XML versus TEX – jemné porovnání

Martin Tůma

31. srpna 2004

Obsah

1 Úvod

Srovnání TEXu, robusního a časem prověřeného typografického systému s poměrně mladou, ale o to více se rozvíjející technologií XML by jistě vydalo na celou knihu, tento článek je proto pouhým shrnutím možností obou technologií při řešení nejzákladnějších uživatelských činností. Především jde o tvorbu klasických článků a knih, vědeckých článků s matematickými výrazy a elektronické publikování. Na úvod krátké shrnutí obou technologií.

1.1 TEX

TEX je sázecí systém, který umožňuje vytvářet dokumenty složité struktury i vysoké typografické kvality. Jeho autorem je Donald Ervin Knuth, profesor počítačových věd na Standfordské univerzitě, a jeho vznik se datuje již do roku 1978.

Názvem TEX se často označuje kompletní instalace TEXového prostředí. Ta se skládá z nástrojů pro manipulaci s písmy a virtuálními písmy, nástrojů pro generování bitmapových instancí písem, nástrojů pro manipulaci s metrikami písem, ze systému pro popis písem METAFONT, z vlastního programu TEX, i maker se základními nadstavbami TEXu – plain TEX, LATEX či ConTEXt a z volně šiřitelných písem. V distribucích TEXu, jako je například teTEX[1] či TEXLive[2], dále bývají zahrnuta i národní rozšíření, pro české dokumenty jsou to například *csplain* a *cslatex*.

TEX však je nejenom sázecí systém, ale i programovací jazyk. Patří do kategorie makrojazyků a makra jsou základní programovací strukturou TEXu. Makra v TEXu začínají znakem \ a končí před prvním nepísmenným znakem. Závorky { a } pak ohraničují oblast platnosti lokálních definic, definice a argumenty makra, znak # označuje číslo argumentu. Speciální význam mají také znaky % a $ – % uvozuje komentář a znak $ přepíná sazbu do matematického módu. Příklad makra pro nadpis dokumentu pak může vypadat například následovně:

```
\def\nadpis#1{              % definice nadpisu
   \removelastskip\bigskip  % odmaže poslední vert. mezeru a přidá vlastní
   \indent{\titulfont #1}   % odsazený text nadpisu větším fontem
   \par\nobreak\medskip}    % konec řádku, zakázaný zlom, menší mezera
```

Jeho použití v textu pak vypadá takto:

```
\nadpis{Toto je nadpis}
```

Základních příkazů TEXu, tzv. primitiv, ze kterých se makra skládají, je pouze něco přes 300. Vše ostatní jsou již složená makra, která ale mohou být díky možnostem jazyka velice mocná a umožňují tak poměrně dobře rozlišit v dokumentu obsah a formu.

[1] http://www.tug.org/teTeX
[2] http://www.tug.org/texlive

Výstupním formátem TEXu je obvykle `dvi`, ale existují i jiné možnosti – výstup do `pdf` pomocí programu pdfTEX[3] či do HTML pomocí LATEX2HTML[4].

1.2 XML

Na rozdíl od TEXu není XML (eXtensible Markup Language) typografickým systémem, ale jedná se o značkovací jazyk. XML vzniklo v roce 1998 výběrem nejužitečnějších vlastností SGML (Standard Generalized Markup Language) a jeho doplněním o některé prvky. Důležitou vlastností XML převzatou právě z SGML je možnost tvorby vlastních *definic typů dokumentů* – DTD. Důsledkem toho je, že si každý může vytvořit svůj vlastní formát dokumentu zcela vyhovující jeho potřebám.

Základní věci jsou však pro všechny XML dokumenty stejné. Každý XML dokument se skládá z *elementů*, které jsou do sebe navzájem vnořené. Elementy se v textu vyznačují pomocí tzv. *tagů*. Názvy tagů se zapisují mezi znaky < a >, v případě, že je element párový, tzn. uzavírá nějaký text, má ukončovací tag před svým názvem ještě znak /, který musí být v XML uveden i na konci každého nepárového tagu. Každý element může mít dále své *atributy*. Atributy se zapisují dovnitř tagu ve formátu `atribut="hodnota"`. Zápis nadpisu by v XML mohl vypadat třeba takto:

```
<nadpis jazyk="cz">Toto je nadpis</nadpis>
```

Otázkou zůstává, jak bude takovýto element vlastně v konečném dokumentu vypadat, samotné XML tento problém nijak neřeší. K tomuto účelu vždy slouží některý ze stylových jazyků, nejčastěji CSS (Cascading Style Sheets) nebo XSL (eXtensible Stylesheet Language) a příslušný stylový procesor. Jako příklad lze uvést procesor formátovacích objektů FOP[5] či XSLT procesor Saxon[6].

