

XML Schema

Irena Mlýnková

irena.mlynkova@mff.cuni.cz

Karlova Univerzita

Matematicko-fyzikální fakulta

Katedra softwarového inženýrství

Úvod

- **XML – nástroje pro popis dat + nástroje pro popis jejich (přípustné) struktury**
- **DTD (Document Type Definition)**
 - **Stále nejpoužívanější nástroj**
 - **Jednoduchý**
 - **Prozatím postačující (?) vyjadřovací síla**
- **Složitější aplikace, požadavky na přesnější vyjádření struktury => DTD nestačí => nutnost použít silnější nástroj => např. XML Schema**

DTD – připomenutí

- **Možnost definovat:**
 - **elementy a atributy**
 - **vztahy element-podelement a element-atribut,**
 - **přípustný obsah elementů (prázdný, textový, elementový, smíšený...)**
 - **pořadí (, |) a počet výskytů (? + *) elementů v rámci nadelementu**
 - **(v omezené míře) datové typy, povinnost výskytu a implicitní hodnoty atributů**
- **Pro jednodušší aplikace to obvykle stačí**

DTD – příklad (1)

```
<!ELEMENT zamestnanci ( osoba )+>
<!ELEMENT osoba ( jmeno, email*, vztahy? )>
  <!ATTLIST osoba id ID #REQUIRED>
  <!ATTLIST osoba poznamka CDATA #IMPLIED>
  <!ATTLIST osoba dovolena ( ano | ne ) "ne">
<!ELEMENT jmeno ( (krestni, prijmeni) | (prijmeni, krestni) )>
<!ELEMENT krestni ( #PCDATA )>
<!ELEMENT prijmeni ( #PCDATA )>
<!ELEMENT email ( #PCDATA )>
<!ELEMENT vztahy EMPTY>
  <!ATTLIST vztahy nadrizeny IDREF #IMPLIED>
  <!ATTLIST vztahy podrizeni IDREFS #IMPLIED>
```

DTD – XML dokument (2)

```
<zamestnanci>  
  <osoba id="nacelnik" poznamka="Nemecky Cech"  
    dovolena="ano">  
    <jmeno>  
      <krestni>Karel</krestni>  
      <prijmeni>Nemec</prijmeni>  
    </jmeno>  
    <email>karel.nemec@cimr.cz</email>  
    <vztahy podrizeni="lekarnik ucitel nosic"/>  
  </osoba>  
  ...  
</zamestnanci>
```

XML Schema – přínosy (1)

- **Nevyžaduje speciální syntaxi (XML schémata = XML dokumenty)**
- **Silná podpora datových typů (vestavěné x uživatelsky definované)**
- **Možnost opakovaně využívat již definované části (tzv. globální prvky)**
- **Možnost přesně vyjádřit přípustné počty výskytů elementů v rámci nadelementu**

XML Schema – přínosy (2)

- Rozlišuje mezi posloupností a množinou elementů
- Objektově-orientované prvky (dědičnost, substituovatelnost apod.) => pro modelování přirozené
- Možnost specifikovat unikátnost elementu, atributu nebo jejich kombinace v rámci požadované části XML dokumentu

XML Schema – úvod

- Jazyk pro definici přípustné struktury XML dokumentů
- Specifikace W3C z r. 2001 (3 části)
- Problémy v pojmech:
 - XML schéma = schéma XML dat v libovolném jazyce (nadále v jazyce XML Schema)
 - XML schémata = XML dokumenty => přípustné elementy a atributy XML dokumentu definujeme pomocí elementů a atributů jazyka XML Schema

XML Schema – základy (1)

- **Definice XML schématu:**
 - **Kořenový element schema**
 - **Využití jmenných prostorů (JP) – JP jazyka XML Schema, cílový JP, implicitní JP**

<xs:schema

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

targetNamespace="http://www.mff.cuni.cz/MojeSchema"

xmlns="http://www.mff.cuni.cz/MojeSchema">

... <!-- definice XML schématu --> ...

