

XML Schema

Iztok Savnik

XML-Schema, verzija 2

- Po izdaji XML 1.0 je DTD hitro postal premalo izrazen
- Delo na novem standardu se začne 1998
- Različne firme so predlagale različne verzije formata
 - XML Data (MS, Arbortext, Inso), January 1998
 - DCD (MS & IBM), June 1998
 - XDR (XML Data Reduced), July 1998
 - SOX (Schema for OO XML), July 1999

XML-Schema, verzija 2

- W3C delovna skupina za shemo je definirana 1999
- XML Schema postane uradno priporočilo v maju 2001, 2. verzija je izdana 2004
 - Primer
 - Structures
 - DataTypes

XML-Schema, verzija 2

- Validacija podatkov
 - Strukturo elementov in atributov
 - Urejenost elementov
 - Podatkovne vrednosti elementov in atributov
 - Unikatnost vrednosti
- Definira pogodbo s poslovnimi partnerji
- Dokumentacija
- Uporaba privzetih vrednosti
- Shramba aplikacijskih podatkov

Primer 1

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="product" type="ProductType"/>
  <xsd:complexType name="ProductType">
    <xsd:sequence>
      <xsd:element name="number" type="xsd:integer"/>
      <xsd:element name="size" type="SizeType"/>
    </xsd:sequence>
    <xsd:attribute name="effDate" type="xsd:date"/>
  </xsd:complexType>
  <xsd:simpleType name="SizeType">
    <xsd:restriction base="xsd:integer">
      <xsd:minInclusive value="2"/>
      <xsd:maxInclusive value="18"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>

<product effDate="2001-04-02">
  <number>557</number>
  <size>10</size>
</product>
```

Deklaracije in definicije

- Deklaracija se uporablja za napoved komponent v instanci (npr. Elementi in atributi)
- Definicija je za komponente, ki so interne v dani shemi (npr., podatkovni tipi in modeli skupin)
- Urejenost v dokumentu sheme ni pomembna

Globalne in lokalne komponente

- Globalne komponente
 - Pojavijo se na najvišjem nivoju shema (otroci `xsd:schema`)
 - Ime mora biti unikatno v shemi
- Lokalne komponente
 - Omejene na definicijo, ki vsebujejo lokalne komponente
 - Npr. elementi, ki so deklarirani v okviru kompleksnega tipa ali tipi deklarirani anonimno v drugih gradnikih

Atributi in elementi

- Osnovni gradniki XML dokumentov
- Vsak gradnik je povezan s podatkovnim tipom
- Uporabi različni imeni za podatke, ki so strukturno enaki in imajo isti tip vendar imajo različen pomen v različnem kontekstu
 - shipTo in billTo imata oba tip USAddress
- Uporabi ista imena in različne tipe za isti podatek v različnem kontekstu
 - size ima isto ime v primeru majice in klobuka, ima pa različen tip (celo število in naštevni tip 'S', 'L', 'X', 'XL')

Enostavni in kompleksni tipi

- Elementi enostavnih tipov imajo vsebino, ki so znaki nimajo pa atributov ali otrok

```
<size>10</size>  
<comment>Extra trim on sides</comment>  
<availableSizes>10 large 2</availableSizes>
```

- Elementi kompleksnih tipov imajo lahko otroke in attribute

```
<size system="US-DRESS">10</size>  
  <comment>Extra <b>trim</b> on sides</comment>  
  <availableSizes>  
    <size>10</size>  
    <size>2</size>  
  </availableSizes>
```

- Atributi so vedno enostavnega tipa

Imenovani in neimenovani tipi

- Imenovani tip je vedno definiran globalno
- Anonimen tip nima imena in je viden samo lokalno na mestu kjer je definiran

```
<xsd:element name="size">
  <xsd:simpleType>
    <xsd:restriction base="xsd:integer">
      <xsd:minInclusive value="2"/>
      <xsd:maxInclusive value="18"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

