

---

# Validace a schémata XML dat (XML Schema, Relax NG, Schematron...)

## Obsah

Specifikace a validita XML .....	2
Aktuální specifikace XML .....	2
Jakou verzi použít? .....	2
Validita XML dokumentů .....	2
XML Schema .....	2
XML Schema - základní zdroje informací .....	3
XML Schema - motivace .....	3
XML Schema - hlavička definice schématu .....	3
XML Schema - přiřazení typu elementu s daným názvem .....	3
XML Schema - definice jednoduchého typu .....	4
XML Schema - definice jednoduchého typu - příklad 1 .....	4
XML Schema - definice jednoduchého typu - příklad 2 .....	4
XML Schema - jednoduché typy - "union" .....	4
XML Schema - jednoduché typy - seznam hodnot .....	5
XML Schema - definice složeného typu .....	5
XML Schema - definice složeného typu - skupiny .....	5
XML Schema - definice složeného typu - skupiny atributů .....	6
XML Schema - použití skupin .....	6
XML Schema - kompozitor "sequence" .....	6
XML Schema - kompozitor "choice" .....	7
XML Schema - kompozitor "all" .....	7
XML Schema - jednoduchý obsah elementu .....	7
XML Schema - smíšený obsah elementu .....	8
XML Schema - další možnosti .....	8
XML Schema - anotace schémat .....	8
XML Schema - znovupoužití definice schématu .....	9
XML Schema - abstraktní a konečné typy .....	9
XML Schema - jmenné prostory .....	9
XML Schema - nspecifikované elementy a atributy .....	9
XML Schema - odkaz na definici schématu .....	10
Relax NG .....	10
Relax NG - motivace .....	10
Relax NG - základní zdroje informací .....	10
Jazyky schémat používající vzory .....	11
Schematron .....	11
Exampatron .....	11
Ostatní jazyky schémat .....	11
DSD 2.0 .....	11

Vyjadřovací síla těchto modelů, jejich nedostatky .....	11
Nástroje na validaci XML dat modelovaných podle těchto standardů .....	11

## 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/xml/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 může 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 schémata (RelaxNG, Schematron).

## XML Schema

## XML Schema - základní zdroje informací

Specifikace XML Schema - <http://www.w3.org/XML/Schema>

Tutoriál *Using W3C XML Schema*: <http://www.xml.com/pub/a/2000/11/29/schemas/part1.html> - stručný

*XML Schema Tutorial* - <http://www.w3schools.com/schema/default.asp> - obsáhlejší

vynikající komplexní tutoriál na <http://www.xfront.com>

## XML Schema - motivace

Dát silnější prostředek pro specifikaci modelu XML dat než je DTD; mít možnost:

- Oddělit koncept *typu* (např. typu elementu) od jeho *výskytu* (instance, např. elementu s určitým názvem) - to DTD neumí
- Poskytnout bohatší škálu *primitivních datových typů*
- Umožnit použití *jmenných prostorů*
- Umožnit jemnější specifikaci *modelu obsahu* (elementů)
- Umožnit *odvozování* nových typů (*dědičnosti*)
- Umožnit *modularizaci* a znovupoužitelnost schémat
- Zapisovat schéma v XML

## XML Schema - hlavička definice schématu

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  .../...  
</xs:schema>
```

## XML Schema - přiřazení typu elementu s daným názvem

```
<xs:element name="element_name">  
  ... definice typu - je přímo zde - tzv. "local" nebo daná odkazem - tzv. "global"
```

```
</xs:element>
```

## XML Schema - definice jednoduchého typu

- Neobsahuje dceřinné elementy, lze použít jako typ obsahu elementu nebo atributu
- Lze definovat restrikcí z existujícího typu

```
<xs:simpleType name="TypeName">  
  <xs:restriction base="BaseTypeName"> ... </xs:restriction>  
</xs:simpleType>
```

## XML Schema - definice jednoduchého typu - příklad 1

Restrikce délky obsahu

