



SDXML VT2024

Models and languages for semi-structured data and XML

DTD & XML Schema

DTD and XML schema

nikos dimitrakas
nikos@dsv.su.se
08-161295

Corresponding reading

Excerpt from Data on the Web

Chapter 1, 4, 5, 6, 10 (especially 10.6) of the course book

Parts of chapter 30 of Database Systems (Connolly, Begg) 6th edition (chapter 31 in 5th edition)



DTD (Document Type Definition)

- **Defines elements and their structure**

- **subelements**
- **content**
- **attributes**

- **Own syntax**

Defines elements and their structure

- subelements
- content
- attributes

Own syntax

DTD - elements

- **Empty element**
<!ELEMENT Person EMPTY> Element with any type of content
<!ELEMENT Person ANY>
- **Subelements**
<!ELEMENT Person (Name, Address)>
- **Optional subelements (zero or one occurrences)**
<!ELEMENT Person (Name, Address?)>
- **Subelements with zero or more occurrences**
<!ELEMENT Person (Name, Address, Phone*)>
- **Subelements with one or more occurrences**
<!ELEMENT Person (Name, Address, Phone+)>
- **Elements with text content**
<!ELEMENT Person (#PCDATA)> <!ELEMENT Person (#PCDATA)>
- **Element with any type of content**
<!ELEMENT Person ANY>

DTD - elements

- **Alternative subelements**
<!ELEMENT Person (Name, (Employee|Student))>

Name, Address, Phone, Employee, Student must of course be defines as well.

DTD - attributes

Attribute with specific possible values - `<!ATTLIST Person gender ("m", "f") #REQUIRED>`

- Attributes are defined per element type
- One or more attributes can be defined in the same ATTLIST
- Compulsory attribute
`<!ATTLIST Person pnr CDATA #REQUIRED>`
- Optional attributes
`<!ATTLIST Person length CDATA #IMPLIED>`
- Optional attribute with default value
`<!ATTLIST Person birthplace CDATA "Stockholm">`
- Attribute with fixed value (constant)
`<!ATTLIST Person employer CDATA #FIXED "IBM">`
- Attribute with specific possible values
`<!ATTLIST Person gender ("m", "f") #REQUIRED>`

Compulsary attribute `<!ATTLIST Person pnr CDATA #REQUIRED>`

DTD - attributes

- More attributes in the same ATTLIST
`<!ATTLIST Person
pnr CDATA #REQUIRED Compulsary attribute
birthplace CDATA "Stockholm"
gender ("m", "f") #REQUIRED
length CDATA #IMPLIED>`
- Attributes with unique values
`<!ATTLIST Person pnr ID #REQUIRED>`
- Attributes that refer to unique ID attributes
`<!ATTLIST Person boss IDREF #IMPLIED>`

DTD - link to the XML document

7

- **DOCTYPE**

- `<!DOCTYPE Courses SYSTEM "courses.dtd">`

- **Inline**

- `<!DOCTYPE Courses [
declarations ELEMENT, ATTLIST, etc.`

Link in the DOCTYPE - `<!DOCTYPE Courses SYSTEM "courses.dtd">`

XML Schema

8

- **Defines elements and their structure**

- subelements
 - content
 - attributes
 - data types
 - number of occurrences
 - complex sequence rules

XML schema
defines elements and their structure
subelements
content
attributes
data types
number of occurrences
complex sequence rules

- **Uses XML syntax**

- XML Schema is an XML-based language

XML Schema document

- **Namespace**

- `http://www.w3.org/2001/XMLSchema`
- recommended alias: `xs`

- **Root element**

- `schema`
- attribute `targetNamespace`

Root element - schema,
attribute `targetNamespace`

- **Defines**

- `elements`
- `attributes`
- `types`

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

XML Schema document

- **The root element schema**

- `elementFormDefault`
 - » `qualified`
 - » `unqualified (default)`
- `attributeFormDefault`
 - » `qualified`
 - » `unqualified (default)`
- Are used to decide whether namespaces must be specified or not

```
<sdxml:Course xmlns:sdxml="http://ns.dsv.su.se/SDXML">
    <Teacher name="nikos"/>
</sdxml:Course>

<sdxml:Course xmlns:sdxml="http://ns.dsv.su.se/SDXML">
    <sdxml:Teacher sdxml:name="nikos"/>
</sdxml:Course>
```

Only relevant when planning to combine multiple namespaces and local elements/attributes

XML Schema - element

- The element element defines elements
 - the attribute name defines the element's name
 - the attribute type or the content defines the element's type
- Elements may be of the following types
 - A base type (string, integer, date, etc.)
 - » `<element name="Firstname" type="string" />`
 - A type defined somewhere else
 - » `<element name="Firstname" type="MyType" />`
 - A type defined in the content
 - » `<element name="Firstname">`
type definition
`</element>`

XML Schema - attribute

- The element attribute defines attributes
 - The attribute name defines the attribute's name
 - The attribute type or the content defines the attribute's type
 - The attribute use defines whether the attribute is optional (default) or required
- Attributes may be of the following types
 - A base type (string, integer, date, etc.)
 - » `<attribute name="Lastname" type="string" />`
 - A type defined somewhere else
 - » `<attribute name="Lastname" type="MyType" />`
 - A type defined in the content
 - » `<attribute name="Lastname">`
 `<simpleType ... />`
 `</attribute>`

XML Schema - types

- **Base types**

- string, integer, date, etc.

