Zpravodaj Československého sdružení uživatelů TeXu

Jiří Kosek

Sazba XML

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 13 (2003), No. 1, 3–6

Persistent URL: http://dml.cz/dmlcz/149910

Sazba XML

Jiří Kosek

V posledních měsících proběhlo v texové diskusní skupině i jinde několik diskusí o tom, zda je lepší při přípravě dokumentů používat TEX nebo XML. Pokusím se příliš nezneužít možnosti zaplnit stránky Zpravodaje názory, které jsem v diskuzích zastával já. Konec konců celá diskuse neměla moc smysl, protože TEX a XML spolu mohou žít bok po boku bez nějakých velkých problémů.

Asi se všichni shodneme, že TEX je výtečný nástroj pro sazbu dokumentů. Po dlouhá léta se dokumenty pro sazbu připravovaly přímo v TEXu. TEX tak sloužil jako jazyk pro zápis dokumentu a také jako systém pro jeho sazbu a zlom. Pokud je naším cílem jen dokument připravit, pěkně vysázet a vytisknout, je potom TEX vhodným a postačujícím nástrojem.

Pokud však chceme s připraveným dokumentem naložit i jinak, než jej jen vytisknout, třeba převést do HTML nebo z něj dotazem vybrat nějaká data, není už TEX jako formát pro primární pořízení dokumentu nejvhodnějším kandidátem. Vhodnější pro tyto účely jsou značkovací jazyky, v dnešní době především XML. S tím by možná někteří diskutéři nesouhlasili, ale zkusme to brát jako fakt. XML se zkrátka hodí pro ukládání informací, protože je lze dále poměrně snadno zpracovávat. TEX se hodí pro sazbu dokumentů, protože je rychlý a má implementovány kvalitní typografické algoritmy. Nic nám tedy nebrání ve využití výhod obou technologií – dokumenty ukládat do XML a pro potřeby tisku je zpracovávat TEXem. Získáme tak výhody obou světů.

Možností, jak sazbu dokumentu XML provést pomocí TEXu, je několik a my se je pokusíme přiblížit. V navazujících článcích jsou pak ty nejpoužívanější metody rozebrány více do hloubky včetně příkladů použití.

Přímá sazba XML

Asi nejjednodušší variantou je přímá sazba XML pomocí TEXu. Nepotřebujeme žádné přídavné programy, XML se zpracovává čistě prostředky TEXu. Tento přístup má však dva poměrně nepříjemné nedostatky.

Napsat parser XML přímo v TEXu je velmi obtížný úkol. Existující parsery jsou schopné načítat jen podmnožinu XML, obvykle nepodporují entity, které umožňují jeden dokument XML rozložit do více fyzických souborů.

Druhým nedostatkem je samotný princip, kdy musíme mapovat výskyty značek v XML na texové sekvence. V mnoha případech toto mapování můžeme poměrně jednoduše vytvořit, ale jsou případy, kdybychom v místě výskytu značky

potřebovali znát informace z jiné části dokumentu XML. Lze to samozřejmě obejít pomocí ukládání stavových informací nebo víceprůchodovým zpracováním dokumentu apod. Ale v porovnání se speciálními nástroji, které během zpracování umožňují přístup k celému dokumentu, je to velmi nepohodlné.

Pro přímou sazbu XML pomocí TEXu se dnes používají dva systémy. Prvním z nich je xmltex. Jedná se o jednoduchý parser implementovaný v TEXu, kterému můžeme v konfiguračním souboru určit, na jaké TEXové sekvence se mají převádět jednotlivé značky v dokumentu XML. Podrobnější popis xmltexu naleznete v samostatném článku *Použití parseru XML v TEXu* v tomto čísle Zpravodaje.

Druhým produktem, který si s XML poradí přímo, je ConTEXt. Tato uživatelsky příjemná makronadstavba nad TEXem obsahuje i nástroje na zpracování XML.

Konverze XML do TEXu

Přímou sazbu XML TEXem lze použít jen v jednoduchých situacích. I při složitějších požadavcích však nemusíme na použití TEXu rezignovat. Pro převod XML do podoby TEXového kódu můžeme použít sofistikovanější nástroje, které nám umožní pohodlné načtení XML a jeho převedení.

Asi nejvhodnějším nástrojem pro tuto úlohu je jazyk XSLT. Tento jazyk primárně slouží k popsání transformaci dokumentu XML do podoby HTML stránky, případně XML dokumentu s jinou strukturou. XSLT však s jistými omezeními umí generovat i obyčejné textové soubory, a tedy i TEXové dokumenty. XSLT pomocí dotazovacího jazyka XPath nabízí přístup k libovolné části dokumentu XML kdykoliv během zpracování a řeší tak pohodlně problémy jako je generování obsahu, křížových odkazů apod.

Napsání stylu XSLT pro převod XML do TEXu je poměrně jednoduché. Musíme si dát pozor jen na správné ošetření a převod znaků, které mají v TEXu speciální význam (např. `#`, `$`, `&`).