Definicija hierarhije tipov

- Podatkovni tipi so lahko izpeljani iz drugih tipov z omejitvijo in razširitvijo
 - Tip `SizeType` omeji zalogo vrednosti integer
- Kompleksni tip `UKAddressType` je razširitev `AddressType` z dodajanjem novih otrok
 - Princip zamenljivosti
 - Primerek `UKAddressType` je sprejemljiv povsod kjer se pričakuje primerek `AddressType`

Enostavni tipi

- Niz:
 - ENTITIES, ENTITY, ID, IDREF, IDREFS, language, Name, NCName, NMTOKEN, NMTOKENS, normalizedString, QName, string, token
- Datum:
 - date, dateTime, duration, gDay, gMonth, gMonthDay, gYear, gYearMonth, time
- Število:
 - base64Binary, byte, decimal, double, float, hexBinary, int, integer, long, negativeInteger, nonPositiveInteger, positiveInteger, short, unsignedLong, unsignedInt, unsignedShort, unsignedByteErr, unrelated anyURI, boolean, NOTATION, ...

Enostavni tipi (1)

- Predefinirani enostavni tipi
- Iz predefiniranih tipov lahko definiramo nove enostavne tipe z omejitvijo predefiniranega tipa
- Imamo tudi strukturirane enostavne tipe:
 - Sezname enostavnih vrednosti
 - Unije enostavnih vrednosti različnih tipov

Kompleksni tipi

- Vsebina elementa so lahko tekst in elementi
 - Štiri različni tipi vsebine:
 - Simple, Element, Mixed, Empty
- Model vsebine opiše urejenost in strukturo elementov kompleksnega tipa
 - Sekvenca določa urejenost
 - Izbira omogoča uporabo ene od večih opcij
 - Skupina all zahteva da se uporabijo vsi elementi v poljubnem vrstnem redu

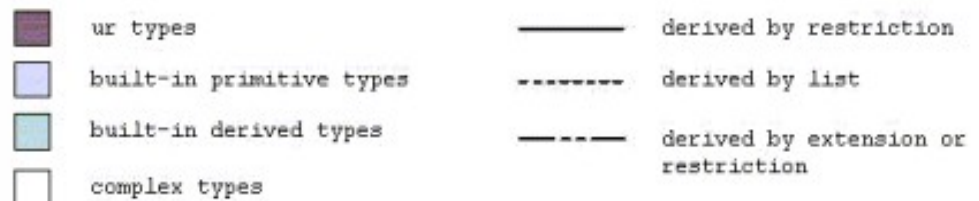
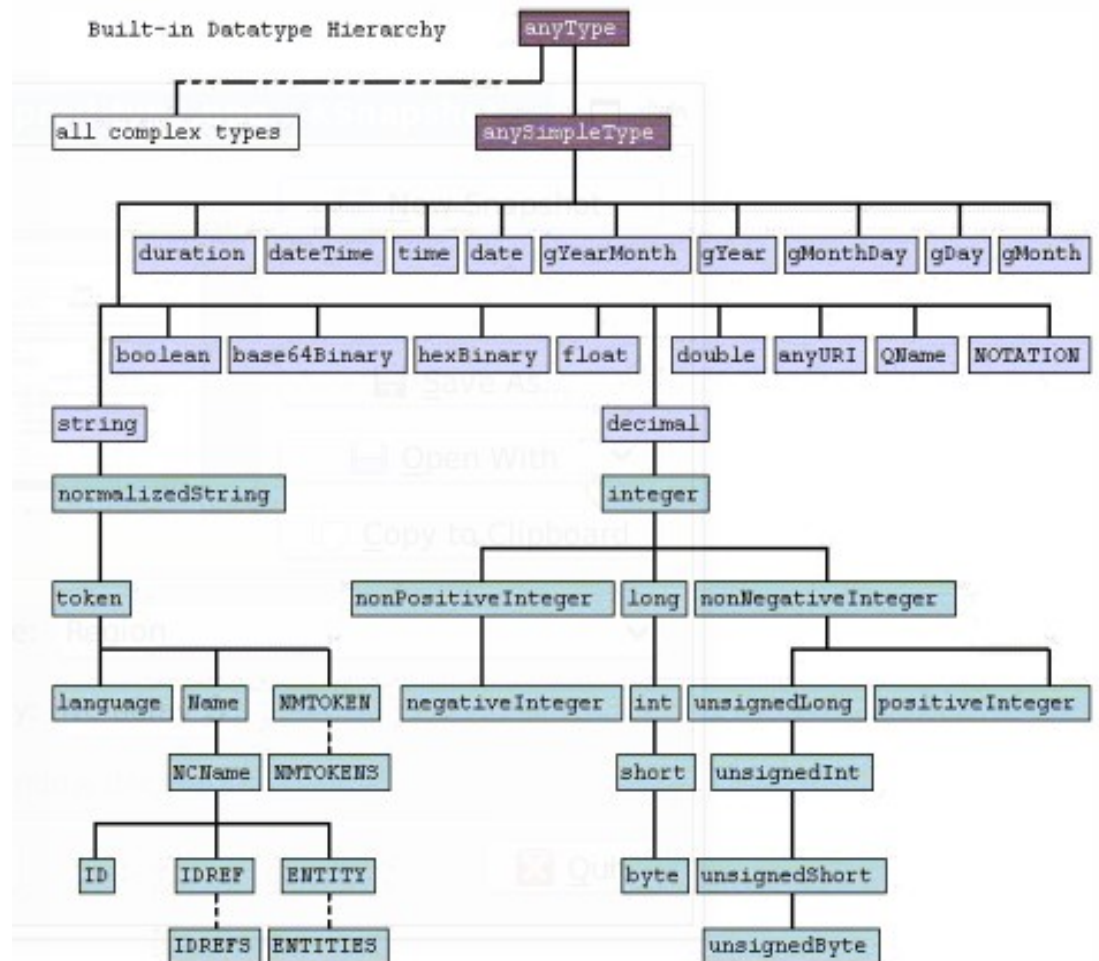
Kompleksni tipi: primer

```
<xsd:complexType name="ProductType">
  <xsd:sequence>
    <xsd:element name="number" type="xsd:integer"/>
    <xsd:choice minOccurs="0" maxOccurs="3">
      <xsd:element name="size" type="SizeType"/>
      <xsd:element name="color" type="ColorType"/>
    </xsd:choice>
    <xsd:any/>
  </xsd:sequence>
  <xsd:attribute name="effDate" type="xsd:date"/>
</xsd:complexType>
```

