

---

# Formátování, vizualizace. Docbook

## Obsah

Konceptuální, logické a fyzické formátování .....	2
Co a k čemu je formátování? .....	2
Úrovně formátování .....	2
Odkud kam sahají úrovně formátování? .....	2
Fáze formátování .....	3
Postup formátování a příklady nástrojů .....	3
data -> požadovaná data (filtrace) .....	3
filtrovaná data -> konceptuální formát .....	4
Konceptuální formát -> logický formát .....	4
Logický formát -> fyzický formát .....	5
Formátování a výstupní média .....	5
Výstup na běžnou obrazovku, web (HTML, plaintext, RTF) .....	6
(X)HTML, RTF, plaintext (2) .....	6
Výstup pro tisk (PDF, TeX) .....	6
Výstup na malé displeje (WAP, PDA) .....	7
Hlasový výstup (VoiceXML) .....	7
Podrobněji k formátovacím objektům (XSL:FO) .....	7
Co a k čemu jsou XSL:FO .....	8
Informační zdroje k XSL:FO .....	8
Motivace k Docbooku .....	8
DocBook: příklad složitějšího značkování .....	8
Co je Docbook? .....	9
Výhody Docbooku .....	9
Původ .....	9
Základní struktury Docbook .....	9
Ukládání .....	10
Druhy dokumentů/strukturálních prvků .....	10
Blokové prvky .....	10
Prvky na řádku .....	11
Příklad dokumentu v Docbook 5 .....	11
Totéž v Docbooku 4.4 .....	11
Varianty Docbook .....	12
Používat verze 5 nebo 4.x? .....	12
DocBook: vrstvy a přizpůsobení .....	12
Vrstvy Docbooku - Simplified .....	12
Docbook Slides .....	13
Další moduly .....	13
Nástroje pro Docbook .....	13
Požadavky na editory .....	13
Dostupné editory .....	14
Transformační nástroje .....	14

Docbook prakticky .....	14
Balík xslt2 .....	14
Dokumentace ke xslt2 .....	15
Co je Darwin Information Typing Architecture (DITA)? .....	15
Darwin Information Typing Architecture (DITA) .....	15
Historie a současnost .....	15
Základní pojmy .....	16
Příklad .....	16

## Konceptuální, logické a fyzické formátování

### Co a k čemu je formátování?

XML data jsou sice částečně „lidsky čitelná“, ale pro zpracování člověkem vyžadují transformaci do kvalitně zobrazitelné podoby - *formátování*.

Formátování je proces transformace primárních dat do lidsky čitelné „prezentovatelné“ podoby, určené k zobrazení, hlasové interpretaci, atd.

### Úrovně formátování

Rozlišujeme formátování na úrovni:

konceptuální	maximum sémantiky v datech je zachováno, je přidána formátovací sémantika - např. struktura publikace, členění na kapitoly, sekce; datový obsah je vyfiltrován pro účely zobrazení (např. metadata jsou redukována). Příklad formátu - DocBook [ <a href="http://docbook.org">http://docbook.org</a> ].
logické	původní sémantiku již nelze rekonstruovat, je přidána další formátovací sémantika - např. členění na nadpisy, tabulky, styly, přibývají instrukce pro volbu písma, barvu, odsazení, mezery mezi odstavci... Formátovaný dokument je stále relativně univerzálně zobrazitelný a přenositelný. Příklad formátu - HTML [ <a href="http://w3.org/HTML">http://w3.org/HTML</a> ].
fyzické	formátovaný dokument je v podstatě přesným popisem, jak vytisknout, zobrazit, hlasově přečíst dokument. Příklad formátu - PostScript, PDF.

### Odkud kam sahají úrovně formátování?

- Hranice mezi jednotlivými úrovněmi nejsou ostré.
- Dokument na *vyšších formátovacích úrovních* (konceptuální) lze dále transformovat, zpracovávat...  
Část sémantiky původních dat je však již ztracena a naopak formátovací sémantika je doplněna.  
Viz např. formát DocBook - dokument v DocBooku už je článek, kniha, přednáška ve formě slidů...

což původní zdrojová data nebyla.

