XML Schema

XML Schema ? What is it ?

XML Schema is a formal notation for defining a schema for a class of XML documents

XML Schema is a vocabulary for expressing data dependencies

XML Schema document models are designed to define the usage and relationships of various schema components, such as the following:

- Datatypes
- Elements and their content
- Attributes and their values
- Reusable components and their content
- Notations

Example

Is the following data valid ?

```
<location>
          <latitude>32.904237</latitude>
          <longitude>73.620290</longitude>
          <uncertainty units="meters">2</uncertainty>
</location>
```

Constraints:

(1) the location must be comprised of a latitude, followed by a longitude, followed by an indication of the uncertainty of the lat/lon measurements,
(2) the latitude must be a decimal with a value between -90 to +90,
(3) the longitude must be a decimal with a value between -180 to +180,
(4) for both latitude and longitude the number of digits to the right of the decimal point must be exactly six digits,
(5) the value of uncertainty must be a non-negative integer,
(6) the uncertainty units must be either meters or feet.

XML Schema: an overview

Properties:

- Enhanced data types (44+)
- Type definition mechanisms
- The same syntax as XML documents
- Object-orientation (inheritance)
- Create type constructor
- Ability to specify uniqueness (keys or content) constraints
- Ability to define the multiple elements with the same name but different content
- Ability to define the elements with nil contents
- Ability to define substitutable elements, e.g. the "Book" element is substitutable for the "Publication" element.

DTD versus XML Schema

```
<?xml version="1.0"?>
<!DOCTYPE BookStore [
<!ELEMENT BookStore (Book+)>
<!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Author (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ELEMENT Publisher (#PCDATA)> ]>
<BookStore>
  <Book>
    <Title>Java my way !</Title>
    <Author>Janusz R. Getta</Author>
    <Date>05-APR-2005</Date>
    <ISBN>09-345673</ISBN>
    <Publisher>Addison Wesley Ltd. Pty.</Publisher>
  </Book>
</BookStore>
```


DTD versus XML Schema

```
<?xml version="1.0"?>
<BookStore
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="bookstore.xsd">
  <Book>
    <Title>Java my way !</Title>
    <Author>Janusz R. Getta</Author>
    <Date>05-APR-2005</Date>
    <ISBN>09-345673</ISBN>
    <Publisher>Addison Wesley Ltd. Pty.</Publisher>
  </Book>
</BookStore>
```

DTD versus XML Schema

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="BookStore">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Book" minOccurs="1"
                                        maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
   ...          ...          ...          ...
```

<!ELEMENT BookStore (Book+)>

DTD versus XML Schema

```
...
<xsd:element name="Book">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Title" type="xsd:string"/>
<xsd:element name="Author" type="xsd:string"/>
<xsd:element name="Date" type="xsd:string"/>
<xsd:element name="ISBN" type="xsd:string"/>
<xsd:element name="Publisher" type="xsd:string"/>
</xsd:schema>
```

<!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>

DTD versus XML Schema

```
 ...          ...          ...          ...
<xsd:element name="Book">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Title" type="xsd:string"/>
<xsd:element name="Author" type="xsd:string"/>
<xsd:element name="Date" type="xsd:string"/>
<xsd:element name="ISBN" type="xsd:string"/>
<xsd:element name="Publisher" type="xsd:string"/>
</xsd:schema>
```

<!ELEMENT Title (#PCDATA)>

DTD versus XML Schema

```
 ...          ...          ...          ...
<xsd:element name="Book">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Title" type="xsd:string"/>
<xsd:element name="Author" type="xsd:string"/>
<xsd:element name="Date" type="xsd:string"/>
<xsd:element name="ISBN" type="xsd:string"/>
<xsd:element name="Publisher" type="xsd:string"/>
</xsd:schema>
```

<!ELEMENT Author (#PCDATA)>

XML Schema styles

XML Schema can be created using one of the following 3 styles:

Inlined element declarations:

Element declaration precisely follow an order of elements in a document. No new types are defined

Typed element declarations:

A hierarchy of types is defined and the elements are declared using new types

Global element declarations:

Hypothetical global elements are declared and referenced in either Inlined or Typed declarations

"Inlined" element declarations

<!ELEMENT BookStore (Book+)>

<!ELEMENT Title (#PCDATA)>

<!ELEMENT Author (#PCDATA)>

<!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>

"Typed" element declarations

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

    <xsd:element name="BookStore" type="BookStoreType"/>

        <xsd:complexType name="BookStoreType">
            <xsd:sequence>
                <xsd:element name="Book" maxOccurs="unbounded" type="BookType"/>
            </xsd:sequence>
        </xsd:complexType>

        <xsd:complexType name="Book">
            <xsd:sequence>
                <xsd:element name="Title" type="xsd:string"/>
                <xsd:element name="Author" type="xsd:string"/>
                <xsd:element name="Date" type="xsd:string"/>
                <xsd:element name="ISBN" type="xsd:string"/>
                <xsd:element name="Publisher" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>

</xsd:schema>
```

