XMLSchema, RelaxNG

T. Pitner, L. Bártek, A. Rambousek. L Grolig
FI MU Brno 2020

XMLSchema - Motivation

Stronger tool for XML data model specification **than DTD**, it allows:

- Separate **type** (e.g. *element type*) from its **occurrence** (i.e. element with particular name)
- More **primitive data types**.
- Allows to use **namespaces**.
- Allows to specify content model (elements) more **accurate way**.
- Allows new **type inheritance**.
- Allows modular schema design and schema **reuse**.
- XML Schema has an **XML syntax**.

XMLSchema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <!-- here the schema definition goes -->
</xs:schema>
```

XMLSchema - Element type

```
<xs:element name="element_name">
 <!-- here comes the type definition --
 placed either right here (so called "local")
 or as a referenced one (so called "global") -->
</xs:element>
```

XMLSchema - Element type (short)

```
<xs:element name="element_name" type="element_type"/>
```

XMLSchema - Simple type

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

e.g.

```
<xs:simpleType name="MyString32">
   <xs:restriction base="xs:string"> <xs:maxLength value="32"/>
</xs:restriction> </xs:simpleType>
```

XMLSchema - Union

```
<xs:simpleType name="isbnType">
  <xs:union>
    <xs:simpleType>
     <xs:restriction base="xs:st
      <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>
```

Union of two simple types - **string restricted** by regular expression and **enumeration**

XMLSchema - inheritance

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

isbnTypes - basic type,
isbnTypes10 - **derived** type with restriction

XMLSchema - Complex type

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

complex type - applies to elements typically defines **sequence** of child elements

XMLSchema - Group of elements

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

group type - reusable **sequence** of child elements

```
<xs:choice> and <xs:all>
```

choice or **all** type - may be used instead of **sequence**

XMLSchema - Attribute group

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

attribute group - reusable **set** of attributes

XMLSchema - Namespaces

```
<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>
```

XMLSchema - reference

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

or

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

RelaxNG - Alternative to Schema

```
<addressBook>
   <card>
      <name>John Smith</name>
      <email>js@example.com</email>
   </card>
   <card>
      <name>Fred Bloggs</name>
      <email>fb@example.net</email>
   </card>
</addressBook>
```

RelaxNG vs DTD

```
<!DOCTYPE addressBook [
<!ELEMENT addressBook (card*)>
<!ELEMENT card (name, email)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT email (#PCDATA)>
]>
```

Do you like this or that?

```
element addressBook {
   element card {
      element name { text },
      element email { text }
   }*
}
```

RelaxNG - Cardinality indication

```
element addressBook {
   element card {
      element name { text },
      element email { text }
   }*
}
```

could be **+** or **?** or ***** or nothing

RelaxNG - Options

```
element addressBook {
   element card {
      # here is the option
      (element name { text }
         | (element givenName { text },
            element familyName { text })),
      element email { text },
      element note { text }?
   }*
}
```

RelaxNG - Options with attributes

```
element addressBook {
   element card {
      (element name { text }
         | attribute name { text }),
      (element email { text }
         | attribute email { text })
   }*
}
```

RelaxNG - Grammar

```
grammar {
   start = AddressBook
   AddressBook = element addressBook { Card* }
   Card = element card { Name, Email }
   Name = element name { text }
   Email = element email { text }
}
```

RelaxNG - Enumerations

```
element card {
   attribute name { text },
   attribute email { text },
   attribute preferredFormat { "html" | "text" }
}

element card {
   element name { text },
   element email { text },
   element preferredFormat { "html" | "text" }
}
```