- Dokument formátovaný do formátů *nižších úrovní* (logický, fyzický) není obvykle určen k dalšímu zpracování vyjma přípravy pro tisk, zobrazení.

Viz např. HTML, PDF... (s jistými výjimkami)

## Fáze formátování

### Postup formátování a příklady nástrojů

Budeme uvažovat XML data a adekvátní nástroje se zaměřením na open-source.

#### data -> požadovaná data (filtrace)

Z primárních dat extrahujeme data potřebná. Vhodné nástroje:

filtrování proudu (SAX) událostí	poměrně nekomfortní, ale za běhu efektivní zpracování dat; pro pouhé filtrace elementů vyhoví; lze použít <code>XMLFilter</code>
manipulace se stromem dokumentu	DOM, <code>dom4j</code> a další reprezentace; náročnější na paměť i čas; dovoluje složitější manipulace
XSLT transformace	mocné, ale poněkud náročnější na zvládnutí
dotazovacím jazykem	vhodné řešení, ale vyžaduje stroj na interpretaci

#### Příklad 1. Příklad XSLT extrakce

Ze všech programátorů v Devguru chceme zobrazit jen mladší 30 let:

```
<?xml version="1.0" encoding="Windows-1250"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0" xmlns:saxon="http://icl.com/saxon"
  extension-element-prefixes="saxon">
  <xsl:strip-space elements="*" />
  <xsl:output method="xml" encoding="Windows-1250" indent="no" />
  <xsl:template match="programmer[number(age) < 30]">
    <xsl:copy-of select="." />
  </xsl:template>
  <xsl:template match="programmer"/>
  <xsl:template match="/">
    <young_programmers>
      <xsl:apply-templates/>
    </young_programmers>
  </xsl:template>
</xsl:stylesheet>
```

```
    </young_programmers>
  </xsl:template>
</xsl:stylesheet>
```

## filtrovaná data -> konceptuální formát

### Příklad 2. XSLT transformace do DocBooku

O všech mladších programátorech z Devguru napiš souhrnnou zprávu (styl je zkrácen):

```
<xsl:output method="xml" encoding="Windows-1250" indent="no"
  doctype-system="http://www.oasis-open.org/docbook/xml/simple/1.0/sdocbook.dtd"
  doctype-public="-//OASIS//DTD Simplified DocBook XML V1.0//EN"/>
<xsl:template match="programmer">
  <row>
    <entry>
      <xsl:value-of select="name"/>
    </entry>
    <entry>
      <xsl:value-of select="age"/>
    </entry>
  </row>
</xsl:template>
<xsl:template match="/">
  <article>
    <title>Seznam pracovníků mladších 30 let</title>
    <table>
      <title>Tabulka pracovníků mladších 30 let</title>
      <tgroup cols="2">
        <tbody>
          <row><entry>jméno</entry><entry>věk</entry></row>
          <xsl:apply-templates/>
        </tbody>
      </tgroup>
    </table>
  </article>
</xsl:template>
...
```

