Technologie XML

Úvod: Validace XML

Jiří Měska (jiri.meska@gmail.com)

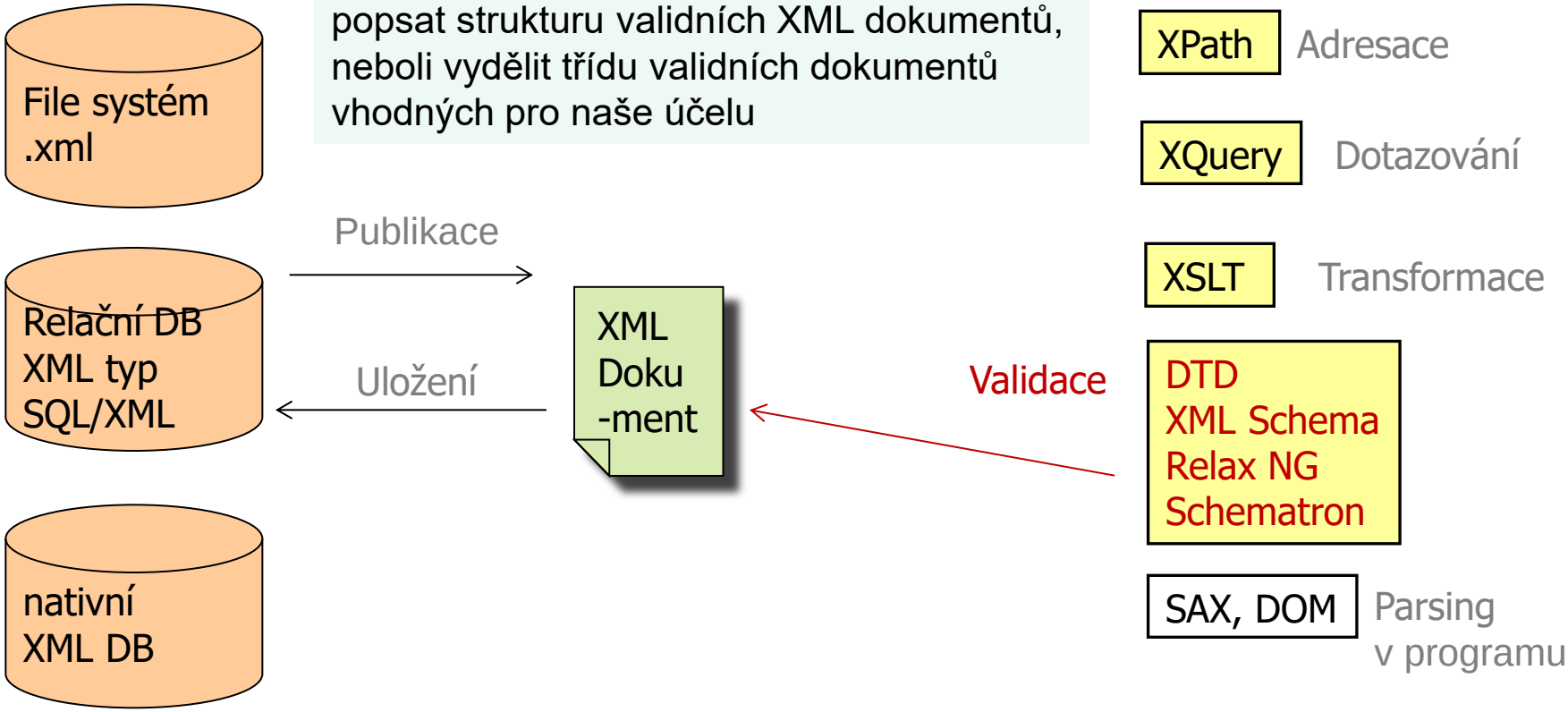
MIB008, Datové a procesní modely,

MFF UK Praha, 2020

Perzistence XML dat

Zpracování XML dat

V této presentaci se budeme snažit popsat strukturu validních XML dokumentů, neboli vydělit třídu validních dokumentů vhodných pro naše účelu



Validace XML dokumentu

XML dokument je "**well formed**" (správně formátovaný), jestliže dokument odpovídá standardu XML 1.0, v podstatě vyhovuje syntaxi, kterou jsme probrali.

XML dokument je „**valid**„ (validní), jestliže odpovídá nějakému syntaktickému předpisu (definici, specifikaci, gramatice). Validní přehled panovníků musí odpovídat např. nějaké DTD definici (už jsem se seznámili), validní XHTML stránka musí splňovat určité syntaktické požadavky.

