



Baze podataka 2

XML i XML Schema



XML

- XML je jezik za predstavljanje polustrukturiranih podataka.
- Skraćenica od eXtensible Markup Language.
- Jezik označavanja sličan HTML-u.
- Razvijen je od strane W3C (World Wide Web Consortium) sa ciljem prevazilaženja nedostataka HTML-a.
- XML nije zamena za HTML. Tendencija je da se HTML koristi za prikazivanje, a XML za opisivanje podataka.



XML

- Ne mora se definisati shema.
- Fleksibilan je i proširiv, omogućava postojanje različitih tipova podataka u okviru jednog dokumenta.
- Opisuje podatke stavljanjem akcenta na to šta podaci jesu a ne kako oni izgledaju.
- Ima format koji je čitljiv za čoveka.
- Nezavistan je od platforme odnosno od softvera i hardvera koji se koristi.
- Postoji veliki broj gotovih aplikacija za procesiranje XML-a koje se mogu koristiti.
- Manipulisanje podacima često sporije.
- Optimizacija je kompleksnija zahvaljujući bogatstvu i velikoj izražajnoj moći upitnih jezika koje koristi.



XML dokumenti

- centralni koncept XML jezika
- konstruiše se pomoću dva osnovna koncepta:
 - element
 - glavni gradivni element XML dokumenta
 - opisuje se pomoću početnog i krajnjeg taga (oznake)
 - moguće napraviti hijerarhiju elemenata
 - atribut
 - karakteristika koja detaljnije opisuje element i reprezentuje njegovu semantiku
 - upotreba atributa ne mora uvek odgovarati upotrebi atributa u relacionom modelu podataka



Element

- složeni element
 - konstruisan pomoću drugih elemenata
- primitivni (prosti) element
 - sadrži atomične vrednosti
- glavne razlike između XML-a i HTML-a
 - nazivi elemenata
 - u XML dokumentu reprezentuju semantiku
 - u HTML dokumentu definišu isključivo način prezentacije podataka
 - specifikacija elemenata
 - sadržana je u XML šema dokumentu i korisnički definisana
 - u HTML jeziku, svi elementi su unapred definisani



Atribut

- najčešće se koristi za opis podataka smeštenih u elementima
 - slično atributima u HTML-u
- može kao vrednosti da sadrži osnovne ili dodatne podatke koje ne sadrži sam element
- tri pristupa strukturiranja podataka
 - svi podaci se pridružuju direktno elementu
 - svi podaci se pridružuju isključivo atributima elementa
 - neki ("osnovni") podaci se pridružuju elementu, a ostali ("dodatni") atributima elementa



XML - Primer

```
<messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```



Struktura XML dokumenta

- Hijerarhijska struktura (stablo) koja se sastoji iz elemenata, atributa i znakovnih podataka
- XML dokument ima jedan i samo jedan koreni (root) element
- Svi ostali elementi u strukturi su elementi "deca" korenog elementa
 - Dozvoljeno je višestruko ugnježdavanje elemenata



XML deklaracija

- Svaki XML dokument mora da sadrži XML deklaraciju, odnosno instrukciju obrade kojom se dokument identifikuje kao XML dokument.
 - Osnovni oblik XML deklaracije:

```
<?xml version="1.0"?>
```
 - Opcioni oblik XML deklaracije:

```
<?xml version="1.0"encoding="UTF-8"?>
```



XML deklaracija

- `<?xml version="1.0"encoding="UTF-8"?>`
- ? Oznaka za instrukciju obrade (poruka programima koji procesiraju XML document)
- Atribut `version` specificira XML verziju
- Atribut `encoding` definiše znakovni kod u kome je XML dokument napisan
 - UTF-8 (kompresovana verzija Unicode-a)
 - UTF-16 (Unicode)



Validacija XML dokumenta

- Definisiranje tipova XML dokumenata
- W3C je ponudio dva standarda za definiranje tipova XML dokumenta, odnosno opisivanje strukture XML dokumenta:
 - Document Type Definition (DTD)
 - XML Schema Definition (XSD)
- Validni XML dokument
 - Dobro-odformljen
 - Konzistentan sa strukturom definisanom u opisu tipa dokumenta



XML Schema

- XML Schema predstavlja (XML) dokument koji sadrži opis strukture i pravila koja čine validan XML dokument.
- XML dokument ako zadovoljava sva ograničenja zadana šemom, je validan za tu šemu.
- XML Schema dokumenti imaju ekstenziju .xsd - pišu se u XSD (XML Schema Definition) jeziku.



XML Schema

- omogućava validaciju XML dokumenta prema detaljnoj specifikaciji
- podržava sintaksu XML-a
- omogućava tipizaciju podataka i prikaz
- ograničenja
- koristi prostor imena (namespace)
- prezentuje veze koje postoje između elemenata



Struktura XML Schema-e