Výstupní formát záleží pouze na zvoleném stylovém procesoru a jeho možnostech. Nejčastěji používaným formátem pro tištěný výstup je formát *pdf* ale výstup je obvykle možné získat i v dalších formátech jako RTF, HTML či PostScript.

2 Srovnání v jednotlivých oblastech použití

2.1 Dokumenty všedního dne

Pod tímto poněkud netypickým názvem jsou myšleny klasické články, dopisy, a jiné „kancelářské“ záležitosti. Zde je možností jaký nástroj zvolit k tvorbě takovéhoto dokumentu opravdu nepřeberně a to jak v případě TEXu, tak v případě XML.

V obou případech lze při tvorbě takovéhoto dokumentu jít v dané technologii velice do hloubky. V TEXu to znamená kompletní nadefinování všech maker v plainTEXu, v případě

[3] http://www.tug.org/applications/pdftex
[4] http://www.latex2html.org
[5] http://xml.apache.org/fop
[6] http://saxon.sourceforge.net

XML by pak mohlo jít o návrh vlastního DTD i XLS stylu. Tento přístup je však vhodný pouze pro opravdu rozsáhlá díla a to ještě ne vždy. Navíc je ještě nutné dodat, že pro opravdu profesionální typografii se nehodí ani jeden ze systémů a v současné době se téměř výhradně používají různé propertiální DTP systémy.

Je možné ale zvolit i opačný „extrém“ a jít cestou WYSIWYG (What You See Is What You Get – Co vidíš to dostaneš) editorů, a nutno přiznat, že takto v současnosti vzniká většina dokumentů. TEX nabízí pod GPL šířený LyX[7] a TeXmacs[8], či komerční Scientific Word[9]. Na straně XML jsou „nejtěžším kalibrem“ bezpochyby OpenOffice.org[10], jejichž nativním datovým formátem je právě XML. Výhodou takovéhoto přístupu je potřeba minimálních či spíše žádných znalostí TEXu či XML. Nevýhodou je pak horší kvalita výstupu a prakticky nulové oddělení obsahu dokumentu od jeho formy, což podstatně ztěžuje případné vyhledávání relevantních informací v dokumentu.

Rozumným kompromisem, kdy za určitou snahu získáme téměř profesionální výstup je pak použití některé makronástavby TEXu, například LATEXu. Na straně XML je pak srovnatelným řešením použití DocBooku[11]. Jak dokumenty v LATEXu tak dokumenty v DocBooku jdou psát velmi dobře „ručně“, tzn. bez nutnosti použití programu doplňujícího formátování, pouze v textovém editoru. Následující fragmenty ukazují typickou strukturu dokumentů v DocBooku a LATEXu.

DocBook:

```
<?xml version='1.0' encoding='iso-8859-2'?>
<!DOCTYPE book PUBLIC '-//OASIS//DTD DocBook XML V4.2//EN'
  'http://www.oasis-open.org/docbook/xml/4.2/docbookx.dtd'>

<article lang="cs">
  <artheader>
    <title>Můj článek</title>
    <author>
      <firstname>Martin</firstname>
      <surname>Tůma</surname>
    </author>
  </artheader>

  <sect1>
    <title>První část</title>
    <para>První odstavec</para>
    <para> ... </para>
```

[7] http://www.lyx.org
[8] http://www.texmacs.org
[9] http://www.mackichan.com
[10] http://www.openoffice.org
[11] http://www.docbook.org

```
    <sect2>
      <title>První podčást </title>
      <para> ... </para>
    </sect2>
</sect1>

<sect1>
    ...
</sect1>

<bibliography> ... </bibliography>

</article>
```

LaTeX:

```
\documentclass[a4paper, 12pt]{article}
\usepackage{czech}
\usepackage{a4wide}

\title{Můj článek}
\author{Martin Tůma}

\begin{document}
\maketitle

\section{První část}
První odstavec

...

\subsection{První podčást}
...

\section{ ... }

\begin{thebibliography}
...
\end{thebibliography}

\end{document}
```

