

XML Schémas

Laurent Réveillère

Enseirb
Département Télécommunications

`Laurent.Reveillere@enseirb.fr`
`http://www.enseirb.fr/~reveille`

XML Schémas

- ◆ Alternative aux DTDs
- ◆ Langage de définition de schéma pour XML
- ◆ Permet de définir
 - Des types simples
 - Des types complexes
 - Structures d'éléments / attributs
- ◆ Recommandation W3C du 02/05/2001
- ◆ Syntaxe XML
 - Utilisation d'outils XML standards
 - » Éditeurs, ...

Avantages

- ◆ Support des types de données
- ◆ Permet de
 - Décrire plus finement la structure d'une famille de documents
 - Valider le contenu textuel
 - Échanger des données avec des bases de données
 - Définir des types de données simples, par restriction des types de bases (facettes)
 - Convertir des données de types différents
 - Composer des schémas par importation de fragments

Exemple

note.xml

```
<?xml version="1.0"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

- ◆ Élément note: élément *complexe*
 - Contient d'autres éléments
- ◆ Autres
 - éléments *simples*

DTD

note.dtd

```
<!ELEMENT note (to, from, heading, body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>
```

XML Schema

note.xsd

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"
          minOccurs='1' maxOccurs='1' />
        <xs:element name="from" type="xs:string" />
        <xs:element name="heading" type="xs:string" />
        <xs:element name="body" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Référence au schéma

note.xml

```
<?xml version="1.0"?>
<note
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://../note.xsd">

  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Définition d'éléments

- ◆ Définition d'éléments simples

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

- ◆ Définition d'éléments complexes

```
<xs:element name="name">  
  <xs:complexType>  
    element content  
  </xs:complexType>  
</xs:element>
```

Référence à des définitions

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="to"/>
        ...
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<xs:element name="to" type="xs:string" minOccurs='1'
maxOccurs='1' />
```

Éléments simples

- ◆ Éléments ne pouvant contenir que du texte
- ◆ Possibilité de restreindre la syntaxe du contenu textuel à l'aide de types simples

```
<xs:element name="lastname" type="xs:string"/>  
<xs:element name="age" type="xs:number"/>  
<xs:element name="dateborn" type="xs:date"/>
```

```
<lastname>Refsnes</lastname>  
<age>34</age>  
<dateborn>1988-03-27</dateborn>
```

Types simples

- ◆ xs:string
- ◆ xs:decimal
- ◆ xs:integer
- ◆ xs:boolean
- ◆ xs:date
- ◆ xs:time
- ◆ ...

Restriction des types simples

- ◆ Définition de facettes
- ◆ Restriction sur un domaine de valeurs
- ◆ Exemple
 - Age : entier compris entre 0 et 120

```
<xs:element name="age">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="120"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Restriction des types simples

- ◆ Définition d'ensembles en compréhension

```
<xs:element name="monnaie" type="devise"/>

<xs:simpleType name="devise">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EUR"/>
    <xs:enumeration value="USD"/>
    <xs:enumeration value="YEN"/>
  </xs:restriction>
</xs:simpleType>
```

Exemples

◆ Montant monétaire

```
<xs:simpleType name='montantMonétaire'  
base='xs:decimal'>  
  <xs:scale value='2' />  
  <xs:minInclusive value='0' />  
</xs:simpleType>
```

- Remarque: la facette `scale` définit le nombre de chiffre après la virgule

◆ Code postal

```
<xs:simpleType name='codePostal' base='xs:string'>  
  <xs:pattern value='\d{5}' />  
</xs:simpleType>
```

Types complexes

- ◆ Définis à l'aide d'un élément `complexType`
 - `elementOnly` contenu composé d'un ou plusieurs éléments
 - » valeur par défaut
 - `textOnly` contenu de type string
 - `empty` contenu vide
 - `mixed` contenu mixte

Exemple

◆ Définition d'un élément prix

```
<xs:element name='prix'>
  <xs:complexType content='empty'>
    <xs:attribute name='montant' type='montantMonétaire' />
    <xs:attribute name='devise' type='devise' />
  </xs:complexType>
</xs:element>
```

◆ Instance

```
<prix devise='EUR' montant='4.52' />
```

Contenu d'un type complexe

- ◆ Choice
 - choix parmi plusieurs
 - `minOccurs` et `maxOccurs` portent sur l'ensemble du choix
- ◆ Sequence
 - chaque élément doit apparaître dans l'ordre spécifié
 - `minOccurs` et `maxOccurs` portent sur l'ensemble de la séquence
- ◆ All
 - 0 ou 1 fois, ordre quelconque

Groupes nommés

- ◆ Commodité d'écriture permettant de nommer un ensemble d'éléments pour y faire référence ultérieurement
- ◆ Exemple de déclaration
 - `<xs:group name="foo">`
 - `<xs:element .../>`
 - `</xs:group>`
- ◆ Exemple d'utilisation
 - `<xs:group ref="foo">`