

# APLIKACE XML PRO INTERNET

**Jaroslav Ráček**

Fakulta Informatiky, Masarykova Universita Brno

## Abstrakt

Text je věnován možnostem využití XML technologie pro prezentaci dokumentů pomocí Internetu. V úvodu je stručně popsán vznik a vývoj jazyka XML. Na to navazuje část věnována DTD (Definice typu dokumentu). Další část příspěvku je věnována transformaci XML dokumentů pomocí stylů (CSS a XSL). V závěru jsou pak představeny dva standardy na bázi XML. Jedná se o jazyk XHTML pro oblast webu a jazyk WML pro WAP mobilní komunikaci.

## Úvod

Jazyk XML (eXtensible Markup Language) vznikl v devadesátých letech jako podmnožina jazyka SGML (Standard Generalized Markup Language). Hlavním důvodem vzniku jazyka XML byla snaha nahradit jazyk HTML (HyperText Markup Language). Ačkoli si HTML, který je asi nejznámější aplikací SGML, získal díky tvorbě webových stránek velkou popularitu, začalo se v polovině 90. let ukazovat, že jazyk HTML již nevyhovuje novým požadavkům. Problém byl v tom, že HTML je poměrně jednoduchý a má velmi malé možnosti pro vymezení obsahu textu.

Oproti tomu jazyk SGML byl pro tyto účely až příliš komplexní a složitý. Proto byla ze SGML vyčleněna jeho nejpoužívanější část, čímž vznikl nový jazyk pojmenovaný jako XML. Syntaxe XML je přísnější než syntaxe obecnějšího SGML, což umožňuje snazší vývoj aplikací.

Díky své otevřenosti a flexibilitě nachází XML mnoho možností svého využití. Je to formát, který je nezávislý na platformě, umožňuje vysoký informační obsah dokumentu a je poměrně snadno převoditelný do dalších formátů. Nezanedbatelná je také skutečnost, že jako znaková sada je v XML podporována 32bitová ISO 10646, která bez problémů pokrývá všechny znaky současných jazyků.

## DTD

Při dodržení všech pro XML definovaných syntaktických pravidel, může být obecný XML dokument vytvořen za použití libovolných elementů. Jednotlivé XML aplikace však používají vlastní omezené sady elementů, kde každý element a jeho atribut má svůj předdefinovaný význam, což podstatně zvyšuje informační obsah dokumentů.

Vznikají tak nové značkovací jazyky, které jsou podmnožinami obecnějšího XML. Každý takto vzniklý jazyk může být (a v naprosté většině případů také je) popsán pomocí tzv. Definice Typu Dokumentu (DTD), která obsahuje sadu povolených elementů, jejich atributů a popis vzájemných vztahů, v nichž mohou být jednotlivé elementy použity.

Založení dokumentu na základě nějakého DTD má tu výhodu, že jeho strukturu můžeme kontrolovat pomocí parseru (programu pro kontrolu správnosti XML dokumentů). Další výhodou je, že pro spoustu standardních DTD existuje množství již vytvořených nástrojů, které můžeme pro daná DTD s úspěchem použít. Jedná se například o styly, pomocí nichž jsou dokumenty konvertovány do formátu HTML.

Jazyk, v němž jsou jednotlivá DTD psána, je rovněž odvozen od XML. Vlastní DTD pak lze rozdělit do čtyř základních částí. Jsou to:

- deklarace elementů
- deklarace atributů
- deklarace entit
- deklarace notací

Deklarace elementu má tvar:

```
<!ELEMENT <<název_elementu>> <<obsah_elementu>>>
```

Jak může deklarace elementů vypadat v praxi ukazuje následující příklad, který lze použít pro označení jednotlivých částí nějakého článku.

*Příklad 1 – Deklarace elementů:*

```
<!ELEMENT clanek (navez, autor?, odstavec+)>
<!ELEMENT navez (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT odstavec (#PCDATA)>
```

Předchozí zápis definuje elementy clanek, navez, autor a odstavec. Element clanek obsahuje uvnitř element navez, může obsahovat element autor a dále obsahuje alespoň jeden element odstavec. Elementy navez, autor a odstavec již obsahují samotný text.

Také deklarace atributů pro element má poměrně jednoduchý tvar

```
<!ATTLIST <<název_elementu>> <<deklarace_atributů>>>.
```

Atributy elementu clanek by mohly být například deklarovány následujícím způsobem:

*Příklad 2 – Deklarace atributů:*

```
<!ATTLIST clanek autor CDATA
                jazyk MNTOKEN
                zarovnani(vlevo|nastred|vpravo) "vlevo">
```

Element clanek může mít tři atributy. Jsou to autor, jazyk a zarovnani. Atribut autor může obsahovat libovolný textový řetězec a atribut jazyk pouze jedno slovo. Atribut zarovnani nabývá jednu z hodnot vlevo, nastred nebo vpravo, přičemž implicitně je nastavena hodnota vlevo. Všimněte si, že narozdíl od příkladu č.1 je autor článku v příkladu č.2 uveden pomocí atributu autor a ne pomocí samostatného elementu. V příslušném XML dokumentu by pak mohl být element clanek použit takto:

*Příklad 3 – Použití atributů:*

```
<clanek autor="Jaroslav Ráček" jazyk="cz"
      zarovnani="nastred">
  ...
</clanek>
```

DTD může být umístěno přímo v dokumentu, častější je však případ, kdy je DTD uloženo v samostatném souboru. To je možné provést například následujícím způsobem, kdy DTD je obsaženo v souboru clanekek.dtd.

