Kapitola 1. Elektronické publikování

Obsah

Elektronické publikování a XML

V současnosti jsme svědky stále probíhající 3. inoformační (komunikační) revoluce, která spočívá v širokém zavádění informačních technologií. Tento proces začal v 50. létech 20. století, kdy začaly pracovat první počítače. Nyní počátkem 21. století informační revoluce vrcholí, nebot počítačová gramotnost představuje jednu ze základních složek vzdělání moderního člověka. Rozvoji informačních technologií předcházeli dvě další události (procesy), které hrály rozhodující roli při formování komunikačních dovedností a technologí - šlo o vznik písma (1. informační revoluce, 4 500 let před naším letopočtem) a vynález knihtisku (2. informační revoluce, 15. století).

Elektronické publikování (e-publishing) je v současnosti fenoménem v oblasti šíření informací. Elektronické publikování můžeme definovat jako přípravu, tvorbu, zachycení, transformaci, ukládání a diseminaci dokumentů směřující k jejich zpřístupnění v elektronické podobě [Jed2001a]. Produktem elektronického publikování je elektronická publikace neboli dokument[1] zpřístupnění v elektronické formě určitému okruhu uživatelů [Jed2001a].

Přenos informací elektronickou cestou je rychlý a má díky celosvětové počítačové síti široký (globální) dosah a také většina informačních zdrojů je k dispozici v elektronické podobě. Proto elektronické publikování proniklo téměř do všech oblastí lidské činnosti - s elektronickými publikacemi se můžeme setkat v oblasti komerce (e-business, e-banking), vědy, kultury, umění, státní správy (e-demokracie, e-goverment) i vzdělávání (e-learning), ale nesmíme zapomenout ani na oblast běžné komunikace - elektronickou poštu (e-mail). Tento článek by měl pojednávat pouze o technickém řešení publikace dat.

Elektronický dokument se od klasického analogového dokumentu liší v mnoha směrech, především v uživatelském komfortu - elektronický dokument může obsahovat navigační prvky, vyhledávací prostředky, zpětnou vazbu od uživatele k autorovi (diskuze, ankety, hlasování…), propojení na multimediální prvky[2], hyperlinkové a interaktivní prvky. Novinkou jsou také nosiče publikací, kdy již nejsme omezeni tištěnou knihou, ale e-publikace se mohou objevovat na CD-ROM, DVD apod. Výhodou elektronických publikací je také absence tzv. časoprostorových bariér, tedy možnost kontinuální aktualizace, zpětných zásahů do textu a okamžitého publikování. V neposlední řadě jsou elektronické publikace oproti analogovým také mnohem levnější. Další odlišnosti, které pro někoho mohou představovat nevýhody elektronických dokumentů, představuje legislativa a počítačová gramotnost. Vyšší úroven počítačových znalostí se sice předpokládá na straně autora, ale také uživatel by měl být schopný minimálně spustit příslušný

[1] Asociace amerických vydavatelů definuje dokument jako organizovanou kolekci malých částí textu (například kapitoly nebo odstavce) a obrázků, které se nazývají elementy. Elementy v dokumentu mají mezi sebou vzájemné vztahy, které definují logickou strukturu dokumentu [Chle2000].

[2] Multimédia představují kombinaci více typů médií na jediném nosiči (například kombinace textu, zvuků, statické grafiky a dynamické grafiky).

prohlížeč a také ho nakonfigurovat. Nedostatečná legislativa v oblasti informačních technolgií je široce známá - dochází k tzv. e-kriminalitě spočívající v porušování autorských práv a také v odcizování osobních údajů.

Budeme-li se zabývat klasifikací e-dokumentů, můžeme vytvořit dvě základní kategorie - off-line dokumenty a on-line dokumenty. Off-line dokumenty jsou šířeny pomocí pamětových médií (CD ROM, DVD…), on-line dokumenty jsou k dispozici pomocí sítových služeb (Internet, Intranet…). Další možností je rozdělovat e-dokumenty na primární (dokumenty vytvořené v elektronické formě) a sekundární (zdigitalizované dokumenty).

