in another shape. This article aims to point out these notions and the differences between them. It ends with some proposals about the evolution of the tools belonging to TeX’s world.

Keywords Structured documents, semantics-oriented writing, TeX, LaTeX, PassiveTeX, xml, xslt, xsl-fo.

Streszczenie

Słowa kluczowe Dokumenty strukturalne, pisanie zorientowane semantycznie, TeX, LaTeX, PassiveTeX, xml, xslt, xsl-fo.

Introduction
The notion of document has deeply changed since the introduction of sgml. A document markup only depends on what users want to express by their own tags, regarding questions that are relevant for them. Besides, the notion of document transformation also appeared at this time with dsssl: from the same sgml document, we can derive a printable document sent to a laser printer as well as a hyper-text document in html for the Web. sgml being too complex for defining specialised markup

Introduction
The notion of document has deeply changed since the introduction of sgml. A document markup only depends on what users want to express by their own tags, regarding questions that are relevant for them. Besides, the notion of document transformation also appeared at this time with dsssl: from the same sgml document, we can derive a printable document sent to a laser printer as well as a hyper-text document in html for the Web. sgml being too complex for defining specialised markup

easily, a subset of this meta-language has been defined as xml. This meta-language has succeeded: nowadays it is used as a central formalism for data interchange, some related to networking use configuration files written according to xml’s syntax, ...

On another point, this markup notion also existed within word processors such as Plain TeX or LaTeX. So this article aims to point out different kinds of markup, and how they are put into action in xml and TeX.

Marking documents up
If we consider a HTML document, many tags used throughout it are related to questions of style: good examples are definitions of headings by means of tags h1, h2, etc. Even if the layout may be refined

* Title in Polish: Konstruowanie dokumentów strukturalnych i zorientowanych semantycznie: TeX versus XML.

2 Standard Generalized Markup Language. Now this meta-language has only historical interest, a good introduction to it can be found in [1].

3 Document Style Semantics and Specification Language.

4 HyperText Markup Language. See [1, Ch. 12] about the relationship between sgml and html.

5 eXtensible Markup Language. A good introduction to it is [10].
by means of CSS6 files [3, § 7.4], such an approach is related to the shape of document. In fact, HTML tags related to speech structuration, like p or div, are rarely used in practice.

A good example of a structured document is given by a poem, as shown in Figure 1: this document is obviously given a tree structure. Besides, repetitions are easily implemented: a repetition of a verse or stanza can be implemented syntactically, by means of an entity, or structurally, by sharing subtrees labelled by identifiers. Such an approach yields a very strict hierarchy among tags.

If we look at Figure 2, the DocBook tags of this bibliography express semantic information. Such an approach is more conformant to XML's philosophy, but the questions related to style may be more difficult to implement. For example, we see that title tags are used for three purposes: the bibliography's title, a bibliographical entry's title, and the title of a

---

6 Cascading StyleSheets.
<xml version="1.0" encoding="ISO-8859-1">
<!DOCTYPE bibliography PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN" "http://docbook.org/xml/4.2/docbookx.dtd">
<bibliography>
<title>Example of a small bibliography for the Bachotex 2006 conference</title>
<biblioentry id="bib.donaldson1982" lang="en" xreflabel="Donaldson1977">
  <author id="donaldson">
    <firstname>Stephen</firstname>
    <surname>Donaldson</surname>
    <othername role="mi">R.</othername>
  </author>
  <copyright><year>1982</year><holder><link linkend="donaldson"></holder></copyright>
  <isbn>0-00-615239-2</isbn>
  <pagenums>658</pagenums>
  <publisher>
    <publishername>Fontana</publishername>
    <address id="harper-collins">
      Harper Collins Publishers
      <street>77–85 Fulham Palace Road</street>
      <otheraddr role="district">Hammersmith</otheraddr>
      <city>London</city>
      <postcode>W6 8JB</postcode>
      <country>United Kingdom</country>
    </address>
  </publisher>
  <title>The One Tree</title>
  <seriesvolnums>2</seriesvolnums>
  <biblioset relation="seriesinfo">
    <title>The Second Chronicles of Thomas Covenant</title>
  </biblioset>
</biblioentry>
<biblioentry id="bib.feist-wurts1990" lang="en" xreflabel="Feist-Wurts1987">
  <authorgroup id="feist-wurts">
    <author>
      <firstname>Raymond</firstname>
      <surname>Feist</surname>
      <othername role="mi">E.</othername>
    </author>
    <author><firstname>Janny</firstname><surname>Wurts</surname></author>
  </authorgroup>
  <copyright><year>1990</year><holder><link linkend="feist-wurts"></holder></copyright>
  <isbn>0-586-07481-3</isbn>
  <pagenums>827</pagenums>
  <publisher>
    <publishername>Grafton Books</publishername>
    <address><otheraddr role="link"><link linkend="harper-collins"></link></otheraddr></address>
  </publisher>
  <title>Daughter of the Empire</title>
</biblioentry>
</bibliography>

(DocBook is a system for writing documents. It was initially designed from SGML [15], but recent versions have been reformulated from XML. We use the conventions of [11].)

