Zpravodaj Československého sdružení uživatelů TeXu

Jiří Kosek
PassiveTeX

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 10 (2000), No. 4, 132–143

Persistent URL: http://dml.cz/dmlcz/150250

Model (DOM) Level 2 Core Specification [http://www.w3.org/TR/DOM-Level-2-Core/]. W3C, 2000.

[SAX] Megginson, David: SAX 2.0: The Simple API for XML [http://www.megginson.com/SAX/]. 2000.

[W3C] Stránky konsorica W3C. http://www.w3.org/

[XLINK] DeRose Steve, Maler Eve, Orchard David: XML Linking Language (XLink) [http://www.w3.org/TR/xlink]. W3C, 2000.

[XMLCS] Kosek, Jiří: XML pro každého [http://www.kosek.cz/xml/]. Grada Publishing, 2000.

[XMLSPEC] Bray Tim, Paoli Jean, Sperberg-McQueen C. M.: Extensible Markup Language (XML) 1.0 [http://www.w3.org/TR/REC-xml]. W3C, 1998.

[XPTR] DeRose Steve, Daniel Ron Jr., Maler Eve: XML Pointer Language (XPointer) [http://www.w3.org/TR/xptr]. W3C, 2001.

[XSDSPEC] Fallside, David C.: XML Schema Part 0: Primer [http://www.w3.org/TR/xmlschema-0/]. W3C, 2000.

[XSL] Adler Sharon, Berglund Anders, Caruso Jeff, Deach Stephen, Grosso Paul, Gutentag Eduardo, Milowski Alex, Pernell Scott, Richman Jeremy, Zilles Steve: Extensible Stylesheet Language (XSL) – Version 1.0 [http://www.w3.org/TR/xsl]. W3C, 2000.

[XSLT] Clark, James: XSL Transformations (XSLT) Version 1.0 [http://www.w3.org/TR/xslt]. W3C, 1999.

Summary: Gentle Introduction to XML

XML language brings many revolutional changes into the area of electronic publishing, information sharing and interchange and e-business. This article introduces basic XML principles and its relation to related technologies (stylesheet langauges, document type definition langauges, query languages and linking languages).

PassiveTEX

Jiří Kosek

PassiveTEX je sada maker, která umožňuje formátování XML dokumentů. Pro popis výsledného vzhledu dokumentu se přitom používá standardní stylový jazyk XSL (eXtensible Stylesheet Language). Tvorba stylů v XSL je pro

mnoho uživatelů snazší než přímé programování v TEXu. PassiveTEX zcela odstiňuje uživatele od makrojazyka TEXu a umožní tak využít kvalitní výstup TEXu širšímu okruhu uživatelů než dnes. Během přednášky se posluchači seznámí se základními principy XSL a shlédnou praktické ukázky použití PassiveTEXu.

Úvod

XML je formát vhodný pro ukládání mnoha různých druhů dat. Data je však nutné prezentovat uživateli, XML dokument proto musíme zformátovat. Pro formátování XML dokumentů existuje speciální jazyk XSL (eXtensible Stylesheet Language). Jazyk XSL obsahuje dvě části – transformační jazyk XSLT a formátovací objekty (FO). Transformační část jazyka umožňuje definovat transformaci do XML dokumentu s jinou strukturou i značkami. Formátovací část jazyka pak definuje speciální jazyk s XML syntaxí, který definuje jednotlivé objekty zformátovaného dokumentu a jejich vizuální vlastnosti.

Formátování XML dokumentu probíhá tedy tak, že se nejprve provede transformace do formátovacích objektů. V dalším kroku se zpracují formátovací objekty a dostaneme výsledek – ať už přímo na obrazovce nebo třeba v PDF souboru.

Programů, které umějí provádět transformace definované pomocí XSLT, je dnes již mnoho. O to větší nouze je však o programy, které umějí dokument s formátovacími objekty zalomit do výsledných stránek. Jedním z těchto programů je i PassiveTEX, který při formátování využívá schopnosti typografického systému TEX. V tomto příspěvku se nejprve podíváme na princip fungování XSL stylů a poté si ukážeme některé možnosti PassiveTEXu.

