Základní standardy rodiny XML (DTD, XML Base, jmenné prostory...)

Obsah

Specifikace a validita XML

Aktuální specifikace XML

- Původní specifikace (W3C Recommendation) XML 1.0 na W3C: http://www.w3.org/XML/
- 4th Edition (aktualizace, opravy, ne změny) na Extensible Markup Language (XML) 1.0 (Fourth Edition) [http://www.w3.org/TR/2006/REC-xml-20060816/]
- výborná komentovaná verze téhož na XML.COM (Annotated XML): http://www.xml.com/pub/a/axml/axmlintro.html
- XML 1.1 (Second Edition) [http://www.w3.org/TR/2006/REC-xml11-20060816/] - změny indukované zavedením *UNICODE 3*, lepší možnosti *normalizace*, upřesnění postupu manipulace se znaky *ukončení řádku*. XML 1.1 není už vázaný na konkrétní verzi UNICODE, ale vždy na verzi poslední.

Jakou verzi použít?

Jakou verzi specifikace bychom měli v nových aplikacích používat?

Odpověď dává W3C XML Core Working Group [http://www.w3.org/XML/Core/#Publications]:

- nepíšeme-li parser, ale aplikaci, která generuje nebo vytváří XML (editor), používejme XML 1.0 (zpětná kompatibilita)
- nové parsery by měly umět XML 1.1

Validita XML dokumentů

- Opakování: každý XML dokument MUSÍ být správně utvořený (*well formed*)
- Nové: XML dokument můřže být platný (*valid*) dokument:

  Platný podle specifikace znamená *přísnější* omezení než správně utvořený.

  Obvykle se validitou myslí soulad s *DTD* (Document Type Definition) dokumentu nebo

  (moderněji) - soulad s XML Schematem, případně jinými schématy (RelaxNG, Schematron).

Document Type Definition (DTD)

Document Type Definition (DTD)

- Definice typu dokumentu (použití této definice je pak **deklarace typu dokumentu**).
- Specifikována přímo (základním) standardem XML 1.0.
- Popisuje přípustný **obsah elementů**, **atributů**, jejich implicitní (default) hodnoty, definuje použité **entity**.
- Může být uvedena jako **interní** nebo **externí** DTD (*internal and external subset*) nebo "napůl" - tam i tam.
- Dokument vyhovující DTD je označován jako *valid* (platný).
- DTD a podobným jazykům se také říká *modelovací* -- modelují/definují konkrétní značkování.
- Syntaxe DTD *není* XML (na rozdíl od většiny dalších modelovaních jazyků).

Motivace pro DTD, srovnání, výhody a nevýhody

Jaká jsou úskalí DTD?

- Zásadním úskalím DTD je "nekompatibilita" se jmennými prostory (XML Namespaces) a
- slabá modelovací schopnost - řadu omezení v modelu nelze pomocí DTD popsat.
- "Přímým" a mocnějším, ale také podstatně složitějším, nástupcem DTD je W3C XML Schema [http://www.w3.org/XML/Schema].
- Existují rovněž zdařilé, jednoduché a mocné alternativy ke XML Schematu - jako je RelaxNG [http://relaxng.org]. (on Wikipedia:RELAX_NG [http://en.wikipedia.org/wiki/RELAX_NG])

Proč vůbec (ještě) používat DTD?

- Jednoduché, všechny validující parsery ho umí.
- Pro mnoho jednoduchých značkování nepoužívajících jmenné prostory DTD dostačuje.

DTD - tutoriály

- Webreview: http://www.webreview.com/2000/08_11/developers/08_11_00_2.shtml
- ZVON: http://www.zvon.org/xxl/DTDTutorial/General/contents.html
- XML DTD Tutorial (101): http://www.xml101.com/dtd/
- W3Schools DTD Tutorial: http://www.w3schools.com [http://www.w3school.com]

DTD - deklarace typu dokumentu podrobněji

Uvádí se těsně před kořenový elementem konstrukcí

- `<!DOCTYPE jméno-kořenového-elt Externí-ID [ interní část DTD ]>`

Interní nebo **externí** část (*internal or external subset*) nemusí být uvedena nebo mohou být uvedeny obě.

