Významné aplikace (2), Docbook; značkovací architektury, DITA

Obsah

Motivace k Docbooku

DocBook: příklad složitějšího značkování

- rozsáhlý projekt - poskytnout jednotný komplexní značkovací jazyk pro „veškerou“ programátorskou dokumentaci
- nyní používáno k celé řadě jiných účelů - psaní článků (article), knih (book), jednotlivých kapitol (chapter), sekcí (section, sectX)

- autorem je Norman Walsh (Sun Microsystems Inc.)
- podrobnosti, DTD, help, software, styly k dispozici viz docbook.org [http://docbook.org]
- pravděpodobně nejrozsáhlejší existující značkování pro logický popis dokumentu
- k DB existuje TDG (DocBook: The Definitive Guide) - také jako Windows Help [/~tomp/xml/tdg-en-2.0.7.chm]

Co je Docbook?

- Docbook je XML (a SGML) značkování pro psaní dokumentů, především technické povahy (počítačové manuály, technická dokumentace).
- Vznikl původně jako nástroj pro zvládnutí rozsáhlé dokumentace unixových systémů.
- Principem je logické (sémantické) značkování, text vzniká s vyznačením logických celků:
  - větších bloků textu (kniha, článek, kapitola, sekce, odstavec, výpis obrazovky...)
  - menších částí textu na řádku (zdůrazněná část, odkaz, název produktu, příkaz,...)
  - multimediální prvky (obrázky, videa, zvuky...)
  - pomocné prvky a metadata (název, autorství, datum vzniku, copyright, vyznačení položek rejstříku, obsahu...)

Výhody Docbooku

- Předností bylo a je, že z dobře označkované dokumenty je možné zpracovávat:
  - směrem k vizuální podobě (pomocí CSS, pomocí XSLT do HTML, přes LaTeX nebo XSL:FO do PDF, ale i PostScript, PDF, RTF, DVI a prosté ASCII...), speciální důraz na výstup do formátů dokumentace/nápovědy (HTML Help, Microsoft CHM, man-stránky)
  - lze z něj extrahovat požadované části nebo prvky (vezmi kapitolu úvod, vygeneruj obsah knihy...) nebo více textů spojovat do jednoho

Původ

- Docbook se objevil počátkem 90. let (1991) tehdy jako SGML značkování.
- Od zavedení XML jako de-facto standardu pro semistrukturovaná data (W3C specifikace XML v roce 1998) se Docbook začíná reprezentovat převážně v XML -- i kvůli rozvoji a dostupnosti stále více nástrojů
- O vývoj se nyní stará konsorcium OASIS [http://www.oasis-open.org] (The Organization for the Advancement of Structured Information Standards).
- Na vývoji se podílí mj. Jirka Kosek [http://www.kosek.cz], editorem specifikací je Norm Walsh [http://norman.walsh.name].

Základní struktury Docbook

Ukládání

Ukládáme-li Docbookové dokumenty do souborů, je obvyklou příponou `.dbk` [http://www.google.com/search?q=.dbk] [http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.dbk], `.xml` [http://www.google.com/search?q=.xml] [http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.xml]

MIME type pro Docbook je `application/docbook+xml`

Druhy dokumentů/strukturálních prvků

Povaha dokumentu je určena zejména jeho základní strukturou danou použitím příslušných *strukturálních prvků*.

Podle rozsahu (velikosti, logického uspořádání) dokumentu je možným typem:

set	kolekce knih (`book`) nebo dalších kolekcí -- kolekce lze vnořovat.
book	kniha, sestává z kapitol (`chapter`), článků (`article`) nebo částí (`part`), smí obsahovat rejstříky, přílohy atd.
part	část, soubor jedné či více kapitol, části se smí vnořovat a mohou obsahovat úvodní texty.
article	pojmenovaný soubor blokových prvků.
chapter	pojmenovaný a obvykle číslovaný soubor blokových prvků vyskytující se ve větším celku (kniha, článek).
appendix	příloha
dedication	text představující určení vnořeného elementu

Blokové prvky

Jsou dalšími, jemnějšími stavebními kameny dokumentu:

- odstavce
- tabulky
- seznamy
- příklady
- obrázky, atd.

Tyto a další blokové prvky jsou v dokumentu uvedeny v pořadí, v jakém jsou následně čteny -- v západních jazycích jsou tedy vizualizovány zhora dolů, ale např. v čínštině zleva doprava.

Prvky na řádku

Neboli in-line elements se vyskytují v blokových elementech a vyznačují blíže povahu textu, který obklopují:

- zdůraznění (`emphasis`...)
- odkazy (např. `link`, `ulink`, `olink`...)
- význam (klíčové slovo, příkaz, název souboru...)