XSLT aneb není nad dobrého sluhu

Základní myšlenka XSLT je velice jednoduchá. Styl obsahuje šablony, které slouží pro zpracování určité části (většinou určitého elementu) vstupního dokumentu. XSLT procesor postupně prochází vstupní dokument, a pokud najde šablonu, která umí obsloužit danou část dokumentu, provede ji. Toto jednoduché chování lze samozřejmě ovlivňovat mnoha způsoby – části šablon lze provádět podmíněně, části vstupního dokumentu můžeme před zpracováním setřídit apod. Na podrobný popis možností XSLT zde bohužel nemáme prostor.

Styl v XSLT nepoužívá žádnou speciální syntaxi, jedná se o XML dokument. V tomto dokumentu se používají vždy dvě sady značek – instrukce pro XSLT procesor a značky, které se mají objevit ve výstupním dokumentu. Když používáme XSLT pro převod z XML do HTML, obsahuje styl XSLT instrukce a HTML

tagy. Pokud provádíme transformaci do formátovacích objektů, obsahuje XSLT styl kromě instrukcí ještě elementy formátovacích objektů. Kombinování více různých sad značek v jednom dokumentu je umožněno díky standardnímu mechanismu jmenných prostorů, který je součástí XML.

Základní princip fungování XSLT si ukážeme na jednoduchém příkladě. Dejme tomu, že máme v XML dokumentu uloženy informace o přednáškách konaných v rámci SLT (viz příklad 1). Nyní chceme vytvořit HTML stránku, kde bude program ve formě tabulky se čtyřmi sloupci, ve kterých bude postupně název přednášky, jméno autora, vysílající firma a abstrakt.

Příklad 1: XML podoba seznamu přednášek – slt.xml

```
<?xml version="1.0" encoding="iso-8859-2"?>
<prednasky>
  <akce>SLT 2001</akce>
  <prednaska>
    <nazev>Clusterová řešení na Linuxu</nazev>
    <autor>Jan Kasprzak</autor>
    <firma>FI MU Brno</firma>
    <abstrakt>Operační systém Linux je svojí modularitou a dostupností
      zdrojových textů vhodný i jako součást rozsáhlejších systémů.
      V přednášce budou popsány základní typy clusterů &#x2013;
      výpočetní clustery, load-balancing clustery a high availability
      clustery. Dále software, pomocí kterého je možno jednotlivé typy
      clusterů realizovat a některé zkušenosti s prací
      clusterů.</abstrakt>
  </prednaska>
  <prednaska>
    <nazev>BIRD Internet Routing Daemon</nazev>
    <autor>Martin Mareš</autor>
    <firma>SuSE ČR</firma>
    <abstrakt>Jádro Linuxu od nepaměti (i z pohledu kernelových
      hackerů, a to už je co říci) disponuje velice rychlou a flexibilní
      implementací TCP/IP. K tomu, aby se z Linuxu stal špičkový router
      použitelný i ve velkých sítích, ovšem chybí kvalitní daemon
      implementující dynamické routovací protokoly (BGP, OSPF, ...)
      a předávající zkonstruované routovací tabulky jádru. Před časem
      vznikl na MFF UK projekt BIRD, který se snaží tuto mezeru zaplnit
      a přinést (nejen) Linuxu i další síťové vymoženosti, ve světě
      &#x201e;opravdových&#x201c; routerů zatím velice řídké: dynamické
      rekonfigurování, více routovacích tabulek, BGP multihoming
      a programovatelnou filtraci, to vše jak pro IPv4, tak i IPv6. Více
      se dočtete na http://bird.network.cz/.</abstrakt>
  </prednaska>
  <prednaska>
    <nazev>Využití XML rozhraní při tvorbě aplikací</nazev>
    <autor>Jan Pazdziora</autor>
    <firma>FI MU Brno</firma>
    <abstrakt>Oddělení obsahu a prezentace je módní slogan, bohužel
```

```
      většina řešení skončí u některého z template
      přístupů. Prezentované řešení nad AxKitem využívá XML jako vrstvu,
      která generování dat od formátování striktně odstiňuje. Zaměříme
      se zejména na aplikace pracující s dynamickým obsahem
      a prozkoumáme, jak vypadá dobře strukturované XML a jakých rozličných
      výstupů je možno z jedněch databázových dat dosáhnout.</abstrakt>
  </prednaska>
  <prednaska>
    <nazev>IP verze 6</nazev>
    <autor>Pavel Satrapa</autor>
    <firma>TU Liberec</firma>
    <abstrakt>Základní seznamení s vlastnostmi nové verze
      IP. Adresovaní, směrovaní, bezpečnostní mechanismy, podpora
      mobilních zařízení. Aktuální stav implementací IPv6 a především
      jeho implementace v Linuxu.</abstrakt>
  </prednaska>
</prednasky>
```