Externí identifikátor může být buď

- `PUBLIC "PUBLIC ID" "URI"` (hodí se pro "veřejná", obecně uznané DTD) nebo
- `SYSTEM "URI"` - pro soukromá nebo jiná "ne zcela standardizovaná" DTD ("URI" nemusí být jen URL na síti, může být i jméno souboru, vyhodnocení se děje podle systému, na němž se vyhodnocuje)

Význam interní a externí části je rovnocenný (a nesmí si odporovat - např. dvě definice téhož elementu).

Obsahem DTD je seznam deklarací jednotlivých prvků - *elementů, seznamů atributů, entit, notací*

DTD - podmíněné sekce

Slouží k "zakomentářování" úseků DTD např. při experimentování.

- `<![IGNORE[ toto se bude ignorovat ]]>`
- `<![INCLUDE[ toto se zahrne do DTD (tj. nebude se ignorovat)]]>`

DTD - definice typu elementu

Popisuje možný obsah elementu, má formu `<!ELEMENT jméno-elementu ... >`, kde `...` může být

- `EMPTY` - prázdný element, může být zobrazen jako `<element/>` nebo `<element></element>` - totéž
- `ANY` - povolen je libovolný obsah elementu, tj. text, dceřinné elementy, ...
- může obsahovat **dceřinné elementy** - `<!ELEMENT jméno-elementu (specifikace dceřinných elementů)>`
- může být **smíšený** (MIXED) - obsahující text i dceřinné elementy dané výčtem `<!ELEMENT jméno-elementu (#PCDATA | přípustné dceřinných elementy)*>`. Nelze specifikovat pořadí nebo počet výskytů konkrétních dceřinných elementů. Hvězdička za závorkou je *povinná* - vždy je možný libovolný počet výskytů.

Pro specifikaci dceřinných elementů používáme:

- operátor **sekvence** (*sequence*, *follow with*) `,`
- operátor **volby** (výběru, *select*, *choice*) `|`
- závorky `()` mají obvyklý význam
- nelze kombinovat v jedné skupině různé operátory `, |`
- počet výskytů dceřinného elementu omezujeme specifikátory "hvězdička", "otazník", "plus" s obvyklými významy. Bez specifikátoru znamená, že je povolen právě jeden výskyt.

DTD - definice atributu

Popisuje (datový) typ, případně implicitní hodnoty atributu u daného elementu.

Má tvar `<!ATTLIST jméno-elementu jméno-atributu typ-hodnoty implicitní-hodnota>`

DTD - definice typu hodnoty atributu

Přípustné *typy hodnot* jsou:

- `CDATA`
- `NMTOKEN`
- `NMTOKENS`
- `ID`
- `IDREF`
- `IDREFS`
- `ENTITY`
- `ENTITIES`
- výčet hodnot - např. `(hodnota1|hodnota2|hodnota3)`
- výčet notací - např. `NOTATION (notace1|notace2|notace3)`

Atribut (i nepovinný) může mít implicitní hodnotu:

- `"implicitní hodnota"` - atribut je nepovinný, ale není-li uveden, chápe se to, jako by měl

hodnotu `implicitní hodnota`

DTD - předpis kardinality (počtu výskytů) atributu

Atributy mohou mít předepsán (povinný) výskyt:

- `#REQUIRED` - atribut je povinný
- `#IMPLIED` - atribut je nepovinný
- `#FIXED "pevná-hodnota"` - atribut je povinný a musí mít právě hodnotu `pevná-hodnota`

DTD - implicitní hodnota atributu

Atribut (i nepovinný) může mít implicitní hodnotu:

- `"implicitní hodnota"` - atribut je nepovinný, ale není-li uveden, chápe se to, jako by měl hodnotu `implicitní hodnota`

Fyzická struktura (entity)

Entita - deklarace a použití

Rozlišuje se:

- deklarace
- reference (tj. použití) dané (již deklarované) entity.

Entity obecné (general) - mohou být

- *parsované* - soubory se (správně utvořeným) značkováním,
- *neparsované* - např. binární soubory,
- *znakové* - znaky, např. `>` je referencí na znakovou entitu

Entity parametrické (parametric)

- mohou být použity *jen v rámci DTD*

- hodí se při např. deklaracích *seznamu atributů* (pokud je dlouhý a vícekrát použitý, nahradíme ho referencí na parametrickou entitu)
- viz např. DTD pro HTML 4.01 - http://www.w3.org/TR/html4/sgml/dtd.html
- definicí parametrické entity je např. `<!ENTITY % heading "H1|H2|H3|H4|H5|H6">`

XML Base

XML Base

- XML Base, W3C Recommendation 27 June 2001: http://www.w3.org/TR/xmlbase/
- Standard pro vyhodnocování relativních URL v odkazech z/na XML dokumenty.
- Definuje použití vyhrazeného atributu `xml:base` označujícího základ pro vyhodnocování relativních URL.
- Doplňuje se se standardem *XLink*.
- Respektuje princip "překrývání" bázové adresy nastavené v nadřazeném elementu.