Jestliže chceme tvořit a publikovat kartografická díla a příslušné doprovodné materiály (texty, grafy apod.), měli bychom dodržovat následující pravidla[3]:

1. Před tvorbou vlastního dokumentu by si měl autor vytvořit podrobný scénář, který by měl obsahovat přehledný vývojový diagram zobrazující strukturu dokumentu, popis jednotlivých částí budoucího dokumentu a také odpovědi na následující otázky:

    - Jaký dokument vytvářím?
    - Proč vytvářím dokument?
    - Jakým způsobem bude dokument vznikat?
    - Jaké budou použity zdroje a prameny?
    - Kde je možné si příslušné zdroje opatřit?
    - Jaký bude rozpočet projektu?
    - Jakým způsobem s dokumentem naložím po jeho vytvoření - způsob zveřejnění, autorská práva, testování, marketing, distribuce apod.

    Scénář by měl sloužit k ujasnění koncepce dokumentu a také k vyřešení některých problémů v komunikaci mezi zadavatelem a autorem dokumentu.

2. Uživatel by měl mít možnost si materiály prohlédnout, případně editovat na jakémkoli typu počítače (platformě), ať už se jedná o Intel x86, Motorola, Alpha (Digital) nebo SPARC (Sun), s libovolným operačním systémem.

3. Získání nástroje k prohlížení, případně také k editaci by nemělo uživatele finančně zruinovat - autor by měl brát v potaz softwarovou a hardwarovou náročnost projektu[4].

4. Informace mohou být nejen tištěny na papír, ale také on-line prohlíženy a to dokonce na různých platformách (PDA, WAP…).

5. Každý dokument by měl být přehledně a přísně logicky strukturován.

6. **Celý dokument a také zásady popisované v předchozích bodech by měly být přizpůsobeny především uživateli, resp. cílové skupině uživatelů.**

[3] Další pravidla lze nalézt také v kapitolách věnovaných tvorbě www stránek

[4] 12.7.2004 byl na serveru www.zive.cz [http://www.zive.cz/] uveřejněn krátký příspěvek popisující snižování rozpočtu ve Francii - k tomu by měl být použit přechod od software poskytovaného firmou Microsoft k Open Source software (konkrétně Linux Mandrake), který bude nainstalován spolu s příslušnými programy na 1500 počítačů úředníků státní správy...kompletní článek [http://www.zive.cz/h/Bleskovky/AR.asp?ARI=117367].

Velice často se však setkáme s porušováním těchto základních pravidel. Dokumenty se často zapisují konkrétním komerčním software, který je úzce svázán s konkrétním operačním systémem (v některých případech dokonce pouze s jedinou verzí operačního systému). Formáty takových programů jsou většinou binární (nečitelné pro různé operační systémy), uzavřené (nejdou implementovat do programů jiných výrobců) a často také zpětně nekompatibilní (starší verze programu si neporadí s novým formátem). Problematický bývá také export do jiných formátů určených pro prezentaci na jiném médiu.

Logická struktura dokumentu bývá v takových formátech sice na první pohled patrná – ovšem jde pouze o první pohled. Struktura takových dokumentů je totiž závislá na zobrazovacích pravidlech, která sice jdou modifikovat, ale většinou nelze vytvořit strukturu typu – všechny obrázky zobraz písmem určité velikosti a kurzívou.

Z předchozích požadavků jasně vyplývá, že řešením kvalitní dokumentace by byl formát vytvářený širokou komunitou, nezávislou na jediném výrobci software, který bude obsahovat nejen vlastní dokument, ale také popisovat strukturu dokumentu.

Takovým formátem je **eXtensible Markup Language (XML)**. Vlastní XML je podmnožinou výrazně širšího jazyka Standard Generalized Markup Language (SGML). Do této skupiny jazyků (formátů), které nazýváme značkovacími patří také například HyperText Markup Language (HTML) – jazyk pro tvorbu webových stránek, který je nejrozšířenější aplikací XML/SGML na světě[5].

Pokusme se ukázat, zda XML splňuje pravidla definovaná v úvodu tohoto odstavce:

1. Dokumenty zapsané v libovolné aplikaci XML jsou jednoduché ASCIII (American Standard Code for Information Interchange) textové dokumenty, které jsou multiplatfomní.