*Příklad 4 – Způsoby připojení DTD k dokumentu:*

```
<!DOCTYPE clanekek SYSTEM "clanekek.dtd">
<clanekek>
  ...
</clanekek>
```

Pokud by DTD bylo obsaženo přímo v dokumentu mohl by mít dokument následující tvar:

```
<!DOCTYPE clanekek [
  <!ELEMENT clanekek ...>
  <!ELEMENT ... >
  ... ]>
<clanekek>
  ...
</clanekek>
```

Další možností je uvést v dokumentu odkaz na nějaké existující veřejné DTD. Příklady této varianty jsou uvedeny v následujícím textu (části věnované XHTML a WML).

## **Styly a XML**

Styly se používají v případě, kdy chceme oddělit obsah dokumentu od jeho vzhledu. To přináší dvě základní výhody. První je, že pro více dokumentů, můžeme používat jeden styl, čímž zajistíme stejný vzhled těchto dokumentů. V okamžiku, kdy se rozhodneme změnit jejich společný vzhled, provedeme změny pouze v daném stylu, přičemž v dokumentech nic měnit nemusíme.

Druhá výhoda je založena na opačné situaci, kdy k jednomu stylu používáme více dokumentů. Daný dokument tak můžeme zobrazovat s různým vzhledem na různých výstupních zařízeních, přičemž jej uchováváme pouze v jedné verzi. Pokud se rozhodneme obsah dokumentu změnit, stačí provést změny pouze jednou.

Stylových jazyků, které lze použít pro formátování XML dokumentů je poměrně velké množství. Vedle CSS (Cascading Style Sheets) a XSL (eXtensible Stylesheet Language), kterým je věnována část tohoto článku existují například ještě DSSSL nebo FOSI.

Styl je možné k XML dokumentu připojit pomocí elementu

```
<?xml-stylesheet href="<<URI>>" type="<<typ>>"?>
```

kde parametr href udává odkaz na soubor, v němž je styl uložen a parametr type specifikuje typ (jazyk) stylu. Jak by mohlo vypadat připojení konkrétního stylu k dokumentu v praxi ukazuje následující příklad.

*Příklad 5 – Připojení stylu k dokumentu:*

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet href="styl.css" type="text/css"?>
<dokument>
...
</dokument>
```

## CSS

Pomocí kaskádových stylů je možné upravovat nejen XML dokumenty, ale rovněž dokumenty psané v jazyce HTML. CSS existují ve verzích 1 a 2. V době přípravy tohoto článku se pracovalo na verzi 3. Syntaxe CSS dokumentů je velmi jednoduchá. Jedná se o soubor pravidel, z nichž každé se skládá ze dvou částí. První část, se nazývá selektor a určuje na kterou část dokumentu bude pravidlo použito. Druhá část deklaruje vzhled části určené selektorem. Použití CSS stylů více osvětlí příklad stylu, pomocí kterého je možné určit vzhled obsahu atributů clank, nazev a autor z příkladu č.1.

*Příklad 6 – Ukázka CSS:*

```
clanek      { font-family: Arial;
              background-color: white }
nazev       { font-size: 24pt }
autor       { font-size: 18pt;
              color: blue
              text-align:center }
```

Předchozí příklad určuje, že text uvnitř elementu clank bude zobrazen fontem Arial na bílém pozadí. Dále velikost textu uvnitř elementu nazev bude 24 bodů. Text uvnitř elementu autor bude o velikosti 18 bodů, modré barvy a vycentrovaný. Jako selektory byly v tomto případě použity přímo názvy jednotlivých elementů. Kaskádové styly, zvláště pak verze CSS2 však nabízejí širší možnosti selektorů. Malá ukázka těchto možností je uvedena v následujícím příkladu:

*Příklad 7 – selektory CSS2:*

*	-všechny elementy.
autor	-všechny elementy autor.
nazev + autor	-všechny elementy autor bezprostředně za elementem nazev.
clanek > autor	-všechny elementy autor, které jsou dětmi elementu clank.
clanek[zarovnani]	-všechny elementy clank s nastaveným atributem zarovnani
clanek[jazyk="cz"]	-všechny elementy clank, které mají nastavený atribut jazyk na cz.
nazev:first-child	-všechny elementy nazev, které jsou první dítě svého rodiče.

## XSL

Narozdíl od CSS, pro jejichž zápis se používá speciální jazyk, je syntaxe XSL založena na XML. Standard XSL má dvě části. První část slouží pro transformaci XML dokumentů a je označována jako XSLT (XSL Transformations). Druhá část standardu se používá pro abstraktní popis vzhledu dokumentu. Říká se jí XSL formátovací objekty.

Základními stavebními kameny XSLT jsou šablony. Šablonu tvoří element