Pregled gradnikov XSD

- Enostavni tipi
 - Atributi enostavnih tipov
 - Omejitve enostavnih tipov
 - Strukture enostavnih tipov
- Kompleksni tipi
 - Omejitve kompleksnih tipov
 - Oblikovanje skupin atributov
 - Definicija skupin atributov
 - Razširitev kompleksnih tipov

Enostavni tipi



Elementi

```
<element
abstract = boolean : false
block = (#all | List of (extension|restriction|substitution))
default = string
final = (#all | List of (extension | restriction))
fixed = string
form = (qualified | unqualified)
id = ID
maxOccurs = (nonNegativeInteger | unbounded) : 1
minOccurs = nonNegativeInteger : 1
name = NCName
nillable = boolean : false
ref = QName
substitutionGroup = QName
type = QName
{any attributes with non-schema namespace
...}>
Content: (annotation?, ((simpleType|complexType)?,
      (unique|key|Keyref)*))
</element>
```

Atributi

```
<attribute
default = string
fixed = string
form = (qualified | unqualified)
id = ID
name = NCName
ref = QName
type = QName
use = (optional | prohibited | required) :
optional
{any attributes with non-schema namespace ...}>
Content: (annotation?, (simpleType?))
</attribute>
```

Elementi enostavnih tipov

- Osnovna forma

```
<xsd:element name="..." type="..." />
```

- Primeri instanc

```
<name>John Harvard</name>
```

```
<year>1636</year>
```

- Primeri deklaracij

```
<xsd:element name="lastname" type="xsd:string" />
```

```
<xsd:element name="age" type="xsd:integer" />
```