```
<xs:simpleType name="nameType">  
  <xs:restriction base="xs:string">  
    <xs:maxLength value="32"/>  
  </xs:restriction>  
</xs:simpleType>
```

## XML Schema - definice jednoduchého typu - příklad 2

Restrikce obsahu regulárním výrazem

```
<xs:simpleType name="isbnType">  
  <xs:restriction base="xs:string">  
    <xs:pattern value="[0-9]{10}"/>  
  </xs:restriction>  
</xs:simpleType>
```

## XML Schema - jednoduché typy - "union"

Zhruba odpovídá konceptu "union" v C

Výsledkem je jednoduchý typ

Lze spojovat bázevý typ a výčet hodnot

Příklad:

```
<xs:simpleType name="isbnType">  
  <xs:union>
```

```
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]{10}"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType>
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="TBD"/>
    <xs:enumeration value="NA"/>
  </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
```

## XML Schema - jednoduché typy - seznam hodnot

Lze definovat typ jako seznam hodnot oddělených bílými znaky

Dalším odvozením lze omezit počet prvků seznamu

Příklad

```
<xs:simpleType name="isbnTypes">
  <xs:list itemType="isbnType"/>
</xs:simpleType>
<xs:simpleType name="isbnTypes10">
  <xs:restriction base="isbnTypes">
    <xs:minLength value="1"/>
    <xs:maxLength value="10"/>
  </xs:restriction>
</xs:simpleType>
```

## XML Schema - definice složeného typu

```
<xs:complexType name="TypeName">
  <xs:sequence>
    <xs:element ...>
    ...
    <xs:attribute ...>
  </xs:sequence>
</xs:complexType>
```

Místo sekvence lze použít <xs:choice> a <xs:all>

## XML Schema - definice složeného typu - skupiny

při definici složeného typu lze použít skupiny (group)

Skupina elementů:

```
<xs:group name="GroupName">
  <xs:sequence>
    <xs:element ... />
    ...
  </xs:sequence>
</xs:group>
```

Místo sekvence lze použít <xs:choice> a <xs:all>

## XML Schema - definice složeného typu - skupiny atributů

Skupina atributů:

```
<xs:attributeGroup name="AttributesGroupName">
  <xs:attribute ... use="required"/>
  ...
</xs:attributeGroup>
```

Může být uvedena povinnost výskytu (use=required)

## XML Schema - použití skupin

Příklad použití skupin elementů a atributů

```
<xs:complexType name="bookType">
  <xs:sequence>
    <xs:group ref="mainBookElements"/>
    <xs:element name="character" type="characterType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attributeGroup ref="bookAttributes"/>
</xs:complexType>
```

## XML Schema - kompozitor "sequence"

Předepisuje výskyt dceřinných elementů v určitém pořadí

```
<xs:element name="element_name">
  <xs:complexType>
    <xs:sequence>
      .../...
```

```
</xs:sequence>
  .../...
</xs:complexType>
</xs:element>
```

sequence označuje model obsahu připouštějící výskyt dané posloupnosti (sekvence) dceřinných elementů

xs je prefix vázaný na NS s URL <http://www.w3.org/2001/XMLSchema>

Místo `<xs:sequence>` lze použít `<xs:choice>` nebo `<xs:all>`

## XML Schema - kompozitor "choice"

Předepisuje výskyt jednoho z dceřinných elementů nebo skupin elementů

```
<xs:element name="element_name">
  <xs:complexType>
    <xs:choice>
      .../...
    </xs:choice>
    .../...
  </xs:complexType>
</xs:element>
```

## XML Schema - kompozitor "all"

Předepisuje výskyt dceřinných elementů bez určeného pořadí

Smí být jen na nejvyšší úrovni definice obsahu

Dceřinné elementy nesmí mít kardinalitu větší než 1

Příklad:

```
<xs:complexType name="bookType">
  <xs:all>
    <xs:element name="title" type="xs:string"/>
    <xs:element name="author" type="xs:string"/>
    <xs:element name="character" type="characterType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:all>
  <xs:attribute name="isbn" type="isbnType" use="required"/>
</xs:complexType>
```

## XML Schema - jednoduchý obsah elementu

Příklad:

```
<xs:element name="book">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="isbn" type="isbnType"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

