

PB138/07 - Modern Markup Languages and Their Applications

Lab 02 [2.3.2020]
XML Schema (XSD)

Bruno Rossi

*Department of Computer Systems and Communications,
Lasaris (Lab of Software Architectures and Information Systems)
Masaryk University, Brno*



lasaris

Quick Summary (1/2)

- XSD (XML Schema Definition): how to **formally** describe the elements in an XML document. It is used to validate that a specific XML conforms to the XSD definition
- XSD: just another XML document

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="classroom" type="classroomtype" />
  <xs:complexType name="classroomtype">
    <xs:sequence>
      <xs:element name="teacher" type="xs:string" />
      <xs:element name="members" type="memberstype" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  ....
</xs:schema>
```

Quick Summary (2/2)

- Different ways to create an XSD
- See “Divide the Schema” and “Using Named Types”
https://www.w3schools.com/xml/schema_example.asp
https://www.w3schools.com/xml/schema_example.asp

1. Defining Data Types

`<xs:simpleType name="yyy">` → text, numbers, restrictions

or

`<xs:complexType name="yyy">` → other elements, attributes, text

2. Assigning Data types to elements

`<xs:element name="xxx" type="yyy">` → type is a reference to the name used to define the type

XML Schema (XSD)

- We will use the same `continent.xml` example we used before
- You can download the file named `02-xsd-ex.zip`
- In these tasks, you will have to modify the file `continent.xsd`

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

- Run `xmllint` to check what's the results from schema validation using file `continent.xml`

```
$> xmllint --noout --schema continent.xsd continent.xml
```

We have an empty XSD, will the validation pass?

Exercise (1/6)

- As you see, validation is not passing
- Try to make the smallest change to the XSD that will make validation pass
- You can use
`<xs:element name="xxx" type="yyy">`

Do you need to declare the type here?
What is the type in this case?

Common types

`xs:string`

`xs:decimal`

`xs:integer`

`xs:boolean`

`xs:date`

`xs:time`

Exercise (2/6)

- Update the XSD to represent the structure of the XML file

Remember:

- attributes can only be defined for complex types
`<xs:attribute name="xxx" type="yyy"/>`
- Manage elements occurrence with `minOccurs="0..n"`
`maxOccurs="0..n"` (`maxOccurs="unbounded"`)
- Difference between `<xs:sequence>`, `<xs:choice>`, `<xs:all>`
- Look at
<https://www.w3.org/TR/xmlschema-0/> (2.1 the Purchase Order Schema)

Exercise (3/6)

- Try to place a restriction on the maximum population of a city (min:0, max:14200000)
- Consider using

```
<xs:restriction base="xs:integer">
<xs:minInclusive value="xxx"/>
<xs:maxInclusive value="yyy"/>
```

Exercise (4/6)

- Try to place a restriction on the type of pollution ('low','medium','high')
- Consider using

```
<xs:restriction base="xs:string">  
<xs:enumeration value="xxx"/>
```


Exercise (5/6)

- Try to place a restriction on initial letter of the city name: it must be starting with capital letter (note that this should also allow cities such as 'Los Angeles')
- Consider using

```
<xs:restriction base="xs:string">  
<xs:whiteSpace value="preserve"/>  
<xs:pattern value="REGEXXX"/>
```

Exercise (6/6)

- Modify continent.xml to give both Tokio and Mumbai the same id:

```
...
<city id="1">
  <name>Tokio</name>
  ...
</city>
  <city id="1">
    <name>Mumbai</name>
    ...
  </city>
```

- Check that the file still passes validation
- Define either a **key** or **unique** constraint for the city id, the structure is:

```
<xs:key name="<constraint name>" >
  <xs:selector xpath="<xpath selector, e.g., element>" />
  <xs:field xpath="<xpath field, e.g. @attr>" />
</xs:key>
```

- Validation has not to pass. You should be getting something like Element 'city': Duplicate key-sequence ['1'] in key identity-constraint 'id'.

For next time (optional)

Checkout the code at <https://gitlab.fi.muni.cz/pb138/seminars/seminar-02>

- 1) Use either the Java or Javascript implementation to validate the schema created today – it should pass as with xmllint :)
- 2) in the *src/* folder, complete *schema.xsd* for *data.xml* in similar way we did today (you can also see a reminder of the syntax in *schema-ukazka.xsd*)

References

Suggested material:

- W3C XML Schema Definition Language (XSD) W3C Recommendation:
→ <https://www.w3.org/TR/xmlschema-0/>
- XML Schema Tutorial on w3schools.com:
→ https://www.w3schools.com/xml/schema_intro.asp