XML et Architectures Logicielles Distribuées

Baabda :
M. E. ABOU ZEID
M. E. MATTA

Zahlé :
Mlle. C. SAAD

Mejdelaya:
M. R. ABI NADER

PLAN DU COURS

- Chapitre 1: Le Langage XML
- Chapitre 2: XPath et APIs XML
- Chapitre 3: Les services web
- Chapitre 4: Les architectures distribuées
- **Chapitre 5: Le Schéma XML**

CHAPITRE 5

Le Schéma XML

PLAN DU CHAPITRE

- Introduction
- **X**ML **S**chema **D**efinition (XSD)
  - Avantages
  - Structure de base
  - Les types de données
- Les types de données simples
- Les types de données complexes

Introduction

Introduction

- **XML** est un **format** permettant de **stocker** et **communiquer** les données entre des systèmes distribués indépendamment de la plateforme

- Besoin de **contrôler la structure** du document XML
  - **Définir** cette structure et **valider** le document XML par rapport à cette structure (« document valide »)

  - Surtout lorsque des groupes indépendants de développeurs veulent s'entendre sur un standard pour échanger les données

  - **Solutions Possibles:**
    - DTD
    - XSD

Introduction

- **DTD**
  - **D**éfinition de **T**ype de **D**ocument
  - **Composants d'une DTD:**
    - Éléments
    - Attributs
    - Entités
    - CDATA
    - Commentaires

Introduction

- **DTD**
  - **Limitations**
    - Ne sont pas au format XML
      - Langage spécifique
      - Besoin d'un outil spécial pour manipuler un tel fichier
    - Le typage des données est limité
  - **Propositions d'alternatives**
    - Le schéma XML ou **X**ML **S**chema **D**efinition (XSD)

XML Schema Definition (XSD)

Avantages

- **Les XSD sont au format XML**
  - Les schémas créés avec XSD sont codés dans un fichier *.xsd*
- **Typage et structuration des données**
  - attributs simples
  - éléments simples et complexes
- **Définir des contraintes**
  - Existence, obligatoire, optionnel, domaines, cardinalités, références
- **Indicateurs d'occurrences des éléments peuvent être tout nombre non négatif**
  - Dans une DTD, on est limité à 0, 1 ou un nombre infini.
- **L'extensibilité**
  - schémas facilement concevables par modules

Structure de base

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <!--déclarations d'éléments, d'attributs et de types-->
</xsd:schema>
```

Les types de données

- **Les éléments** peuvent être **de type..**
  - **Complexe**: éléments pouvant contenir des éléments enfants ou posséder des attributs
  - **Simple:** dans le cas contraire

- **Les attributs**:
  - Ne peuvent être que **de type simple**
  - Ne peut apparaitre que **dans des éléments de type complexe**

Les types de données

- **Exemple d'un élément simple:**

```
<xsd:schema>
        <xsd:element name="nom" type="xsd:string" />
        <xsd:element name="dateDeNaissance" type="xsd:date" />
</xsd:schema>
```

- **Exemple d'un Attribut simple:**
  - <xsd:attribute name= "datenaissance" type= "xsd:date"/>

Les types de données simples

Les types de données simples

Type	Déclaration
Type chaîne	xsd : string
Type booléen	xsd : boolean
Type numérique	> xsd : integer > xsd : decimal > xsd : float > xsd : double
Type date/ heure	> xsd : time > xsd : date > xsd : month > xsd : year > … > Etc.
Type personnalisé	Exemple : limiter la plage numérique d'un type numérique ou limiter une chaine à un ensemble de chaines possibles, etc.

Les types de données simples

- **Les valeurs par défaut:**
  - Doivent être conformes au type déclaré
  - <xsd:element name= "solde" type= "xsd:decimal" default ="0.0" />
  - <xsd:attribute name= "expire" type= "xsd:boolean" default ="false" />

- **Les valeurs fixes:**
  - La seule valeur que peut prendre l'élément/l'attribut
  - <xsd:element name= "pi" type= "xsd:decimal" fixed ="3.14" />
  - <xsd:attribute name= "titre" type= "xsd:string" fixed ="Mr." />

Les types de données simples

- **Les valeurs requises/optionnelles:**
  - Pour les attributs; ***optional*** *est la valeur par défaut*
  - <xsd:attribute name="maj" type="xsd:date" use="optional" default="2003-10-11"/>
  - <xsd:attribute name="unite" type="xsd:string" use="required"/>

- **Les types personnalisés :**
  - création par dérivation des types simples
  - Déclaration **<xsd : simpletype>**
  - On définit une contrainte avec la déclaration **<xsd : restriction>**
    - Une restriction possède un attribut nommé **Base** qui spécifie le type de base à personnaliser.

Les types de données simples

- **Les types personnalisés :**
  - **Les listes :**
    - Des suites de types simples, autorisant plusieurs valeurs pour un élément
    - **Exemple :**

