

XML Schema (eXtensible Markup Language Schema) une synthèse

Christian Soutou

<http://w3.univ-tlse2.fr/icare/soutou>



Plan

- Généralités
- Types de données
- Relations entre schémas
- Patrons (*Design patterns*)

- Généralités

<http://www.w3.org/XML/Schema>

XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content and semantics of XML documents in more detail.

Généralités

- Spécification approuvée en 2001 (univ. Édimbourg, Oracle, Microsoft et autres), révisions en 2004
- Un schéma XML est un document XML validant un autre document XML (structures, types et *namespaces*)
- Un document ayant une grammaire XML Schema est considéré comme une instance
- Les *namespaces* suivants seront utilisés (préfixe `xsd` aussi utilisé par `xs`)
 - `xmlns:xsd="http://www.w3.org/2001/XMLSchema"`
 - `xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`
- Alternative aux DTD, remplaçant à long terme

Généralités



- *XML Schema* est plus puissant que le mécanisme des DTD (types plus riches, cardinalités, etc.)
- Les outils transforment une DTD ou un document XML en un *XML Schema* (minimal)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<voiture >
  <immat>358-ALZ-31</immat>
  <prix>32190</prix>
</voiture>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema elementFormDefault="qualified"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="voiture">
    <xsd:complexType mixed="true">
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="immat"/>
        <xsd:element ref="prix"/>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="immat" type="xsd:string"/>
  <xsd:element name="prix" type="xsd:string"/>
</xsd:schema>
```

C. Soutou - 2008

5

Association à un doc. XML



- Sans prendre en compte les *namespaces* de la grammaire

```
<?xml version="1.0" encoding="UTF-8"?>
<compagnie
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="compagnie.xsd">
  <comp>AB</comp>
  ...
</compagnie>
```

- En prenant en compte des *namespaces* des grammaires

```
<?xml version="1.0" encoding="UTF-8"?>
<stylesheet xmlns="http://www.w3.org/1999/XSL/Transform"
  xmlns:html="http://www.w3.org/1999/xhtml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/1999/XSL/Transform
  http://www.w3.org/1999/XSL/Transform.xsd
  http://www.w3.org/1999/xhtml
  http://www.w3.org/1999/xhtml.xsd">
  ...
</stylesheet>
```

C. Soutou - 2008

6

- Types de données

<http://www.w3.org/TR/xmlschema-2/>

...

C. Soutou - 2008

7

Types de données



- Description d'un élément ou d'un attribut (en composition et en valeur)
- Toujours relié au *namespace* de la spécification *XML Schema* (préfixe *xsd* ou *xs*)
- Simple : type primitif (chaîne, numérique et date)


```
<xsd:element name="immat" type="xsd:string"/>
```
- Dérivé : construit en ajoutant des contraintes (de toute sorte, expressions régulières permises)
 - Restriction (sous-type)
 - Union (alternative entre plusieurs types)
 - Liste (ensemble de types semblables)
- Complexe (contenu d'un élément) : construit par composition d'éléments et/ou d'attributs

C. Soutou - 2008

8

Types dérivés par restriction (1)

xml10

- Numérique de 7 positions dont 2 décimales

```
<xs:element name="salary">
  <xs:simpleType>
    <xs:restriction base="xs:decimal">
      <xs:totalDigits value="7" />
      <xs:fractionDigits value="2" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- Chaîne de taille 8

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- Note comprise entre 0 et 20

```
<xs:element name="note">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="20"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Types dérivés par restriction (2)

xml10

- Restriction par énumération

```
<xs:element name="car" type="carType"/>
...
<xs:simpleType name="carType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Audi"/>
    <xs:enumeration value="Golf"/>
    <xs:enumeration value="BMW"/>
  </xs:restriction>
</xs:simpleType>
```

- Expression régulière

```
<xs:simpleType name="ISBN">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d-\d{4}-\d{4}-d"/>
  </xs:restriction>
</xs:simpleType>
```

Types dérivés (listes)

xml10

- Liste de chaînes de caractères

```
<xsd:simpleType name="listeChaine">
  <xsd:list itemType="xs:string"/>
</xsd:simpleType>
```

- Liste de références de bouquins