Pro popis XML jazyků pro dokumenty existuje více standardů/technologii nejdůležitější z nich jsou:

DTD (Document Type Definition) – součást standardu XML, definuje gramatiku dokumentu

XML Schema – **v současné době nejrozšířenější jazyk pro popis XML dokumentů, definuje gramatiku dokumentu**

Většina standardů W3C konsorcia (internetové standardy) v souvislosti s XML používá XML Schémata. Zbytek prezentace je věnován přehledu syntaxe těchto dvou validačních prostředků a návodem na procvičení - příklady jsou součástí cvičení. **Cílem je porozumět těmto konstrukcím, nemusíte je umět psát!**

DTD - příklad odkazu na externí definici

DTD definice Panovnici.dtd →

Dokument Panovnici2a.xml vyhovující zadanému Panovnici.dtd

```
<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE seznam_panovniku SYSTEM "Panovnici.dtd">
<seznam_panovniku>
  <panovnik rod="přemyslovec">
      <jmeno>Přemysl Otakar II.</jmeno>
      <titul>král český</titul>
      <panoval>
          <od>1253</od>
          <do>1278</do>
      </panoval>
  </panovnik>
</seznam_panovniku>
```

```
<!ELEMENT seznam_panovniku (panovnik*)>
<!ELEMENT panovnik (jmeno+, titul, panoval?)>
<!ELEMENT panoval (od, do)>
<!ELEMENT jmeno (#PCDATA)>
<!ELEMENT titul (#PCDATA)>
<!ELEMENT od (#PCDATA)>
<!ELEMENT do (#PCDATA)>
<!ATTLIST panovnik rod CDATA #REQUIRED>
```

DTD - Interní definice

V minulém případě jsme se setkali s externí definicí,

definice DTD může být i interní.

Viz Panovnici2b.xml

```
<!DOCTYPE seznam_panovniku[
<!ELEMENT seznam_panovniku (panovnik*)>
<!ELEMENT panovnik (jmeno+, titul, panoval?)>
<!ELEMENT panoval (od, do)>
<!ELEMENT jmeno (#PCDATA)>
<!ELEMENT titul (#PCDATA)>
<!ELEMENT od (#PCDATA)>
<!ELEMENT do (#PCDATA)>
<!ATTLIST panovnik rod CDATA #REQUIRED>
]>
<seznam_panovniku>
  <panovnik rod="přemyslovec">
        <jmeno>Přemysl Otakar II.</jmeno>
        <titul>král český</titul>
        <panoval>
              <od>1253</od>
              <do>1278</do>
        </panoval>
  </panovnik>
</seznam_panovniku>
```

DTD – deklarace elementu - kvantifikátory

Element je tvořen seznamem jiných elementů a textů

<!ELEMENT panovnik (jmeno+, titul, panoval?)>

Kvantifikátory výskytu:

```
*       … 0 a více
+       … 1 jedna a více
?       … 0 nebo 1
Bez     … právě jednou
```

DTD – deklarace elementu – struktura

Element **note** obsahuje seznam elementů **to**, **from**, **header** a buď element **message** nebo element **body**

<!ELEMENT note (to, from, header, (message | body) >

Element **hodnoceni** obsahuje pouze text

<!ELEMENT hodnoceni (#PCDATA) >

PCDATA – textová data zpracovávaná (parsovaná) validátorem

Prázdný **zemrel** element

<!ELEMENT zemrel (EMPTY)>

Element **poznamka** může obsahovat cokoliv (nespecifikujeme to)

<!ELEMENT poznamka (ANY)>

DTD – Příklad

DTD předpis:

<!ELEMENT panovnik(jmeno, #PCDATA, panoval, #PCDATA)>

<!ELEMENT jmeno (#PCDATA)>

<!ELEMENT panoval (ANY)>

XML Data:

```
<panovnik rod="přemyslovec"> Data vloženy 28.11.2012
        <jmeno>Přemysl Otakar II.</jmeno>
        <titul>král český</titul>
        <panoval>
            <od>1253</od>
            <do>1278</do> Bylo to na Moravském poli
        </panoval>
        Jeho teta ... abatyše kláštera ... Svatořečená 1989
</panovnik>
```

Otázky:

Je či není XML validní? Kde jsou chyby?

Jak se jmenovala jeho teta?

