Syntaxe XML

XML – teorie a praxe značkovacích jazyků (4IZ238)

Jirka Kosek

Poslední modifikace: 2020-10-08 08:34:28 UTC

Obsah

Základy syntaxe

Elementy a atributy

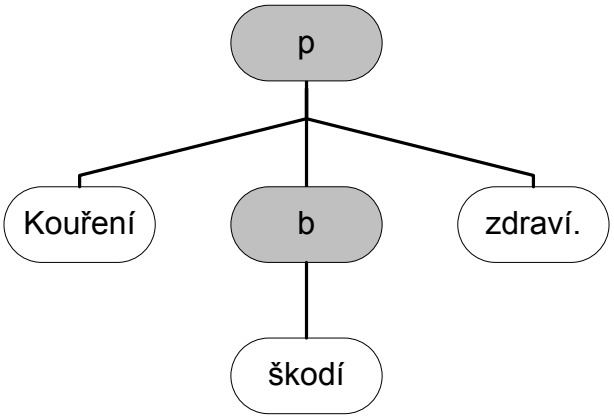
- element, tag, počáteční tag, ukončovací tag, obsah elementu
- elementy se nesmí křížit
- tagy musí být spárované nebo má element prázdný obsah
- kořenový element – obaluje celý dokument
- atribut, název atributu, hodnota atributu
- názvy elementů a atributů – přesná definice ve specifikaci XML[1]
  - začínají písmenem, podtržítkem nebo dvojtečkou
  - další znaky jsou písmena, číslice, tečka, pomlčka, podtržítko, dvojtečka a některé další znaky
- zápis vyhrazených znaků – pomocí entit (< & > ' ")

[1] http://www.w3.org/TR/REC-xml

Datový model dokumentu

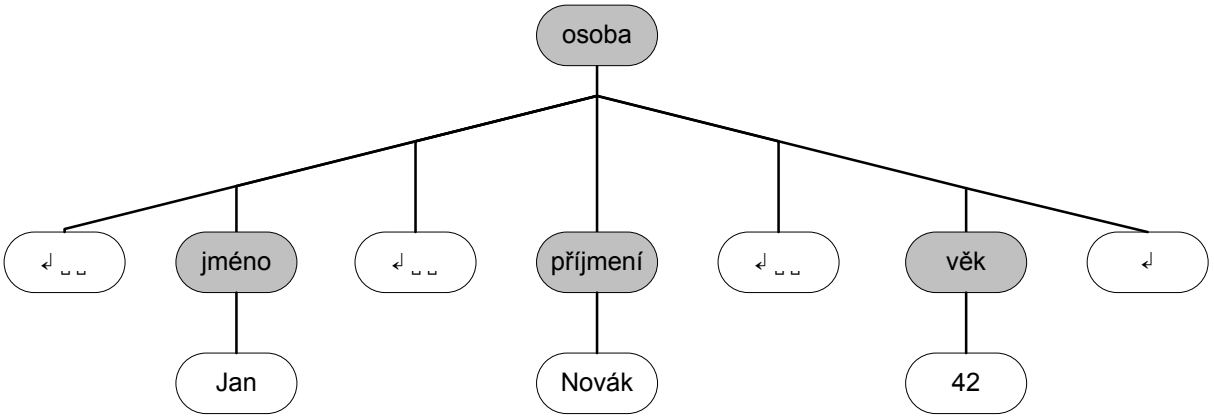
- každý dokument si lze představit jako strom
- smíšený obsah

```
<p>Kouření
<b>škodí</b> zdraví.</p>
```



- bílé znaky

```
<osoba>
  <jméno>Jan</jméno>
  <příjmení>Novák</příjmení>
  <věk>42</věk>
</osoba>
```



Znakový model XML

- XML dokumenty používají znakovou sadu ISO 10646
- 21bitová znaková sada, v současné době je definováno přes 100 tisíc znaků
- zcela shodné s Unicode
- kódování: UTF-16 a UTF-8
- UTF-16
  - 16bitové slovo přímo obsahuje kód znaku
  - další téměř milión znaků je dostupných pomocí „surrogates" – 1 znak = dvě 16bitová slova
- UTF-8
  - text může být kódován jako sekvence bajtů
  - 1 znak = 1 až 4 bajty
  - kompatibilní s ASCII

Obrázek 1. Kódování UTF-8

Scalar Value	UTF-16	1st Byte	2nd Byte	3rd Byte	4th Byte
00000000 0xxxxxxx	00000000 0xxxxxxx	0xxxxxxx			
00000yyy yyxxxxxx	00000yyy yyxxxxxx	110yyyyy	10xxxxxx		
zzzzyyyy yyxxxxxx	zzzzyyyy yyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	
000uuuuu zzzzyyyy yyxxxxxx	110110ww wwzzzzyy 110111yy yyxxxxxx	11110uuu	10uuzzzz	10yyyyyy	10xxxxxx

- Where uuuuu = wwww + 1 (to account for addition of 10000_{16} as in *Section 3.7, Surrogates)*.