## Konceptuální formát -> logický formát

### Příklad 3. XSLT transformace do DocBooku

---

Zprávu o mladších programátorech formátuj do HTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Seznam pracovn&iacute;k&#367; mlad&scaron;&iacute;ch 30 let</title>
    <link rel="stylesheet" href="html.css" type="text/css">
    <meta name="generator" content="DocBook XSL Stylesheets V1.60.1">
  </head>
  <body bgcolor="white" text="black" link="#0000FF" vlink="#840084" alink="#0000FF">
  <div class="article" lang="en">
  <div class="titlepage">
    <div><div>
      <h2 class="title">
        <a name="d0e1"></a>Seznam pracovn&iacute;k&#367; mlad&scaron;&iacute;ch 30 let</h2>
      </div></div>
    <hr>
  </div>
  <div class="table">
    <a name="d0e4"></a>
    <p class="title">
      <b>Table 1. Tabulka pracovn&iacute;k&#367; mlad&scaron;&iacute;ch 30 let</b>
    </p>
    <table summary="Tabulka pracovn&iacute;k&#367; mlad&scaron;&iacute;ch 30 let">
    <colgroup>
      <col>
      <col>
    </colgroup>
    <tbody>
      <tr><td> jm&eacute;no </td> <td> v&#283;k</td> </tr>
      <tr> <td> Minnie Mouse</td> <td> 24 </td> </tr>
    </tbody>
    </table>
  </div>
</div>
</body>
</html>
```

## Logický formát -> fyzický formát

Výsledek zobrazený v prohlížeči

## Formátování a výstupní média

---

## Výstup na běžnou obrazovku, web (HTML, plaintext, RTF)

- k transformaci pro tato média lze použít XSLT (do plaintextu, HTML, XHTML...), speciálních nástrojů (např. RTF FormattingKit, viz RTF FormattingKit web [<http://www.schema.de/sitehtml/site-e/xmlnach0.htm>] pro RTF)
- pokud je formátování určeno k prohlížení na webu (prohlížečem), je možné pro specifikaci stylu použít konvenci (<?xml-stylesheet type="text/xsl" href="somestylesheet.xml"?>)
- totéž lze použít i pro CSS styly
- pro XML je použitelná verze CSS2

## (X)HTML, RTF, plaintext (2)

- k transformaci pro tato média lze použít XSLT (do plaintextu, HTML, XHTML...), speciálních nástrojů (např. RTF FormattingKit, viz RTF FormattingKit web [<http://www.schema.de/sitehtml/site-e/xmlnach0.htm>] pro RTF)
- pokud je formátování určeno k prohlížení na webu (prohlížečem), je možné pro specifikaci stylu použít konvenci typu <?xml-stylesheet type="text/xsl" href="somestylesheet.xml"?>.
- přesný popis této konvence [<http://web3.w3.org/TR/xml-stylesheet/>]
- totéž lze použít i pro styly CSS [<http://www.w3.org/Style/CSS>] (pro XML je použitelná verze CSS2)
- dobré informační zdroje k použití CSS pro XML:
  - specifikace CSS2 [<http://www.w3.org/Style/CSS>]
  - CSS 2 Tutorial (zvon.org) [<http://www.zvon.org/xx1/CSS2Tutorial/General/htmlIntro.html>]
  - Building Documents with XML, XSL, and CSS [<http://www.siteexperts.com/tips/xml/ts01/page1.asp>]
  - displaying xml: working with cascading style sheets [<http://www.javertising.com/webtech/cssxml.htm>]

Validace dokumentů s CSS styly: CSS Validátor na W3C: <http://jigsaw.w3.org/css-validator/>

## Výstup pro tisk (PDF, TeX)

- obvyklý postup:
  - transformace XML -- XSL:FO pomocí XSLT

- rendering XSL:FO do PDF/PS pomocí Apache FOP nebo jiného (komerčního) nástroje
- nebo přes PassiveTex do TeXového zdroje a pak standardní TeXovou cestou (TeX -- DVI -- PS nebo TeX -- PDF s pomocí pdfTeXu)
- jak může vypadat celý postup zpracování, ukazuje návod J. Pavloviče k modulu xslt2 [<http://www.fi.muni.cz/~xpavlov/xml/>]

## Výstup na malé displeje (WAP, PDA)

- pro mobilní telefony s WAP: v podstatě obdobné jako pro web, jazykem popisu WAP stránek je WML [<http://www.wapforum.org/what/technical.htm>]
- použijí se typicky opět XSLT transformace
- pro PDA: typicky nemají on-line spojení, připojují se "občas" přes stolní počítač zapojený do internetu
- proto potřebují systém na stažení, kompresi a off-line prohlížení webových stránek
- možným řešením je AvantGo.com [<http://avantgo.com>]: pro uživatele zdarma, pro poskytovatele obsahu je to komerční systém (omezené použití (málo uživatelů) zdarma)
