

## XSL

# XSL = nástroj pro prezentaci XML

**Nástroj XSL slouží pro prezentaci dokumentů zapsaných v XML. Lze jej však využít i pro řadu jiných účelů...**

## Proč XML potřebuje XSL?

Formát XML (eXtensible Markup Language) byl navržen konsorciem W3C (World Wide Web Consortium) pro reprezentaci obecných dokumentů. Je navrženo tak, aby dokumenty zapsané v XML byly jednoduše zpracovatelné programem – elementy dokumentu musí být snadno rozpoznatelné. XML je značkovací jazyk, a proto používá pro vyznačení elementů značky (markups). Značky představují zobecněné závorky, např.:

```
<blok> ... obsah elementu typu blok ... </blok>
```

kde <blok> je otevírací závorka (start tag) a </blok> je zavírací závorka (end tag). Použitelná sada závorek není formátem XML definována – je pouze definován způsob zápisu závorek. Uvažme např. fragment XML-dokumentu typu faxy: zpráva:

```
<fax> FAX Message
  <odesilatel>Karel</odesilatel>
  <adresat>Jana</adresat>
  <text> ... </text>
</fax>
```

XML umožňuje, aby popis použitelné sady závorek byl součástí dokumentu – k dokumentu lze připojit tzv. definici typu dokumentu (DTD – Document Type Definition). Tato "metadata" stanoví předepsaný obsah dokumentu. Např. DTD pro faxy by mohla vypadat následovně:

```
<?xml version="1.0" encoding="iso-8859-2" ?>
<!DOCTYPE fax [
  <!ELEMENT fax (odesilatel,adresat,text)>
  <!ELEMENT odesilatel #PCDATA >
  <!ELEMENT adresat #PCDATA >
  <!ELEMENT text #PCDATA >
]>
<fax> ... </fax>
```

Vyznačením zajímavých položek v obsahu zprávy jsme získali možnost jednoduché extrakce informací o odesilatelích, adresátech a vlastním textu zprávy. V XML však není nijak určeno, jak se mají odesilatelé, adresáti a text zobrazovat. Není tedy zcela zřejmé, jak se má dokument typu "fax" naformátovat při prezentaci. To je problém, který se snaží vyřešit XSL.

## Co to je XSL?

XSL (eXtensible Stylesheet Language) je doporučení vytvořené konsorciem W3C jako nástroj pro prezentaci dokumentů zapsaných v XML. XSL je tedy svým původním určením nástroj pro popis "stylesheets" – formátované (zobrazené nebo tištěné) verze XML-dokumentů. Je ovšem možné jej využít i pro řadu jiných účelů.

Definice XSL má dvě části:

jazyk pro popis transformace XML-dokumentů, definici světa formátovaných objektů (flow objects), ve kterém se vyjádří naformátovaný dokument (flow object tree) – XML-slovník popisující sémantiku formátování (výstupu).

Jeden XSL-dokument (XSL stylesheet) specifikuje prezentaci celé třídy XML-dokumentů – popisuje, jak se libovolná instance této třídy transformuje do výstupního formátovaného dokumentu. Naformátovaný dokument patří do světa formátovaných objektů. Formátovaný svět má své typy objektů – má svůj slovník elementů. Pro jeho vyjádření je opět použit formát XML – formátované dokumenty mají své DTD. XSL popisuje transformaci vstupního XML-dokumentu (např. libovolné faxy zprávy) do výstupního XML-dokumentu, který používá určitý formátovací slovník [např. do formátované posloupnosti stránek – používá se XML "namespace" fo (flow objects), tj. standardně se použije značka <fo:page-sequence>].

Transformace je v XSL popsána pomocí sady pravidel. Libovolný vstupní XML-dokument dané třídy je podle těchto pravidel zpracován a prezentován. Proces prezentace má dva kroky:

zdrojový XML-dokument (source tree) je transformován na výsledný dokument (result tree);

výsledný dokument je interpretován a výsledkem je formátovaný výstup (na obrazovce, papíru atd.).

