

# TP 2 schema XML

## Notions fondamentales de XML Schema

*Objectif : vous introduire aux principes de base de XML schéma, créer les schémas de documents XML.*

23 septembre 2013  
Créé par : Neila Ben Lakhal

# TP 2 schema XML

## Notions fondamentales de XML Schema

### Exercice 1. Validation d'un document XML par rapport à un schéma

```
<?xml version="1.0" encoding="utf-8" ?>
<Club clubId="ACM">
  <Member id="_12345" clubs="ACM">
    <Name>
      <FirstName>WEB</FirstName>
      <MiddleInitial>H</MiddleInitial>
      <LastName>DO</LastName>
    </Name>
  </Member>
</Club>
```

- Enregistrer le document xml ci-dessous sous le nom **club.xml** et vérifiez qu'il est conforme.
- Enregistrer le schéma suivant et reliez-le au fichier xml **club.xml** pour qu'il soit valide par rapport à ce schéma :

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- Define the elements that make up a Member's Name -->
  <!-- Define a Member -->
  <xs:element name="Member">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Name" type="NameType" />
      </xs:sequence>
      <xs:attribute name="id" type="xs:ID" use="required" />
      <xs:attribute name="clubs" type="xs:IDREFS" use="optional" />
    </xs:complexType>
  </xs:element>
  <!-- Define the Club root element -->
  <xs:element name="Club">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Member" />
      </xs:sequence>
      <xs:attribute name="clubId" type="xs:ID" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:complexType name="NameType" >
    <xs:sequence>
      <xs:element name="FirstName" type="xs:string" />
      <xs:element name="MiddleInitial" type="xs:string" />
      <xs:element name="LastName" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

- Faites les modifications suivantes au schéma et au document xml :
  - **MiddleInitial** d'un **Membre** est optionnel
  - Ajouter un élément **Email** optionnel après l'élément **Name**

- Faire en sorte qu'un club peut contenir zéro ou un nombre illimité de membre (utiliser unbounded)

## Exercice 2. Réalisation d'un schéma

- Soit un document XML contenant un nombre indéterminé d'éléments sous la forme :

```
<contact titre="..." techno="...">
<nom>...</nom>
<prenom>...</prenom>
<telephone> ...</telephone>
<email>...</email>
<email>...</email>
...
</contact>
```

- L'élément telephone et l'attribut techno sont en option. Les textes seront des chaînes simples xs:string.
- Vous utiliserez les types complexes numerosType et contactType pour construire un schéma nommé annuaire.xsd.

## Exercice 3. Construction de types simples

- Créez un schéma annuaire2.xsd à partir du schéma de l'exercice précédent.
  - Définissez et utilisez les types simples suivants :
    - technoType : énumération dont les valeurs possibles sont XML, Java, Autre.
    - telType : liste de 5 entiers (attention : créez d'abord un type pour la liste d'entiers).

Validez ce nouveau schéma sur un document de votre conception.

## Exercice 4. Réécriture de DTDs en Schémas XML

Proposez un Schéma XML pour le document XML suivant donné avec sa DTD :

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE gdt SYSTEM "wg.dtd">
<gdt month="February" year="2004">
  <expose day="5">
    <speaker>Nicolas Baudru</speaker>
    <title>Netcharts et HMSC</title>
    <time>12h45</time>
    <salle>102</salle>
  </expose>
  <expose day="12">
    <comment>Exposé proposé par François Denis, équipe BDA</comment>
    <speaker>Liva Ralaivola</speaker>
    <title>apprentissage statistique, noyaux et applications à la
    bio-informatique</title>
    <time>14h</time>
    <salle>104</salle>
  </expose>
  <expose day="19">
```

```

    <comment>pas de groupe de travail</comment>
  </expose>

  <expose day="26">
    <speaker>Pedro d'Argenio</speaker>
    <title>Secure Information Flow by Self-Composition</title>
    <time>12h45</time>
    <salle>102</salle>
  </expose>
</gdt>

```

```

<!ELEMENT gdt (expose*)>
<!ATTLIST gdt
  month CDATA #REQUIRED
  year CDATA #REQUIRED>
<!ELEMENT expose (comment?, (speaker, title, time, salle)?)>
<!ATTLIST expose day CDATA #REQUIRED>
<!ELEMENT comment (#PCDATA)>
<!ELEMENT speaker (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT time (#PCDATA)>
<!ELEMENT salle (#PCDATA)>

```

## Exercice 5. Déclarations d'éléments

Rédiger un Schema XML pour une bibliographie. Cette bibliographie :

- contient des livres et des articles ;
- les informations nécessaires pour un livre (élément `livre`) sont :
  - son titre général (élément `titre`) ;
  - les noms des auteurs (éléments `auteur`) ;
  - ses tomes (élément `tomes`) et pour chaque tome (éléments `tome`), leur nombre de pages (élément `pages`) ;
  - des informations générales sur son édition (élément `infosEdition`) comme par exemple le nom de l'éditeur (élément `editeur`), le lieu d'édition (élément `lieuEdition`), le lieu d'impression (élément `lieuImpression`), son numéro ISBN (élément `ISBN`) ;
- les informations nécessaires pour un article (élément `article`) sont :
  - son titre (élément `titre`) ;
  - les noms des auteurs (éléments `auteur`) ;
  - ses références de publication (élément `infosPublication`) : nom du journal (élément `nomJournal`), numéro des pages (élément `pages`), année de publication (élément `anneePublication`) et numéro du journal (élément `numeroJournal`)
- on réservera aussi un champ optionnel, pour chaque livre et chaque article, pour un avis (élément `avis`) personnel.

Tester ce Schema XML avec un fichier XML que l'on et validera.

## Exercice 6. Déclarations d'attributs

---

Modifier le Schéma précédent. On ne déclarera, pour le moment, que des types de chaînes de caractères.

- ... en ajoutant un attribut optionnel `soustitre` à l'élément `titre` ;
- ... en faisant de l'élément `tome` un élément vide et en lui ajoutant un attribut requis `nbPages` et un attribut optionnel `sousTitre` ;
- ... en faisant de l'élément `nomJournal` un attribut de l'élément `infosPublication` et en lui donnant comme valeur par défaut `Feuille de Chou` ;

Utiliser ce Schéma pour créer un fichier XML valide.