- AvantGo systém je rozdělen na serverovou část (na výše uvedené adrese) a klientskou část rozdělenou na desktop (menší část) a PDA počítač (větší část)
- technicky je AvantGo schopno přizpůsobit a zobrazit "skoro normální HTML"
- problémy jsou pochopitelně s interaktivními aplikacemi (formuláře) -- částečně lze řešit tzv. Form Managerem
- bližší informace [<http://www.avantgo.com/support/>] k technice tvorby webů pro AvantGo

## Hlasový výstup (VoiceXML)

VoiceXML je značkovací jazyk určený k popisu dialogu vedeného hlasovým rozhraním

Dave Raggett's Introduction to VoiceXML 2.0 [<http://www.w3.org/Voice/Guide>]

VoiceXML - Tutorials [<http://www.voicexml.org/tutorials/>]

VoiceXML Tutorial [<http://cafe.bevocal.com/docs/tutorial/>]

...další informace spolu se systémem Elvira (vyvíjeným na FI) najdete na domovské stránce [[http://gin2.itek.norut.no/elvira/\\_elvira.php?p=introduction](http://gin2.itek.norut.no/elvira/_elvira.php?p=introduction)] projektu

## Podrobněji k formátovacím objektům (XSL:FO)

---

## Co a k čemu jsou XSL:FO

- *XSL:Formatting Objects* (XSL:FO) je standard pro platformově neutrální, v XML zapsaný popis (tiskového) formátu publikací
- dány standardem W3C - viz <http://www.w3.org/Style/XSL/>
- k transformaci z XML do XSL:FO se typicky využívá XSLT (proto původně v jedné specifikaci)
- dalším krokem bývá transformace XSL:FO do PDF/PS např. nástrojem Apache FOP [<http://xml.apache.org/fop/index.html>].
- obvykle kvalitnějšího výstupu dosáhneme použitím komerčního nástroje, který také podporuje širší škálu formátovacích objektů. Viz např. RenderX XEP [<http://www.renderx.com/>].

## Informační zdroje k XSL:FO

- dobrý úvodní článek What is XSL:FO [<http://www.xml.com/pub/a/2002/03/20/xsl-fo.html>] (Ken Hollman, XML.COM)
- kvalitní XSL FO Tutorial [<http://www.renderx.com/tutorial.html>] (firma RenderX)
- Stylesheet Tutorial, Sample Files of Formatting Objects and Sample Stylesheets [<http://www.antennahouse.com/XSLsample/XSLsample.htm>]

## 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](http://~/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 označované 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 spojit 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://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=.dbk>], `.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 ( <code>book</code> ) nebo dalších kolekcí -- kolekce lze vnořovat.
book	kniha, sestává z kapitol ( <code>chapter</code> ), článků ( <code>article</code> ) nebo částí ( <code>part</code> ), 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 západ-

ních 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>!</para>
  </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 nevhodících 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) XML dokument.
- 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](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 po-

doby prostředky TeXu.

- Autorem je *Jan Pavlovič*.
- Balík `xslt2` je dostupný na unixových strojích FI zadáním `module add xslt2`  [\[http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=module+add+xslt2\]](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 [\[http://www.ics.muni.cz/zpravodaj/clanky\\_tisk/306.pdf\]](http://www.ics.muni.cz/zpravodaj/clanky_tisk/306.pdf) Tomáš Pitner, Jan Pavlovič, FI MU
- Návod k modulu xslt2 [\[http://www.fi.muni.cz/~xpavlov/xml/\]](http://www.fi.muni.cz/~xpavlov/xml/) Jan Pavlovič

# Co je Darwin Information Typing Architecture (DITA)?

## Darwin Information Typing Architecture (DITA)

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

- Nástroj pro tvorbu tematicky orientovaného značkováného 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	<i>strukturální specializace</i> -- umožňuje tvořit nové typy témat (topic types) nebo map (map types)  <i>doménová specializace</i> -- 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>].