Např. pravidlo pro transformaci faxové zprávy na posloupnost stránek by mohlo vypadat následovně:

```
<xsl:template match="fax">
  <fo:page-sequence
    font-family="times new roman,serif"
    font-size="12pt">
    <xsl:process-children/>
  </fo:page-sequence>
</xsl:template>
```

## Jak to řeší HTML a SGML?

XSL je navržen pro XML. Formát XML vychází ze staršího a obecnějšího standardu SGML (Standard Generalized Markup Language – ISO 8879). Pro definici zobrazení SGML-dokumentů byl rovněž vytvořen standard – DSSSL (Document Style Semantics and Specification Language). Avšak podobně jako je SGML složitější než XML, je DSSSL složitější a komplikovanější než XSL, což je pravděpodobně příčina, proč zatím nedošlo k jeho širšímu užití.

DSSSL, podobně jako XSL, definuje svět formátovaných objektů (flow objects), ve kterém se vyjádří naformátovaný dokument (flow object tree). Současně zavádí transformační prostředky pro převod SGML-dokumentu do tohoto vyjádření. Pro vyjádření formátovaného dokumentu a pro popis transformace používá DSSSL syntaxi podobnou jazyku Lisp. Naproti tomu XSL využívá syntaxi XML. Pro ukázkou jenom fragment obdobného stylu zapsaného v DSSSL:

```
(element (FAX)
  (make page-sequence
    font-size: 12pt
    font-family: 'times new roman,serif'
    (process-children)))
```

Podle standardu SGML byl navržen i formát HTML (Hyper-Text Markup Language). HTML je jazyk používající pevnou sadu značek. Sada značek HTML slouží k vyjádření přibližné prezentační podoby dokumentu. Např. značka <H1> ... </H1> slouží pro označení nadpisu na nejvyšší úrovni. Tento nadpis pak všechny prohlížeče HTML zobrazí (a vytisknou se) přibližně stejným způsobem. Přesná výstupní forma dokumentu však prostřednictvím HTML určena není.

Pro možnost přesného vyjádření stylistické podoby dokumentu byl konsorciem W3C vytvořen nástroj CSS (Cascading Style Sheets) – nástroj umožňující připojit k HTML-dokumentu i přesnou definici jeho podoby. Pokud bychom např. požadovali, aby elementy typu "H1" byly zobrazeny modrým písmem "times new roman,serif" o velikosti 14 bodů, definujeme pravidlo CSS:

```
H1 { color: blue; font-size: 14pt; font-family: times new roman,serif }
```

Taková pravidla uložíme do samostatného souboru a připojíme k HTML-dokumentu vazbou, nebo je přímo vložíme do HTML-dokumentu. Následující příklad ilustruje různé možnosti – vazbu na externí soubor (element LINK), pomocí elementu STYLE v hlavičce nebo jako atribut STYLE odstavce v těle HTML:

```
<HTML>
  <HEAD>
    <TITLE>title</TITLE>
    <LINK REL=STYLESHEET TYPE="text/css"
      HREF="http://style.com/cool" TITLE="Cool">
    <STYLE TYPE="text/css">
      H1 { color: blue }
    </STYLE>
  </HEAD>
  <BODY>
    <H1>Nadpis je modrý</H1>
    <P STYLE="color: green">Paragraf je zobrazen zeleně.</P>
  </BODY>
</HTML>
```

Poznamenejme, že definice XSL je z hlediska formátovaných objektů do značné míry konformní s CSS a s DSSSL.

## Formát XSL-dokumentu

XSL-dokument se skládá ze sady pravidel, která popisují transformaci vstupního dokumentu na výstupní. Pravidla obsahují vzorek (pattern) a akci (template). Vzorek se porovnává se zdrojovým dokumentem, při srovnání se modifikuje akce pravidla a aplikuje se na výstupní strom (zdrojový a výstupní strom jsou dva různé dokumenty). Struktura výstupního dokumentu může být zcela jiná než struktura zdrojového dokumentu (lze přidávat elementy, přeuspořádat pořadí, opakovat atd.).

