Data-centric XML

XML Schema (Part 2)

DTD Example: Recipes

Document element

```
<!ELEMENT rezepte-sammlung (titel, rezept+)>
```

```
<xs:element name="rezepte-sammlung">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="titel" />
      <xs:element ref="rezept" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Recipes: Title

```
<!ELEMENT titel (#PCDATA)>
```

```
<xs:element name="titel" type="xs:string" />
```

Recipes: Individual Recipe

```
<!ELEMENT rezept
   (titel, zutaten?, zubereitung, idee*)>

<xs:element name="rezept">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="titel" />
      <xs:element ref="zutaten" minOccurs="0" />
      <xs:element ref="zubereitung" />
      <xs:element ref=”idee” maxOccurs=”unbounded” minOccurs=”0”/>
   </xs:sequence>
  </xs:complexType>
</xs:element>
```

Recipes: Ingredients

```
<!ELEMENT zutaten (zutat)+>
<!ELEMENT zutat (#PCDATA)>
<!ATTLIST zutat
        id    ID    #IMPLIED
        menge CDATA #IMPLIED>
```

Recipes: Ingredients (2)

```
<xs:element name="zutaten">
  <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
      <xs:element name="zutat">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="id" type="xs:ID" />
              <xs:attribute name="menge" type="xs:string" />
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Recipes: Preparation

```
<!ELEMENT zubereitung (#PCDATA | zutat)*>
<!ATTLIST zubereitung dauer CDATA #IMPLIED>
<!ATTLIST zutat
        ref IDREF #IMPLIED>
```

Recipes: Preparation (2)

```
<xs:element name="zubereitung">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="zutat">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="ref" type="xs:IDREF" use="required" />
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:choice>
    <xs:attribute name="dauer" type="xs:string" />
  </xs:complexType>
</xs:element>
```

Element Groups

- Allow re-use of element definitions
- Can be used in a similar way as parameter entities in DTDs
- Must specify sequence, choice, or all

```
<xs:group name="person">
  <xs:sequence>
    <xs:element name="titel" type="xs:string" minOccurs="0" />
    <xs:element name="vorname" type="xs:string" />
    <xs:element name="name" type="xs:string" />
  </xs:sequence>
</xs:group>
```

Element Groups: Usage

```
<xs:complexType name="...">
  <xs:sequence>
    <xs:group ref="person" />
    <xs:element name="alter" type="xs:short" />
  </xs:sequence>
</xs:complexType>
```

Attribute Groups

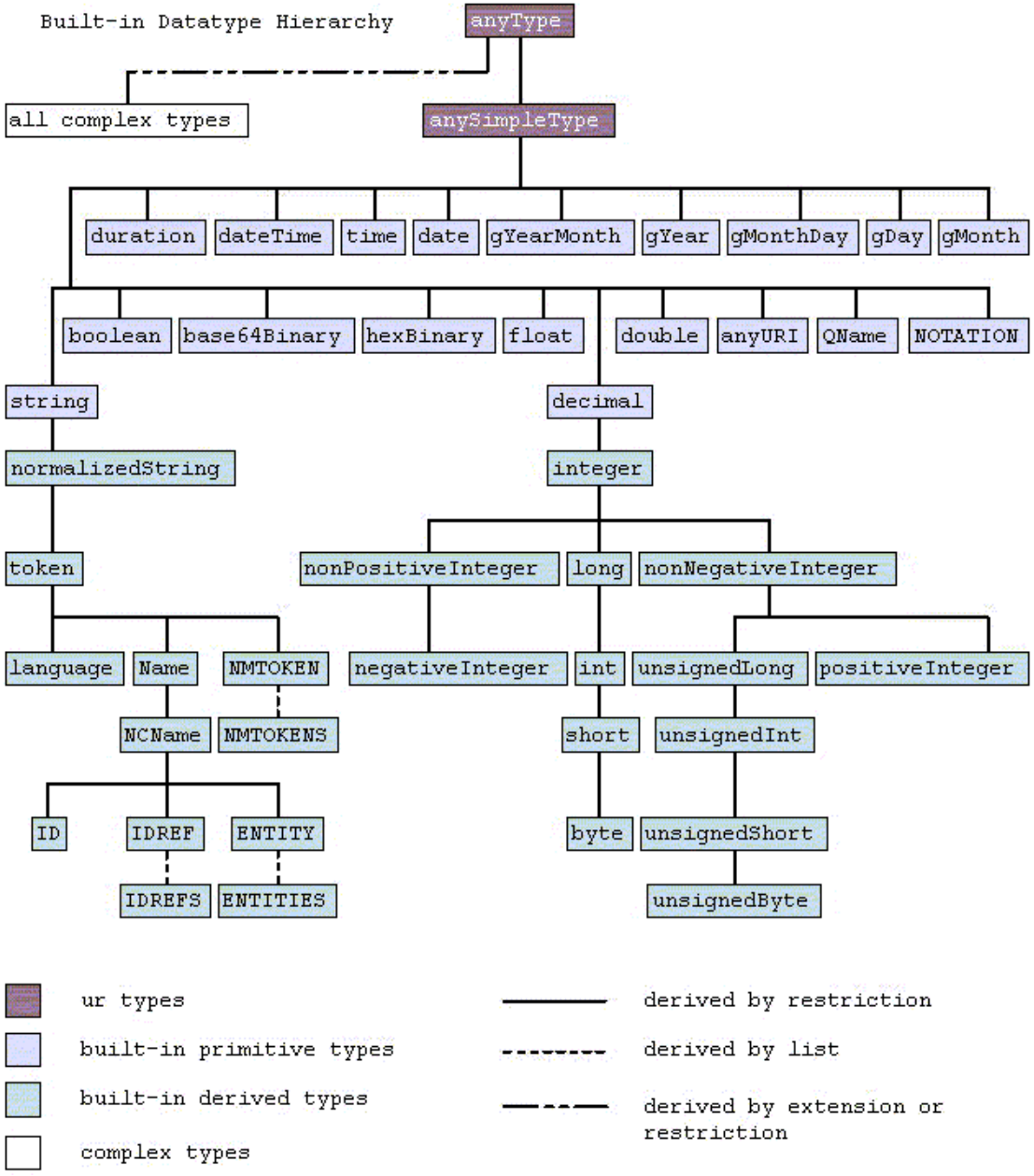
- Similar to element groups, grouping attributes

```
<xs:attributeGroup name="common.atts">
  <xs:attribute name="id" type="xs:ID" />
  <xs:attribute name="class" type="xs:string" />
</xs:attributeGroup>

<xs:element name="p">
  <xs:complexType>
    ... <!– Content model for p omitted -->
    <xs:attribute ref="align" />
    <xs:attributeGroup ref="common.atts" />
  </xs:complexType>
</xs:element>
```