Figure 2: Bibliography using DocBook.
series an entry belongs to. That is a nice use of such a tag name w.r.t. semantics-oriented approach, but these three kinds of titles are not to be displayed the same way when this bibliography is listed: according to English-speaking conventions [2, §§ 15 & 16], the title of a book should be displayed using italicised characters, whereas the title of a series is just displayed using ‘normal’ font. Last, the bibliography’s title should be emphasised as a ‘general’ title.

These two different approaches coexist within the description of a mathematical expression using MathML\textsuperscript{7}, either by its presentation or by its contents [14, §§ 2.3.1 & 2.3.2]. Let us consider:

\[
f(x) = \frac{(ax + b)^2}{\sqrt{\pi}}
\]

The specification of Figure 3 emphasises its graphical structure, whereas the version of Figure 4 directly refers to mathematical operations.

---

\textsuperscript{7} \texttt{MATH}ematical \texttt{Markup} \texttt{Language}.

**Figure 3:** The equation (1) in presentation mode.

**Figure 4:** The equation (1) in content mode.

---

**Structure and semantics within \TeX**

At a first glance, the programs related to \TeX & Co. only put a notion of structured document into action. Let us not forget that end-users are responsible for their semantics. Besides, such an semantics-oriented approach is encouraged by conceptors. For example, Leslie Lamport recommends to define a \texttt{\textbackslash{B}}\TeX command for an inner product, in order to decide its layout at only one place [7, § 1.5]. To give a second example from our documents, we systematically use a \texttt{\textbackslash{pgname}} command for programming languages’ names that are not logos. That allows us a unified layout for such names and we can know which programming languages are cited throughout one of our texts by a quick search. Such a command can be easily changed, as shown by our \texttt{\textbackslash{logo}} command: our logos are displayed using small capitals, except for the articles for TUGboat, where an \texttt{ad hoc} command is used.

\[
\newcommand{\pgname}[1]{\textsc{#1}}
\def{\logo}{\iffortugboat{\acro{\uppercase{#1}}}\else{\textsc{#1}}}\
\acro{\uppercase{#1}}\else{\textsc{#1}}\fi
\]

So these simple examples show that semantics-oriented writing is possible with \TeX, even if it is not always practised. Besides, grouping such commands into packages improve interchange among users.
Directions

A well-known drawback about programs belonging to \TeX's world: they recognise only their own formats. Let us consider the text given in Figure 1, it cannot be processed with \LaTeX. In that case, this is not a problem, we can write an \xslt\footnote{\textit{eXtensible Stylesheet Language} Transformations, the language of transformations used for XML texts.} program \footnote{\textit{eXtensible Stylesheet Language} — \textit{Formatting Objects}: this language aims to describe high-quality print outputs. See \cite{9} for an introduction, the official document being \cite{13}.} whose output would be suitable for \LaTeX. But this output will be in \textit{text} mode, that is, there will be two checks from a syntactic point of view. If \LaTeX accepted XML inputs, we could ensure that the result of such an \xslt program would be syntactically suitable for \LaTeX. Con\TeXxt \cite{4} can do some import, \LaTeX should do so, too.

On another point, a good ‘recycling’ of \TeX into XML's world is \textit{Passive} \TeX \cite{9, p. 180}, which processes \xsl-fo\footnote{\textit{XSL-FO}: Version 1.0. W3C Recommendation. Edited by James Clark. October 2001. \textcolor{blue}{http://www.w3.org/TR/2001/REC-xsl-20011015/}.} documents. \TeX is unrivalled as a typeset engine, so this approach allows some XML texts to take advantage of \TeX's power. From our point of view, this project should be developed and expanded, in the sense that \textit{Passive} \TeX should be able to include and mix fragments written w.r.t. \TeX syntax as well as XSL-FO documents.

\TeX and the programs related to it often work within a closed work. As an example, \LaTeX users often get used to put \LaTeX commands inside values handled by Bib\LaTeX \cite{8}, the bibliography processor usually associated with it. That is often needed, but makes difficult the use of Bib\LaTeX for another target than \LaTeX. We think that a successor of Bib\LaTeX should be based on XML as an interchange format and be able to replace \LaTeX commands by ‘semantic’ tags of XML, that is what we plan in \cite{5}. Such a choice would allow the layout corresponding to semantic tags to be deferred until the final step.

As a conclusion, we think that structured and semantics-oriented approaches are complementary. \TeX and XML can be complementary, too. The programs related to \TeX’s world are sometimes viewed as old products, but they can get new youth if they succeed in taking advantage of XML features.

Acknowledgements

Many thanks to Jerzy B. Ludwichowski, who has written the Polish translation of the abstract in a very short time, and who waited it for a longer time.

References

\begin{itemize}
\item \cite{1} Neil Bradley: \textit{The Concise SGML Companion}. Hayden Books. April 1998.
\item Oren Patashnik: \textit{BibTeXing}. February 1988. Part of Bib\TeX’s distribution.
\end{itemize}