Struktura tedy vypadá v obou případech podobně, jen způsob zápisu je odlišný. Výhoda LaTeXu spočívá v dalším zpracování takovéhoto dokumentu. Takto zapsaný text se jednoduše „prožene“ interpretem a výsledkem je perfektní výstup ve formátu *dvi*. Podrobný postup viz například [1]

U DocBooku je situace o trochu složitější. Před finální transformací do konečné podoby je potřeba ještě, jak již bylo zmíněno, vytvořit odpovídající styl. Naštěstí existují již vytvořené kvalitní styly, které lze rovnou na dokument aplikovat a získat tak požadovaný výstup. Více o problematice aplikace stylů lze nalézt v [4].

Výhoda XML v tomto případě spočívá v rozmanitosti možných výstupních formátů, které lze aplikováním různých stylů z jednoho zdroje získat. Toho lze v omezené míře dosáhnout i u LaTeXu, ale již né tak efektivně jako v případě XML – v mnoha případech je nutné upravovat i původní dokument.

2.2 Sazba matematiky a vědecké literatury

Jestliže v případě tvorby klasických dokumentů lze bez větších problémů použít jak TeXu tak XML, v případě tvorby matematických či odborných publikací s vysokým podílem matematických výrazů, je situace zcela odlišná – zde má TeX zatím navrch.

XML sice nabízí MathML, ale jeho využití je spíše ve výměně dat mezi matematickými programy. MathML podporuje například Maple[12] či Mathematica[13]. Je však potřeba přiznat, že v poslední době množství software využívající MathML pro sazbu matematiky roste, obsáhlý seznam je možné najít na stránkách organizace W3C[14].

TeX má však oproti všem řešením založeným na MathML dvě obrovské výhody. Tou první je perfektní kombinace sazby matematiky se sazbou ostatního textu. Druhou, a možná ještě mnohem důležitější výhodou, je pak zápis matematických vzorců v TeXu. Zápis matematiky je velice efektivní a přirozený a hlavně podobný „ručnímu“ zápisu. Oproti tomu zápis v MathML je velice nepřehledný a většina editorů MathML proto nabízí „klikací“ zápis, tedy stejný druh zápisu jako má například editor rovnic v MS Wordu. Ten je však ve většině případů mnohem zdlouhavější, než přímý zápis ve formě TeXového kódu.

Že je tato forma zápisu výhodná pak potvrzuje i to, že jí převzal i kancelářský balík OpenOffice.org, který umožňuje zápis matematiky se syntaxí velice podobnou TeXu. OpenOffice.org však jednak používají upravené MathML, což výrazně ztěžuje přenositelnost takto vytvořených vzorců, a navíc je typografická kvalita matematického výstupu velice nízká.

2.3 Elektronické publikování

V dnešní době je stále více potřeba vytvořený dokument exportovat do několika různých formátů, aby jej bylo možné nejenom vytisknout, ale také publikovat na internetu. Te-

[12] http://www.maplesoft.com
[13] http://www.wolfram.com
[14] http://www.w3.org/Math/implementations.html

oreticky je sice možné dokument na internetu vystavit ve formátu PDF, chceme-li však dokument zpřístupnit opravdu „on-line“, je potřeba zvolit HTML, či ještě lépe XHTML.

Pokud pro tvorbu dokumentu použijeme TEX, je potřeba pro převod do HTML použít nějaký externí převaděč, ostatně stejně jako pro převod do jakéhokoliv jiného formátu než dvi. Vyčerpávající seznam HTML/XML převaděčů lze nalézt na internetové stránce: `http://www.cse.ohio-state.edu/~gurari/TeX4ht/mn.html`, nejznámější a nejpoužívanější je však již zmíněný LATEX2HTML pro LATEX. Za zmíňku stojí HtmlTEX[15], jehož autorem je Sergey Brin, jeden ze zakladatelů Googlu. Při použití LATEX2HTML získáme poměrně dobrý HTML výstup, i když pravda, standartní vzhled se mě osobně příliš nelíbí. Dokument vyhovuje zvolenému DTD, které však lze zvolit pouze z množiny HTML, na modernější XHTML LATEX2HTML transformovat zatím neumí. Přetransformovat lze velká většina LATEXových dokumentů, problémy jsou hlavně s matematikou a vlastnímy makry, přičemž u matematiky to není vinou LATEX2HTML, ale bídnou podporou MathML v současných prohlížečích.

