XML Documents

Miao Qiao

SEAT
Massey University

XML: Schema Definition Language

- Well-formed XML: adheres to basic structural requirements.
- Valid XML: adheres to content-specific specification.

Two Schema Definition Languages for XML:

- Document Type Definition (DTD)
- XML Schema Definition (XSD)

XML: Schema Definition, Schema, and Instance Document

- XML Schema: the most widely used schema standard. XML schema is developed and maintained by W3C, and is designed to define a broad range of document structures.
  - URL: `http://www.w3.org/2001/XMLSchema`
- XML Schema Definition: an XML document that defines an XML vocabulary.
- Instance Document: the document to be validated when apply XML schema to a specific XML document.

XML Schema and DTD

XML schema is more powerful than DTD.

	XML Schema	DTD
Document language	XML	Extended Backus-Naur Form (EBNF)
Standard	multiple standard	single standard
Supported data type	44(19 primitives + 25 derived)	10
Mixed content	easy to develop	difficult to develop
Namespace	completely supported	only namespace prefix are supported
Entities	no	yes
Supported Keys	global/typed keys	only global keys

XML: Namespace

- Purpose: to avoid duplicated tag names in merging XML files.
- Method: prepend each tag with a Unique Resource Identifier (URI).
- Namespace: define an abbreviation for URIs.

An instance document with namespace:

```
<xsi:Bookstore xmlns:xsi="http://www.MasseyBookStore.com">
   <xsi:Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
      <xsi:Title>A First Course in Database Systems</xsi:Title>
      <xsi:Authors>
         <xsi:Author>
            <xsi:First_Name>Jeffrey</xsi:First_Name>
            <xsi:Last_Name>Ullman</xsi:Last_Name>
         </xsi:Author>
         <xsi:Author>
            <xsi:First_Name>Jennifer</xsi:First_Name>
            <xsi:Last_Name>Widom</xsi:Last_Name>
         </xsi:Author>
      </xsi:Authors>
   </xsi:Book>
```

XSD: Start a Schema File

- By convention, the namespace prefix, typically xsd or xs, is assigned to the XML Schema namespace to identify elements and attributes that belong to the XML Schema vocabulary.

- The usual form of an Schema document is:

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

XSD: an Example

See documents "Bookstore-XSD.xml" and "Bookstore.xsd".

XSD: Element Type

- Simple type: an XML element that can contain only text — no attributes / subelements

- Complex type: an XML element that contains subelements or attributes.

XSD: Define a Simple Type Element

Define a simple type element:

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

An instance of "Last_Name":

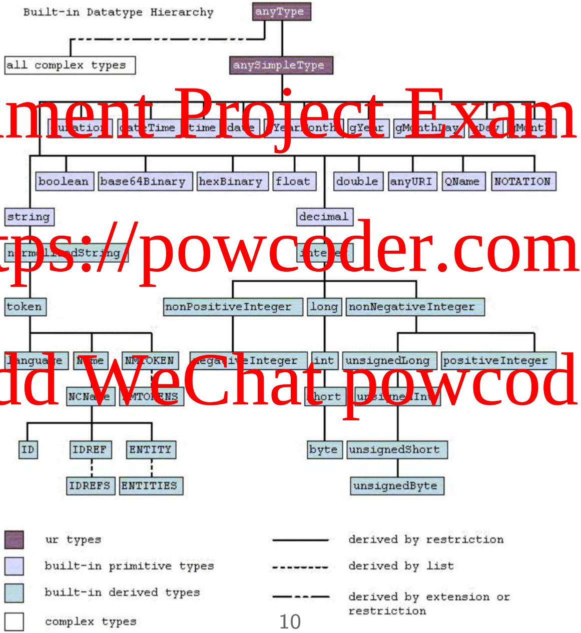
```
<Last_Name>Ullman</Last_Name>
```

Common built-in types:

- xsd:string
- xsd:decimal
- xsd:integer
- xsd:boolean
- xsd:date
- xsd:time

Their instances are all texts.