</xs:schema>

XML Schema – základy (2)

- Připojení XML schématu k dokumentu:
 - JP instancí XML schémat, implicitní JP
 - Schéma bez cílového JP
(noNamespaceSchemaLocation)

<KorenovyElement

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

xsi:schemaLocation="http://www.mff.cuni.cz/MojeSchema
schema1.xsd"

xmlns="http://www.mff.cuni.cz/MojeSchema">

...

</KorenovyElement>

XML Schema – základy (3)

- **Princip definice schématu:**
 - **Definice datových typů**
 - **Definice elementů a atributů (= název + datový typ)**
- **Prvky XML schématu:**
 - **Jednoduché – jednoduché + složené datové typy, elementy, atributy, skupiny atributů**
 - **Pokročilé – omezení identity, substituční skupiny, zástupci, externí schémata**

Jednoduché datové typy (1)

- **Element simpleType + jeho atributy**
- **Omezení textových hodnot**
- **Vestavené – předdefinované**
 - **Základní x odvozené (ze základních)**
- **Uživatelsky definované – specifikované uživatelem prostřednictvím tzv. odvození:**
 - **Restrikcí x seznamem x sjednocením**

Jednoduché datové typy – vestavěné (2)

- **Základní:**
 - př. string, boolean, decimal, float, dateTime (YYYY-MM-DDThh:mm:ss.ss), date, time, gYear, gMonth, hexBinary...
- **Odvozené – od dvou základních typů:**
 - string – př. normalizedString (bez CR, LF a tabulátoru), token, language (př. en, en-GB...), ID, IDREF, IDREFS (z DTD)...
 - decimal – př. integer, positiveInteger, negativeInteger, long, int, short, byte, unsignedLong, unsignedInt...

Jednoduché datové typy – uživatelsky definované (3)

- **Odvození restrikcí => podmnožina hodnot
původního typu**

```
<xs:simpleType name="NeprazdnyRetezec">  
  <xs:restriction base="xs:string">  
    <xs:minLength value="1"/>  
    <xs:maxLength value="100"/>  
  </xs:restriction>  
</xs:simpleType>
```

- **length, pattern, enumeration, maxInclusive,
minInclusive, maxExclusive, minExclusive,
totalDigits, fractionDigits...**

Jednoduché datové typy – uživatelsky definované (4)

- **Odvození seznamem => seznam hodnot původního typu oddělených mezerou**
 - **Vícehodnotové datové typy**
 - **Nelze odvozovat z jiných vícehodnotových typů**

```
<xs:simpleType name="SeznamRealnychCisel">  
  <xs:list itemType="xs:float"/>  
</xs:simpleType>
```

Jednoduché datové typy – uživatelsky definované (5)

- Odvození sjednocením => sjednocení
hodnot všech sjednocovaných typů

```
<xs:simpleType name="NenulovaCelaCisla">
```

```
  <xs:union
```

```
    memberTypes="xs:positiveInteger xs:negativeInteger"/>
```

```
</xs:simpleType>
```


Jednoduché datové typy – globální x lokální (6)

```
<xs:simpleType name="SjednoceníTypu">  
  <xs:union>  
    <xs:simpleType>  
      <xs:restriction base="xs:positiveInteger">  
        <xs:minInclusive value="8"/>  
        <xs:maxInclusive value="72"/>  
      </xs:restriction>  
    </xs:simpleType>  
    ...  
  </xs:union>  
</xs:simpleType>
```

Atributy (1)

- **Element attribute + jeho atributy**
- **Název + jednoduchý typ**
 - Vestavěný x uživatelsky definovaný
 - Globální x lokální (viz. příklad)
- **Atributy: default, fixed, use**
- **Globálně x lokálně definované (podobně jako jednoduché typy)**
- **Elementům přiřadíme pomocí složených datových typů (viz. dále)**