DTD – deklarace atributu

<!ATTLIST panovnik rod CDATA #REQUIRED>

Typem atributu může být:

CDATA – libovolný text, který nebude zpracováván (parsován)

ID- hodnotou je jednoznačný identifikátor v dokumentu

IDREF – hodnotou je reference na jednoznačný ID identifikátor použitý v dokumentu

Dvojice ID, IDREF reprezentují v XML referenční integritu jako foreign key

IDREFS - je seznam IDREF oddělený mezerami

ENTITY – hodnotou atributu je odkaz na v rámci DTD definovanou entitu (&autor;)

<!ENTITY autor „Julius Verne">

ENTITIES – hodnotou je seznam ENTITY

MTOKEN – hodnotou je slovo pro který platí stejné pravidlo jako pro název elementu nebo atributu s výjimkou, že na počátku může být číslice (nepoužívá se)

MTOKENS - seznam MTOKEM oddělený mezerami (nepoužívá se)

NOTATION – hodnotou je název notace (nepoužívá se)

Možno uvést výčet typů oddělený |

Povinný / nepovinný atribut ... #REQUIRED / #IMPLIED

Příklady

Pokyny pro cvičení:

1.Spustěte NetBeans IDE (dostupné v učebnách na MFF)

2.Otevřete soubory z adresáře přednáška xml 1/priklady.

3.Pravým tlačítkem na ploše xml dokumentu:

- Check XML – testuje well form XML dokument
- Validace XML – validuje dokument podle validačního schématu (interní DTD, externí DTD, Schemata)

Přehled DTD: **www.w3scools.com/dtd**

Vyzkoušet:

Panovnici2a.xml

Panovnici2b.xml

XML Schema

XML Schema Definition (XSD), zkráceně XML Schema je standard W3C sloužící k popisu struktury XML.

Oproti DTD má XML Schema následující výhody:

- **XML Schema podporuje datové typy** – lze lépe popsat povolený obsah elementů a atributů a validovat jej, vychází tím vstříc požadavkům na přenos dat mezi programy

- **XML Schema je rozšiřitelné** – XML Schema lze použít a rozšířit v dalším XML Schematu

- **XML Schema je XML dokument** – lze s ním pracovat jako s jakýmkoliv XML

Oproti DTD je XML Schema výrazně komplikovanější a hůře čitelnější.

Pro tvorbu XML Schemat se používají specializované editory (oXygen, Altova, Microsoft XML Schema Designer)

XML Schema popisuje XML „jazyk“ vždy svázaný s nějakým jmenným prostorem

XML Schema se používá např. pro validaci zpráv předávaných webovými službami

XML dokument svázaný s XML schématem – příklad

Soubory: panovnici3.xml, panovnici.xsd

```
<?xml version="1.0" encoding="windows-1250"?>
<seznam_panovniku
xmlns ="http://www.historie.cz"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.historie.cz Panovnici.xsd">
  <panovnik rod="přemyslovec">
      <jmeno>Přemysl Otakar II.</jmeno>
      <titul>král český</titul>
      <panoval>
          <od>1253</od>
          <do>1278</do>
      </panoval>
    </panovnik>
</seznam_panovniku>
```

XML Schema – Připojení schématu k XML dokumentu

Jmenný prostor „jazyka" našeho dokumentu obsahujícího seznam českých panovníků:

xmlns =http://www.historie.cz

Naše nástroje/programy pracující s dokumenty „české historie" by měly zpracovávat prvky z tohoto jmenného prostoru a ignorovat prvky z jiných jmenných prostorů, např. z jmenného prostoru schémat.

Jmenný prostor jazyka XML Schemat

Xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance

Validační nástroje využívají XML schémat musí umět zpracovat elementy tohoto jmenného prostoru.

Atribut schemaLocation je z jmenného prostoru XML Schémat a jeho význam je odkaz na soubor XML Schémat, který má být použit k validaci našeho dokumentu .

xsi:schemaLocation="http://www.historie.cz panovnici.xsd"

XML Schema – část 1

Soubor: Panovnici.xsd

```
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.historie.cz"
elementFormDefault="qualified">
<xs:element name="seznam_panovniku">
<xs:complexType>
   <xs:sequence>
   <xs:element name="panovnik">
..... další slide
   </xs:element>
   </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

XML Schema – část 2