XSD: Built-in Data Type https://www.w3.org/TR/xmlschema-2/



XSD: Define a Complex Type

Define a complex type called BookType:

```
<xs:complexType name="BookType">
   Declaration...
</xs:complexType>
```

Define an element of the complex type:

```
<xs:element name="Book" type="BookType"/>
```

Or define the element directly:

```
<xs:element name="Book">
   <xs:complexType>
      Declaration...
   </xs:complexType>
</xs:element>
```

XSD: Contents of a Complex Type

Contents of a complex type:

- Attribute
- Basic text
- Nested child element

XSD: Contents of a Complex Type

Contents of a complex type:

- Attribute
- Basic text
- Nested child element

XSD: Define an Attribute

Define an attribute:

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

Define an instance with the attribute:

```
<Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
    ...
</Book>
```

XSD: Attribute Use Indicators

Use indicator: the requirement on the appearance of an attribute.

Use is one of the following three values:

- required: the attribute must always appear with the element.
- optional (default): the use of the attribute is optional with the element.
- prohibited: the attribute cannot be used with the element.

Example:

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

See documents “Bookstore.xsd” and “Bookstore-XSD.xml”.

XSD: Contents of a Complex Type

Contents of a complex type:

- Attribute
- Basic text
- Nested child element

A complex-typed element can have any combination of the three above:

1. Empty element: contain only attributes.
2. Basic text + attributes
3. Nested child elements
4. Nested child elements + attributes
5. Mixed: Nested child elements + attributes + basic text

XSD: Contents of a Complex Type

Contents of a complex type:

- Attribute
- Basic text
- Nested child element

A complex-typed element can have any combination of the three above:

1. Empty element: contain only attributes.
2. Basic text + attributes
3. Nested child elements
4. Nested child elements + attributes
5. Mixed: Nested child elements + attributes + basic text

XSD: Define a Empty Element

Define an empty element of Book:

```
<xsd:element name="Book">
   <xsd:complexType>
      <xsd:attribute name="ISBN" type="xsd:string" />
      <xsd:attribute name="Price" type="xsd:integer" />
      <xsd:attribute name="Edition" type="xsd:string" />
   </xsd:complexType>
</xsd:element>
```

Define an instance of Book:

```
<Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd"/>
```

XSD: Contents of a Complex Type

Contents of a complex type:

- Attribute
- Basic text
- Nested child element

A complex-typed element can have any combination of the three above:

1. Empty element: contain only attributes.
2. Basic text + attributes
3. Nested child elements
4. Nested child elements + attributes
5. Mixed: Nested child elements + attributes + basic text

XSD: Define an Basic Text with Attributes

Define an element of Book:

```
<xsd:element name="Book">
   <xsd:complexType>
      <xs:simpleContent>
         <xs:extension base="xs:string">
            <xsd:attribute name="ISBN" type="xsd:string" />
            <xsd:attribute name="Price" type="xsd:integer" />
            <xsd:attribute name="Edition" type="xsd:string" />
         </xs:extension>
      </xs:simpleContent>
   </xsd:complexType>
</xsd:element>
```

Define an instance of Book:

```
<Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
   A First Course in Database Systems
</Book>
```

XSD: Contents of a Complex Type

Contents of a complex type:

- Attribute
- Basic text
- Nested child element

A complex-typed element can have any combination of the three above:

1. Empty element: contain only attributes.
2. Basic text + attributes
3. Nested child elements
4. Nested child elements + attributes
5. Mixed: Nested child elements + attributes + basic text

XSD:Define an Element with Nested Child Elements

Define an element of Author:

```
<xs:element name="Author">
  <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="First_Name" type="xsd:string" />
         <xsd:element name="Last_Name" type="xsd:string" />
      </xsd:sequence>
   </xsd:complexType>
</xs:element>
```

Define an instance of Author:

```
 <Author>
   <First_Name>Jeffrey</First_Name>
   <Last_Name>Ullman</Last_Name>
</Author>
```

XSD: Order Indicators for Nested Child Elements

Order indicators: to specify the order of the appearances of the child elements.