Jak již bylo řečeno, XSL-dokument se reprezentuje jako XML-dokument. Lze jej tedy zpracovávat pomocí XML-processoru. Používají se přitom následující konvence:

XSL-značky se píší malými písmeny (např. <xsl> ).

Značky pro výstup v HTML (HTML "Flow Object") se píší VELKÝMI PÍSMENY.

XSL-dokument je dobře vytvořený XML-dokument.

Hodnoty atributů se píší v uvozovkách:

```
<TABLE BORDER = "1">
```

```
<OL COMPACT = "YES PLEASE">
```

Syntaxe prázdných elementů je <HR/>.

Skripty se do dokumentu vkládají pomocí datové sekce CDATA.

Protože je XSL-dokument dobře vytvořený XML-dokument, musí být kořen uzavřen v závorkách (používá se dvojice <xsl:stylesheet> a </xsl:stylesheet>) a všechny ostatní elementy musí být do něj vnořeny a závorky musí být správně spárovány.

Specifikace XSL plně využívá tzv. prostorů jmen (name spaces). Pro seznámení s XSL není nutno porozumět tomuto mechanismu do všech podrobností, postačí nám jednoduchý příklad. Podrobný popis lze nalézt v literatuře. Všechny dokumenty odpovídající specifikaci XSL jsou vloženy do kořene XSL-dokumentu, který má značku:

```
<xsl:stylesheet> ... </xsl:stylesheet>
```

Příklad:

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/TR/WD-xsl"
  xmlns:fo="http://www.w3.org/TR/WD-xsl/FO"
  result-ns="fo">
```

.....

```
</xsl:stylesheet>
```

Kořen dokumentu má v tomto případě atributy: xmlns:xsl (prostor jmen XSL-dokumentu jako XML-dokumentu), xmlns:fo (formátovací slovník) a result-ns="fo" (výstupní prostor jmen). Prvý atribut je povinný a jeho hodnota musí být:

```
xmlns:xsl="http://www.w3.org/TR/WD-xsl".
```

Tento atribut slouží pro procesor XSL, který tak pozná, že všechny elementy s prefixem 'xsl:' mají být interpretovány dle specifikace XSL (jsou v prostoru 'xsl'). Druhý atribut je teoreticky volitelný, ale ve skutečnosti bývá většinou uveden v naznačeném tvaru. Říká XSL-processoru, že elementy s prefixem 'fo:' (zkratka pro flow objects) jsou v prostoru 'fo' a že formátování využívá formátovací slovník dle specifikace XSL.

Narazí-li pak procesor na <fo:block font-size="12pt">, řídí se syntaxí specifikovanou v prostoru 'fo'. Je samozřejmě možné používat jiný slovník (viz např. [www.mystyle.org](http://www.mystyle.org)), je pak ale třeba přesvědčit tvůrce aplikací, aby tento formát respektovali. Třetí atribut result-ns="fo" říká XSL-processoru, že konstruovaný výstupní strom (result tree) používá slovník 'fo'.

## Pravidla XSL-dokumentu

Každé pravidlo XSL obsahuje vzorek a akci. Vzorek pravidla slouží pro selekci uzlu (elementu) vstupního stromu. Selekcí je možno předepsat podle: jména elementu, předchůdce elementu, ID-elementu, lze využít zástupné znaky, lze odkazovat na atributy elementu, relativní pozici elementu ve stromě nebo na unikátnost elementu. Přesnější vysvětlení by vyžadovalo popis jazyka XPath, ve kterém se dají zapisovat vzorky a akce pravidel.

Akce pravidla obsahují především elementy konstruovaného (formátovaného) objektu, např. <fo:block></fo:block> vytvoří ve formátovaném objektu blok. Lze nastavit i atributy formátovaného objektu, např. <fo:block font-size="16pt"> indikuje, že ve výstupním bloku má být použito písmo velikosti 16 bodů. Způsob zpracování vstupního stromu je vyjádřen formou speciálního prázdného elementu; např. <xsl:apply-templates/> způsobí, že procesor XSL aplikuje v tomto místě všechna pravidla a zpracuje všechny potomky.

Procesor XSL prochází vstupní dokument (vstupní strom) a pokouší se hledat vzorky pravidel

XSL-dokumentu. Pokud se podaří nalézt shodu, aplikuje akci pravidla na výstupní dokument (výstupní strom). Pokud je aplikovatelných pravidel více, vybere se pravidlo dle stanoveného postupu. Pravidla XSL mají následující obecný formát:

```
<xsl:stylesheet>
  <!-- zde je vzorek -->
  <xsl:template ...
    <!-- zde je akce -->
  ...
</xsl:template>
</xsl:stylesheet>
```

Příklad pravidla pro element typu "fax":

```
<xsl:stylesheet>
  <xsl:template match="fax"> <!-- to je vzorek -->
    <!-- a zde je akce -->
    <fo:page-sequence color="red" font-size="16pt">
      <apply-templates/>
    </fo:page-sequence>
  </xsl:template>
</xsl:stylesheet>
```

Pokud podle tohoto pravidla zpracujeme dříve uvedenou faxovou zprávu, vznikne výstupní dokument:

```
<fo:page-sequence color="red" font-size="16pt">
  FAX Message
  <!-- zde bude výstup podle ostatních pravidel -->
</fo:page-sequence>
```

## Převod XML do HTML pomocí XSL

Jednou z možností, jak popsat zobrazení XML-dokumentů, je popsat pomocí XSL jejich převod do HTML. Předpokládejme, že vstupní XML-dokument musí mít strukturu dle DTD pro faxové zprávy. Předpis pro transformaci do HTML v XSL (XSL stylesheet) může vypadat následovně:

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/TR/xhtml1/strict">
  <xsl:strip-space elements="fax"/>
  <xsl:output method="xml" indent="yes" encoding="iso-8859-2" />

  <xsl:template match="fax">
    <html>
      <head>
        <title>
          <xsl:text>FAX Message</xsl:text>
        </title>
      </head>
      <body>
        <b>FAX Message</b>
        <xsl:apply-templates/>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="fax/odesilatel">
    <h1> Od:
      <xsl:apply-templates/>
    </h1>
  </xsl:template>

  <xsl:template match="fax/adresat">
    <h1> Komu:
      <xsl:apply-templates/>
    </h1>
  </xsl:template>
```

```
</h1>
</xsl:template>
```

```
<xsl:template match="fax/text">
  <p>
    <xsl:apply-templates/>
  </p>
</xsl:template>
```

```
</xsl:stylesheet>
```

Pokud XSL-procesoru předložíme následující dokument:

```
<!DOCTYPE fax SYSTEM "fax.dtd">
```

```
<fax> FAX Message
```

```
  <odesilatel>Karel</odesilatel>
```

```
  <adresat>Jana</adresat>
```

```
  <text> Nazdar </text>
```

```
</fax>
```

a XSL-dokument uvedený výše, bude produkován následující výstup (v XHTML 1.0):

```
<?xml version="1.0" encoding="iso-8859-2"?>
```

```
<html xmlns="http://www.w3.org/TR/xhtml1/strict">
```

```
<head>
```

```
<title>FAX Message</title>
```

```
</head>
```

```
<body>
```

```
<b>FAX Message</b>
```

```
<h1>Od: Karel</h1>
```

```
<h1>Komu: Jana</h1>
```

```
<p>Nazdar</p>
```

```
</body>
```

```
</html>
```

*Karel Richta, richta@fel.cvut.cz*