- standard XML vyžaduje, aby všechny aplikace podporovaly alespoň UTF-8 a UTF-16
- lze použít i jiné kódování, musí se uvést v XML deklaraci

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

  musí být první řádka dokumentu
- znak s libovolným kódem z ISO 10646 můžeme zapsat pomocí entity `&#kód;` (*kód* je číslo v desítkové soustavě) nebo `&#xkód;` (*kód* je číslo v šestnáctkové soustavě)

Komentáře

- slouží pro vkládání poznámek nebo pro dočasné vyřazení části dokumentu
- nesmí obsahovat -- a nemohou být navzájem vnořené

```
<!-- ukázkový komentář -->
```

Instrukce pro zpracování

- instrukce pro zpracování = processing instructions
- standardní způsob pro začlenění nestandardních dat
- interpretace závisí na aplikaci:
  - připojování stylu k dokumentu
  - příkazy pro různé preprocesory
  - poslední pozice editace, ruční zlom řádky/stránky

```
<?xml-stylesheet href="styl.css" type="text/css"?>


<datum>
  <?php echo Date("d.m.Y")?>
</datum>


<?myDTP page-break?>
```

Sekce CDATA

- pro delší úseky dokumentu, kde nechceme interpretovat pro XML významné znaky jako '<', '&', …
- použití:
  - ukázky XML a HTML kódu
  - zařazování programů přímo do XML dokumentů

```
<výpis>
<![CDATA[
Můžu použít klidně <tag> a nikomu to
nevadí. Ani firmě Son & Brother.
]]>
</výpis>
```

je zcela ekvivalentní s

```
<výpis>
Můžu použít klidně &lt;tag&gt; a nikomu to
nevadí. Ani firmě Son &amp; Brother.
</výpis>
```

- omezení – CDATA sekce nesmí obsahovat `]]>`

Kontrola správné syntaxe XML dokumentů

- well-formed = správně strukturovaný dokument
  - dokument splňuje základní syntaktická pravidla
  - takový dokument by měla zpracovat každá aplikace s podporou XML
  - lze ověřit pomocí parseru

Jmenné prostory

Jmenné prostory

- slouží k rozlišení elementů a atributů se shodnými jmény v případech kdy by mohlo dojít ke konfliktům
- aplikace si vybere jen ty části dokumentu, které umí zpracovat
  - kombinování více „sad značek“ dohromady
  - např.: XSLT styly (XSLT instrukce × HTML kód), XHTML stránka s obrázky v SVG, …
- jména a elementů a atributů se skládají ze dvou částí – ze jmenného prostoru a z lokálního názvu
- jmenné prostory se identifikují pomocí URI adresy, ale nic konkrétního se na ní nevyskytuje, slouží pouze jako identifikátor
- pro zkrácení zápisu se při deklaraci jmenného prostoru vytvoří prefix, který jmenný prostor zastupuje:

```
<prefix:element xmlns:prefix="http://nekde.com/neco"> …
</prefix:element>
```

- prefixy je možné použít i u atributů a elementů obsažených v elementu s deklarací

Příklad 1. Dokument se jmennými prostory

```
<ceník:nabídka
    xmlns:ceník="http://www.ecena.cz/e-cenik"
    xmlns:bib="http://www.book.org/bibliography">
  <ceník:položka ceník:dph="22%">
    <ceník:název>
      <bib:book>
        <bib:author>Jiří Kosek</bib:author>
        <bib:title>HTML - tvorba dokonalých
                   WWW stránek</bib:title>
      </bib:book>
    </ceník:název>
    <ceník:cena měna="CZK">259</ceník:cena>
  </ceník:položka>
</ceník:nabídka>
```

Zkracování zápisu

- implicitní jmenné prostory (prefix je prázdný)

```
<nabídka xmlns="http://www.ecena.cz/e-cenik"
    xmlns:bib="http://www.book.org/bibliography">
  <položka>
    <název>
      <bib:book>
        <bib:author>Jiří Kosek</bib:author>
        <bib:title>HTML - tvorba dokonalých
                   WWW stránek</bib:title>
      </bib:book>
    </název>
    <cena měna="CZK">259</cena>
  </položka>
</nabídka>
```

- jmenné prostory se stejným prefixem se navzájem překrývají

```
<nabídka xmlns="http://www.ecena.cz/e-cenik">
  <položka>
    <název>
      <book xmlns="http://www.book.org/bibliography">
        <author>Jiří Kosek</author>
        <title>HTML - tvorba dokonalých
               WWW stránek</title>
      </book>
    </název>
    <cena měna="CZK">259</cena>
  </položka>
</nabídka>
```

Pár poznámek

- prefix + lokální jméno = kvalifikované jméno (QName)
- atribut bez prefixu nepatří přímo do žádného jmenného prostoru:
  - ani když je jeho element v implicitním jmenném prostoru
  - patří tam však „nepřímo“, protože se vždy vztahuje k elementu, u kterého je uvedený

```
<nabídka xmlns="http://www.ecena.cz/e-cenik">
  <položka>
    <název>
      ...
    </název>
    <cena měna="CZK">259</cena>
  </položka>
</nabídka>
```