Atributi enostavnih tipov

- Osnovna forma

```
<xsd:attribute name="..." type="..." />
```

- Primerki

```
<student gender="male">John Harvard</student>
```

- Primeri deklaracij

```
<xsd:attribute name="gender" type="xsd:string" />
```

Atributi s privzeto ali fiksno vrednostjo

- Če ni vrednosti se lahko uporabi privzeta vrednost

```
<xsd:attribute name="country" type="xsd:string"
  default="US"/>
```

- Vrednost je lahko fiksna t.j. se mora pojaviti kot je deklarirano

```
<xsd:attribute name="country" type="xsd:string"
  fixed="US"/>
```

Opcijski in obvezen atribut

- Privzeto so atributi opsijski

```
<xsd:attribute name="country" type="xsd:string"  
  use="optional"/>
```

- Če so definirani kot obvezni se morajo pojaviti

```
<xsd:attribute name="country" type="xsd:string"  
  use="required"/>
```

Omejitve enostavnega tipa

Facet	Description
enumeration	Defines a list of acceptable values
fractionDigits	Specifies the maximum number of decimal places allowed. Must be equal to or greater than zero
length	Specifies the exact number of characters or list items allowed. Must be equal to or greater than zero
maxExclusive	Specifies the upper bounds for numeric values (the value must be less than this value)
maxInclusive	Specifies the upper bounds for numeric values (the value must be less than or equal to this value)
maxLength	Specifies the maximum number of characters or list items allowed. Must be equal to or greater than zero
minExclusive	Specifies the lower bounds for numeric values (the value must be greater than this value)
minInclusive	Specifies the lower bounds for numeric values (the value must be greater than or equal to this value)
minLength	Specifies the minimum number of characters or list items allowed. Must be equal to or greater than zero
pattern	Defines the exact sequence of characters that are acceptable
totalDigits	Specifies the exact number of digits allowed. Must be greater than zero
whiteSpace	Specifies how white space (line feeds, tabs, spaces, and carriage returns) are handled

Omejitve enostavnega tipa (1)

- Omejitev z vrednostjo

```
<xsd:element name="year">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:integer">  
      <xsd:minInclusive value="2007"/>  
      <xsd:maxInclusive value="2010"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

```
<xsd:element name="year">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:integer">  
      <xsd:minExclusive value="2006"/>  
      <xsd:maxExclusive value="2011"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

Omejitve enostavnega tipa (2)

- Omejitev z vrednostjo

```
<xsd:element name="major">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="English"/>
      <xsd:enumeration value="Math"/>
      <xsd:enumeration value="Physics"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

Omejitve enostavnega tipa (3)

- Omejitev z vrednostjo

```
<xsd:element name="major" type="majors"/>

<xsd:simpleType name="majors">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="English"/>
    <xsd:enumeration value="Math"/>
    <xsd:enumeration value="Physics"/>
  </xsd:restriction>
</xsd:simpleType>
```

Omejitve enostavnega tipa (4)

- Omejitev z vzorcem

```
<xsd:element name="choice">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[abcd]"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

```
<xsd:element name="initials">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[A-Z][A-Z][A-Z]?"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

Omejitve enostavnega tipa (5)

- Omejitev z vzorcem

```
<xsd:element name="gender">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:pattern value="male|female"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

```
<xsd:element name="password">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:pattern value="[a-zA-Z0-9]{8}"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

Omejitve enostavnega tipa (6)

- Omejitev z vzorcem

```
<xsd:simpleType name="ProdNumType">  
  <xsd:restriction base="xsd:string">  
    <xsd:pattern value="\d{3}-[A-Z]{2} | \d{7}"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Omejitve enostavnega tipa (7)

- Omejitev belih znakov

```
<xsd:element name="name">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:whiteSpace value="preserve"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

```
<xsd:element name="name">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:whiteSpace value="replace"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

```
<xsd:element name="name">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:whiteSpace value="collapse"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

Omejitve enostavnega tipa (8)

- Omejitev z dolžino

```
<xsd:element name="password">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:length value="8"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

```
<xsd:element name="password">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="5"/>
      <xsd:maxLength value="8"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```


Enostavni tipi - strukture

- Sezname