Příklad dokumentu v Docbook 5

Docbook 5 je posledním, dosud však nehotovým standardem. Od předchozích verzí se kromě úprav značkování liší používáním XML *jmenných prostorů* a absencí DOCTYPE deklarace.

```
<?xml version="1.0" encoding="UTF-8"?>
<book id="simple_book" xmlns="http://docbook.org/ns/docbook" version="5.0">
   <title>Very simple book</title>
   <chapter id="chapter_1">
     <title>Chapter 1</title>
     <para>Hello world!</para>
     <para>I hope that your day is proceeding <emphasis>splendidly</emphasis>!</pa
   </chapter>
   <chapter id="chapter_2">
     <title>Chapter 2</title>
     <para>Hello again, world!</para>
   </chapter>
</book>
```

Poznámka

I vzhledem k neusazenosti DB 5 se stále převážně používají verze 4.x

Totéž v Docbooku 4.4

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN"
                      "http://www.oasis-open.org/docbook/xml/4.4/docbookx.dtd">
<book id="simple_book">
   <title>Very simple book</title>
   <chapter id="chapter_1">
     <title>Chapter 1</title>
     <para>Hello world!</para>
     <para>I hope that your day is proceeding <emphasis>splendidly</emphasis>!</pa
   </chapter>
   <chapter id="chapter_2">
     <title>Chapter 2</title>
     <para>Hello again, world!</para>
   </chapter>
</book>
```

Varianty Docbook

Používat verze 5 nebo 4.x?

1. V zásadě je vhodnější zatím používat verze 4.x (pokud možno tu nejnovější, např. V4.5) -- jsou dostupné editory, transformační styly, další nástroje.
2. Konverze do DB 5 je možná kdykoli později pomocí standardně dodávaného XSLT stylu...

DocBook: vrstvy a přizpůsobení

- DocBook lze používat jako základní (Full)
- zjednodušený (Simplified) nebo
- si jej přizpůsobit

přizpůsobení znamená:

- upravit DTD (přes parametrické entity)
- evt. upravit (XSL) styly
- XSL styly jsou upravovány na základě importu původního stylu a překrytí vybraných šablon

Vrstvy Docbooku - Simplified

Z Docbooku lze omezováním (redukcí množiny povolených elementů), rozšiřováním (přidáváním elementů) nebo obojím vytvářet odvozené jazyky:

Simplified Docbook

Omezení redukující značkování např. tak, že se z rodiny příbuzných elementů zachoval jen jeden -- např. `programlisting`, ale není povolen `screen`

Žádné "velké" věci, tzn. žádné knihy (book), jen články (article)

Krátké DTD (aby se dalo i online stáhnout)

Každý dokument ve značkování Simplified Docbook je automaticky i ve značkování Docbook.

Dokumentace k Simplified Docbook online [http://www.docbook.org/schemas/simplified]

Docbook Slides

- Je naopak rozšířením :-) Simplified Docbook (existuje ale i varianta Slides pro full-Docbook).
- Určeny pro psaní prezentací -- "fólií" (stejně jako např. Powerpoint).
- Dostupné XSLT umějí transformovat do HTML prezentací buďto prostých statických nebo ovládaných JavaScriptem.
- Moderní prohlížeče umějí nad takovými prezentacemi navigaci na standardní strukturální body dokumentu -- začátek, obsah, rejstřík, další či předchozí fólii.

Další moduly

Rozšíření Docbooku se technicky děje pomocí tzv. *modulů*.

Jelikož DB 4.x je definován pomocí DTD, jsou moduly de-facto překrytím nehodících se částí DTD a doplněním nových.

Příklady:

EBNF	Adds support for EBNF diagrams
HTML Forms	Adds support for HTML forms
MathML	Adds support for MathML in equations
SVG	Adds support for SVG in graphics

Nástroje pro Docbook

Požadavky na editory

Čím docbookové soubory vytvářet a upravovat?

- K editaci lze použít v nouzi libovolný *textový editor* s podporou požadovaných znakových sad a s možností ukládání ve zvoleném kódování.
- Lepší volbou je XML editor alespoň s poloautomatickým uzavíráním elementů -- získáme jistě dobře utvořený (well-formed) XMLdokument.
- Ještě lepší je editor s podporou psaní dokumentů vymezených DTD nebo schématem -- získáme validní dokument.
- Ideální je vizuální nástroj, kde píšeme jako v běžném textovém editoru a výstup je validní Docbook.

Dostupné editory

V současnosti je několik specializovaných editorů pro Docbook i zdarma dostupných:

xmlmind	http://xmlmind.com [???] fy `Pixware` je výkonný vizuální editor v základní verzi zdarma (nelze jej např. integrovat do aplikací), "umí" i jiná značkování nebo se je může naučit :-)
eDE	>e-novative> Docbook Environment [http://www.e-novative.info/software/ede.php] fy `e-novative` je prostředí pro MS Windows určené k vizuální tvorbě docbookových dokumentů.

Transformační nástroje

Smyslem převážně většiny běžných transformačních nástrojů (např. XSLT stylů) je převod do vizuální podoby (HTML, XSL:FO a následně PDF...)

- základním nástrojem jsou Docbook XSL [http://en.wikipedia.org/wiki/DocBook_XSL] styly