Vytvořit takovou transformaci je velice jednoduché. V podstatě stačí „přejmenovat“ naše XML tagy na HTML tagy pro tvorbu tabulek. Příklad 2 obsahuje ukázku jednoduchého stylu, který můžeme použít pro převod do HTML. Na obrázku 1 si pak můžeme prohlédnout výsledek v prohlížeči. (Pevně doufám, že fundamentální příznivci Linuxu přimhouří oko nad obrázkem pořízeným ve Windows, pro jejich uklidnění jsem alespoň použil prohlížeč Mozilla místo Internet Exploreru;–)

Příklad 2: Styl pro konverzi do HTML

```
<?xml version="1.0" encoding="iso-8859-2"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

<xsl:output method="html"/>

<xsl:template match="prednasky">
  <html>
    <head><title><xsl:value-of select="akce"/></title></head>
    <body>
      <xsl:apply-templates select="akce"/>
      <table border="1">
        <tr>
          <th>Název</th>
          <th>Přednášející</th>
          <th>Firma</th>
          <th>Abstrakt</th>
        </tr>
        <xsl:apply-templates select="prednaska"/>
      </table>
    </body>
  </html>
</xsl:template>
```

```
<xsl:template match="akce">
  <h1 align="center"><xsl:apply-templates/></h1>
  <h2 align="center">Program akce</h2>
</xsl:template>

<xsl:template match="prednaska">
  <tr><xsl:apply-templates/></tr>
</xsl:template>

<xsl:template match="nazev">
  <td><b><xsl:apply-templates/></b></td>
</xsl:template>

<xsl:template match="autor">
  <td><xsl:apply-templates/></td>
</xsl:template>

<xsl:template match="firma">
  <td><xsl:apply-templates/></td>
</xsl:template>

<xsl:template match="abstrakt">
  <td><small><xsl:apply-templates/></small></td>
</xsl:template>

</xsl:stylesheet>
```

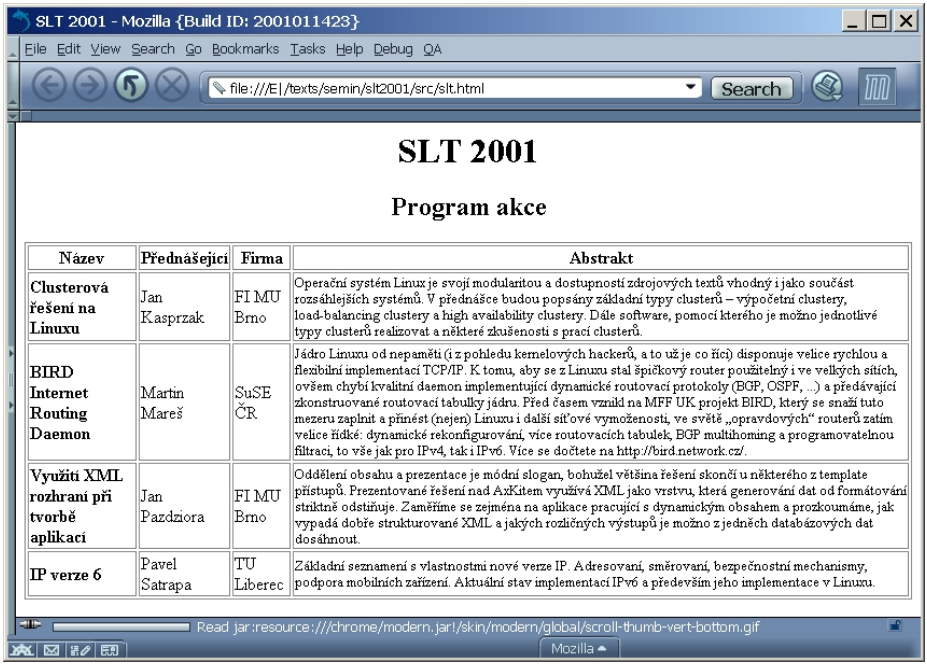
Výhoda tohoto přístupu spočívá v tom, že stylů si můžeme vytvořit několik. Můžeme vytvořit styl, který místo do tabulky uspořádá údaje přehledně pod sebe, nebo například příspěvky nejprve seřadí podle jejich názvu.