2. Vzhledem k široké komunitě uživatelů XML jsou k dispozici nejen komerční verze programů, ale také velké množství nástrojů, které jsou poskytovány zdarma. Používání programů s bezplatnými licencemi (open source, freeware, shareware apod.) je v souladu s deklarovanou nezávislostí XML.

3. Formát XML a jeho deriváty pouze popisují konkrétní data – v čistém XML se nesetkáme s pravidly, která udávají jakým způsobem budou data zobrazena. K tomu slouží tzv. transformační jazyky (jsou často vytvořené v také XML), pomocí nichž lze převádět dokumenty do jiných formátů, bez toho, že bychom ve vlastním dokumentu museli měnit jeho obsah nebo některé komponenty. Tímto způsobem lze snadno převést data zapsaná v XML do jiných formátů.

4. Dokument XML jasně a přehledně popisuje strukturu dat.

Dokument zapsaný v XML může obsahovat následující prvek:

```
<para>Odstavec, který obsahuje další text.</para>
```

Jde skutečně o pouhou textovou informaci, která uživateli říká, že text (data) uzavřený v elementu (tagu) para tvoří odstavec. V příslušném transformačním jazyku je pak zapsáno, jakým způsobem bude odstavec zobrazen. Tento příklad ilustruje základní filosofii XML.

[5] Není striktně definováno, zda se aplikace XML/SGML nazývají jazyky nebo formáty. Většinou se tyto termíny používají v závislosti na tom, zda pomocí XML/SGML popisujeme pouze datovou strikturu (formát) nebo jestli je popis datové strukturu spojený s nějakými vuzualizačními pravidly (jazyk).

Důležité

XML striktně odděluje obsah (vlastní data) od formy (vizualizace).

K představě jakým způsobem lze zpracovat XML dokument může sloužit následující schéma:

Obrázek 1.1. Princip XML

K dokumentu v XML se vztahuje další dokument, který obsahuje popis struktur, které mohou být ve vlastním XML použity. To je důležité pro provádění tzv. validace dokumentu (automatické kontroly správnosti dokumentu) pomocí programů, které se nazývají parsery. Některé XML editory obsahují vestavěný parser, časté jsou také samostatné parsery, například nsgmls [http://www.jclark.com], Xerces [http://xml.apache.org/index.html], MSXML [http://msdn.microsoft.com/downloads] nebo xmllint. Existuje více jazyků, které určené pro popis definic typů a atributů – ze známějších můžeme vyjmenovat Document Type Definition (DTD), XML Schema, Schematron nebo Relax NG. Transformační procesory slouží k převodu XML dokumentů pomocí transformačních stylů do výsledného formátu.

DocBook

Druhou nejpoužívanější aplikací XML/SGML je DocBook [http://www.oasis-open.org/docbook]. SGML verze vznikla v roce 1991. Od roku 1998 je DocBook vyvíjen na základě XML, přičemž tato verze DoCBooku dnes převažuje jak u uživatelů tak u vývojářů. DocBook byl v první řadě určen pro výměnu unixové dokumentace. V současnosti je na stránkách sdružení OASIS, které DocBook spravuje, k dispozici verze 4.2 (připravuje se verze 4.3).

Vlastní DocBook je vlastně DTD. To znamená, že definuje syntaxi dokumentu, tedy elementy a jejich atributy, které můžeme v dokumentu používat. Docbook je poskytován zdarma. Volně jsou také k dispozici styly pro transformaci dokumentů do jiných formátů zapsané v jazycích XSL (XML Style Language) a také DSSSL[6] (Document Style Semantics and Specification Language). Pomocí těchto jazyků lze transformovat dokumenty zapsané v DocBooku (Dokumenty napsané v DocBooku ve většině standardních aplikací nejdou zobrazit, protože se jedná pouze o obsah dokumentu a ne o popis způsobu prezentace dokumentu.) do mnoha jiných formátů – HTML stránky (jednotlivé i vzájemně provázané), soubory s on-line nápovědou (HTML Help, Java Help), RTF (Rich Text Format), TeX, TeXinfo (styl pro TeX,včetně dokumentace), FrameMaker MIF, UNIX man pages, PDF (Portable Document Format), Postscript nebo nápověda pro Eclipse (open-source vývojové prostředí).

