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:

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)

- 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
  - uses XPath

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)

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

* NB: values in documents represented by strings
Disadvantages of XSDL

- Complexity of XSDL (esp. of Rec. Part 1)
  - a long learning curve
  - possible rejection by users
- Possible immaturity of implementations (?)
  - Open-source Apache XML parsers (Xerces C++ 2.0 and Xerces 2. Java) claim full XML Schema support

XSDL through Example

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

```xml
<purchaseOrder instanc...mitty)/city)
<state>CA/state>
<zip>90905</zip>
</shipTo>
</item>
</purchaseOrder>

End of the example instance

<item partNum="926-AAA">
  <productName>Baby Phone</productName>
  <quantity1>1</quantity1>
  <USPrice>148.95</USPrice>
  <comment>Hurry, my lawn is wild!</comment>
</item>
</item>
</purchaseOrder>

Next: A schema for such purchase orders

The Purchase Order Schema (1/5)

```xml
<xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema"
<xs:element name="purchaseOrder" type="xsd:complexType"/>
<xs:complexType name="purchaseOrder">
  <xs:sequence>
    <xs:element name="billTo" type="xsd:complexType"/>
    <xs:element name="shipTo" type="xsd:complexType"/>
  </xs:sequence>
</xs:schema>
```

The Purchase Order Schema (2/5)

```xml
<xs:complexType name="USAddr">
  <xs:sequence>
    <xs:element name="name" type="xsd:string"/>
    <xs:element name="street" type="xsd:string"/>
    <xs:element name="city" type="xsd:string"/>
    <xs:element name="state" type="xsd:string"/>
    <xs:element name="zip" type="decimal"/>
    <xs:element name="country" type="xsd:token" fixed="US"/>
  </xs:sequence>
</xs:complexType>
```
The Purchase Order Schema (3/5)

```xml
<xsd:complexType name="item">
  <xsd:sequence>
    <xsd:element name="item" minOccurs="0" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:anonymous type="xsd:string"/>
        <xsd:element name="quantity" type="xsd:integer"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
```

The Purchase Order Schema (4/5)

```xml
<xsd:complexType name="SKU">
  <xsd:restriction base="xsd:string">
    <!-- defined by a regular expr: -->
    <xsd:pattern value="^[A-Z]+$"/>
  </xsd:restriction>
</xsd:complexType>
```

The Purchase Order Schema (5/5)

```xml
<!-- Type for Stock Keeping Units, (codes for identifying products): -->
<xsd:simpleType name="SKU">
  <xsd:restriction base="xsd:string">
    <!-- 3 digits, hyphen, 2 letters -->
  </xsd:restriction>
</xsd:simpleType>
```

XSDL 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 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:
  - minOccurs="5" maxOccurs="5"
- 10 to 900 occurrences of element A:
  - minOccurs="10" maxOccurs="900"

Regular expression vs an XSDL content model

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

- XSDL all group is a restricted version of the SGML &-connector
  - E_1 B ... A_1 En allows sequences corresponding to any permutation of E_1, ..., En
  - XSDL all restricted to appear as the only content model group of a complexType, and its children to (possibly optional) elements

XML Schema: Summary

- XSDL: an XML-based grammar formalism
  - W3C Recommendation
  - more powerful than DTDs (esp. content and attribute datatypes)

- Future of alternative schema languages?
  - complexity of XSDL criticized
  - other XML-based schema languages proposed and implemented, too
  - e.g., WSDL (Web Services Description Language, W3C Working Draft, July 02) recommends XSDL for describing messages exchanged between web services