## XML Schema - smíšený obsah elementu

Nelze validovat textový obsah (textové dceřinné uzly)

Lze validovat dceřinné elementy

Příklad:

```
<xs:element name="book">
  <xs:complexType mixed="true">
    <xs:all>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="author" type="xs:string"/>
    </xs:all>
    <xs:attribute name="isbn" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

## XML Schema - další možnosti

Možnost specifikace integritních omezení: hodnota je jedinečná - `xs:unique` hodnota je klíčem - `xs:key` hodnota je odkazem na klíč - `xs:keyref`

## XML Schema - anotace schémat

Anotace je (lidsky čitelná) poznámka-komentář ke schématu

Může též obsahovat informace pro zpracování - viz příklad - `xs:appinfo`

Další obsah není předepsán (omezen) - viz příklad - `bind`, `class`

Příklad

```
<xs:annotation>
  <xs:documentation xml:lang="en">Top level element.</xs:documentation>
  <xs:documentation xml:lang="fr">Element racine.</xs:documentation>
</xs:annotation>
```



```
<xs:appinfo source="http://example.com/foo/">
  <bind xmlns="http://example.com/bar/">
    <class name="Book"/>
  </bind>
</xs:appinfo>
</xs:annotation>
```

## XML Schema - znovupoužití definice schématu

Přímo:

```
<xs:include schemaLocation="character.xsd"/>
```

S předefinováním:

```
<xs:redefine schemaLocation="character12.xsd">
  <xs:simpleType name="nameType">
    <xs:restriction base="xs:string">
      <xs:maxLength value="40"/>
    </xs:restriction>
  </xs:simpleType>
</xs:redefine>
```

## XML Schema - abstraktní a konečné typy

*abstract* - nelze instanciovat, pouze jako základ k odvozování dědičností

*final* - nelze rozšiřovat/odvozovat dědičností

## XML Schema - jmenné prostory

Příklad

```
<xs:schema targetNamespace="http://example.org/ns/books/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:bk="http://example.org/ns/books/"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  .../...
</xs:schema>
```

## XML Schema - nspecifikované elementy a atributy

Umožní připustit i něco, co předem neznáme

#### Příklad

```
<xs:complexType name="descType" mixed="true">
  <xs:sequence>
    <xs:any namespace="http://www.w3.org/1999/xhtml"
      processContents="skip"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Pro atributy - xs:anyAttribute

## XML Schema - odkaz na definici schématu

```
<book isbn="0836217462"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file:library.xsd">
```

```
<book isbn="0836217462"
  xmlns="http://example.org/ns/books/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="file:library.xsd">
```

## Relax NG

### Relax NG - motivace

XML Schema:

- Je (zbytečně) složité (specifikace má přes 200 stran)
- Může vést v jistých situacích k nejednoznačnostem.
- Snaží se o pokrytí všech aplikačních oblastí (dokumentové i databázové použití XML a všechno mezi tím).
- Obtížně (úplně) implementovatelné.
- Dále viz <http://www.xml.com/lpt/a/2002/01/23/relaxng.html>

### Relax NG - základní zdroje informací

Vznikl z RELAXu při skupině OASIS-OPEN:

- <http://www.oasis-open.org/committees/relax-ng>

## Jazyky schémat používající vzory

### Schematron

Schematron home page [<http://www.ascc.net/xml/resource/schematron/schematron.html>]

### Examplotron

Examplotron home page [<http://examplotron.org>]

## Ostatní jazyky schémat

### DSD 2.0

Vznikl na Univerzitě v Aarhusu, DK

Podobně jako RELAX NG je jednodušší než XML Schema

viz <http://www.brics.dk/~amoeller/XML/>

Spiše akademický charakter, skutečnými soupeři zůstávají XML Schema a RELAX NG

## Vyjadřovací síla těchto modelů, jejich nedostatky

viz <http://www.xml.com/lpt/a/2001/12/12/schemacompare.html>

## Nástroje na validaci XML dat modelovaných podle těchto standardů

Nástroje na validaci XML dat modelovaných podle těchto standardů