XML Schema

Prof. Weining Zhang

XML Schema

- W3C Recommendation
- A language to define schema of a collection of XML documents
  - to declare various types of elements in the XML documents
- The schema is itself in XML syntax
- Instance documents may be linked to their XML schema in various ways
- XML Schema provides built-in types and allow users to define new types and constraints

Conformance to Schema

- An XML Schema defines syntactic rules and constraints to be satisfied by instance documents
- An application can check if an instance document conforms to any given XML schema
- The W3C recommendation specifies
  - Syntax of the schema definitions
  - Semantics of schema rules
  - Conformance of instance documents

Types and Namespace

- Each element/attribute has a type
- A type is complex if it contains other elements or attributes, otherwise, it is simple
- Attributes can only be of a simple type, so is an element containing value only
- Special tags in XML Schema namespace are used in the schema document to define types, elements, attributes, etc. of instance documents

XML Schema of the Bank

```xml
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="bank" type="BankType"/>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="account" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

XML Schema of the Bank

```xml
.. definitions of customer and depositor ..
<xsd:complexType name="BankType">
  <xsd:sequence>
    <xsd:element ref="account" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="customer" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="depositor" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```
Element Declaration
- Declared using xsd:element element
- Two way to declare:
  - With a name and a type
  - With a reference to an existing element
- The element type can be declared
  - By a named type, or
  - An inline anonymous type definition
- Can appear within a type declaration or outside of any type declaration (global)

Attribute Declaration
- Declared in an xsd:attribute element
- Specify a name and a simple type
- EX:
  - `<xsd:attribute name="country" type="xsd:NMTOKEN" fixed="US"/>
  - XML Schema has 40+ built-in simple types

User-Defined Simple Type
- Using an xsd:simpleType element and
  specify a restriction of a built-in simple types
  - `<xsd:simpleType name="myInteger">
    - `<xsd:restriction base="xsd:integer">
      - `<xsd:minInclusive value="10000"/>
      - `<xsd:maxInclusive value="99999"/>
    - `/xsd:restriction>
  - Fifteen facets such as minInclusive, enumeration, and pattern are predefined to specify restrictions

Define A Complex Type
- Using an xsd:complexType element
- Declare sub-elements and/or attributes
  - Not the types of sub-elements and attributes, but the occurrence and ordering of them
  - Can declare sub-elements as references, too
  - Can specify default values for elements & attributes
  - Can specify number of occurrence of elements

Other User-Defined Types
- List Type: allow a list of values of the same simple type
  - `<xsd:simpleType name="listOfMyIntType">
    - `<xsd:listItemTypes="myInteger"/>
    - `/xsd:simpleType>
- Union Type: allow values of a union of simple types
  - `<xsd:simpleType name="zipUnion">
    - `<xsd:unionMemberTypes="USState listOfMyIntType"/>
    - `/xsd:simpleType>
- Anonymous Type: allow types without names

Element Content
- Contains only attributes & a simple value
  - Use xsd:simpleContent to derive from a simple type
- Contains both simple value & sub-elements
  - Use attribute mixed in xsd:complexType
- Contains only attributes (no element value)
  - Specify xsd:complexTypeContent, but specify no element
- Xsd:anyType
  - Does not constrain the content
  - Is the default content type
Examples
- Simple value + attribute
  `<xsd:element name="internationalPrice">`
  `<xsd:complexType>`
  `<xsd:simpleContent>`
  `<xsd:extension base="xsd:decimal">`
  `<xsd:attribute name="currency" type="xsd:string" />`
  `<xsd:extension>`
  `<xsd:simpleContent>`
  `<xsd:complexType>`

- Element with no content
  `<xsd:element name="internationalPrice">`
  `<xsd:complexType>`
  `<xsd:complexContent>`
  `<xsd:restriction base="xsd:schema">`
  `<xsd:attribute name="currency" type="xsd:string" />`
  `<xsd:restriction>`
  `<xsd:complexContent>`
  `<xsd:complexType>`

Element Groups
- Types of groups
  - Xsd:sequence. Fixed ordering
  - Xsd:choice. Only one in the group can appear
  - Xsd:all. All must appear, but in any order
- Named Groups
  - Assign a name to a group of element using `xsd:group`

Attribute Groups
- Similar to element groups
- Can be declared in element by references
  `<xsd:attributeGroup name="ItemDelivery">`
  `<xsd:attribute name="partNum" type="SKU" use="required" />`
  `<xsd:attribute name="weightKg" type="xsd:decimal" />`
  `<xsd:attribute name="shipBy">`
  `<xsd:simpleType>`
  `<xsd:restriction base="xsd:string">`
  `<xsd:enumeration value="air" />`
  `<xsd:enumeration value="land" />`
  `<xsd:restriction>`
  `<xsd:simpleType>`
  `<xsd:attributeGroup>`

Default vs Target Namespaces
- EX:
  `<xsd:element name="shipAndBill" />`
  `<xsd:element name="shipTo" type="USAddress" />`

Using Other XML Schemas
- A schema can include definitions from other schema files
  `<include schemaLocation="http://.../schemas/addr.xsd" />`
  - All these files must define the same target namespace
- A schema can import definitions from other schema files that define different target namespaces
  `<import namespace="http://www.example.com/IPO" />`

Qualification of Local Names
- Controlled by
  - `elementFromDefault`
  - `attributeFromDefault`
- If set to "unqualified", names declared locally in complex types must be unqualified
- If set to "qualified", local names must be qualified
  - Directly by a prefix
  - Indirectly by defining a default namespace in instance documents
Specify Uniqueness & Keys

- Specify unique code attribute in elements reachable with a path regions/zip
  <unique name="dummy1"> <selector
    xpath="r:regions/r:zip"/> <field xpath="@code"/>
  </unique>
- Specify that attribute reachable with a path parts/part/@number is a key with a unique and non-null value, and the name pNumKey is a reference to the key
  <key name="pNumKey"> <selector xpath="r:parts/r:part"/>
    <field xpath="@number"/> <key/>

Other XML Schema Languages

- XML-Data (XDR)
- Document Content Description (DCD)
- Schema for Object-oriented XML (SOX)
- Document Definition Markup Language (DDML)
- Schematron
- Document Structure Description (DSD)
- Regular Language Description for XML (RELAX)
- TREX (Tree Regular Expressions for XML)
- RELAX NG
- Examplotron
- Hook
- Document Schema Definition Language (DSDL)
- STEP/EXPRESS and XML

Tools Related to XML Schema

- Schema Validator
  ▲ XSV, XML Schema Validator, XSDValid
- Schema Models
  ▲ Caster: an OO model in Java
  ▲ MSXML: DOM & SAX + XSLT
- Schema Translator
  ▲ Dtd2xs
- Schema User Interface
  ▲ JaxFront, Schema Viewer, SQC Schema Quality Checker, Xbuilder Schema builder
- Many others are available

Examples of XML Schema

- Log Markup Language (LOGML)
  ▲ describes log reports of web servers
- Geography Markup Language (GML)
  ▲ OpenGIS Consortium
- XGMML (eXtensible Graph Markup and Modeling Language)
  ▲ used for graph description
- Many DTD or XML Schema definitions in real-world applications at xml.org
  ▲ http://www.xml.org/xml/registry.jsp