DocBook je hojně používán k vytváření nejrůznější technické dokumentace, například k tvorbě dokumentace k velkému množství softwarových produktů – Linux, FreeBSD, PHP, uživatelská rozhraní KDE, Gnome… DocBook používají také známé firmy Sun a Novell a známé nakladatelství O'Reilly, které se na vývoji DocBooku od počátku podílelo.

Vzhledem k tomu, že soubory v XML jsou ukládány jako ASCII (American Standard Code for Information Interchange) znaky, lze tyto dokumenty editovat ve kterémkoli textovém editoru. Existují specializované editory pro práci s XML – mezi ně patří například TextPad, PSPad, vim nebo v Javě napsaný JEdit. Pokročilí uživatelé zvláště preferují klasický editor Emacs (s balíčkem PSGML). Existují i WYSIWYG (What You See Is What You Get) editory, které většinou nejsou poskytovány bezplatně, napří-

[6] Čti "dýzl" styly.

klad XML Spy, Corel XMetaL nebo Epic. Jediným známým WYSIWYG editorem, který je poskytován v bezplatné verzi, je XMLmind XML Editor [http://www.xmlmind.com/xmleditor].

K nevýhodám DocBooku řadíme především poměrně náročnou instalaci a konfiguraci systému. Existuje ovšem velké množství dokumentace. Pro některé uživatele, zvyklé na komfortní prostředí jiných textových editorů, může být používání DocBooku poměrně složité, především zápis dokumentů. Na Internetu se také nachází velké množství studijních materiálů a příkladů.

Podíváme-li se na obrázek popisující transformaci XML dokumentů, vidíme, že pro DocBook jsme se zmínili o používaných editorech pro zápis a modifikaci souborů, některých parsovacích programech, jazyků pro popis syntaxe dokumentu a možných výstupních formátech.

Pro zpracování XML dokumentů se používají kromě zmíněných XSL a DSSSL stylů ještě styly CSS (Cascading Style Sheets, kaskádové styly) a styly FOSI (Formatting Output Specification Instance).

- XSL - v současnosti nejperspektivnější stylový jazyk. Jak vyplývá z názvu, jedná o speciální stylový jazyk pro převod dokumentů v XML.
- DSSSL - jazyk původně určený pro SGML, jsou k dispozici volně šiřitelné styly pro DocBook. Dnes je tento jazyk na ústupu – existuje pro něj pouze jediná funkční aplikace (resp. její modifikace). DSSSL styly se v současnosti používají především pro převod do formátu RTF, se kterým si XSL styly neporadí.
- CSS - jsou velice jednoduché, ale pro převod složitějších dokumentů nedostačují. Okrajově se používají pouze pro převod stránek do HTML.
- FOSI - nejméně používaný stylový jazyk, vyskytuje se víceméně v komerčních aplikacích (Epic).

Chceme-li převádět dokument pomocí DSSSL stylů, lze použít pouze jediný transformační procesor Jade [http://www.jclark.com/jade] nebo OpenJade [http://openjade.sourceforge.net]. Pro generování dokumentu ve formátu TeX existuje balík maker JadeTeX.

Jestliže budeme dokument zpracovávat pomocí XSL stylů je nejvýhodnější použít XSLT procesor Saxon [http://users.iclway.co.uk/mhkay/saxon]. Jinou možností jsou programy xsltproc [http://www.fh-frankfurt.de/~igor/projects/libxml/index.html], XT [http://www.jclark.com/xml/xt.html] nebo Xalan [http://xml.apache.org].

Zajímavá internetová služba se nachází na adrese firmy Schema Software Inc. [http://www.schemasoft.com] - jedná se o převod documentu ve formátu DOC (Microsoft Word) do formátu DocBook - nedostatkem je především vlastní DTD (aplikace nepoužívá originální DTD DocBooku).

Vzhledem k faktu, že DocBook je poskytován zdarma, lze na Internetu najít poměrně velké množství informací, tutorialů, návodů a jiných materiálů, které se DocBooku týkají.