Data Types

- Type Hierarchy of simple and complex data types
- Simple types:
  - string: unconstrained Unicode character data (#PCDATA, CDATA)
  - language: language code, e.g. de-AT
  - NMTOKEN, ID, IDREF, IDREFS, ENTITY, ENTITIES: like DTDs
  - float, double: “scientific” notation (mantissa, exponent)
    - represents closest IEEE-754 number
  - decimal: arbitrary number of digits, including arbitrarily-precise fraction
  - integer, long, int, short, …: derived from decimal, with range constraints
  - boolean: false/true, 0/1
  - …



More Simple Types

- anyURI: an arbitrary URL
- QName: prefix:localname (with the prefix being declared)
- duration: a time duration, in format PnYnMnDTnHnMnS
  - e.g. P1DT12H, PT140M, -P120D, PT9M8.1S
- dateTime, date, time, gYearMonth, gYear, gMonthDay, gDay, gMonth
  - representation of points in time, and parts thereof
    - full format -?CCYY-MM-DDThh:mm:ss(Z | [+-]hh:mm)?
  - e.g. 2002-07-05T10:36:59+02, 2002-07-05, 10:36:59+02:00, 2002-07, 2002, --07-05, ---05, --07--

User-Defined Types

- constraining built-in types
- resulting type is a subset of the original type

```
<xs:simpleType name="Name">
  <xs:restriction base="Base-Type”>
                    <-- facets -->
  </xs:restriction>
</xs:simpleType>
```

  defines a type *Name* as a subset of *Base-Type*
- Facets describe specific aspects of the type

User-Defined Types (2)

- Facets depend on the base type
- For string, ID, anyURI, QName, Lists, …
  - <xs:length value="..." />
  - <xs:minLength value="..." />
  - <xs:maxLength value="..." />
- For numeric types (float, double, decimal), duration, dateTime,…
  - <xs:minInclusive value="..." />
  - <xs:minExclusive value="..." />
  - <xs:maxInclusive value="..." />
  - <xs:maxExclusive value="..." />

User-Defined Types (3)

- For decimal and derived types
  - <xs:totalDigits value="..." />
  - <xs:fractionDigits value="..." />
- For all types except boolean: an enumeration of included values
  - <xs:enumeration value="..." />
- For all types: a regular expression matching all included values
  - <xs:pattern value="..." />

User-Defined Types (4)

- Examples

```
<xs:simpleType name="alterTyp">
  <xs:restriction base="xs:positiveInteger">
    <xs:maxInclusive value="150" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="alterTyp">
  <xs:restriction base="xs:decimal">
    <xs:minExclusive value="0" />
    <xs:maxInclusive value="150" />
  </xs:restriction>
</xs:simpleType>
```

User-Defined Types (5)

- User-defined types can be further derived

```
<xs:simpleType name="ageType">
 <xs:restriction base="xs:decimal">
        <xs:minExclusive value="0" />
        <xs:maxInclusive value="150" />
 </xs:restriction>
</xs:simpleType>
<xs:simpleType name="childAgeType">
 <xs:restriction base="ageType">
        <xs:maxExclusive value="14" />
 </xs:restriction>
</xs:simpleType>
```

User-Defined Types (6)

- Enumerations

```
<xs:attribute name="farbe">
  <xs:simpleType>
    <xs:restriction base="xs:token">
      <xs:enumeration value="russisch-grün" />
      <xs:enumeration value="schilf" />
      <xs:enumeration value="eierschale" />
      <xs:enumeration value="mauve" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

User-defined Types (7)

- Combination of multiple facets

```
<xs:element name="inventar-code">
  <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:minLength value="5" />
     <xs:maxLength value="10" />
     <xs:pattern value="[a-z]{2}-[0-9]+" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

User-defined Types (8)

- Multiple pattern facets

```
<xs:element name="ISBN">
  <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:pattern value="\d-\d{3}-\d{5}-\d" />
     <xs:pattern value="\d-\d{5}-\d{3}-\d" />
     <xs:pattern value="\d-\d{2}-\d{6}-\d" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```