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

Używanie składni XML-owej do opisu dokumentów nadaje im strukturę drzewiastą i indukuje w ten sposób pojęcie dokumentu strukturalnego. Definiowanie znaczników domenowo zależnych wprowadza pojęcie pisania zorientowanego semantycznie. Oba elementy łącznie dają nowe spojrzenie na tworzenie dokumentów. W rzeczy samej istniały one już w TEX-u, ale w innym kształcie. W artykule staramy się omówić wymienione pojęcia oraz różnice między nimi. Kończymy propozycjami dotyczącymi rozwoju narzędzi należących do świata TEX-owego.

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[2]. 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[3] [6]: 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[4] for the Web. SGML being too complex for defining specialised markup easily, a subset of this meta-language has been defined as XML[5]. 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] **S**tandard **G**eneralized **M**arkup **L**anguage. Now this meta-language has only historical interest, a good introduction to it can be found in [1].

[3] **D**ocument **S**tyle **S**emantics and **S**pecification **L**anguage.

[4] **H**yper**T**ext **M**arkup **L**anguage. See [1, Ch. 12] about the relationship between SGML and HTML.

[5] e**X**tensible **M**arkup **L**anguage. A good introduction to it is [10].

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<!--  This encoding allows Polish accents and special letters to be typed directly.    -->
<?xml-stylesheet href="poem.css" type="text/css"?>

<!DOCTYPE poem0 SYSTEM "poem0.dtd"
  [<!ENTITY refren-1 "<verse>Czuj, czuj, czuwaj,</verse>">]>

<poem0>
  <preamble>   <!--  This element groups some subtrees for metadata.    -->
    <title>Płonie ognisko</title>
  </preamble>
  <body>
    <stanza>
      <verse>Płonie ongisko w lesie,</verse>
      <verse>Wiatr smętną piosnkę niesie.</verse>
      <verse>Przy ogniu zaś drużyna</verse>
      <verse>Gawędę rozpoczyna</verse>
    </stanza>
    <stanza label="refren">   <!--  label is an optional attribute being type ID.    -->
      &refren-1;&refren-1;   <!--  Syntactical replacement.    -->
      <verse>Rozlega się dokoła,</verse>
      &refren-1;&refren-1;
      <verse>Najstarszy druh zawoła.</verse>
    </stanza>
    <stanza>
      <verse>Przestańciesię już bawić</verse>
      <verse>I czas swój marnotrawić.</verse>
      <verse>Niechj każdy z was się szczerze,</verse>
      <verse>Do pracy swej zabierze</verse>
    </stanza>
    <stanza>
      <!--  A stanza is a non-empty list of verses, but can be a repetition of a previous stanza, in which
            case we use the resume element with a required attribute, ref. This works only if we make
            sure that the value associated with this attribute unambiguously identifies a subtree, which
            is ensured by attributes being IDREF type.
        -->
      <resume ref="refren"/>
    </stanza>
  </body>
</poem0>
```

Figure 1: Example of a Polish song as an XML text.

by means of CSS[6] 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 *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] **C**ascading **S**tyle**S**heets.

```
<?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"/></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.

```
<math>
  <mi>f</mi>
  <mo stretchy="false">(</mo>
  <mi>x</mi>
  <mo stretchy="false">)</mo>
  <mo>=</mo>
  <mstyle displaystyle="true">
    <mfrac>
      <mrow>
        <msup>
          <mrow>
            <mfenced open="(" close=")"
                     separators="">
              <mi>a</mi>
              <mi>x</mi>
              <mo>+</mo>
              <mi>b</mi>
            </mfenced>
          </mrow>
          <mrow><mn>2</mn></mrow>
        </msup>
        <mo>+</mo>
        <mn>2</mn>
      </mrow>
      <mrow>
        <msqrt>
          <mrow><mi>&pi;</mi></mrow>
        </msqrt>
      </mrow>
    </mfrac>
  </mstyle>
</math>
```