Atributy (2)

```
<xs:attribute name="Vek" type="xs:positiveInteger"/>
```

```
<xs:attribute name="Nazev" type="NeprazdnyRetezec"/>
```

```
<xs:attribute name="TelefoniCislo">
```

```
  <xs:simpleType>
```

```
    <xs:restriction base="xs:string">
```

```
      <xs:length value="10"/>
```

```
      <xs:pattern value="\d{3}-\d{6}"/>
```

```
    </xs:restriction>
```

```
  </xs:simpleType>
```

```
</xs:attribute>
```

Elementy (1)

- **Element element + jeho atributy**
- **Název + jednoduchý / složený typ**
 - **Vestavěný x uživatelsky definovaný**
 - **Globální x lokální (viz. příklad)**
- **S jednoduchým x složeným typem (atributy, podelementy, smíšený obsah)**
- **Globálně x lokálně definované => kořenové elementy XML dokumentů**

Elementy (2)

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

```
<xs:element name="Prijmeni">  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:minLength value="1"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

Složené datové typy (1)

- **Element complexType + jeho atributy**
- **Pro definici složitějších typů elementů**
 - **Vztahy element-podelement a element-atribut**
 - **Počty a pořadí elementů v rámci nadelementu**
- **Obsahuje (viz. příklad):**
 - **Specifikaci obsahu složeného typu (prázdná => složený typ elementu bez obsahu)**
 - **Specifikaci množiny atributů (prázdná => složený typ elementu bez atributů)**
- **Atributy: mixed**

Složené datové typy – příklad (2)

```
<xs:complexType name="Adresa">  
  <xs:sequence>  
    <xs:element name="Ulice" type="xs:string"/>  
    <xs:element name="CDomu" type="xs:integer"/>  
    <xs:element name="Mesto" type="xs:string"/>  
  </xs:sequence>  
  <xs:attribute name="Zeme" type="xs:NMTOKEN" default="CZ"/>  
</xs:complexType>  
  
<xs:element name="MojeAdresa" type="Adresa"/>
```

Složené datové typy (3)

- **Typy obsahu složeného typu:**
 - s jednoduchým obsahem (simpleContent)
 - posloupnost elementů (sequence)
 - výběr z elementů (choice)
 - množina elementů (all)
 - modelová skupina (group)
 - se složeným obsahem (complexContent)
- **Globálně (viz. příklad) x lokálně definované (stejně jako u jednoduchých typů)**

Složené datové typy – s jednoduchým obsahem (4)

- Obsahem je jednoduchý typ + atributy

```
<xs:complexType name="Typ">
```

```
  <xs:simpleContent>
```

```
    <xs:extension base="xs:string">
```

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

```
    </xs:extension>
```

```
  </xs:simpleContent>
```

```
</xs:complexType>
```

Složené datové typy – posloupnost elementů (5)

- Obsahem jsou všechny specifikované prvky v daném pořadí

```
<xs:complexType name="Osoba">  
  <xs:sequence>  
    <xs:element name="Jmeno" type="xs:string" maxOccurs="5"/>  
    <xs:element name="Prijmeni" type="xs:string"/>  
    <xs:element name="DatumNar" type="xs:date"/>  
    <xs:element name="Pozn" type="xs:string" minOccurs="0"/>  
  </xs:sequence>  
  <xs:attribute name="Id" type="xs:ID"/>  
</xs:complexType>
```

Složené datové typy – výběr z elementů (6)

- Obsahem je jeden ze specifikovaných prvků

```
<xs:complexType name="DopravniProstredok">
```

```
  <xs:choice>
```

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

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

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

```
  </xs:choice>
```

```
</xs:complexType>
```

Složené datové typy – množina elementů (7)

- Obsahem jsou specifikované elementy v libovolném pořadí

```
<xs:complexType name="Kniha">
```

```
  <xs:all>
```

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

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

```
    <xs:element name="Vydani" type="xs:date"/>
```

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