```
<grades>90 85 77 100 99 45</grades>
```

```
<xsd:element name="grades">  
  <xsd:simpleType>  
    <xsd:list itemType="xsd:nonNegativeInteger"/>  
  </xsd:simpleType>  
</xsd:element>
```

Enostavni tipi – strukture (1)

- Unije

```
<xsd:element name="jeans_size">  
  <xsd:simpleType>  
    <xsd:union memberTypes="sizebyno sizebystring"/>  
  </xsd:simpleType>  
</xsd:element>
```

```
<xsd:simpleType name="sizebyno">  
  <xsd:restriction base="xsd:positiveInteger">  
    <xsd:maxInclusive="42"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

```
<xsd:simpleType name="sizebystring">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="small"/>  
    <xsd:enumeration value="medium"/>  
    <xsd:enumeration value="large"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Kompleksni tipi

- Omejitev vsebine
 - Enostavni: vsebina je samo tekst
 - `<name>Jerry Seinfeld</name>`
 - Samo-elementi: vsebina so samo elementi
 - `<name><first>Jerry</first><last>Seinfeld</last></name>`
 - Mešani: vsebina je tekst in elementi
 - `<p><name>Jerry Seinfeld</name> is a
<title>comedian</title>.</p>`
 - Prazni: ni vsebine
 - `<comedian name="Jerry Seinfeld"/>`

Kompleksni tipi (1)

- Enostavna vsebina

```
<xsd:element name="...">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:restriction base="...">
        ....
      </xsd:restriction>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="...">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="...">
        ....
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

Kompleksni tipi (2)

- Vsebina = samo elementi

```
<student>  
  <name>John Harvard</name>  
  <year>1636</year>  
</student>
```

```
<xsd:element name="student">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element name="name" type="xsd:string"/>  
      <xsd:element name="year" type="xsd:integer"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

Kompleksni tipi (3)

- Vsebina = samo elementi

```
<name>  
  <first>John</first>  
  <last>Harvard</last>  
</name>
```

```
<xsd:element name="name">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element name="first" type="xsd:string"/>  
      <xsd:element name="last" type="xsd:string"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

Kompleksni tipi (4)

- Mešana vsebina

```
<letter>
Dear Mr.<name>John Smith</name>.
Your order <orderid>1032</orderid>
will be shipped on <shipdate>2001-07-13</shipdate>.
</letter>
```

```
<xsd:element name="letter">
  <xsd:complexType mixed="true">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"/>
      <xsd:element name="orderid" type="xsd:positiveInteger"/>
      <xsd:element name="shipdate" type="xsd:date"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Kompleksni tipi (5)

- Prazna vsebina

```
<foo bar="baz" />
```

```
<xsd:element name="foo">  
  <xsd:complexType>  
    <xsd:attribute name="bar" type="xsd:string"/>  
  </xsd:complexType>  
</xsd:element>
```


Kompleksni tipi (5)

- Oblikovanje skupin
 - `sequence` : skupina določa elemente, ki se morajo uporabiti po vrstnem redu pojavitve
 - `choice` : skupina določa elemente od katerih se mora uporabiti samo eden
 - `all` : skupina določa elemente, ki se morajo pojaviti v kakršnemkoli vrstem redu vendar ne več kot enkrat

Kompleksni tipi (6)

- Sekvenca

```
<xsd:element name="name">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="first" type="xsd:string"/>
      <xsd:element name="last" type="xsd:string"/>
      <xsd:element name="nick" type="xsd:string"
        maxOccurs="unbounded" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Kompleksni tipi (7)

- Izbira

```
<xsd:complexType name="ProductType">  
  <xsd:choice minOccurs="0" maxOccurs="3">  
    <xsd:element name="size" type="SizeType"/>  
    <xsd:element name="color" type="ColorType"/>  
  </xsd:choice>  
  <xsd:attribute name="effDate" type="xsd:date"/>  
</xsd:complexType>
```

Kompleksni tipi (8)