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

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[7], either by its *presentation* or by its *contents* [14, §§ 2.3.1 & 2.3.2]. Let us consider:

$$f(x) = \frac{(ax+b)^2}{\sqrt{\pi}} \tag{1}$$

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

```
<math>
  <apply>
    <eq/>
    <apply><fn><ci>f</ci></fn><ci>x</ci></apply>
    <apply>
      <divide/>
      <apply>
        <power/>
        <apply>
          <plus/>
          <apply>
            <times/>
            <ci>a</ci>
            <ci>x</ci>
          </apply>
          <ci>b</ci>
        </apply>
        <cn>2</cn>
      </apply>
      <apply><root/><ci>&pi;</ci></apply>
    </apply>
  </apply>
</math>
```

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 LATEX 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 `\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 `\logo` command: our logos are displayed using small capitals, except for the articles for TUGboat, where an *ad hoc* command is used.

```
\newcommand{\pgname}[1]{\textsf{#1}}
\def\logo#1{\iffortugboat%
 \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.

[7] **MATH**ematical **M**arkup **L**anguage.

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[8] program [12] whose output would be suitable for LATEX. But this output will be in `text` mode, that is, there will be two ckecks 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. ConTEXt [4] can do some import, LATEX should do, too.

On another point, a good 'recycling' of TEX into XML's world is Passive TEX [9, p. 180], which processes XSL-FO[9] 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 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 BIBTEX [8], the bibliography processor usually associated with it. That is often needed, but makes difficult the use of BIBTEX for another target than LATEX. We think that a successor of BIBTEX 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 [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

[1] Neil BRADLEY: *The Concise* SGML *Companion.* Addison-Wesley. 1997.

[2] *The Chicago Manual of Style.* The University of Chicago Press. The 14th edition of a manual of style revised and expanded. 1993.

[3] Michel GOOSSENS, Sebastian RAHTZ, Eitan M. GURARI, Ross MOORE and Robert S. SUTOR: *The LATEX Web Companion.* Addison-Wesley Longmann, Inc., Reading, Massachusetts. May 1999.

[4] Hans HAGEN: *ConTEXt, the Manual.* November 2001. `http://www.pragma-ade.com/general/manuals/cont-enp.pdf`.

[5] Jean-Michel HUFFLEN: "MlBIBTEX: beyond LATEX". In: Apostolos SYROPOULOS, Karl BERRY, Yannis HARALAMBOUS, Baden HUGUES, Steven PETER and John PLAICE, eds., *International Conference on TEX,* XML, *and Digital Typography,* Vol. 3130 of LNCS, pp. 203–215. Springer, Xanthi, Greece. August 2004.

[6] International Standard ISO/IEC 10179:1996(E): DSSSL. 1996.

[7] Leslie LAMPORT: *LATEX. A Document Preparation System. User's Guide and Reference Manual.* Addison-Wesley Publishing Company, Reading, Massachusetts. 1994.

[8] Oren PATASHNIK: *BIBTEXing.* February 1988. Part of BIBTEX's distribution.

[9] Dave PAWSON: XSL-FO. O'Reilly & Associates, Inc. August 2002.

[10] Erik T. RAY: *Learning* XML. O'Reilly & Associates, Inc. January 2001.

[11] Thomas SCHRAITLE: *DocBook*-XML — *Medienneutrales und plattformunabhändiges Publizieren.* SuSE Press. 2004.

[12] W3C: XSL *Transformations (*XSLT*). Version 1.0.* W3C Recommendation. Edited by James Clark. November 1999. `http://www.w3.org/TR/1999/REC-xslt-19991116`.

[13] W3C: *Extensible Stylesheet Language (*XSL*). Version 1.0.* W3C Recommendation. Edited by James Clark. October 2001. `http://www.w3.org/TR/2001/REC-xsl-20011015/`.

[14] W3C: *Mathematical Markup Language (MathML) Version 2.0*, 2nd edition. W3C Recommendation. Edited by David Carlisle, Patrick Ion, Robert Miner, and Nico Poppelier. October 2003. `http://www.w3.org/TR/2003/REC-MathML2-20031021`.

[15] Norman WALSH and Leonard MUELLNER: *DocBook: The Definitive Guide.* O'Reilly & Associates, Inc. October 1999.

[8] e**X**tensible **S**tylesheet **L**anguage **T**ransformations, the language of transformations used for XML texts.

[9] e**X**tensible **S**tylesheet **L**anguage — **F**ormatting **O**bjects: this language aims to describe high-quality print outputs. See [9] for an introduction, the official document being [13].