```
<xsd:simpleType name="listeLivre">
  <xsd:list itemType="ISBN"/>
</xsd:simpleType>
```

- Restriction d'une liste (chaînes de taille 8 max)

```
<xsd:simpleType name="listeChaineT8">
  <xsd:list>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:length value="8"/>
      </xs:restriction>
    </xs:simpleType>
  </xsd:list>
</xsd:simpleType>
```

Types dérivés (union)

xml10

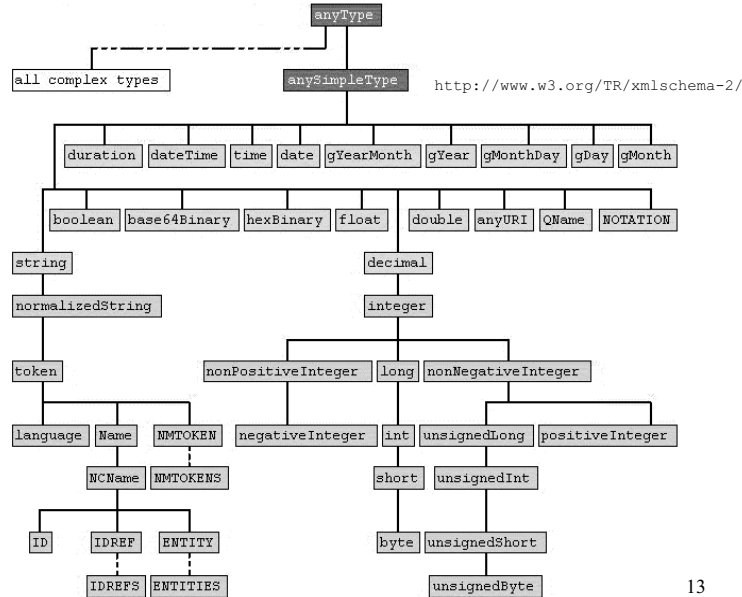
- La taille en nombre ou en nom

```
<xs:element name="jeans_size">
  <xs:simpleType>
    <xs:union memberTypes="sizebyno sizebyname" />
  </xs:simpleType>
</xs:element>

<xs:simpleType name="sizebyno">
  <xs:restriction base="xs:positiveInteger">
    <xs:minInclusive value="36"/>
    <xs:maxInclusive value="42"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="sizebyname">
  <xs:restriction base="xs:string">
    <xs:enumeration value="small"/>
    <xs:enumeration value="medium"/>
    <xs:enumeration value="large"/>
  </xs:restriction>
</xs:simpleType>
```

Types de données



13

Types complexes



- Concerne le contenu d'un élément

- Contre-exemples

```
<element1>Cours XML</element1>
</element2>
```

- Exemples

```
<element3 nom="Cours XML"/>
<element4 nom="Cours SGBD"/>Oracle 11g</element4>
<element5>
  <element6>Cours SGBD</element6>
  <element7>Oracle 11g</element7>
</element5>
```

C. Soutou - 2008

14

Connecteur « séquence »



- Donne l'ordre des sous-éléments dans un ordre précis
- Equivalent DTD (... , ... , ...)
- Exemple (déclaration locale et globale)

```
<xs:element name="etatCivil">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="prenom" type="xs:string"/>
      <xs:element name="nom" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:complexType name="etatCivilType">
  <xs:sequence>
    <xs:element name="prenom" type="xs:string"/>
    <xs:element name="nom" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="etatCivil" type="etatCivilType"/>
```

C. Soutou - 2008

15

Connecteur « choix »



- Permet une alternative aux sous-éléments
- Equivalent DTD (... | ... | ...)
- Exemple (déclaration locale et globale)

```
<xs:element name="login">
  <xs:complexType>
    <xs:choice>
      <xs:element name="pseudo" type="xs:string"/>
      <xs:element name="nom" type="xs:string"/>
    </xs:choice>
  </xs:complexType>
</xs:element>

<xs:complexType name="loginType">
  <xs:choice>
    <xs:element name="pseudo" type="xs:string"/>
    <xs:element name="nom" type="xs:string"/>
  </xs:choice>
</xs:complexType>
<xs:element name="login" type="loginType"/>
```

C. Soutou - 2008

16