Pokud si chceme práci s XSLT vyzkoušet, musíme mít k dispozici nějaký XSLT procesor. Dnes jich existuje několik, mezi asi tři nejznámější open-source procesory patří Saxon, Xalan a XT. Odkazy na ně jsou uvedeny na konci příspěvku. Novější verze Mozilly mají XSLT procesor zabudovaný přímo do sebe.

Formátovací objekty

Formátovací objekty abstraktním způsobem popisují výsledný vzhled dokumentu. Formálně jsou objekty zapsány jako XML dokument, který obsahuje speciální elementy odpovídající jednotlivým formátovacím objektům.

Pokud situaci zjednodušíme, obsahuje soubor s formátovacími objekty nejprve definici layoutu jednotlivých typů stránek použitých v dokumentu – jejich rozměry, velikost okrajů, počet sloupců apod. Za ní jsou už uloženy elementy, které zastupují obsah jednotlivých stránek. XSL procesor pak stránky na výstupu vytváří tak, že do prostoru definovaného layoutem stránky umisťuje obsah těchto elementů.



Obrázek 1: HTML stránka vytvořená z XML pomocí XSLT stylu

Na nejnižší úrovni se pak používají objekty, které zastupují základní druhy formátovaného výstupu. Mezi ty nejpoužívanější patří:

`block` Objekt odpovídá blokovým elementům, které známe z kaskádových stylů. Typicky se používá se pro odstavce, nadpisy apod.

`external-graphic` Objekt zastupuje obrázek, který je uložen mimo výsledný dokument formátovacích objektů. Obvykle je obrázek uložen v externím souboru (např. GIF, JPEG, PNG, EPS apod.).

`float` Plovoucí objekt – umístí se na vhodné místo stránky. Obvykle se používá pro obrázky a tabulky případně pro sazbu poznámek vedle textu (marginálií).

`footnote`, `footnote-citation` Objekty se používají pro poznámky pod čarou.

`inline` Formátovací objekt nezpůsobující vznik nového odstavce. Používá se například pro změny druhu písma uvnitř odstavce.

`leader` Objekt se používá pro čáry nebo opakované znaky (nejčastěji tečky), které mají vyplnit daný prostor. Používá se například v obsahu pro oddělení názvu kapitoly od čísla strany.

`list-block`, `list-item`, `list-item-body`, `list-item-label` Objekty se používají pro seznamy.

`basic-link` Umožňuje do výsledného dokumentu zařadit odkazy.

`table`, `table-*` Několik objektů, které umožňují vytváření tabulek.

`marker`, `retrieve-marker` Objekty umožňují vytváření záhlaví a zápatí, které obsahují proměnlivé texty – např. názvy kapitol a podkapitol.

`page-number`, `pagenumber-citation` Objekty umožňují generování čísla stránky a čísla stránky s určitým objektem.

`wrapper` Objekt se používá v případech, kdy je potřeba pro několik objektů nastavit společné vlastnosti.

U každého z těchto objektů je možné nastavovat několik desítek parametrů, které ovlivní formátování – velikost a typ písma, okraje, různé velikosti odsazení, způsob zarovnání textu, barvy atd. Možnosti formátovacích objektů jsou kompletně popsány ve specifikaci XSL [XSL].

Specifikace XSL se již blíží do finálního stádia, a existuje již i několik jejích implementací. Komerční implementace nabízí firmy RenderX, Unicorn a ArborText, mezi open-source implementace patří FOP a PassiveTEX. Míra podpory formátovacích objektů je v jednotlivých implementacích různá, žádná implementace zatím není úplná. S formátovačem od ArborTextu jsem nepracoval, z těch ostatních je pouze PassiveTEX schopen zpracovat dokumenty obsahující české znaky s diakritikou. Ostatní procesory neobsahují potřebné mapovací tabulky ze znakové sady XML do jednotlivých fontů.

PassiveTEX

Pro první pokusy s formátovacími objekty je PassiveTEX jedním z nejvhodnějších kandidátů. Kromě toho, že je zadarmo a obsahuje poměrně slušnou podporu formátovacích objektů, je schopen sázet české znaky a používat české (a samozřejmě i slovenské) vzory pro dělení slov. Náskok PassiveTEXu oproti ostatním technologiím je možný díky tomu, že PassiveTEX využívá formátovací jádro TEXu a některé možnosti LATEXu.