  - atribut `měna` v tomto případě nepatří do jmenného prostoru `http://www.ecena.cz/e-cenik`
  - na druhou stranu patří k elementu `cena`, který již do daného jmenného prostoru patří
- specifikace jmenných prostorů doplňuje standard XML
- dokumenty se jmennými prostory nelze validovat oproti DTD bez toho, aby se vytvořil speciální kříženec všech použitých DTD
- jmenný prostor neodpovídá DTD ani XML schématu, nicméně obvykle pro každý jmenný prostor někde existuje popis použitelných elementů a atributů nejčastěji právě ve formě DTD nebo XML schématu
- jmenné prostory se hodně používají v dalších jazycích souvisejících s XML – XML schémata, XSLT, …

Přehled vybraných jmenných prostorů

Popis	Jmenný prostor
XHTML	http://www.w3.org/1999/xhtml
SVG	http://www.w3.org/2000/svg
MathML	http://www.w3.org/1998/Math/MathML
W3C XML Schema	http://www.w3.org/2001/XMLSchema
RELAX NG	http://relaxng.org/ns/structure/1.0
DocBook	http://docbook.org/ns/docbook
XSLT	http://www.w3.org/1999/XSL/Transform
XSL-FO	http://www.w3.org/1999/XSL/Format
Katalogový soubor	urn:oasis:names:tc:entity:xmlns:xml:catalog
Atom	http://www.w3.org/2005/Atom
ISDOC (elektronické faktury v ČR)	http://isdoc.cz/namespace/2013

Další informace

URI × URL × URN

- URI se v XML používá k identifikaci jmenných prostorů a pro určení umístění externích entit
- URI = URL + URN
- URI = Uniform Resource Identifier
  - RFC 2396
  - *schéma*:*specifická část*
- URL = Uniform Resource Locator
  - RFC 1738
  - identifikuje zdroj dostupný na určitém místě a určitým protokolem
  - FTP, HTTP, NNTP, Gopher, …
  - lze použít i relativní URL – při odkaz na zdroj ve stejném adresáři stačí použít jméno souboru
- URN = Uniform Resource Name
  - RFC 2141
  - identifikuje zdroj nezávisle na jeho umístění
  - *urn*:*druh URN*:*specifický identifikátor*
  - např.:

```
urn:ietf:rfc:2141
urn:ietf:std:50
```

XML deklarace

- <?xml version="1.0"
  encoding="utf-8"
  standalone="yes"?>

- musí být první řádkou v dokumentu
- standalone
  - standalone="yes" – pro zpracování dokumentu není potřeba číst externí DTD
  - standalone="no" – implicitně

Verze XML

- v praxi se používá téměř výhradně verze 1.0
- existuje verze 1.1, která se v praxi vůbec neujala, ale vznikla spíše kvůli „politické korektnosti“
  - jména elementů a atributů mohou používat téměř jakýkoliv znak z Unicode včetně budoucích verzí Unicode (to je ale možné i v posledním pátém vydání XML 1.0)
  - jako znak konce řádku lze používat i znaky NEL (…) a unicodový oddělovač řádek ( )
  - v dokumentu mohou být použity i znaky s kódem 1 až 31, ale musí být zapsány jako číselný odkaz na znak
  - řetězce uvnitř dokumentu by měly být normalizované (NFC) – tj. preferuje se znak s diakritickým znaménkem, ne znak a za ním kombinující diakritické znaménko
- pragmatický přístup:
  - přijímat dokumenty v obou verzích v libovolném kódování
  - generovat dokumenty ve verzi 1.0 a v kódování UTF-8
- podmnožiny XML
  - obecně není dobré je vytvářet a používat
  - SOAP – nepodporuje entity a instrukce pro zpracování

XML a binární data

- XML samo o sobě binární data nepodporuje, ale v praxi je často potřeba XML doplnit o binární data (např. obrázky)
- v současnosti se používá několik přístupů:
  - bezpečné zakódování do XML dokumentu
    - binární data se překódují do textové podoby (např. Base64) a vloží se přímo do dokumentu
    - výsledný dokument je velký
    - používá například formát WordML v MS Office 2003
  - binární data jsou uložena odděleně
    - XML dokument obsahuje pouze odkaz na binární data
    - pro snadnou práci je možné XML i binární data uložit do jednoho archivu – využívá např. OpenOffice a MS Office 2007
  - SOAP with Attachments
    - binární data jsou přidána za samotnou XML zprávu pomocí mechanismu MIME
  - DIME
    - binární data jsou rovněž samostatná
    - DIME je na rozdíl od MIME binární formát a tudíž efektivnější
- SOAP + XOP
  - binární data jsou rovněž samostatná a využívají MIME
  - na jednotlivé BLOBy se odkazuje přímo z dokumentu XML

Další informace

Povinná četba

- Kapitola o syntaxi XML[2]
- Vysvětlení problematiky znakových sad a kódování[3]

Doporučená četba

- Anotovaná specifikace XML[4]

[2] http://www.kosek.cz/knihy/phpxml/php5xml-ukazka.pdf
[3] https://www.kosek.cz/knihy/phpxml/php5xml-unicode.pdf
[4] http://www.xml.com/axml/testaxml.htm