- jsou bohatě parametrizovatelné, umožňují "naladit" výstup podle potřeb

- dokumentace - kniha vydavatelství Sagehill [http://www.sagehill.net/docbookxsl/index.html]
- zde je kompletní reference [http://docbook.sourceforge.net/release/xsl/current/doc/] k použití Docbook XSL

Docbook prakticky

Balík xslt2

- Umožňuje práci s Docbookem přizpůsobeným pro pořizování *závěrečných prací* (bakalářky, diplomky...) na FI MU.
- Pomáhá v (téměř) celém životním cyklu dokumentu -- psaní, transformace, rendering do tiskové podoby prostředky TeXu.
- Autorem je *Jan Pavlovič*.
- Balík `xslt2` je dostupný na unixových strojích FI zadáním **module add xslt2 [http://www.google.com/search?q=module add xslt2]** WIKIPEDIA [http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=module add xslt2]

Dokumentace ke xslt2

- Článek ve Zpravodaji ÚVT MU -- DocBook a jeho využití [http://www.ics.muni.cz/zpravodaj/clanky_tisk/306.pdf] Tomáš Pitner, Jan Pavlovič, FI MU
- Návod k modulu xslt2 [http://www.fi.muni.cz/~xpavlov/xml/] Jan Pavlovič

Úvod

Co je TEI

Iniciativa směřující k vytvoření a aplikacím podpory zachycování textů různé povahy ve standardizované formě

- dnes v XML syntaxi (P5), dříve SGML (po P3) nebo obojí (P4)
- rozsáhlé značkování (ještě větší počet elementů než např. Docbook)
- lépe podporuje metadata dokumentů a jejich životní cyklus (vznik, revize)
- používá se pro různorodé dokumenty (texty pořizované na počítači, skenované texty, historické dokumenty, dokumenty v neevropských jazycích)
- značkování je modulární - lze sestavit na míru potřebám

Aplikace TEI značkování

- příklady textů v TEI [http://wiki.tei-c.org/index.php/Samples] (především XML)
- Manuál (Guidelines [http://www.tei-c.org/Guidelines/P5/]) pro TEI P5

Co je Darwin Information Typing Architecture (DITA)?

Darwin Information Typing Architecture (DITA)

IBM a následně konsorcium OASIS zavedlo architekturu DITA [http://docs.oasis-open.org/dita/v1.0/archspec/ditaspec.toc.html] jako:

- Nástroj pro tvorbu tematicky orientovaného značkovaného obsahu s možností specializace pro zvláštní účely.
- Není to, na rozdíl např. od Docbooku, jedno pevné značkování.
- Využívá se principů podobných jako v objektových jazycích.
- Specializace znamená podědit vlastnosti (např. formátování) a konkretizovat je.
- Používá se tam, kde se tvoří rozsáhlý, vysoce strukturovaný, znovupoužitelný obsah s přesně vymezenou sémantikou.

Historie a současnost

- od roku 2001 DITA vyvíjena společností IBM (motivace: pevná značkování nestačí...)
- 2004 -- IBM daruje standard do správy OASIS
- O vývoj se stará *OASIS DITA Technical Committee* [http://www.oasis-open.org/committees/dita/].
- Duben 2005 -- Version 1.0 of the DITA specification:
  - OASIS Darwin Information Typing Architecture (DITA) Language Specification [http://xml.coverpages.org/DITAv10-OS-LangSpec20050509.pdf]
  - OASIS Darwin Information Typing Architecture (DITA) Architectural Specification [http://xml.coverpages.org/DITAv10-OS-ArchSpec20050509.pdf]

Základní pojmy

topic	téma -- jednotka informace daná názvem a obsahem; dostatečně malá, aby byla dále nedělitelná z hlediska obsahu a pořízení (menší už by nedávala ucelený smysl) -- např. odpověď na jednu otázku
map	dokument organizující témata do větších jednotek se zachycením vztahu mezi tématy, vč. např. obsahu
specialization	specializace -- je technika umožňující definovat nové strukturální typy nebo nové informační domény) s maximálním znovupoužitím existujícího návrhu a kódu, důraz je kladen na snižování nákladů přechodu na nové typy (výměna dat, migrace, správa)
structural vs. domain specialization	*strukturální specializace* -- umožňuje tvořit nové typy témat (topic types) nebo map (map types)

	doménová specializace -- dovoluje vznik nového značkování použitelného pro více strukturálních typů (např. nové typy klíčových slov, tabulek, seznamů)
integration	integrace -- každá doménová nebo strukturální specializace má svůj návrhový modul. Moduly mohou být při vytváření nových typů dokumentů kombinovány v procesu zvaném integrace.
customization	přizpůsobení -- např. požadujeme-li jen změnu výstupu, lze ji provést bez narušení přenositelnosti a výměny dat, bez nutnosti specializace
generalization	generalizace -- nabízí možnost chápat specializovaný obsah jako obsah nadřazeného (obecnějšího) typu dokonce s možností návrhu zpět ke specializovanému obsahu (round-tripping).

Příklad

CambridgeDocs nabízí řešení pro pořizování a správu dokumentů navržených podle DITA -- xDoc Pro [http://www.cambridgedocs.com/solutions/dita.htm].