- **Custom types**

- **complexType**

- » when there are subelements or attributes

- **simpleType**

- » constraining a base type

XML Schema - simpleType

```
<xs:simpleType name="weekday">  
  <xs:restriction base="xs:string">  
    <xs:enumeration value="Monday"/>  
    <xs:enumeration value="Tuesday"/>  
    <xs:enumeration value="Wednesday"/>  
    <xs:enumeration value="Thursday"/>  
    <xs:enumeration value="Friday"/>  
    <xs:enumeration value="Saturday"/>  
    <xs:enumeration value="Sunday"/>  
  </xs:restriction>  
</xs:simpleType>
```

```
<xs:simpleType name="grade">  
  <xs:restriction base="xs:integer">  
    <xs:minInclusive value="0"/>  
    <xs:maxInclusive value="10"/>  
  </xs:restriction>  
</xs:simpleType>
```

XML Schema - complexType

- May contain one of the following
 - simpleContent
 - » if the content of the element shall be of a base type (a text node)
 - complexContent
 - » if the content of the element is based on another complexType
 - all
 - » defines subelements in unspecified order
 - choice
 - » defines alternative subelements
 - sequence
 - » defines a sequence of subelements
 - group
 - » uses a predefined structure of subelements
- May also contain (when not Content)
 - zero or more elements "attribute" or "attributeGroup"

all - defines subelements in unspecified order

choice - defines alternative subelements

XML Schema - complexType

```
<xs:complexType name="PersonType">  
  </xs:sequence>  
  <xs:element name="Firstname" type="xs:string"/>  
  <xs:element name="Lastname" type="xs:string"/>  
  <xs:element name="Birthdate" type="xs:date"/>  
</xs:sequence>  
</xs:complexType>
```

should be xs:sequence

```
<xs:complexType name="PersonType">  
  <xs:attribute name="Firstname" type="xs:string"/>  
  <xs:attribute name="Lastname" type="xs:string"/>  
  <xs:attribute name="Birthdate" type="xs:date"/>  
</xs:complexType>
```


XML Schema - occurrences

- The attribute **minOccurs** specifies the minimum number of occurrences
 - default 1
- The attribute **maxOccurs** specifies the maximum number of occurrences
 - default 1
 - for no limit use "unbounded"
- Used for
 - choice
 - all (0..1 till 1..1)
 - sequence
 - group
 - element
 - any (arbitrary element)

XML Schema - examples

```
<xs:group name="InfoGrupp">
  </xs:choice>
  <xs:element name="Email" type="xs:string"/>
  <xs:element name="Phone" type="xs:string"/>
  <xs:element name="Website" type="xs:string"/>
</xs:choice>
</xs:group>
```

Can repeat any number of times

```
<xs:complexType name="PersonType">
  <xs:sequence>
    <xs:group ref="InfoGrupp" minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="Firstname" type="xs:string"/>
    <xs:element name="Lastname" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="Birthdate" type="xs:date" use="optional"/>
</xs:complexType>
```

```
<xs:complexType name="EmployeeType">
  <xs:complexContent>
    <xs:extension base="PersonType">
      <xs:attribute name="EmploymentNumber" type="xs:string"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```


XML Schema - various

- **list**
 - defines that a simpleType is a list
- **union**
 - defines a new type that is the union of other types (simpleTypes)
- **annotation**
 - for specifying comments (with documentation or appinfo)
- **key, unique, field, keyref, selector**
 - defines "identifiers"/"candidate keys" and references
- **data type constraints with**
 - length, minLength, maxLength
 - minExclusive, minInclusive, maxExclusive, maxInclusive
 - totalDigits, fractionDigits
 - pattern, whiteSpace

XSD - Link to the XML document

- **With namespace**
 - An attribute (from the XSI namespace) in the root element and a namespace definition (default or with prefix)
 - xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 - xsi:schemaLocation="http://ns.dsv.su.se/SDXML/courses courses.xsd"
 - xmlns="http://ns.dsv.su.se/SDXML/courses"
 - **eller**
xmlns:c="http://ns.dsv.su.se/SDXML/courses"
- **Without namespace**
 - An attribute (from the XSI namespace) in the root element:
 - xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 - xsi:noNamespaceSchemaLocation="courses.xsd"

White space



What to do next

- **Quiz about DTD & XML Schema (Quiz 2)**