Jak tedy PassiveTEX funguje? První věc, kterou je potřeba v PassiveTEXu vyřešit, je načítání XML dokumentu s formátovacími objekty. Využívá se xmltex, což je parser napsaný Davidem Carlislem přímo v TEXu. Tento parser umožňuje definovat pro každý element sekvenci TEXových příkazů, které se mají provést na začátku a na konci elementu.

PassiveTEX pak definuje sadu maker, která jsou vystavená nad nízkoúrovňovými příkazy LATEXu a jsou namapovaná na jednotlivé formátovací objekty. Díky PassiveTEXu můžeme TEX použít jako jeden z procesorů pro zpracování formátovacích objektů. Můžeme tak dostat všechny výstupní formáty, které podporuje TEX. V dnešní době je nejatraktivnější výstup do formátu PDF, který je možný díky pdfTEXu.

Malá ukázka použití

Dejme tomu, že náš přehled přednášek budeme chtít zformátovat do tištěné podoby. Vytvoříme proto styl, který převede XML dokument na formátovací objekty. Styl musí obsahovat XSLT instrukce a formátovací objekty. Instrukce řídí zpracování stylu a v našem případě například seřadí přednášky podle názvu.

Příklad 3: Styl pro převod do formátovacích objektů – styl.xsl

```
<?xml version="1.0" encoding="iso-8859-2"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:fo="http://www.w3.org/1999/XSL/Format"
                version="1.0">

<xsl:template match="prednasky">
  <fo:root>
    <fo:layout-master-set>
      <fo:simple-page-master master-name="page"
                             page-height="100mm" page-width="148.5mm"
                             margin-top="10mm" margin-bottom="10mm"
                             margin-left="10mm" margin-right="10mm">
        <fo:region-body
                        margin-top="0mm" margin-bottom="0mm"
                        margin-left="0mm" margin-right="0mm"/>
      </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence master-name="page">
      <fo:flow flow-name="xsl-region-body">
        <xsl:apply-templates select="akce"/>
        <xsl:apply-templates select="prednaska">
          <xsl:sort select="nazev"/>
        </xsl:apply-templates>
      </fo:flow>
    </fo:page-sequence>
  </fo:root>
</xsl:template>

<xsl:template match="akce">
  <fo:block text-align="center" font-family="Helvetica"
            font-size="12pt" space-after="4pt">
    <xsl:apply-templates/>
  </fo:block>
  <fo:block text-align="center" font-family="Helvetica"
            font-size="8pt" space-after="5pt">Program akce</fo:block>
</xsl:template>

<xsl:template match="prednaska">
  <fo:block text-align="justify" space-before="6pt"
            font-size="7pt" font-family="Times Roman">
    <fo:block text-align="start" font-weight="bold" font-family="Helvetica">
      <xsl:value-of select="nazev"/>
```

```
    </fo:block>
    <fo:block text-align="start">
      <fo:inline font-weight="bold"><xsl:value-of select="autor"/></fo:inline>
      <xsl:if test="firma">
        <fo:inline font-style="italic">
          (<xsl:value-of select="firma"/>)</fo:inline>
      </xsl:if>
    </fo:block>
    <xsl:apply-templates select="abstrakt"/>
  </fo:block>
</xsl:template>

<xsl:template match="abstrakt">
  <fo:block font-size="6pt" space-before="1pt">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>

</xsl:stylesheet>
```

Nejprve musíme pomocí tohoto stylu převést vstupní XML dokument `slt.xml` na XML dokument s formátovacími objekty. K tomu potřebujeme nějaký XSLT procesor. Například s procesorem XT [XTCS] můžeme transformaci provést následujícím příkazem:

```
xt slt.xml styl.xsl slt.fo
```

Dostaneme dokument s formátovacími objekty (viz ukázka 4), který můžeme zformátovat libovolným procesorem formátovacích objektů. V našem případě použijeme PassiveTEX. Pro jeho spuštění je potřeba nějaká novější distribuce TEXu, například TEXLive 5. Ten obsahuje i xmltex, je však dobré stáhnout si vždy nejnovější verzi PassiveTEXu [PTEX], protože se neustále rychle vyvíjí. K zformátování do PDF (výsledek je na obrázku 2) pak použijeme příkaz:

```
pdfxmltex slt.fo
```