XML Base - příklad

Příklad 1. xml:base určuje základ pro relativní URL

```
<!--  Slides RelaxNG locations  -->
- <group xml:base="schema/relaxng/" id="slides-relaxng"
    prefer="public">
    <uri name="slides.rng" uri="slides.rng" />
    <uri name="slides-full.rng" uri="slides-full.rng" />
  </group>
```

Poznámka

Všimněte si použití vyhrazeného prefixu `xml:`

Jmenné prostory

Jmenné prostory (XML Namespaces)

- XML Namespaces (W3C Recommendation, aktuálně *Namespaces in XML 1.0 (Second Edition) W3C Recommendation 16 August 2006*): http://www.w3.org/TR/REC-xml-names
- Existuje také nové *Namespaces in XML 1.1 W3C Recommendation [http://www.w3.org/TR/xml-names11/]* 4th February 2004. Andrew Layman, Richard Tobin, Tim Bray, Dave Hollander
- Definují "logické prostory" jmen (elementů, atributů) v XML dokumentu.
- Dávají uzlům ve stromu XML dokumentu "třetí dimenzi".
- Logickému prostoru jmen odpovídá jeden globálně ("celosvětově") jednoznačný identifikátor, daný URI (URI tvoří nadmnožinu URL).
- NS odpovídající danému URI nemá nic společného s obsahem nacházejícím se případně na tomto URL ("nic se odnikud automaticky nestahuje" - nedochází k tzv. dereferenci daného URI).

Prefixy jmenných prostorů, shoda...

- V rámci dokumentů se místo těchto URL používají zkratky, *prefixy* těchto NS namapované na příslušné URI atributem `xmlns:prefix="URI"`.

  Jméno elementu či atributu obsahující dvojtečku se označuje jako *kvalifikované jméno, QName*.
- Dva NS jsou stejné, jestliže se jejich URI shodují po znacích přesně (v kódování UNICODE).
- NS neovlivňují význam textových uzlů.
- Element/atribut nemusí patřit do žádného NS.
- Deklarace prefixu NS nebo implicitního NS má platnost na všechny dceřinné uzly rekurentně, dokud není uvedena jiná deklarace "přemapující" daný prefix.
- Jeden NS je tzv. *implicitní* (*default NS*), deklarovaný atributem `xmlns=`
- Na atributy se *implicitní NS nevztahuje*!!!, čili atributy bez explicitního uvedení prefixu nejsou *v žádném NS*.

Příklad implicitního jmenného prostoru

V následující ukázce je pro celý úryvek platný deklarovaný implicitní jmenný prostor charakterizovaný URI (URL) `http://www.w3.org/1999/xhtml`

Příklad 2. Implicitní jmenný prostor

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```

```
   <body>
      <h1>Hurááááá</h1>
   </body>
</html>
```

Příklad explicitního jmenného prostoru

V následující ukázce je deklarován a přiřazen prefixu `xhtml` jmenný prostor charakterizovaný URI (URL) `http://www.w3.org/1999/xhtml`

Příklad 3. Jmenný prostor mapovaný na prefix

```
<xhtml:html xmlns:xhtml="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
   <xhtml:body>
      <xhtml:h1>Hurááááá</xhtml:h1>
   </xhtml:body>
</xhtml:html>
```

Obtíže se jmennými prostory

Dosud ne všechny parsery dokážou rozpoznávat NS. ...*i když problémy jsou s tím dnes výjimečné*...

NS jsou nekompatibilní s DTD (DTD přísně rozlišuje např. jméno `xi:include` a `include` i v případě, že patří do stejného NS a mají tedy z hlediska aplikace obvykle stejnou interpretaci/význam).