```
  </xs:all>
```

```
</xs:complexType>
```

- **maxOccurs** elementů i celé množiny max. 1

Složené datové typy – modelová skupina (8)

- **Obsahuje posloupnost / množinu / výběr z elementů**
- **Vždy deklarována globálně => opakované využití typů obsahů (pomocí referencí)**
- **Reference**
 - **Deklarujeme stejným elementem jako referencovaný prvek + atributem ref**
 - **Pro modelové skupiny i elementy, atributy a množiny atributů (viz. dále)**

Složené datové typy – modelová skupina (9)

```
<xs:group name="ElementyProPublikaci">  
  <xs:sequence>  
    <xs:element name="Nazev" type="xs:string"/>  
    <xs:element name="Autor" type="xs:string" maxOccurs="5"/>  
  </xs:sequence>  
</xs:group>  
<xs:complexType name="Kniha">  
  <xs:sequence>  
    <xs:group ref="ElementyProPublikaci"/>  
    <xs:element name="ISBN" type="xs:string"/>  
  </xs:sequence>  
</xs:complexType>
```

Složené datové typy – se složeným obsahem (10)

- **Odvozování nových typů z již existujících**
- **Restrikcí (restriction) – nový typ je podmnožinou původního (např. omezením počtu výskytů elementů, omezením přípustných hodnot použitých jednoduchých typů...)**
- **Rozšířením (extension) – vytvořený typ obsahuje původní i nový typ (tj. jeho prvky) v tomto pořadí (obdoba dědičnosti)**

Složené datové typy – se složeným obsahem (11)

```
<xs:complexType name="Publikace">  
  <xs:sequence>  
    <xs:element name="Název" type="xs:string"/>  
    <xs:element name="Autor" type="xs:string"  
                minOccurs="1" maxOccurs="unbounded"/>  
    <xs:element name="Datum" type="xs:gYear"/>  
  </xs:sequence>  
</xs:complexType>
```


Složené datové typy – se složeným obsahem (12)

```
<xs:complexType name="PublikaceSJednimAutorem">  
  <xs:complexContent>  
    <xs:restriction base="Publikace">  
      <xs:sequence>  
        <xs:element name="Nazev" type="xs:string"/>  
        <xs:element name="Autor" type="xs:string"/>  
        <xs:element name="Datum" type="xs:gYear"/>  
      </xs:sequence>  
    </xs:restriction>  
  </xs:complexContent>  
</xs:complexType>
```

Složené datové typy – se složeným obsahem (13)

```
<xs:complexType name="Kniha">  
  <xs:complexContent>  
    <xs:extension base="Publikace">  
      <xs:sequence>  
        <xs:element name="Vydavatel" type="xs:string"/>  
        <xs:element name="ISBN" type="xs:string"/>  
      </xs:sequence>  
    </xs:extension>  
  </xs:complexContent>  
</xs:complexType>
```

Skupiny atributů

- Opakované využívání množiny atributů

```
<xs:attributeGroup name="SpolecneAtributy">  
  <xs:attribute name="Vypujcen" type="xs:boolean"/>  
  <xs:attribute name="Id" type="xs:ID"/>  
</xs:attributeGroup>  
<xs:complexType name="Kniha">  
  <xs:sequence>  
    <xs:element name="Nazev" type="xs:string"/>  
    <xs:element name="Vydavatel" type="xs:string"/>  
  </xs:sequence>  
  <xs:attributeGroup ref="SpolecneAtributy"/>  
</xs:complexType>
```

Shrnutí

- **Definice XML schématu:**
 - Definice jednoduchých + složených datových typů
 - Přiřazení typů atributům / elementům
 - Opakovaně využívané prvky definovány globálně + využití referencí
 - Elementy, atributy, modelové skupiny a skupiny atributů
- **Větší příklad XML schématu + podrobnější přehled pokročilých prvků – viz. článek ve sborníku**

Závěr

- **Nevýhody jazyka XML Schema:**
 - Délka a (ne)přehlednost zápisu schématu
 - Komplexnost – složitější pro pochopení
- **Hlavní výhoda: objektově-orientované rysy**
 - Přirozený nástroj pro modelování
 - Možnost zpracovávat v OO programovacích jazycích (technologie XML Data Binding) a OR nebo OO DBS (mapping)