- Vgnezdene skupine

```
<xsd:complexType name="ProductType">
  <xsd:sequence>
    <xsd:element name="number" type="xsd:integer"/>
    <xsd:choice minOccurs="0" maxOccurs="3">
      <xsd:element name="size" type="SizeType"/>
      <xsd:element name="color" type="ColorType"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="effDate" type="xsd:date"/>
</xsd:complexType>
```

Kompleksni tipi (9)

- Celotna skupina

```
<xsd:element name="name">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="first" type="xsd:string"/>
      <xsd:element name="last" type="xsd:string"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Kompleksni tipi (10)

- Definicija skupine

```
<xsd:group name="persongroup">
  <xsd:sequence>
    <xsd:element name="firstname" type="xsd:string"/>
    <xsd:element name="lastname" type="xsd:string"/>
    <xsd:element name="birthday" type="xsd:date"/>
  </xsd:sequence>
</xsd:group>
```

```
<xsd:element name="person" type="personinfo"/>
<xsd:complexType name="personinfo">
  <xsd:sequence>
    <xsd:group ref="persongroup"/>
    <xsd:element name="country" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
```

Kompleksni tipi (11)

- Definicija skupine atributov

```
<xsd:attributeGroup name="personattrgroup">  
  <xsd:attribute name="firstname" type="xsd:string"/>  
  <xsd:attribute name="lastname" type="xsd:string"/>  
  <xsd:attribute name="birthday" type="xsd:date"/>  
</xsd:attributeGroup>
```

```
<xsd:element name="person">  
  <xsd:complexType>  
    <xsd:attributeGroup ref="personattrgroup"/>  
  </xsd:complexType>  
</xsd:element>
```

Kompleksni tipi (12)

- Razširitev enostavnega tipa

```
<xsd:simpleType name="size">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="small" />  
    <xsd:enumeration value="medium" />  
    <xsd:enumeration value="large" />  
  </xsd:restriction>  
</xsd:simpleType>
```

```
<xsd:complexType name="jeans">  
  <xsd:simpleContent>  
    <xsd:extension base="size">  
      <xsd:attribute name="sex">  
        <xsd:simpleType>  
          <xsd:restriction base="xsd:string">  
            <xsd:enumeration value="male" />  
            <xsd:enumeration value="female" />  
          </xsd:restriction>  
        </xsd:simpleType>  
      </xsd:attribute>  
    </xsd:extension>  
  </xsd:simpleContent>  
</xsd:complexType>
```


Kompleksni tipi (13)

- Razširitev kompleksnega tipa

```
<xsd:complexType name="ProductType">
  <xsd:sequence>
    <xsd:element name="number" type="ProdNumType"/>
    <xsd:element name="name" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ShirtType">
  <xsd:complexContent>
    <xsd:extension base="ProductType">
      <xsd:choice maxOccurs="unbounded">
        <xsd:element name="size" type="SizeType"/>
        <xsd:element name="color" type="ColorType"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Kompleksni tipi (14)

- Poljuben element dovolimo z any

```
<xsd:element name="name">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="first" type="xsd:string"/>
      <xsd:element name="last" type="xsd:string"/>
      <xsd:any minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Primer: naročilo (1)

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      Purchase order schema.
      Copyright 2007 Harvard University. All rights reserved.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="purchaseOrder" type="PurchaseOrderType"/>

  <xsd:element name="comment" type="xsd:string"/>

  <xsd:complexType name="PurchaseOrderType">
    <xsd:sequence>
      <xsd:element name="shipTo" type="USAddress"/>
      <xsd:element name="billTo" type="USAddress"/>
      <xsd:element ref="comment" minOccurs="0"/>
      <xsd:element name="items" type="Items"/>
    </xsd:sequence>
    <xsd:attribute name="orderDate" type="xsd:date"/>
  </xsd:complexType>

  <xsd:complexType name="USAddress">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"/>
      <xsd:element name="street" type="xsd:string"/>
      <xsd:element name="city" type="xsd:string"/>
      <xsd:element name="state" type="xsd:string"/>
      <xsd:element name="zip" type="xsd:decimal"/>
    </xsd:sequence>
    <xsd:attribute name="country" type="xsd:NMTOKEN" fixed="US"/>
  </xsd:complexType>
```

Primer: naročilo (2)