Pěknou ukázku použití XSLT při sazbě knihy z podkladů v XML jste mohli nalézt v minulém čísle Zpravodaje v článku Zdeňka Wagnera *Využití XML a LaTEXu při sazbě odborných knih.*

Využití standardních stylových jazyků

Výše popsané metody nám umožňují z dokumentů XML získat kvalitní tištěný výstup pomocí TEXu. Skalní příznivce značkovacích technologií by však tomuto přístupu přece jen jedno vytkl: formátovací předpis je úzce svázán se systémem TEX – nelze jej využít s jiným sázecím systémem a nemůžeme tak dostat jiný

výstup než formáty podporované (pdf)TEXem – DVI, PostScript a PDF. Co kdybychom však chtěli výstup v jiném formátu, jako je třeba RTF nebo HTML? Vytvořená konverze do TEXu by nám byla k ničemu.

Tento problém ve světě XML řeší stylové jazyky. Ty umožňují popsat vzhled dokumentu poměrně abstraktně a nezávisle na výstupním formátu a médiu. Takovýmto obecným předpisem – stylem – se může řídit převod do mnoha formátů v různých aplikacích. Stačí, když se standardizuje jazyk, kterým se styl popisuje. Je to jen přirozené pokračování snahy vytvořit nezávislý formát dat – tím je XML. Vzhled dokumentu by měl být také definován nezávisle, a to jak na dokumentu, tak na programu, který se použije pro výsledné formátování.

Stylových jazyků, které se hodí pro formátování XML dokumentů, existuje mnoho, ale mezi dva nejpoužívanější a „nejmocnější“ patří DSSSL a XSL. Shodou okolností pro oba dva jazyky existují i jejich implementace, které pro samotné formátování používají TEX. V případě DSSSL se jedná o JadeTEX a v případě XSL pak o PassiveTEX. I jejich základním principům a použití se věnujeme v samostatných článcích (články o JadeTEXu i o PassiveTEXu jsou v tomto čísle Zpravodaje).

Použití stylových jazyků přináší kromě nezávislosti na použitém formátovacím nástroji i další výhody. Většinu parametrů sazby lze nastavit jednoduše jako hodnoty vlastností a není potřeba pronikat do makrojazyka TEXu. DSSSL i XSL obsahují dotazovací jazyk, který umožňuje kdykoliv pracovat s libovolnou částí vstupního dokumentu XML – nejsme tak omezeni jako u TEXu, který vstupní soubor čte sekvenčně. Své výhody má i TEX – lze jej plně programovat, a jeho možnosti záleží jen na schopnostech uživatele, ne na jednou pevně daných vyjadřovacích možnostech standardizovaného stylového jazyka.

Další vývoj

Sazba XML pomocí TEXu je možná a v mnoha případech můžeme získat výborné výsledky. Nicméně jsou dvě oblasti, které by potřebovaly výrazně zlepšit. TEX je sám o sobě 8bitový a zpracování dokumentů používajících jako znakovou sadu Unicode mu činí potíže. Jde je sice obejít, ale postup je krkolomný a výsledek není vždy zcela robustní. Řešením může být přechod na systém Omega, který do TEXu přidává právě podporu Unicode.

Druhým omezením je samotný vstupní procesor TEXu. Jde s ním sice dělat kouzla, ale napsat v něm plnohodnotný parser XML nebo jiného formátu není příliš jednoduché. Vhodnější by byl modulární sázecí systém, kde by šlo použít uživatelské vstupní moduly speciálně připravené na čtení XML. Něco podobného by mělo umožňovat NTS.

Vše záleží na tom, jak rychle se budou „nové verze TEXu“ jako NTS a Omega vyvíjet, a jak rychle se rozšíří mezi uživateli. Bez jejich rozšíření se asi nikdo nebude obtěžovat s přepisováním stávajících maker pro tyto nové nadějné projekty.

Summary: Typesetting XML

This articles summarizes methods suitable for processing XML documents by the TEX system – direct typeseting (xmltex, ConTeXt), conversion to TEX (XSLT) and TEX based stylesheet language implementations (XSL, DSSSL). The article acts as an introduction for more detailed articles about processing XML with TEX.

Jiří Kosek
jirka@kosek.cz

Použití parseru XML v TEXu

Jiří Kosek

Chceme-li XML dokumenty sázet čistě prostředky TEXu bez nutnosti používání dalších pomocných programů, je asi nejvhodnějším nástrojem xmltex. Jedná se o makro nadstavbu na LATEXem, která implementuje parser XML. Ten umožňuje načítání dokumentů XML a umí podle pravidel definovaných v konfiguračním souboru převádět jednotlivé elementy na sekvence texového kódu, které pak řídí sazbu. Autorem xmltexu je David Carlisle, který je uznávaným odborníkem jak na oblast TEXu, tak i XML a zejména pak stylového jazyka XSL.

Většina moderních distribucí TEXu již xmltex obsahuje jako samostatný formát. Vysázení dokumentu XML je pak otázkou spuštění jediného příkazu:

```
xmltex dokument.xml
```

resp.

```
pdfxmltex dokument.xml
```

v případě, že chceme získat výstup v PDF pomocí pdfTEXu. Nemáme-li xmltex připravený jako samostatný formát, jde využít jednoduchý pomocný dokument v TEXu, který nadefinuje jméno dokumentu ke zpracování a poté aktivuje xmltex:

```
\def\xmlfile{dokument.xml}
\input xmltex.tex
\end{document}
```

Tento dokument se pak zpracuje pomocí klasického LATEXu.