



XML Schema 1.1

Sandy Gao, IBM



Agenda

- Overview
- Key improvements over Schema 1.0
 - Datatypes
 - Structures
- References



Overview

- Schema 1.1 is to address most common issues and requests for schema “users”
- Currently at Last Call for datatypes; “almost” at Last Call for structures
- Compatibility is a major goal
 - All 1.0 valid documents are 1.1 valid
 - With exceptions



Datatypes Improvements

- Alignment with XPath, XSLT, XQuery, etc. specifications
 - AnyAtomicType, YearMonthDuration, DayTimeDuration, date/time normalization, ...
- precisionDecimal
 - To support the anticipated new IEEE-754r type that carries precision
- Bug fixes and clarifications



Structures Improvements

- Co-constraints
 - Assertion
 - Conditional Type Assignment
- Versioning
 - Open content , <xs:all>, weakened wildcard, negative wildcard, conditional inclusion
- Complex type restriction
- Other improvements



Assertion

- Similar to Schematron
- Use XPath **2.0** to specify boolean expressions that have to be satisfied
- Part of complex types
 - Restriction relation is maintained

```
<xs:complexType name="arrayType">
  <xs:sequence>
    <xs:element name="entry" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="length" type="xs:int"/>
  <xs:assert test="@length eq fn:count(./entry)"/>
</xs:complexType>
```



Assertion (Cont')

- Most frequently requested schema feature
- Only the sub-tree rooted at the element being validated is accessible by XPath evaluation
 - Axis like “..” always returns the empty sequence
- Does not completely replace Schematron



Conditional Type Assignment

Schema:

```
<xs:element name="message" type="messageType">
  <xs:alternative test="@kind='string'" type="messageTypeString"/>
  <xs:alternative test="@kind='base64'" type="messageTypeBase64"/>
  <xs:alternative test="@kind='binary'" type="messageTypeBase64"/>
  <xs:alternative test="@kind='xml'" type="messageTypeXML"/>
  <xs:alternative test="@kind='XML'" type="messageTypeXML"/>
</xs:element>
```

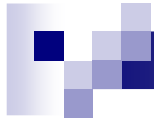
Instance:

```
<message kind="string">
  This is a text message.
</message>
<message kind="XML">
  <some>
    <xml>content</xml>
  </some>
</message>
```




CTA (Cont')

- Type used for validation is selected based on XPath 2.0 boolean expressions
- Only attributes on the current element are accessible by XPath evaluation



Versioning

- Allow earlier/general versions of schemas to leave extension points
- Allow later/specific versions of schemas to customize/extend



Open Content

- Allow any number of any elements to appear
 - Anywhere in the children list (“interleave”)
 - At the end of the children list (“suffix”)
- Leaves extension points in the content model

```
<xs:complexType name="name">
  <xs:openContent>
    <xs:any mode="interleave"/>
  </xs:openContent>
  <xs:sequence>
    <xs:element name="given" type="xs:string"/>
    <xs:element name="middle" type="xs:string" minOccurs="0"/>
    <xs:element name="family" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```



Open Content (Cont')

- “An element with `sml:ref=true` is an instance of the URI scheme if and only if there is one and only one `<sml:uri>` element in its [children].”

```
<xs:complexType>
  <xs:openContent>
    <xs:any processContents="skip" notQName="sml:uri" />
  </xs:openContent>
  <xs:sequence>
    <xs:element ref="sml:uri" />
  </xs:sequence>
  <xs:anyAttribute processContents="skip" />
</xs:complexType>
```



<xs:all> Improvements

- Allow maxOccurs to be > 1
- Allow wildcards
- Allow extensions to add new members



<xs:all> Improvements (Cont')

```
<xs:complexType name="computer"><xs:all>  
  <xs:element name="CPU"/> <!-- other computer parts -->  
  <xs:element name="speaker" minOccurs="0" maxOccurs="7"/>  
  <xs:any minOccurs="0" maxOccurs="unbounded"/>  
</xs:all></xs:complexType>
```

Extension:

```
<xs:extension base="computer"><xs:all>  
  <xs:element name="Joystick"/>  
</xs:all></xs:extension>
```

Restriction:

```
<xs:restriction base="computer"><xs:all>  
  <xs:element name="CPU"/><!-- other computer parts -->  
  <xs:any notQName="speaker" minOccurs="0"  
    maxOccurs="unbounded"/>  
</xs:all></xs:restriction>
```



Weakened Wildcard

```
<xs:complexType>  
  <xs:sequence>  
    <xs:element name="child" minOccurs="0"/>  
    <xs:any maxOccurs="unbounded" />  
  </xs:sequence>  
</xs:complexType>
```

- Violates “Unique Particle Attribution” (UPA) in schema 1.0
- Becomes valid in schema 1.1
- Elements take precedence over wildcards
- Element Declaration -> Wildcard -> Open Content



Negative Wildcard

- Any elements/attributes that
 - Are **not** in the listed namespaces
 - Do **not** match the listed QNames
 - Are **not** defined in this schema

```
<xs:sequence>  
  <xs:any notNamespace="##targetNamespace" />  
  <xs:any namespace="MyNS" notQName="my:e1 my:e2" />  
  <xs:any notQName="##defined" />  
</xs:sequence>
```




Conditional Inclusion

```
<xsd:element name="e" vc:minVersion="3.2">
  <!--* declaration suitable for 3.2
    * and later processors *-->
</xsd:element>
<xsd:element name="e" vc:minVersion="1.1"
  vc:maxVersion="3.1">
  <!--* declaration suitable for processors
    * supporting versions 1.1 through 3.1
    *-->
</xsd:element>
```

- A mechanism to version schema documents
- An element is not “included” if its min/maxVersion attributes don’t match the schema language version of the processor

Complex Type Restriction

- Extensional restriction (as opposed to schema 1.0's intentional restriction) for content models and attributes
- Identity Constraints can be named and reused

Base #1:	<code><xs:complexType name="base"> ...</code>
(a, (b, c))	<code> <xs:element name="e"> ...</code>
Restriction #1:	<code> <xs:key name="myKey"></code>
((a, b), c)	<code> ...</code>
Base #2:	<code><xs:complexType name="restriction"> ...</code>
(a b c)	<code> <xs:element name="e"> ...</code>
Derived #2:	<code> <xs:key ref="myKey"></code>
(b c a)	<code> ...</code>



Complex Type Restriction (Cont')

■ “targetNamespace” on local declarations

```
<xs:schema targetNamespace="base">
  ...
  <xs:complexType name="base">
    ...
    <xs:element name="e" form="qualified">
      ...

<xs:schema targetNamespace="restriction">
  ...
  <xs:complexType name="restriction">
    <xs:complexContent><xs:restriction base="base:base">
      ...
      <xs:element name="e" targetNamespace="base">
        ...
```



References

- Schema 1.1 Last Call working drafts:
 - Structures: <http://www.w3.org/TR/xmlschema11-1/>
 - Datatypes: <http://www.w3.org/TR/xmlschema11-2/>
 - Drafts that show differences from schema 1.0 are also available.
- Guide to versioning using schema 1.1 features:
<http://www.w3.org/TR/2007/WD-xmlschema-guide2versioning-20070720>