Příklad 4: Výsledný dokument s formátovacími objekty

```
<?xml version="1.0" encoding="utf-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="page" page-height="100mm"
                           page-width="148.5mm" margin-top="10mm"
                           margin-bottom="10mm" margin-left="10mm"
                           margin-right="10mm">
      <fo:region-body margin-top="0mm" margin-bottom="0mm"
                           margin-left="0mm" margin-right="0mm"/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-name="page">
    <fo:flow flow-name="xsl-region-body">
```

```
      <fo:block text-align="center" font-family="Helvetica"
                font-size="12pt" space-after="4pt">SLT 2001</fo:block>
      <fo:block text-align="center" font-family="Helvetica"
                font-size="8pt" space-after="5pt">
        Program akce
      </fo:block>
      <fo:block text-align="justify" space-before="6pt"
                font-size="7pt" font-family="Times Roman">
        <fo:block text-align="start" font-weight="bold"
                  font-family="Helvetica">BIRD Internet Routing Daemon</fo:block>
        <fo:block text-align="start">
          <fo:inline font-weight="bold">Martin Mareš</fo:inline>
          <fo:inline font-style="italic">(SuSE ČR)</fo:inline>
        </fo:block>
        <fo:block font-size="6pt" space-before="1pt">Jádro Linuxu od
          nepaměti (i z pohledu kernelových hackerů, a to už je co
          říci) disponuje velice rychlou a flexibilní implementací
          ...
          jak pro IPv4, tak i IPv6. Více se dočtete na
          http://bird.network.cz/.</fo:block>
      </fo:block>

      ...

    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

Další možnosti

PassiveTEX obsahuje dvě velice užitečná rozšíření. Prvním z nich je možnost generovat automaticky záložky (bookmarks) v PDF dokumentu. Kromě formátovacích objektů XSL, rozeznává PassiveTEX další element, který umožňuje specifikovat název záložky, její úroveň a místo v dokumentu, kam má ukazovat.

Druhé rozšíření dovoluje využívat nejsilnější stránku TEXu – sazbu matematických vzorců. Existuje speciální jazyk MathML (Mathematical Markup Language), který umožňuje zapisovat matematické výrazy v syntaxi založené na XML. Této syntaxi dnes rozumí několik editorů vzorců, prohlížeče Mozilla a Amaya a i některé matematické programy. PassiveTEX umí podmnožinu MathML přímo formátovat, takže pokud chceme mít matematické vzorce na Webu i na papíře, je použití MathML a PassiveTEXu jednou z možností.

Závěr

Ačkoliv PassiveTEX zatím neimplementuje celý návrh XSL, na mnoho věcí již stačí a dá se bez problémů použít. Jeho velkou výhodou je využití kvalitních

BIRD Internet Routing Daemon
Martin Mareš *(SuSE ČR)*
Jádro Linuxu od nepaměti (i z pohledu kernelových hackerů, a to už je co říci) disponuje velice rychlou a flexibilní implementací TCP/IP. K tomu, aby se z Linuxu stal špičkový router použitelný i ve velkých sítích, ovšem chybí kvalitní daemon implementující dynamické routovací protokoly (BGP, OSPF, ...) a předávající zkonstruované routovací tabulky jádru. Před časem vznikl na MFF UK projekt BIRD, který se snaží tuto mezeru zaplnit a přinést (nejen) Linuxu i další síťové vymoženosti, ve světě „opravdových" routerů zatím velice řídké: dynamické rekonfigurování, více routovacích tabulek, BGP multihoming a programovatelnou filtraci, to vše jak pro IPv4, tak i IPv6. Více se dočtete na http://bird.network.cz/.

Clusterová řešení na Linuxu
Jan Kasprzak *(FI MU Brno)*
Operační systém Linux je svojí modularitou a dostupností zdrojových textů vhodný i jako součást rozsáhlejších systémů. V přednášce budou popsány základní typy clusterů – výpočetní clustery, load-balancing clustery a high availability clustery. Dále software, pomocí kterého je možno jednotlivé typy clusterů realizovat a některé zkušenosti s prací clusterů.