```
<xs:complexType>
    <xs:sequence>
        <xs:element name="jmeno" type="xs:string"/>
        <xs:element name="titul" type="xs:string"/>
        <xs:element name="panoval">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="od" type="xs:integer"/>
              <xs:element name="do" type="xs:integer"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="rod" type="xs:string" use="required"/>
 </xs:complexType>
```

XML Schema – Deklarace cílového jmenného prostoru

Definice jmenného prostoru použitého pro jazyk XML Schemat (elementy a atributy patřící do popisného světa schémat):

xmlns:xs="http://www.w3.org/2001/XMLSchema"

XML Schema popisuje elementy z našeho jazyka, a to pouze elementy z jmenného prostoru:

targetNamespace= http://www.historie.cz

Atributem

elementFormDefault="qualified"

Specifying "qualified" in the schema, which is nearly always the right thing to do, means that local element declarations (xs:element within xs:complexType) refers to elements in the target namespace of the schema. Without it, they refer to elements in no namespace.

So with qualified, in your case, the name element must be in the namespace **http://www.historie.cz**.

XML Schema – deklarace jednoduchého elementu I

Jednoduchý element neobsahuje žádné další elementy, v XML schématech jeho jméno je popsáno atributem **name**, datový typ textového obsahu elementu je popsán atributem **type**:

```
<xs:element name="jmeno" type="xs:string"/>
<xs:element name="do" type="xs:integer"/>
```

XML Schema předdefinované datové typy. Nejčastěji pracujeme s:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

Na datových typech je nejlépe vidět, že XML schemata oproti DTD vycházejí lépe vstříc potřebám kladeným na datovou validaci dat.

XML Schema – deklarace jednoduchého elementu II

Dále lze atributem **default** definovat implicitní hodnotu elementu (je dosazena pokud není):

<xs:element name="barva" type="xs:string" default="modrá" />

Atributem **fixed** Lze požadovat fixní hodnotu elementu:

<xs:element name="jazyk" type="xs:string" fixed="čeština" />

V rámci deklarace jednoduchého elementu lze omezit minimální a maximální počet výskytů elementu v datech. Např. následující řádek připouští uvedení a opakování 1-5 elementů <name>.

<xs:element name="name" type="xs:string" maxOccurs="5" minOccurs="1" />

XML Schema – deklarace atributu

Deklarace atributu rod z našeho příkladu, opět lze atributem **type** popsat typ textového obsahu atributu:

```
<xs:attribute name="rod" type="xs:string"/>
```

Atribut může mít rovněž defultní nebo fixní hodnotu.

```
<xs:attribute name="lang" type="xs:string" default=„CZ"/>
```

```
<xs:attribute name="version" type="xs:string" fixed="1.1"/>
```

Atribut může být povinný (required) nebo volitelný (optional)

```
<xs:attribute name="rod" type="xs:string" use="required"/>
```

Příklady

Pokyny pro cvičení:

1.Spustěte NetBeans IDE

2.Otevřete soubory z adresáře prednaska xml 1/priklady.

3.Pravým tlačítkem na ploše xml dokumentu:

- Check XML – testuje well form XML dokument
- Validace XML – validuje dokument podle validačního schématu (interní DTD, externí DTD, Schemata)

Vyzkoušet:

Panovnici3.xml

Panovnici.xsd

Vyzkoušet:

XSDUkazka1.xml

XSDUkazka1.xsd

Vyzkoušet:

XSDUkazka2.xml

XSDUkazka2.xsd

XML Schema – restrikce jednoduchého typu I

V našem popisu panovníka v elementech **od, do** specifikujeme období vlády panovníka. Rok lze popsat datovým typem integer.

```
<xs:element name="do" type="xs:integer"/>
```

Rádi bychom tuto kontrolu zpřísnily na období 850-2018. Můžeme k tomu použít restrikce v rámci definice jednoduchého typu:

```
<xs:element name="do">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="850"/>
            <xs:maxInclusive value="2018"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

V podkladech na cvičení 8 naleznete ukázku: XSDUkazka1.xml, XSDUkazka1.xsd

Můžete upravit v XML souboru hodnotu do mimo rozsah a vyzkoušet validaci.

XML Schema – restrikce jednoduchého typu II

Následující popis, říká, že titul může být libovolný znakový řetězec:

```
<xs:element name="titul" type="xs:string"/>
```

Rádi bychom omezily tituly na výběr z nějakého číselníku titulů. Volbu titulů přece nelze ponechat nahodilostem ☺:

```
<xs:element name="titul">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="král český"/>
            <xs:enumeration value="král římský"/>
            <xs:enumeration value="císař německý"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

V podkladech na cvičení 8 naleznete ukázku: XSDUkazka2.xml, XSDUkazka2.xsd

Zkuste zadat jako titul např. prezident. Bude dokument validní?


Thanks