2.5 XML Schemas

- Short introduction to XML Schema
  - W3C Recommendation, 1st Ed. May, 2001; 2nd Ed. Oct, 2004:
    - XML Schema Part 0: Primer (readable non-normative introduction; Recommended)
    - XML Schema Part 1: Structures
    - XML Schema Part 2: Datatypes
  - W3C Draft (didn't lead anywhere?):
    - Formal Description, 9/2001

Advantages of XSDL (1)

- XML syntax
  - schema documents easier to manipulate by programs (than the special DTD syntax)
- Compatibility with namespaces
  - can validate documents using declarations from multiple sources
- Content datatypes
  - 44 built-in datatypes (including primitive Java datatypes, datatypes of SQL, and XML attribute types)
  - mechanisms to derive user-defined datatypes

Advantages of XSDL (2)

- Element names and content types independent; Compare with
  - For example, could define titles
    - of people as "Mr."/"Mrs."/"Ms.", and
  - more powerful than CFGs/DTDs
    - where non-terminal / tag-name determines its allowed content alone
  - relevant for practice?

Advantages of XSDL (3)

- Ability to specify uniqueness and keys within selected parts of the document
  - for example, that titles of chapters should be unique; or key attributes of relations
  - uses XPath
- Support for schema documentation
  - element annotation with sub-elements documentation (for human readers) and appInfo (for applications)

Disadvantages of XSDL

- Complexity of XSDL (esp. Part 1!) vs. added power
  - a long learning curve
  - slow adoption by users
- Immaturity of implementations (?)
  - ~ 60 tools or processors mentioned at the W3C site
  - Apache Xerces claims full XSDL support
  - Some features difficult to implement efficiently; See later
- Alternative schema languages have been suggested, too

XSDL through Example

- Next: walk-through of an XML schema example
  - from Chapter 2 of the XML Schema Primer
  - Consider modelling purchase orders like below:

```xml
<purchaseOrder orderDate="1999-10-20">
  <shipTo country="US">
    <name>Alice Smith</name>
    <street>123 Maple Street</street>
  </shipTo>
  <city>Mill Valley</city>
  <state>CA</state>
  <zip>90952</zip>
</purchaseOrder>
```
The Purchase Order Schema (1/5)

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="purchaseOrder" type="POrdType"/>
  <xs:element name="comment" type="xs:string"/>
  <xs:complexType name="POrdType">
    <xs:complexContent>
      <xs:restriction base="xs:positiveInteger">
        <xs:choice>
          <xs:element ref="item" minOccurs="0" maxOccurs="unbounded"/>
          <xs:sequence>
            <xs:element name="SKU" type="xs:string"/>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="quantity" type="xs:decimal"/>
            <xs:element name="ordDate" type="xs:date"/>
            <xs:attribute name="partNum" type="SKU" use="required"/>
          </xs:sequence>
        </xs:choice>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

The Purchase Order Schema (2/5)

```xml
<xs:complexType name="USAddr">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="street" type="xs:string"/>
    <xs:element name="city" type="xs:string"/>
    <xs:element name="state" type="xs:string"/>
    <xs:element name="zip" type="xs:decimal"/>
  </xs:sequence>
</xs:complexType>
```

The Purchase Order Schema (3/5)

```xml
<xs:complexType name="Items">
  <xs:sequence>
    <xs:element name="item" minOccurs="0" maxOccurs="unbounded">
      <xs:element name="SKU" type="xs:string"/>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="quantity" type="xs:decimal"/>
      <xs:element name="ordDate" type="xs:date"/>
      <xs:attribute name="partNum" type="SKU" use="required"/>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The Purchase Order Schema (4/5)

```xml
<xs:element name="USPrice" type="xs:decimal"/>
<xs:element ref="comment" minOccurs="0"/>
<xs:element name="shipDate" type="xs:date" minOccurs="0"/>
<xs:attribute name="partNum" type="SKU" use="required"/>
</xs:complexType>
```

The Purchase Order Schema (5/5)

```xml
<!-- Type for Stock Keeping Units, (codes for identifying products): -->
<xs:simpleType name="SKU">
  <xs:restriction base="xs:string">
    <!-- defined by a regular expr: -->
    <xs:pattern value="[d3]-[A-Z][2]"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

XML Schema Content Models

- Element content of `complexType` can be regulated using
  - group elements `sequence`, `choice` and `all`, and
  - occurrence constraint attributes `minOccurs` and `maxOccurs`
- Elements `sequence` and `choice` correspond to catenation and alternation `(|)` in regular expressions
XSDL Occurrence Constraints

- optionality (E?) can be expressed by 
  \[ \text{minOccurs} = \text{E} \]
- Iteration (E') can be expressed by 
  \[ \text{minOccurs} = 1 \text{ and maxOccurs} = \text{E'} \]
- Exactly five occurrences of element A:
  \[ \text{xs:element name="A" minOccurs=5 maxOccurs=5} \]
- 10 to 900 occurrences of element A:
  \[ \text{xs:element name="A" minOccurs=10 maxOccurs=900} \]

Challenge of Numeric Occurrence Indicators

- Occurrence indicators can be eliminated
  \[ \text{<element name="A" minOccurs=2 maxOccurs=5} \]
  \[ \text{-> A, A, (A, (A, A?)?)} \]
- XML Schema Parsers, e.g. Apache Xerces, do this before building an automation
- Large enough occurrence values crash them!
- We are trying to do better; See
  Kilpeläinen & Tuhkanen: "Towards efficient implementation of XML Schema content models", ACM Docking 2004

The all group: An example

- For example
  \[ \text{<xs:all}> \]
  \[ \text{<xs:element name="A"} \]
  \[ \text{<xs:element name="B"} \]
  \[ \text{<xs:element name="C" minOccurs=0} \]
  \[ </xs:all> \]
accepts the following element sequences:
A B C; A C B; B A C; B C A; C A B; C B A;
A B; and B A;

XML Schema: Summary

- XSDL: an XML-based grammar formalism
  - W3C Recommendation; Alternative to DTDs
  - support for namespaces
  - richer content and attribute datatypes
- Well accepted(?) in XML industry
  - e.g., to describe messages between clients and servers in Web services; (See, e.g., Web Services Description Language, Vers. 2.0, W3C Draft 3/07)

XML Schema: Future?

- XSDL has also been criticized
  - for overwhelming complexity vs. relatively little added power
  - other XML-based schema languages proposed and implemented, too
  - E.g., RELAX NG (ISO/IEC standard)
    - Simpler and with a rigorous formal base, yet more powerful at certain places (e.g. dependencies between attributes/content, unordered content)