2.5 XML Schemas

- A quick introduction to XML Schema
  - W3C Recommendation, May 2, 2001:
    - XML Schema Part 0: Primer (readable non-normative Introduction; Recommended)
    - XML Schema Part 1: Structures
    - XML Schema Part 2: Datatypes
  - Also under development:

Schema terminology

- Schema: a formal description for the structure and allowed content of a set of data (esp. in databases)
- “XML Schema” is often used for each of ...
  1. XML Schema, the W3C Rec. that defines ...
  2. XML Schema Definition Language (XSDL), an XML-based markup language for expressing ...
  3. schema documents, each of which describes a schema (~DTD) for a set of XML document instances (This may cause some confusion!)

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

XSDL built-in types (Part 2, Chap. 3)

Advantages of XSDL (2)

- Independence of element names and content types; Compare with ...
  - DTDs: 1-to-1 correspondence btw. element type names and their content models
  - CFGs: 1-to-1 correspondence btw. nonterminals and their productions
- For example, could define titles of people as “Mr.”/“Mrs.”/“Ms.” and titles of chapters as strings

Advantages of XSDL (3)

- Support for schema documentation
  - element annotation with sub-elements documentation (for human readers) and appInfo (for applications)
- Ability to specify uniqueness and keys within selected parts of document
  - for example, that titles of chapters should be unique
Disadvantages of XSDL

- Complexity of XSDL (esp. of Rec. Part 1!)
  → a long learning curve

- Possible immaturity of implementations (?)
  – W3C XML Schema Web site mentions a dozen of tools or processors (http://www.w3.org/XMLSchemaTools, March 2002)
  – Open-source Apache XML parsers (Xerces C++ 1.7.0 and Xerces Java 1.4.4) seem reasonable implementations, but also document limitations/problems in their XML Schema support

XSDL through Example

- Next: walk-through of an XML schema example
  – from XML Schema Rec, Part 0, Chap. 2
  – Modelling purchase orders like below:

  ```xml
  <purchaseOrder orderDate="1999-10-20">
    <shipTo country="US">
      <name>Alice Smith</name>
      <street>123 Maple Street</street>
      <city>Mill Valley</city>
      <state>CA</state>
      <zip>90952</zip>
    </shipTo>

    <billTo country="US">
      <name>Robert Smith</name>
      <street>8 Oak Avenue</street>
      <city>Old Town</city>
      <state>PA</state>
      <zip>95819</zip>
    </billTo>

    <comment>Hurry, my lawn is wild!</comment>
    <items>
      <item partNum="872-AA">
        <productName>Lawnmower</productName>
        <quantity>1</quantity>
        <USPrice>148.95</USPrice>
        <comment>Only if electric</comment>
      </item>
      <item partNum="926-AA">
        <productName>Baby Phone</productName>
        <quantity>1</quantity>
        <USPrice>39.98</USPrice>
        <shipDate>1999-05-21</shipDate>
      </item>
    </items>
  </purchaseOrder>
  ```

End of the example instance

Next: A schema for this kind of purchase orders

```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:sequence>
      <xs:element name="shipTo" type="USAddr"/>
      <xs:element name="billTo" type="USAddr"/>
      <xs:element ref="comment" minOccurs="0"/>
      <xs:element name="items" type="Items"/>
    </xs:sequence>
    <xs:attribute name="ordDate" type="xs:date"/>
  </xs:complexType>
</xs:schema>
```
The Purchase Order Schema (3/5)

```xml
<xs:complexType name="Items">
  <xs:sequence>
    <xs:element name="item" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="productName" type="xs:string"/>
          <xs:element name="quantity">
            <xs:simpleType>
              <xs:restriction base="xs:positiveInteger">
                <xs:maxExclusive value="100"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="partNum" type="SKU" use="required"/>
      </xs:complexType>
    </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:sequence>
</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">
    <xs:pattern value="\d{3}-[A-Z]{2}"/>
  </xs:restriction>
</xs:simpleType>
```

XSDL Content Models

- Element content of `complexType` can be regulated using
  - group elements `sequence`, `choice` and `all`
  - occurrence constraint attributes `minOccurs` and `maxOccurs`

- Elements `sequence` and `choice` correspond to catenation and alternativity (`|`) in regular expressions

XSDL occurrence constraints

- optionality (E?) can be expressed by `minOccurs='0'`
- iteration (E*) can be expressed by `minOccurs='0'` and `maxOccurs='unbounded'`
- Exactly five occurrences of element A:
  ```xml
  <xs:element name="A" minOccurs='5' maxOccurs='5'/>
  ```
- 10 to 900 occurrences of element A:
  ```xml
  <xs:element name="A" minOccurs='10' maxOccurs='900'/>
  ```

Regular expression vs an XSDL content model

- A | B (C D)* could be expressed by
  ```xml
  <xs:choice>
    <xs:element name="A" type="typeA" maxOccurs='unbounded'/>
    <xs:choice>
      <xs:element name="B" type="typeB" maxOccurs='unbounded'/>
      <xs:sequence minOccurs='0' maxOccurs='unbounded'>
        <xs:element name="C" type="typeC"/>
        <xs:element name="D" type="typeD"/>
      </xs:sequence>
    </xs:choice>
  </xs:choice>
  ```
The all group

- XSDL all group is a restricted version of the SGML &-connector
  - E1 & ... & En allows sequences corresponding to any permutation of E1, ..., En
  - XSDL all restricted to appear as the only content model group of a complexType, and its children to (possibly optional) element

The all group: An example

- For example
  
  ```xml
  <xs:all>
    <xs:element name="A" />
    <xs:element name="B" />
    <xs:element name="C" minOccurs="0" />
  </xs:all>
  ```
  accepts all of 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;