- All: the child elements can appear in any order, and that each child element must occur only once.
- Choice: either one child element or another can occur.
- Sequence: the child elements must appear in a specific order.

```
<xs:element name="Author">
  <xsd:complexType>
      <xsd:choice>
         <xsd:element name="First_Name" type="xsd:string" />
         <xsd:element name="Last_Name" type="xsd:string" />
      </xsd:choice>
   </xsd:complexType>
</xs:element>
```

See documents “Bookstore.xsd” and “Bookstore-XSD.xml”.

XSD: Occurrence Indicators for Nested Child Elements

Occurrence indicator: to define how often an element can occur.

```
<xsd:element name="Bookstore">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Book" type="BookType"
            minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

The default value for both maxOccurs and minOccurs is 1.

See documents "Bookstore.xsd" and "Bookstore-XSD.xml".

XSD: Contents of a Complex Type

Contents of a complex type:

- Attribute
- Basic text
- Nested child element

A complex-typed element can have any combination of the three above:

1. Empty element: contain only attributes.
2. Basic text + attributes
3. Nested child elements
4. Nested child elements + attributes
5. Mixed: Nested child elements + attributes + basic text

XSD: Define an Element with Child Elements + Attributes

Define an element “Book” with both child elements and attributes:

```
<xsd:element name="Book" type="BookType"
            minOccurs="0" maxOccurs="unbounded" />
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="Title" type="xsd:string" />
         <xsd:element name="Authors">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="Author" type="AuthorType"
                     maxOccurs="unbounded" />
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="ISBN" type="xsd:string" use="required" />
      <xsd:attribute name="Price" type="xsd:integer" use="required" />
      <xsd:attribute name="Edition" type="xsd:string" use="optional" />
   </xsd:complexType>
</xsd:element>
```

See documents “Bookstore.xsd” and “Bookstore-XSD.xml” for an instance.

XSD: Contents of a Complex Type

Contents of a complex type:

- Attribute
- Basic text
- Nested child element

A complex-typed element can have any combination of the three above:

1. Empty element: contain only attributes.
2. Basic text + attributes
3. Nested child elements
4. Nested child elements + attributes
5. Mixed: Nested child elements + attributes + basic text

XSD: Define a Mixed Element

Define a mixed element by setting its complextype attribute mixed="true".

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

An instance of letter:

```
<letter shipdate="2001-07-13">
  Dear Mr.<name>John Smith</name>.
  Your order <orderid>1032</orderid>
</letter>
```

XSD: Keys and Foreign Keys

Specify an attribute (ISBN) of an element (Book) is a key:

```
<xsd:key name="BookKey">
   <xsd:selector xpath="Book" />
   <xsd:field xpath="@ISBN" />
</xsd:key>
```

XSD: Keys and Foreign Keys

Specify an attribute (ISBN) of an element (Book) is a key:

```
<xsd:key name="BookKey">
   <xsd:selector xpath="Book" />
   <xsd:field xpath="@ISBN" />
</xsd:key>
```

Specify an attribute (book) of an element (Book/Remark/BookRef) is a foreign key of key “BookKey” defined above:

```
<xsd:keyref name="BookKeyRef" refer="BookKey">
   <xsd:selector xpath="Book/Remark/BookRef" />
   <xsd:field xpath="@book" />
</xsd:keyref>
```

XSD: Keys and Foreign Keys

Specify an attribute (ISBN) of an element (Book) is a key:

```
<xsd:key name="BookKey">
   <xsd:selector xpath="Book" />
   <xsd:field xpath="@ISBN" />
</xsd:key>
```

Specify an attribute (book) of an element (Book/Remark/BookRef) is a foreign key of key “BookKey” defined above:

```
<xsd:keyref name="BookKeyRef" refer="BookKey">
   <xsd:selector xpath="Book/Remark/BookRef" />
   <xsd:field xpath="@book" />
</xsd:keyref>
```

See documents “Bookstore-key.xsd” and “Bookstore-XSD-key.xml”.

References

- W3Cschool:
  https://www.w3schools.com/xml/schema_intro.asp
- W3Crecommendation:
  - https://www.w3.org/TR/xmlschema-1/
  - https://www.w3.org/TR/xmlschema-2/
- Chapter 10, “Database System Concepts (Fifth Edition)”, A. Silberschatz, H. F. Korth, S. Sudarshan.
- Introduction to Databases, Stanford University.