IP verze 6
Pavel Satrapa *(TU Liberec)*
Základní seznamení s vlastnostmi nové verze IP. Adresovaní, směrovaní, bezpečnostní mechanismy, podpora mobilních zařízení. Aktuální stav implementací IPv6 a především jeho implementace v Linuxu.

Využití XML rozhraní při tvorbě aplikací
Jan Pazdziora *(FI MU Brno)*
Oddělení obsahu a prezentace je módní slogan, bohužel většina řešení skončí u některého z template přístupů. Prezentované řešení nad AxKitem využívá XML jako vrstvu, která generování dat od formátování striktně odstiňuje. Zaměříme se zejména na aplikace pracující s dynamickým obsahem a prozkoumáme, jak vypadá dobře strukturované XML a jakých rozličných výstupů je možno z jedněch databázových dat dosáhnout.

Obrázek 2: Zformátovaný dokument

typografických algoritmů. Uživatel tak může využít typografický potenciál TEXu bez znalosti jeho makrojazyka, který, přiznejme si, není úplně pro každého. Využití XML a XSLT přináší navíc další výhody – úplné oddělení obsahu od vzhledu a pohodlnější transformace vstupních dat před sazbou.

Odkazy

[MATHML] Mathematical Markup Language (MathML) Version 2.0 [`http://www.w3.org/TR/MathML2`]. W3C.

[PTEX] Rahtz, Sebastian: PassiveTeX. `http://users.ox.ac.uk/~rahtz/passivetex/`

[SAXON] Kay, Michael H.: XSLT procesor Saxon. `http://users.iclway.co.uk/mhkay/saxon/`

[XALAN] XSLT procesor Xalan. `http://xml.apache.org/xalan/`

[XSL] Adler Sharon, Berglund Anders, Caruso Jeff, Deach Stephen, Grosso Paul, Gutentag Eduardo, Milowski Alex, Pernell Scott, Richman Jeremy, Zilles Steve: Extensible Stylesheet Language (XSL) – Version 1.0 [`http://www.w3.org/TR/xsl`]. W3C, 2000.

[XSLT] Clark, James: XSL Transformations (XSLT) Version 1.0 [`http://www.w3.org/TR/xslt`]. W3C, 1999.

[XT] Clark, James: XSLT procesor XT. http://www.jclark.com/xml/xt.html
[XTCS] Kosek, Jiří: XT s podporou českých kódování. http://www.kosek.cz/xml/xt/
[ZVON] Tutoriály XSLT a XSL na Zvonu. http://www.zvon.org

Summary: PassiveTEX

PassiveTEX is a macropackage suited for formatting XML documents. Layout of the output document is described by an XSL stylesheet. Creating XSL stylesheets is easier than writing TEX macros for most users. PassiveTEX hides TEX internals to a common user, but a user can still easily use the state of the art typographic features of the TEX system. This article introduces XSL and PassiveTEX usage on a few easy examples.

Nový vizuální styl CSTUGu

Zdeněk Wagner

Začátkem roku 2000 přijal výbor Československého sdružení uživatelů TEXu nové logo. Autorem loga je Antonín Strejc. Logo je složeno z písmen z fontu New Century Schoolbook a dvou linek. Linky a střední písmeno T jsou červené. Písmena symbolizují anglickou zkratku „Czechoslovak TEX Users Group“ (českou zkratku CSTUG nemá) a jsou graficky uspořádána do podoby map ČR a SR.

Černobílou podobu loga jste mohli poprvé spatřit ve Zpravodaji č. 4/1999 na straně 235. Barevná verze byla použita na nových vydáních knih „Typografický systém TEX“ a „TEXbook naruby“.

S novým logem souvisí též nový vizuální styl všech tiskovin. Základní návrh také vytvořil Antonín Strejc. Nejvýznamnější je vzhled hlavičkového dopisního papíru. Všichni členové dostali od výboru dopis v lednu roku 2001. Tento dopis již využívá nový vizuální styl.

Do jubilejního 10. ročníku vkročil i Zpravodaj s novou obálkou, navrženou Antonínem Strejcem. Nová obálka z důvodu zachování kontinuity vychází z grafického vzhledu předchozích ročníků. Nový design nejen že z obálky odstraňuje matoucí neoficiální název, ale hlavně přináší nové logo a uvolňuje dostatek prostoru pro umístění čtyř číslic roku, což řeší problém Y2K.