XML Information Set

XML Information Set (XML Infoset) - cíle

- *XML Infoset 2nd Edition W3C Recommendation* First published.og 24 October 2001, revised 4 February 2004, John Cowan, Richard Tobin, http://www.w3.org/TR/xml-infoset/
- Infoset popisuje "jaké všechny informace lze o uzlu (elementu, dokumentu, atributu...) získat"
- Jinými slovy: aplikace by neměla spoléhat na informace z XML dokumentu, které se po analýze (parsingu) neobjeví v Infosetu.
- Každý správně utvořený XML dokument vyhovující standardu pro jmenné prostory má Infoset.

XML Infoset - struktura

- Infoset se skládá z *Information items*
- Infoset se týká dokumentu s již expandovanými entitami
- Rozlišuje se infoset *dokumentu*, *elementu*, *atributu*, *znaku*, *instrukci pro zpracování*, *neexpandované entitě*, *neanalyzované entitě*, *notaci*
- Podrobněji viz specifikace.

Kanonický tvar XML

Kanonický tvar XML dokumentu

- Canonical XML Version 1.0, W3C Recommendation 15 March 2001, http://www.w3.org/TR/xml-c14n
- Smyslem je popsat kritéria (a algoritmy), které pomohou rozhodnout, zda jsou dva XML dokumenty ekvivalentní, lišící se pouze fyzickou reprezentací (entity, pořadí atributů, kódování znaků)
- Kanonizace "setře" rozdíly mezi takovými dokumenty, k nimž se analyzátor "bude jistě chovat stejně", tj. z pohledu aplikace jsou totožné.
- Použití kanonického tvaru je nutné např. u *elektronického podpisu* XML dat (při výpočtu hodnoty *digest*).
- Bylo by možné nad XML dokumenty definovat i jiné relace ekvivalence než je *Canonical XML*.

Kanonický tvar - zásady konstrukce

Hlavní zásady konstrukce kanonického tvaru XML dokumentu:

- kódování v UTF-8
- zlomy řádků (CR, LF) jsou normalizovány podle algoritmu uvedeného v std. XML 1.0
- hodnoty atributů jsou normalizovány
- reference na znakové a parsované entity jsou nahrazeny jejich obsahem
- CDATA sekce jsou nahrazeny jejich obsahem
- hlavička "xml" a deklarace typu dokumentu jsou odstraněny
- bílé znaky mimo kořenový element jsou normalizovány
- jiné bílé znaky (vyjma normalizace zlomu řádků) jsou zachovány
- hodnoty atributů jsou uvozeny "

- speciální znaky v hodnotách atributů a textovém obsahu elementů jsou nahrazeny referencemi na entity
- nadbytečné deklarace jmenných prostorů jsou z každého elementu odstraněny
- implicitní hodnoty atributů jsou dodány do každého elementu (kde je to relevantní)
- na pořadí atributů a deklarací jmenných prostorů se uplatní lexikografické řazení

Potíže při definici kanonického tvaru

Ztráta řady informací (typicky pocházejících z DTD):

- neparsované entity (např. binární entity) jsou po kanonizaci nepřístupné
- notace
- typy atributů (vč. implic. hodnot)

XML Catalogs

XML Catalogs

- Vycházejí ze starších SGML katalogů
- Jde o prostředek, jak se jednotně odkazovat na entity (dokumenty) umístěné na různých systémech na různých místech.
- Dovoluje také praktické použití identifikátorů URI typu PUBLIC, které neodkazují na žádnou reálnou lokaci na internetu.
- Existuje několik formátů pro katalogy - bohužel.

XML Catalogs - příklad

Příklad 4. Katalog pro styly značkování DocBook Slides

```
<?xml version="1.0"?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog">
<!-- Slides DTD locations -->
<group xml:base="schema/dtd/"
       id="slides-dtd"
       prefer="public">
```

```
  <public
    publicId="-//Norman Walsh//DTD Slides Custom V3.1.0//EN"
    uri="slides-custom.dtd"/>

  <public
    publicId="-//Norman Walsh//DTD slides Full V3.1.0//EN"
    uri="slides-full.dtd"/>
</group>

  <rewriteURI
    uriStartString="http://docbook.sourceforge.net/release/xsl/current/"
    rewritePrefix="file:/c:/devel/docbook-xsl-1.62.4/"/>

<!-- Map web references to DocBook 4.2 DTD -->
   <nextCatalog catalog="file:/c:/devel/docbook4.2/catalog.xml" />
</catalog>
```