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
- Also started (but didn’t lead anywhere?):

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
  » 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
    » of chapters as strings

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. of Rec. Part 1!) vs. added power
  » a long learning curve
  » slow adoption by users
- Immaturity of implementations (?)
  » W3C site mentions ~60 tools or processors (http://www.w3.org/XML/SchemaTools; ~40 in March’04)
  » Apache Xerces claims full XSDL support
    » Some features difficult to implement efficiently; Con later

Schema Terminology

- Schema (kaavio): a formal description for the structure and allowed contents 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 through Example

- Next: walk-through of an XML schema example
  - from Chapter 2 of the XML Schema Primer
  - Consider modelling purchase orders like below:
    <purchaseOrder orderDate="1999-10-20">
      <shipTo country="US">
        <name>Mr. Smith</name>
        <street>123 Maple Street</street>
        <city>Menlo Park</city>
        <state>CA</state>
        <zip>94025</zip>
      </shipTo>
    </purchaseOrder>
purchaseOrder instance continues

```xml
<billTo country="US">
  <name>Robert Smith</name>
  <street>8 Oak Avenue</street>
  <city>Old Town</city>
  <state>PA</state>
  <zip>19519</zip>
</billTo>
```

End of the example instance

```xml
</item>
</item>
</purchaseOrder>
```

Next: A schema for such purchase orders

The Purchase Order Schema (1/5)

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="purchaseOrder" type="POrdType"/>
</xs:schema>
```

The Purchase Order Schema (2/5)

```xml
<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:complexType>
```

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="partNum" type="SKU"/>
          <xs:element name="productName" type="xs:string"/>
          <xs:element name="quantity" type="xs:decimal"/>
          <xs:element ref="comment" minOccurs="0"/>
          <xs:element name="shipDate" type="xs:date" minOccurs="0"/>
          <xs:attribute name="country" type="xs:NMTOKEN" fixed="US"/>
        </xs:complexType>
      </xs:element>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The Purchase Order Schema (4/5)

```xml
<xs:choice name="SKU">
  <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:choice>
```

The Purchase Order Schema (5/5)

```xml
</item>
</item>
</purchaseOrder>
```

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 alternation ( ) in regular
expressions
XSDL Occurrence Constraints

- **optionality (E?)** can be expressed by
  \[ \text{minOccurs}=0' \]
  _default of both is 1_ 
- **iteration (E*)** can be expressed by
  \[ \text{minOccurs}=0' \quad \text{and} \quad \text{maxOccurs}=\text{unbounded}' \]
- **Exactly five occurrences of element A:**
  \[ <\text{xs:element name='A'} \text{ minOccurs}=5' \text{ maxOccurs}=5' /> \]
- **10 to 900 occurrences of element A:**
  \[ <\text{xs:element name='A'} \text{ minOccurs}=10' \text{ maxOccurs}=900' /> \]

SDP 2006 2.5 XML Schemas 17

---

RegExprs vs XSDL Content Models

- **A* \[ B (C D)^+ \]** could be expressed by
  \[<\text{xs:element ref='A'} \text{ maxOccurs='unbounded' } /> \]
  \[<\text{xs:sequence} \text{ ref='B' } /> \]
  \[<\text{xs:sequence minOccurs='0'} \text{ maxOccurs='unbounded' } /> \]
  \[<\text{xs:element ref='C' } /> \]
  \[<\text{xs:element ref='D' } /> \]
  \[<\text{xs:sequence} /> \]
  \[<\text{xs:choice} /> \]

SDP 2006 2.5 XML Schemas 18

---

Challenge of Numeric Occurrence Indicators

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

SDP 2006 2.5 XML Schemas 19

---

Unordered content: the all group

- **XSDL all group is a restricted version of the &-connector in SGML**
  \[ E1 \& \ldots \& En \]
  **allows sequences corresponding to any permutation of E1, ... , En**
- **XSDL restrictions to all:**
  - **cannot be combined with other content model groups**
  - **can have only (non-iterative) elements as children**
  - **can be optional (minOccurs=0) but not repeated (maxOccurs=1)**

SDP 2006 2.5 XML Schemas 20

---

The all group: An example

- **For example**
  \[ <\text{xs:all} /> \]
  \[ <\text{xs:element name='A'} /> \]
  \[ <\text{xs:element name='B'} /> \]
  \[ <\text{xs:element name='C'} minOccurs='0' /> \]

\[<\text{xs:all} />\]

**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;

SDP 2006 2.5 XML Schemas 21

---

XML Schema: Summary

- **XSDL: an XML-based grammar formalism**
  - W3C Recommendation
  - Alternative to the DTD formalism
  - more powerful than DTDs (esp. content and attribute datatypes)
- **Rather well accepted(?) in XML industry**
  - e.g., to describe messages btw clients and servers in **Web services**; (See, e.g., Web Services Description Language, Vers. 2.0, W3C Cand. Rec., 3/06)

SDP 2006 2.5 XML Schemas 22

---

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
  - For example, RELAX NG
    - Draft international ISO/IEC standard
    - Simpler and with a rigorous formal base, yet more powerful at certain places (e.g. dependencies of attributes, unordered content)

SDP 2006 2.5 XML Schemas 23