U XML je situace poněkud odlišná – XHTML je podmnožinou XML, takže dokumenty lze teoreticky tvořit rovnou v HTML. Toto však není dobrý nápad, pokud chceme dokumenty i tisknout. Kaskádové styly používané pro HTML na kvalitný tiskový výstup většinou nestačí. Pokuď tedy dokument netvoříme pouze pro web, je nejefektivnější cestou zvolit obecnější XML formát, například DocBook, a pomocí XSL či DSSL stylů jej konvertovat na HTML/XHTML.

XML má však v oblast elektronického publikování jednu obrovskou výhodu – může sloužit i jako univerzální datový formát. To znamená, že z XML dokumentu lze velice účině extrahovat potřebné informace. Existuje dokonce „dotazovací“ jazyk pro XML – XQuery[16]. A právě možnost efektivně pracovat s daty v XML dokumentech je jednou z nejdůležitějších vlastností XML.

3 Závěr

V předcházejících částech byly naznačeny možnosti obou technologií v jednotlivých oblastech. Kde je tedy výhodné použít TEX a kde XML?

TEX je bezesporu vynikající na sazbu matematiky a tím pádem také vědeckých článků. Zde je jeho pozice stále prakticky neotřesitelná. Velmi dobrým nástrojem je TEX také na psaní článků a referátů v případě, že výsledný dokument bude určen pimárně pro tisk. Zde je však nutno říci, že TEXem je zde myšlen především LATEX, použití „čistého“ TEXu se vyplatí až u rozsáhlejších děl, nebo pokuď vyžadujeme specifický výstup.

Problémem TEXu může být dostupnost dokumentace. Poměrně dobře je zdokumentován LATEX, viz například [2] či *LATEX navigátor* na stránkách cstugu[17]. Horší to již ale je s dokumentací pro plainTEX. Zde dle mého názoru chybí středně odborná literatura. K dispozici jsou buď publikace pro úplné začátečníky, nebo velmi odborné publikace (TEXbook,

[15] http://www-db.stanford.edu/~sergey/htmltex
[16] http://www.w3.org/TR/xquery
[17] http://www.cstug.cz/latex/lm/frames.html

TEXbook naruby). Naopak často zmiňovaná složitá instalace TEXu je dle mého názoru mýtus, tedy alespoň v případě LINUXu. Prakticky všechny moderní distribuce obsahují TEX připravený k okamžitému použití a to většinou i s českými rozšířeními. Pro Windows pak existuje MikTEX[18], který nabízí velice dobře zpracovaný instalátor. Složitější situace je s rozšířeními TEXu jako LATEX2HTML, které potřebují nějaký interpret, v tomto případě Perl, což na Windows představuje další složitou instalaci.

Pokuď dokument potřebujeme ve více výstupních formátech s důrazem na elektronické publikování je výhodnější zvolit XML. Instalace potřebného software (XML parser, XSLT procesor, ...) sice tké není triviální záležitost, je však společná pro všechny výstupní formáty. Volba XML navíc přináší oproti TEXu mnohem lepší možnost zpracování dat v dokumentech. Jako výhodu XML je ještě nutné zmínit nativní podporu `UNICODE` kódování, které TEX vůbec nepodporuje a v neposlední řadě kvalitní dostupnou dokumentaci, především díky konsorciu W3C.

V současnosti osobně využívám jak TEX(LATEX), tak XML, měl-li bych se ale rozhodnout pouze pro jedno z nich, zvolil bych s výhledem do budoucnosti mocně se rozvíjející XML.

[18] http://www.miktex.org

Literatura

[1] Olšák: *První setkání s TEXem*
ftp://math.feld.cvut.cz/pub/cstex/doc/prvni.pdf

[2] Oetiker, Partl, Hyna, Schlegl: *The Not So Short Introduction to LaTeX2e*
www.ctan.org/tex-archive/info/lshort/english/lshort.pdf

[3] Kosek: *XML pro každého* Grada 2000

[4] Kosek: *DocBook, stručný úvod do tvorby a zpracování dokumentů*
http://www.kosek.cz/xml/db

[5] Diskuzní fórum `cz.comp.cstex`
http://usenet.jyxo.cz/cz.comp.cstex