
Formátování a vizualizace

Obsah

Konceptuální, logické a fyzické formátování	1
Co a k čemu je formátování?	1
Úrovně formátování	1
Odkud kam sahají úrovně formátování?	2
Příklad - Docbook	2
DocBook: příklad složitějšího značkování	2
DocBook: vrstvy a přizpůsobení	3
DocBook: styly	3
Fáze formátování	3
Postup formátování a příklady nástrojů	3
data -> požadovaná data (filtrace)	3
filtrovaná data -> konceptuální formát	4
Konceptuální formát -> logický formát	5
Logický formát -> fyzický formát	6
Formátování a výstupní média	6
Výstup na běžnou obrazovku, web (HTML, plaintext, RTF)	6
(X)HTML, RTF, plaintext (2)	6
Výstup pro tisk (PDF, TeX)	7
Výstup na malé displeje (WAP, PDA)	7
Hlasový výstup (VoiceXML)	8
Podrobněji k formátovacím objektům (XSL:FO)	8
Co a k čemu jsou XSL:FO	8
Informační zdroje k XSL:FO	8

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 [http://docbook.org].
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 [http://w3.org/HTML].
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)

Příklad - Docbook

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]

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

DocBook: styly

- k vizualizaci (a převodu do tiskových formátů) z napsaného DocBookového dokumentu lze použít XSL styly
- XSL styly jsou na <http://docbook.sf.net>
- v rámci toho je i přizpůsobení DocBook Slides
- styly formátují do HTML, XHTML, XSL:FO, ...
- zpracování lze parametrizovat bez přepisování stylů

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, dom4j 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>
```

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>
  </row>
</xsl:template>
```

```

        <entry>
          <xsl:value-of select="age"/>
        </entry>
      </row>
    </xsl:template>
  <xsl:template match="/">
    <article>
      <title>Seznam pracovníků mladších 30 let</title>
      <table>
        <caption>Tabulka pracovníků mladších 30 let</caption>
        <tgroup cols="2">
          <tbody>
            <tr><td>jméno</td><td>věk</td></tr>
            <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 pracovníků mladších 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 pracovníků mladších 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<
  </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="somesylesheet.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 ná-

strojů (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/xxl/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] 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>]