```
<?xml version="1.0"?>
```

```
<xs:schema>
```

```
...
```

```
</xs:schema>
```



Atributi elementa schema

```
<?xml version="1.0"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
targetNamespace="http://www.w3schools.com "  
xmlns="http://www.w3schools.com"  
elementFormDefault="qualified">  
</xs:schema>
```

- *xmlns:xs* - Označava da su elementi i tipovi podataka korišćeni u ovoj šemi iz tog prostora imena, za koji će se koristiti prefiks xs
- *targetNamespace* - Označava da su elementi definisani sa ovom šemom iz tog prostora imena.
- *xmlns* - Označava koji je podrazumevani prostor imena.
- *elementFormDefault* - Označava da li elementi korišćeni od XML dokumenata mora biti definisan prostor imena.



Pridruživanje šeme XML dokumentu

```
<note xmlns="https://www.w3schools.com"
```

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

```
xsi:schemaLocation="https://www.w3schools.com note.xsd">
```

- *xmlns* - Označava koji je podrazumevani prostor imena.
- *xmlns:xsi* - Omogućava definisanje šeme.
- *xsi:schemaLocation* - Sadrži dva tokena. Prvi token sadrži URI odredišnog prostora imena. Drugi je fizička lokacija.



Pridruživanje šeme XML dokumentu

```
<note
```

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

```
xsi:noNamespaceSchemaLocation="note.xsd">
```

- *xmlns:xsi* - Omogućava definisanje šeme.
- *xsi:noNamespaceSchemaLocation* - Sadrži adresu šeme koju treba upotrebiti za proveru validnosti elemenata koji ne pripadaju nijednom prostoru imena.



Validacija šeme

- Za proveru validnosti XML dokumenta u odnosu na šemu, potreban je program koji će obaviti validaciju na osnovu šeme.
- BaseX program ima sledeće funkcionalnosti:
 - `validate:xsd("shiporder.xml")` – validira šemu na osnovu definisane XML Schema-e unutar XML dokumenta
 - `validate:xsd("shiporder.xml","sema.xsd")` – validira šemu na osnovu prosledjene XML Schema-e



Deklaracije elemenata

- Najčešći oblik elementa u šemi:

```
<xs:element name="ime" type="xs:string">
```
- Sadržaj elementa može biti:
 1. Prost
 2. Složen



Prost tipovi

- Prost sadržaj se sastoji samo od teksta bez ugnježenih elemenata i ne mogu imati attribute. U tabeli su navedeni najčešće korišćeni prosti tipovi definisani u W3C specifikaciji:

1. integer

2. string

3. decimal

4. anyURI

5. boolean

6. dateTime

7. date

8. time

9. duration



Deklaracije atributa

- Atributi moraju biti deklarirani kao prosti tipovi. Elementi koji sadrže attribute su složenog tipa.
- Atributi se deklariraju pomoću elementa:

```
<xs:attribute name="ime" type="xs:tip">
```
- Atributi su podrazumevano opcioni. Ako je obavezan:

```
use="required"
```

Primer:

```
<xs:attribute name="id" type="xs:integer"/>
```

```
<proizvod id="135">stolica</proizvod>
```



Deklaracije atributa

Primer:

```
<xs:element maxOccurs="unbounded" name="event">  
  <xs:complexType>  
    <xs:simpleContent>  
      <xs:extension base="xs:string">  
        <xs:attribute default="list" name="node" type="xs:string"/>  
      </xs:extension>  
    </xs:simpleContent>  
  </xs:complexType>  
</xs:element>
```



Ograničenja

- Ograničenja kontrolišu moguće vrednosti prostih tipova podataka. Koristi se sintaksa:

`xs:restriction`

- Jedno ograničenje se iskazuje kao podelement unutar ograničenja. Više ograničenja se može kombinovati da bi se dodatno ograničile moguće vrednosti prostog tipa.



Ograničenja

- *length*, *minLength* i *maxLength* - određuju dužinu podataka
- *pattern* - zadaje ograničenja formata stringa koristeći regularni izraz za definisanje mogućih vrednosti.
- *enumeration* - ograničavaju moguće vrednosti na članove unapred definisane liste
- *whiteSpace* - određuje način na koji se obrađuju beline unutar podataka *maxInclusive* i *maxExclusive* određuju najveće vrednosti stavki *minInclusive* i *minExclusive* određuju najmanje vrednosti stavki
- *totalDigits* - ograničava ukupan broj cifara u decimalnom broju
- *fractionDigits* - zadaje obavezan broj cifara u broju desno od decimalne tačke



Ograničenja

Primer 1:

```
<xs:simpleType name="bso">  
  <xs:restriction base="xs:string">  
    <xs:pattern  
      value="\d\d\d-\d\d-\d\d\d\d"/>  
  </xs:restriction>  
</xs:simpleType>
```

Primer 2:

```
<xs:element name="auto">  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:enumeration value="Audi"/>  
      <xs:enumeration value="Golf"/>  
      <xs:enumeration value="BMW"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```



Složeni tipovi

- Elementi koji definišu složene tipove mogu imati attribute i mogu sadržati ugnježdene elemente. Samo elementi mogu biti složenog tipa. Tipovi atributa su uvek prosti.
- Deklaracija složenog tipa: `xs:complexType`



Složeni tipovi

Primer:

- XML kod

```
<zaposleni>
```

```
  <ime>Pera</ime>
```

```
  <prezime>Miric</prezime>
```

```
</zaposleni>
```

- XML Schema kod

```
<xs:element name="zaposleni">
```

```
  <xs:complexType>
```

```
    <xs:sequence>
```

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

```
      <xs:element name="prezime"
```

```
        type="xs:string"/>
```

```
    </xs:sequence>
```

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

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



Složen tipovi

- Složen element može sadržati i referencu na složeni tip (ili ga sadržati).
- Na ovaj način se može definisati više elemenata istog tipa.

Primer:

```
<xs:element name="zaposleni" type="osobainfo"/>
```

```
<xs:complexType name="osobainfo"/>
```

```
  <xs:sequence>
```

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

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

```
  </xs:sequence>
```

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



Složeni tipovi

- Složeni element može proširiti neki postojeći složeni element novim elementima.

Primer:

```
<xs:complexType name="dopunjeninfo"/>
  <xs:complexContent>
    <xs:extension base="osobainfo">
      <xs:sequence>
        <xs:element name="ulica" type="xs:string"/>
        <xs:element name="grad" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```



Ograničenja broja pojavljivanja

- Eksplicitnog zadavanja najmanjeg i najvećeg broja pojavljivanja nekog elementa na nekom mestu u dokumentu.
- Koriste se sledeći atributi elementa `xs:element`:
 1. `minOccurs` – definiše minimalan broj pojavljivanja elementa
 2. `maxOccurs` – definiše maksimalan broj pojavljivanja elementa

Primer:

```
<xs:element name="student">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="brIndeks" type="xs:string"/>
      <xs:element name="telefon" type="xs:string" minOccurs="1" maxOccurs="10"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```



Mešoviti sadržaj elemenata

- Mešoviti sadržaj imaju elementi koji sadrže stringove, tj. tekst ali i druge elemente.
- XML Schema omogućava definisanje ove funkcionalnosti kao i precizno definisanje broja i redosleda elemenata unutar znakovnih podataka.
- Atribut `mixed` elementa tipa *complexType* određuje da li se znakovni podaci smeju pojaviti unutar tela elementa.



Zadavanje redosleda elemenata

- Za definisanje redosleda elemenata, XML šema nudi:
 1. `xs:sequence` - definiše koje elemente sadrži složeni element i tačan redosled.
 2. `xs:choice` - definiše pojavljivanje samo jednog elementa iz niza mogućnosti.
 3. `xs:all` - definiše da se svaki od elemenata mora pojaviti jednom. Redosled pojavljivanja nije važan.
- Ove mogućnosti se mogu ugnježdavati.



Primer

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="item">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="title" type="xs:string"/>
        <xs:element name="note" type="xs:string" minOccurs="0"/>
        <xs:element name="quantity" type="xs:positiveInteger"/>
        <xs:element name="price" type="xs:decimal"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="shiporder">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="item" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="orderid" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```