...

```
<xsd:complexType name="Items">
  <xsd:sequence>
    <xsd:element name="item" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="productName" type="xsd:string"/>
          <xsd:element name="quantity">
            <xsd:simpleType>
              <xsd:restriction base="xsd:positiveInteger">
                <xsd:maxExclusive value="100"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="USPrice" type="xsd:decimal"/>
          <xsd:element ref="comment" minOccurs="0"/>
          <xsd:element name="shipDate" type="xsd:date" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="partNum" type="SKU" use="required"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<!-- Stock Keeping Unit, a code for identifying products -->
<xsd:simpleType name="SKU">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d{3}-[A-Z]{2}"/>
  </xsd:restriction>
</xsd:simpleType>

</xsd:schema>
```

Primer: naročilo (3)

```
<?xml version="1.0"?>
<purchaseOrder orderDate="1999-10-20"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="po.xsd">
  <shipTo country="US">
    <name>Alice Smith</name>
    <street>123 Maple Street</street>
    <city>Mill Valley</city>
    <state>CA</state>
    <zip>90952</zip>
  </shipTo>
  <billTo country="US">
    <name>Robert Smith</name>
    <street>8 Oak Avenue</street>
    <city>Old Town</city>
    <state>PA</state>
    <zip>95819</zip>
  </billTo>
  <comment>Hurry, my lawn is going wild!</comment>
  <items>
    <item partNum="872-AA">
      <productName>Lawnmower</productName>
      <quantity>1</quantity>
      <USPrice>148.95</USPrice>
      <comment>Confirm this is electric</comment>
    </item>
    <item partNum="926-AA">
      <productName>Baby Monitor</productName>
      <quantity>1</quantity>
      <USPrice>39.98</USPrice>
      <shipDate>1999-05-21</shipDate>
    </item>
  </items>
</purchaseOrder>
```

Povzetek elementov

Element	Explanation
all	Specifies that the child elements can appear in any order. Each child element can occur 0 or 1 time
annotation	Specifies the top-level element for schema comments
any	Enables the author to extend the XML document with elements not specified by the schema
anyAttribute	Enables the author to extend the XML document with attributes not specified by the schema
applInfo	Specifies information to be used by the application (must go inside annotation)
attribute	Defines an attribute
attributeGroup	Defines an attribute group to be used in complex type definitions
choice	Allows only one of the elements contained in the <choice> declaration to be present within the containing element
complexContent	Defines extensions or restrictions on a complex type that contains mixed content or elements only
complexType	Defines a complex type element
documentation	Defines text comments in a schema (must go inside annotation)
element	Defines an element
extension	Extends an existing simpleType or complexType element
field	Specifies an XPath expression that specifies the value used to define an identity constraint

Povzetek elementov (1)

group	Defines a group of elements to be used in complex type definitions
import	Adds multiple schemas with different target namespace to a document
include	Adds multiple schemas with the same target namespace to a document
key	Specifies an attribute or element value as a key (unique, non-nullable, and always present) within the containing element in an instance document
keyref	Specifies that an attribute or element value correspond to those of the specified key or unique element
list	Defines a simple type element as a list of values
notation	Describes the format of non-XML data within an XML document
redefine	Redefines simple and complex types, groups, and attribute groups from an external schema
restriction	Defines restrictions on a simpleType, simpleContent, or a complexContent
schema	Defines the root element of a schema
selector	Specifies an XPath expression that selects a set of elements for an identity constraint
sequence	Specifies that the child elements must appear in a sequence. Each child element can occur from 0 to any number of times
simpleContent	Contains extensions or restrictions on a text-only complex type or on a simple type as content and contains no elements
simpleType	Defines a simple type and specifies the constraints and information about the values of attributes or text-only elements
union	Defines a simple type as a collection (union) of values from specified simple data types
unique	Defines that an element or an attribute value must be unique within the scope

Viri

- <http://www.w3.org/XML/Schema>
- David J. Malan, XML with Java, Java Servlet, and JSP, Course CS E-259, Harvard University, 2007
- <http://www.w3schools.com/schema/default.asp>