```
<xsd:element name="pluviometrie">
    <xsd:simpletype>
        <xsd:list base="xsd:decimal">
            <xsd:length value="12"/>
        </xsd:list>
    </xsd:simpletype>
</xsd:element>
```

    - **Utilisation:**

```
<pluviometrie>1.25 2.0 3.0 4.25 3.75 1.5 0.25 0.75 1.25 1.75 2.0 2.25</pluviometrie>
```

Les types de données simples

- **Les types personnalisés :**
  - **Les motifs (pattern)**
    - Une expression régulière
    - L'élément <xsd:pattern> est utilisé

- ., n'importe quel caractère ;
- \d, n'importe quel chiffre ;
- \D, n'importe quel caractère non chiffre ;
- \s, n'importe quel espace vierge ;
- \S, n'importe quel caractère non espace ;
- x ?, Un ou zéro x ;
- x+, un ou plusieurs x ;

- x*, un nombre quelconque de x ;
- (xy), groupe de x et y ensemble ;
- x|y, x ou y ;
- [xyz], un parmi x, y ou z ;
- [x-y], dans la plage x à y ;
- x{n}, n x à la suite ;
- x{n,m}, au moins n x mais pas plus de m.

Les types de données simples

- **Les types personnalisés :**
  - **Les motifs (pattern)**
    - **Exemple :** un numéro de téléphone américain se présente sous la forme x-xxx-xxx-xxxx

```
<xsd:element name="numphone">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
        <xsd:pattern value="\d{1}-\d{3}-\d{3}-\d{4}"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
```

Exemple d'utilisation en XML:

```
<numphone>1-234-567-8901</numphone>
```

Les types de données simples

- **Les types personnalisés :**
  - **Les types énumérés :**
    - pour identifier les valeurs possibles
    - **Exemple :**

```
<xsd:element name="equipe">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Nashville Predators"/>
            <xsd:enumeration value="Columbus Blue Jackets"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
```

<equipe> pourra avoir l'une des 2 valeurs énumérées seulement.

Les types de données simples

- **Les types personnalisés :**
  - **Autres:**
    - Il est aussi possible de limiter la plage numérique :

```
<xsd:element name = "note">
   <xsd:simpleType>
      <xsd:restriction base= "xsd:integer">
         <xsd:minInclusive value = "1" />
         <xsd:maxInclusive value = "10" />
      </xsd:restriction>
   </xsd :simpleType>
</xsd:element>
```

```
<xsd:element name="motdepasse">
  <xsd:simpleType>
     <xsd:restriction base="xsd:string">
        <xsd:minLength value="8" />
        <xsd:maxLength value="12" />
     </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

Les types de données simples

- **Les types personnalisés :**
  - **Précision des nombres :**
    - **xsd:totalDigits** → spécifier le nombre total de chiffres y compris ceux après la virgule
    - **xsd:fractionDigits** → spécifier le nombre total de chiffres après la virgule

```
<xsd:element name = "solde">
   <xsd:simpleType>
      <xsd:restriction base= "xsd:decimal">
         <xsd:totalDigits value = “8"/>
         <xsd:fractionDigits value = “6"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>
```

Les types de données complexes

Les types de données complexes

- **SimpleType:**
  - pas d'attributs, pas d'éléments.