<!ELEMENT BookStore (Book+)>

<!ELEMENT Title (#PCDATA)>

<!ELEMENT Author (#PCDATA)>

<!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>

"Global" element declaration

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

  <xsd:element name="title" type="xsd:string"/>
  <xsd:element name="author" type="xsd:string"/>

  <xsd:element name="BookStore">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="book" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element ref="title"/>
              <xsd:element ref="author"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

</xsd:schema>
```

Global element title

Global element author

Reference to a global element title

Reference to a global element author

"One element" document

```
<Company xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="slide15.xsd">
         Golden Bolts Pty. Ltd.
</Company>
```

"slide15.xsd"

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

    <xsd:element name="Company" type="CompanyType"/>

    <xsd:simpleType name="CompanyType">
       <xsd:restriction base="xsd:string">
                    <xsd:minLength value="1"/>
                    <xsd:maxLength value="60"/>
       </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

Definition of element Company

Definition of type CompanyType

Restrictions of xsd:string type

"One attribute" document

```
<Company xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="slide16.xsd"
         name="Golden Bolts Pty. Ltd."/>
```

"slide16.xsd"

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

    <xsd:element name="Company" type="CompanyType"/>

    <xsd:complexType name="CompanyType">
      <xsd:attribute name="name" type="nameType"/>
    </xsd:complexType>

    <xsd:simpleType name="nameType">
       <xsd:restriction base="xsd:string">
                    <xsd:minLength value="1"/>
                    <xsd:maxLength value="60"/>
       </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

Definition of element Company

Definition of type CompanyType

Definition of attribute name

Definition of type nameType

Restrictions of xsd:string type

"One element + one attribute" document

```
<Company xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="slide17.xsd"
         ticker="GB">Golden Bolts Pty. Ltd.</Company>
```

"slide17.xsd"

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

    <xsd:element name="Company" type="CompanyType"/>

    <xsd:complexType name="CompanyType">
      <xsd:simpleContent>
        <xsd:extension base="nameType">
          <xsd:attribute name="ticker" type="xsd:string"/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>

    <xsd:simpleType name="nameType">
        <xsd:restriction base="xsd:string">
                <xsd:minLength value="1"/>
                <xsd:maxLength value="60"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

Definition of element Company

Definition of type CompanyType

Extension of type nameType

Definition of type nameType

Simple types

anysimpleType:

- duration
- dateTime
- time
- date
- gYearMonth
- gYear
- gMonth
- gDay
- gMonth
- boolean
- base64binary
- hexbinary

Simple types

anysimpleType:

- float
- double
- decimal
- integer
  - nonPositiveInteger
    - negativeInteger
  - nonNegativeInteger
    - unsignedLong
      - unsignedInt
        - unsignedShort
          - unsignedByte
    - positiveInteger

Simple types

anysimpleType (continuation):

- long
  - int
    - short
      - byte
- string
  - normalizedString
  - token
    - language
    - NAME
      - NCName
        - ID
        - IDREF

Simple types

anysimpleType (continuation):

- IDREFS
- ENTITY
  - ENTITIES
- NMTOKEN
  - NMTOKENS

Restriction on a simple type

```
<age xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="slide20.xsd">25</age>
```

"slide20.xsd"

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

    <xsd:element name="age" type="ageType"/>

    <xsd:simpleType name="ageType">
       <xsd:restriction base="xsd:positiveInteger">
          <xsd:maxInclusive value="120"/>
       </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

Definition of element age

Definition of type ageType as restriction of type xsd:positiveInteger

Restriction on a simple type

```
<first-name
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="slide23.xsd">James</first-name>
```

"slide23.xsd"

Definition of element first-name

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

    <xsd:element name="first-name" type="first-nameType"/>

    <xsd:simpleType name="first-nameType">
       <xsd:restriction base="xsd:string">
          <xsd:pattern value="([A-Z][a-z]*)"/>
       </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

Definition of type first-nameType as restriction of type xsd:string

Restriction on a simple type

```
<credits
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="slide24.xsd">12</credits>
```

"slide24.xsd"

Definition of element credits

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

    <xsd:element name="credits" type="creditsType"/>

    <xsd:simpleType name="creditsType">
       <xsd:restriction base="xsd:positiveInteger">
          <xsd:enumeration value="6"/>
          <xsd:enumeration value="12"/>
       </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

Definition of type creditsType as restriction of type xsd:positiveInteger and enumeration of given values

Derivation of a simple type (by union)

```
<int-or-string
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide25.xsd">12</int-or-string>
```

```
<int-or-string
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide25.xsd">"12"</int-or-string>
```

"slide25.xsd"

Definition of element `int-or-string`

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

    <xsd:element name="int-or-string" type="int-or-stringType"/>

    <xsd:simpleType name="int-or-stringType">
       <xsd:union memberTypes="xsd:positiveInteger xsd:string"/>
    </xsd:simpleType>

</xsd:schema>
```