Tabulka 1.1. Odkazy na DocBook

předpřipravená instalace pro Windows [http://badame.vse.cz/jkj/dbinst.zip]
oficiální dokumentace [http://www.docbook.org]

styly pro formátování dokumentů [http://docbook.sourceforge.net]
aktuální informace o DocBooku [http://www.oasis-open.org]
jedna z nejlepších českých stránek (nejen) o DocBooku [http://www.kosek.cz]
české stránky o DocBooku (autor J.Kosek) [http://www.docbook.cz/]
další kvalitní česká stránka o Docbooku [http://www.volny.cz/zampach/dbk]
software pro práci s XML [http://www.xmlsoftware.com]
tutorialy pro XML aplikace [http://www.w3schools.com]
materiály pro XML aplikace [http://www.zvon.org]
internetové konference [news:cz.comp.lang.xml, docbook-apps@lists.oasis-open.org]

DocBook nese především sémantickou informaci - umožňuje členění textu do kapitol, sekcí, podkapitol. K dispozici jsou elementy pro vkládání citací, výpisů programů, rejstříků, bibliografie, dedikací, metainformací, příloh, rovnic, obrázků, seznamů, tabulek… Celkově DocBook zahrnuje více než 400 elementů. V praxi se běžně využívá několik desítek prvků - ostatní jsou zaměřeny například na názvy tříd, metod, příkazů, tlačítek - obecně programátorských elementů.

Zajímavou alternativou použití DocBooku je tvorba prezentací (prezentačních podkladů - slides). Norman Walsh vytvořil DocBook Slides [hhtp://docbook.sourceforge.net/projects/slides] jako podmnožinu tzv. zjednodušeného DocBooku (Simplified DocBook [http://www.oasis-open.org/docbook/xml/simple]). Výhodou je možnost exportu slidů do PDF, díky čemuž jsou podklady přístupné i v tisknutelné podobě. K převodu do PDF slouží XSL styly. Pomocí XSLT je možné soubor slidů převést do standardního DocBooku.

Na počátku roku 2005 se na Internetu objevil článek Normana Walshe [http://norman.walsh.name/2005/01/03/DocBook-05] [Wal2005], který naznačuje změny DocBooku pro rok 2005. V zásadě jde o tři základní body:

1. Publikování DocBooku 5.0 - hlavní změnou v nové verzi bude přechod z DTD na nové stabilní schéma RELAX NG.
2. Publikování kompletní dokumentace pro verzi 5.0 (DocBook 5.0: The Definitive Guide)[7]
3. Vytvoření lepších balíčků pro uživatele, především jako motivační faktor pro širší používání DocBooku.

Na závěr kapitoly věnované e-publikování je vhodné se zmínit o současném moderním trendu v této oblasti. Jedná se o tzv. digitální konvergenci (digital convergence) neboli o propojení různých zařízení do jedné sítě a vytvoření globální informační infrastruktury (Global Information Infrastructure). Pro takovou síť bude typická snadná výměna dat a také tzv. hybridní služby (např. interaktivní TV, Web TV - webcasting…).

V současnosti je elektronické publikování pevně spojeno s používáním metadat, tedy dat popisujících vlastní dokument. Metadata jsou důležitá pro nalezení relevantních informací, pro informační management a také pro ochranu vlastních dat a autorských práv. Metadata můžeme k dokumentu připojit například formou metatagů v HTML dokumentu, formou externích souborů nebo také formou databáze metadat. Pro zápis metadat existují mnohá metadatová schémata:

[7] Zatím je k dispozici první verze [http://docbook.org/tdg5/].

- Dublin Core (Dublin Metadata Core Element)
- EAD (Encoding Archival Description)
- TEI (Text Encoding Initiative) - schémata EAD a TEI jsou založeny na SGML
- DOBM (Description of Old Books and Manuscripts) - založený na HTML
- AACR2 - (Anglo-American Cataloguing Rules)

Pro překonání nekompatibility existujících metadatových schémat byl navržen formát RDF (Resource Description Framework), který je také jednou z aplikací XML.