```
<xsl:template match="<<výraz>>"><<výstup>></xsl:template>
```

kde <<výraz>> určuje elementy na které bude šablona aplikována. Pro zápis se používá dotazovací jazyk Xpath, kterému se však v tomto článku nebudu dále věnovat. Transformujeme-li například nějaký XML dokument do HTML a chceme, aby text uvnitř elementů vybraných pomocí výrazu byl psán tučně ( tj. uvnitř elementu <b>), můžeme vyjít z následujícího příkladu.

*Příklad 8 – Šablona XSLT:*

```
<xsl:template match="<<výraz>>">
  <b><xsl:apply-templates/></b>
</xsl:template>
```

Celý styl (XSLT dokument) je uvnitř jednoho elementu, který definuje jmenný prostor XSLT.

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <<deklarace šablon>>
</xsl:stylesheet>
```

Své praktické využití našlo XSLT především při transformacích XML dokumentů na HTML a WML dokumenty.

## XHTML

Jazyk XHTML (eXtensible HyperText Markup Language) pochází z dílny konsorcia W3C. Podnětem pro jeho vznik byla neutěšená situace v oblasti používání HTML. Jednalo se hlavně o překřížené elementy nebo špatně spárované tagy, které HTML prohlížeče často tolerovaly a snažily se s nimi vypořádat po svém. To sice dává tvůrcům webových stránek určitou volnost a možnost odchýlit se od HTML standardu, na druhou stranu to však velmi komplikuje práci vývojářů aplikací, které mají s takovým dokumenty pracovat. Jako jedno z možných řešení se ukázalo nahradit stávající HTML standard novým standardem na bázi XML, kterým se stal XHTML. Z tohoto pohledu se může zdát, že XHTML není novým jazykem, ale pouhým „přepsáním HTML do XML“.

Stejně jako HTML 4.0, je i XHTML definováno na základě tří DTD. DTD Strict definuje jazyk bez prezentačních elementů a atributů, DTD Transitional obsahuje všechny elementy HTML 4.0 a DTD Frameset se používá pro dokumenty obsahující dělení okna prohlížeče do rámců. DTD jsou veřejná a naleznete je na serveru konsorcia W3C. Připojení těchto DTD k dokumentu ukazuje následující příklad.

*Příklad 9 – Připojení DTD k XHTML dokumentu:*

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/html1/DTD/xhtml1-strict.dtd">

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/html1/DTD/xhtml1-transitional.dtd">

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/html1/DTD/xhtml1-frameset.dtd">
```

Na závěr části věnované XHTML uvádím příklad jednoduchého XHTML dokumentu.

*Příklad 10 – XHTML dokument:*

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/html1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title> Nadpis </title>
  </head>
  <body>
    <p> Odstavec </p>
  </body>
</html>
```

## **WML**

Díky rozvoji mobilních telefonů, je v současnosti věnována nemalá pozornost komunikaci založené na protokolu WAP (Wireless Application Protocol). Pro tvorbu WAP aplikací se často používá jazyk WML (Wireless Markup Language). WML je jednoduchý jazyk na bázi XML podobný HTML, který podporuje elementy pro vkládání textu, základní formátování, zvýrazňování, tvorbu tabulek, vkládání obrázků, tvorbu odkazů, vkládání jednoduchých formulářů a možnost jednoduchých skriptů v jazyce WMLScript. Znaková sada Unicode, kterou WML používá, podporuje rovněž různé národní znaky.

Je mimo rozsah tohoto příspěvku zabývat se podrobně syntaxí jazyka WML, proto uvádím jen několik zásad, které je třeba při tvorbě WML stránek (dokumentů) dodržovat. Především každá stránka musí obsahovat XML deklaraci. Dále se každá stránka skládá z několika karet, což jsou jednotky zobrazované na displeji. Chceme-li definovat akce dostupné na všech kartách, lze tak učinit pomocí elementu `template`. Následující příklad obsahuje ukázkou jednoduchého WML dokumentu, včetně odkazu na příslušné veřejné DTD.

*Příklad 11 – WML dokument:*

```
<?xml version="1.0" ?>
<!DOCTYPE wml
  PUBLIC "-//WAPFORUM//DTD WML 1.1 //EN"
  "http://www.wapforum.org/DTD/WML_1.1.xml">
<wml>
  <template>
    <do type="prev">
      <prev/>
    </do>
  </template>
  <card title="Prvni karta">
    <p>Text prvni karty</p>
    <p>
      <a href="#card2"> Odkaz na dalsi kartu </a>
    </p>
  </card>
  <card id="card2" title="Druha karta">
    <p>
      <b>Tucny text druhe karty</b>
    </p>
  </card>
</wml>
```

**WWW zdroje:**

1. <http://www.kosek.cz/> - stránky Jiřího Koska
2. <http://www.oasis-open.org/> - stránky konsorcia OASIS
3. <http://www.wapforum.org/> - stránky věnované WAP technologiím
4. <http://www.w3c.org/> - stránky konsorcia W3C