Definition of type `int-or-stringType` as union of types `xsd:positive Integer` and `xsd:string`

Derivation of a simple type (by list)

```
<list-of-int
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide26.xsd">1 2 3 4 5</list-of-int>
```

"slide26.xsd"

Definition of element `list-of-int`

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

    <xsd:element name="list-of-int" type="list-of-intType"/>

    <xsd:simpleType name="list-of-intType">
       <xsd:list itemType="xsd:positiveInteger"/>
    </xsd:simpleType>

</xsd:schema>
```

Definition of type `list-of-intType` as list of `xsd:positiveInteger`

Derivation of a simple type (by list)

```
<sample-attribute
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide27.xsd"
   list-of-values="1 2 3 4 5"/>
```

"slide27.xsd"

Definition of attribute `list-of-values`

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

    <xsd:element name="sample-attribute
      <xsd:complexType>
        <xsd:attribute name="list-of-values" type="list-of-intType"/>
      </xsd:complexType>
    </xsd:element>

    <xsd:simpleType name="list-of-intType">
       <xsd:list itemType="xsd:positiveInteger"/>
    </xsd:simpleType>

</xsd:schema>
```

Definition of type `list-of-intType` as list of `xsd:positiveInteger`

Complex types with simple content

Simple types provide a description of leaf element nodes and attribute values

Complex types provide a description of a markup structure

There are two way to define complex types: one **for simple content** model and one for complex content model

Complex types with simple content model are created by adding a list of attributes to a simple type (extension)

Complex types with **simple content** model are also created by restricting scope of a text node and scope of an attribute (restriction)

Derivation by extension increases a number of child node elements or attributes

Derivation by restriction limits a scope of text node and limits a scope and number of attributes

Derivation by extension

```
<volume
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide29.xsd"
   units="gallons">25</volume>
```

"slide29.xsd"

Definition of element `volume`

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

    <xsd:element name="volume">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:nonNegativeInteger">
            <xsd:attribute name="units" type="xsd:string"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>

</xsd:schema>
```

Extension of type `xsd:nonNegativeInteger` with an attribute `units`

Derivation by extension

```
<volume
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide29.xsd"
   units="gallons" country="US">25</volume>
```

"slide30.xsd"

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

    <xsd:element name="volume">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:nonNegativeInteger">
            <xsd:attribute name="units" type="xsd:string"/>
            <xsd:attribute name="country" type="xsd:string"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>

</xsd:schema>
```

Definition of element `volume`

Extension of type `xsd:nonNegativeInteger` with the attributes `units` and `country`

Derivation by extension

```
<volume
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide31.xsd"
   units="gallons" country="US">25</volume>
```

Definition of element volume

"slide31.xsd"

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

   <xsd:element name="volume" type="volumeType"/>

     <xsd:complexType name="volumeType">
       <xsd:simpleContent>
         <xsd:extension base="xsd:nonNegativeInteger">
           <xsd:attribute name="units" type="xsd:string"/>
           <xsd:attribute name="country" type="xsd:string"/>
         </xsd:extension>
       </xsd:simpleContent>
     </xsd:complexType>

</xsd:schema>
```

Definition of type volumeType

Extension of type xsd:nonNegativeInteger with the attributes units and country

Derivation by restriction

```
<short-first-name
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide32.xsd"
   country="UK">James</short-first-name>
```

Definition of element short-first-name

"slide32.xsd"

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

   <xsd:element name="short-first-name" type="short-first-nameType"/>

   <xsd:complexType name="first-nameType">
         <xsd:simpleContent>
               <xsd:extension base="xsd:string">
                    <xsd:attribute name="country" type="xsd:string"/>
               </xsd:extension>
         </xsd:simpleContent>
   </xsd:complexType>

     <xsd:complexType name="short-first-nameType">
       <xsd:simpleContent>
         <xsd:restriction base="first-nameType">
           <xsd:maxLength value="30"/>
           <xsd:pattern value="([A-Z][a-z]*)"/>
         </xsd:restriction>
       </xsd:simpleContent>
     </xsd:complexType>

</xsd:schema>
```

Definition of type first-nameType

Definition of type short-first-nameType as restriction of type first-nameType

References

Elmasri R., Navathe S., Fundamentals of Database Systems, 6th edition, chapter 12 XML: Extensible Markup Language, pp. 420-448

http://www.uow.edu.au/~jrg/235/HOMEWORK
8.4 How to create XML Schemas and how to validate XML documents against XML Schemas ? (1)

http://www.uow.edu.au/~jrg/235/HOMEWORK
8.5 How to create XML Schemas and how to validate XML documents against XML Schemas ? (2)

http://www.uow.edu.au/~jrg/235/SLIDES/XML/Appendix-2-XMLSchema-1-Reference.pdf

http://www.uow.edu.au/~jrg/235/SLIDES/XML/Appendix-3-XMLSchema-2-Reference.pdf