- **ComplexType:**
  - 2 formes possibles:
    - **SimpleContent:** peut posséder des attributs et du character data
    - **ComplexContent:** peut posséder des attributs et des éléments enfants

*N.B: La syntaxe par défaut pour les types complexes est **ComplexContent**.*

Les types de données complexes

- On crée un élément complexe à l'aide de l'élément : **xsd:complexType** permettant de déclarer des
  - Éléments vides
  - Éléments « élément uniquement »
  - Séquences d'éléments (*xsd:sequence*)
  - Des choix (*xsd:choice*)
  - Contraintes d'occurrences (*xsd:all*)
  - Éléments mixtes
  - Etc.

Les types de données complexes

- **Elément vide:**
  - Un élément vide ne contient ni texte, ni élément enfant, mais peut posséder des attributs.
  - **Exemple:**

```
<xsd:element name=«departement»>
    <xsd:complexType>
        <xsd:attribute name=«nom» type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>
```

```
<departement nom=“dept_10”/>
```

Les types de données complexes

- **Elément « élément uniquement »:**
  - Ne contient que des éléments enfants, mais pas de texte.
  - Il peut bien sûr posséder des attributs.

```
<xsd:complexType name="automobileType">
    <xsd:attribute name="code" type="xsd:string/>
    <xsd:attribute name="annee" type="xsd:year/>
</xsd:complexType>

<xsd:element name="biens">
   <xsd:complexType>
         <xsd:element name="automobile" type="automobileType"/>
   </xsd:complexType>
</xsd:element>
```

Les types de données complexes

- **Elément xsd:sequence:**
  - Il indique que les éléments enfants doivent apparaître **0 ou plusieurs fois mais l'ordre doit être respecté**

```
<xsd:complexType>
 <xsd:sequence>
  <xsd:element name="nom" type="xsd:string" />
  <xsd:element name="prénom" type="xsd:string" />
  <xsd:element name="dateDeNaissance" type="xsd:date" />
  <xsd:element name="adresse" type="xsd:string" />
 </xsd:sequence>
</xsd:complexType>
```

Les types de données complexes

- **Élément <xsd:choice>**

```
<xsd:complexType name="typePersonne">
 <xsd:sequence>
  <xsd:element name="nom" type="xsd:string" />
  <xsd:choice>
   <xsd:element name="adresse" type="xsd:string" />
   <xsd:element name="adresseElec" type="xsd:string" />
  </xsd:choice>
 </xsd:sequence>
</xsd:complexType>
```

Les types de données complexes

- **L'élément xsd:all**
  - Il indique que les éléments enfants doivent apparaître **0 ou 1 fois, dans n'importe quel ordre**

```
<xsd:complexType name="typePersonne">
 <xsd:all>
  <xsd:element name="nom" type="xsd:string" />
  <xsd:element name="prénom" type="xsd:string" />
  <xsd:element name="dateDeNaissance" type="xsd:date" />
  <xsd:element name="adresse" type="xsd:string" />
  <xsd:element name="adresseElec" type="xsd:string" />
  <xsd:element name="téléphone" type="numéroDeTéléphone" />
 </xsd:all>
</xsd:complexType>
```

Les types de données complexes

- **Les attributs**

```
<xsd:complexType name="typePersonne">
        <xsd:sequence>
          <xsd:element name="dateDeNaissance" type="xsd:date" />
          <xsd:element name="adresse" type="xsd:string" />
        </xsd:sequence>
        <xsd:attribute name="nom" type= "xsd:string" />
        <xsd:attribute name="prénom" type= "xsd:string" />
</xsd:complexType>
```

Les types de données complexes

- **Les indicateurs d'occurrences**
  - Tout nombre entier non négatif

```
<xsd:element name="livre">
<xsd:complexType>
    <xsd:sequence>
     <xsd:element name="titre" type="xsd:string" minOccurs="1"  maxOccurs="3"/>
     <xsd:element name="auteur" type="xsd:string" minOccurs="1" maxOccurs="unbounded"/>
    <xsd:element name="editeur" type="xsd:string“ minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
```

*NB: La valeur par défaut de **minOccurs** et **maxOccurs** est**1***

Questions?