

1^{ère} partie : XML SCHEMA

- ❑ Structure globale d'un schéma XML
- ❑ Description des éléments
- ❑ Déclaration d'un élément simple
- ❑ Déclaration d'un élément complexe
- ❑ Déclaration d'un élément quelconque
- ❑ Validation d'un document XML par un Shéma XML

XML SCHEMA

Définition:

XML Schéma est un langage de définition de la structure et du typage des documents XML.

XML SCHEMA

□ La norme (recommandation depuis le 2 mai 2001 et une révision en 2004) :

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

□ Définition :

<http://www.w3.org/TR/xmlschema-0/>

□ Structures :

<http://www.w3.org/TR/xmlschema-1/>

□ Types :

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

XML SCHEMA

- ❑ Les *schémas XML* permettent comme les *DTD* de définir des modèles de documents (éléments et attributs).

- ❑ Les schémas ont été introduits pour combler les lacunes des DTD.

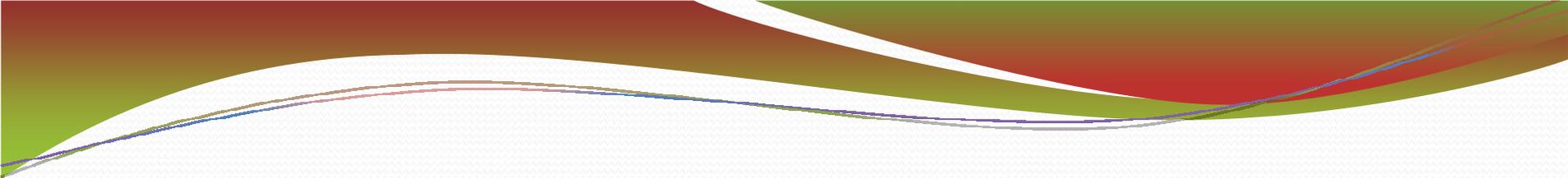
XML SCHEMA

❑ Inconvénients des DTD

- Syntaxe non XML
- Manque de types
- Pas de gestion des espaces de noms.

❑ Apports des schémas XML

- Syntaxe XML
- Nombreux types de données prédéfinis (booléens, intervalles de temps, nombres, dates, ...)
- Possibilité de définir de nouveaux types
- Prise en compte des espaces de noms.
- Les indicateurs d'occurrences des éléments peuvent être tout nombre non négatif.



Structure globale d'un schéma

Un schéma XML se compose essentiellement de déclarations d'éléments et d'attributs et de définitions de types.

Chaque élément est déclaré avec un type qui peut être:

- soit un type prédéfini,
- soit un nouveau type défini dans le schéma.

Le type spécifie quels sont les contenus valides de l'élément ainsi que ses attributs.

Structure globale d'un schéma

Un nouveau type est obtenu:

- soit par construction, c'est-à-dire une description explicite des contenus qu'il autorise,
- soit par dérivation, c'est-à-dire modification d'un autre type.

Structure de base

- Comme tout document XML, un Schéma XML commence par **un prologue**, et a un élément **racine**.

```
<?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 ici -->  
</xsd:schema>
```

- L'élément racine est l'élément **xsd:schema**.
- Tout élément du langage XML Schema doit être préfixé par **xsd:**.
- Les éléments d'un Schéma XML peuvent être de *type simple* ou *complexe*.

Description des éléments

Un élément, dans un schéma, se déclare avec la balise **<xsd:element>**

L'attribut **name** de **xsd:element** contient le nom de l'élément.

Exemple: un élément *livre*

```
<xsd:element name="livre">  
    <!-- déclarations de types ici -->  
</xsd:element>
```

Les éléments composés d'attributs et de sous éléments devront inclure une balise **xsd:complexType**

Déclaration d'éléments simples

- Un **élément simple** est un élément qui ne peut contenir qu'une **chaîne de caractères**.
(il ne peut pas contenir de sous éléments, ni d'attributs)
- Cette chaîne de caractères peut correspondre à :
 - o Des types prédéfinis : *xsd:string*, *xsd:decimal*, *xsd:integer*, *xsd:boolean*, *xsd:date*...
 - o Des types de données propres à l'utilisateur.
- Dans un schéma XML, un élément simple se déclare avec la balise **xsd:element**.

Déclaration d'éléments simples

Exemple1: note.xsd

```
<?xml version="1.0"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="note" type="xsd:float" />
</xsd:schema>
```

L'élément **note** est du type **xsd:float** qui est un type simple prédéfini de XML Schema.

Déclaration d'éléments simples

Exemple1: note.xml

```
<?xml version="1.0"?>
<note xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
      xsi:noNamespaceSchemaLocation="note.xsd">
  16.5
</note>
```

On fait référence au schéma dans le document XML en utilisant l'attribut **noNamespaceSchemaLocation**

Déclaration d'éléments simples, exemple 2

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="contacts" type="typeContacts" />
  <xsd:element name="remarque" type="xsd:string" />
  <!-- déclarations de types simples ou complexes ici -->
</xsd:schema>
```

Déclare deux éléments : un élément **contacts** et un élément **remarque**.

- L'élément **contacts** du type **typeContacts**, qui est un type complexe défini par l'utilisateur.
- L'élément **remarque** est du type **xsd:string** qui est un type simple prédéfini de XML Schema.

Cas particulier d'un élément simple

Exemple: élément vide sans attribut

Document XML Schéma: élément sans attributs

```
<xsd:element name="vide" type="xsd:anyType"/>
```

Document XML:

```
<vide/>
```

Déclaration d'éléments complexes

- Un **élément complexe** est un élément qui peut contenir d'autres éléments ou bien des attributs.
- Il existe 4 types d'éléments complexes :
 - 1- Les éléments vides avec attributs.
 - 2- Les éléments qui contiennent d'autres éléments.
 - 3- Les éléments (avec des attributs) qui contiennent uniquement du texte.
 - 4- Les éléments qui contiennent du texte et d'autres éléments.
- Dans un schéma XML, un élément complexe se déclare en utilisant la balise **xsd:complexType**.

```
<xsd:element name="personne">  
  <xsd:complexType>  
  .....  
</xsd:complexType>
```

1- Déclaration d'éléments complexes: éléments vides

- Les **éléments vides** ne contiennent pas de texte, ni d'autres éléments.
- Un **élément vide** peut contenir des **attributs**.
- La définition d'un attribut se fait grâce à l'élément **xsd:attribute**

Exemple 1: élément vide avec attributs

```
<xsd:element name="personne" type="personnelInfo" />
  <xsd:complexType name="personnelInfo">
    <xsd:attribute name="nom" type="xsd:string" />
    <xsd:attribute name="prenom" type="xsd:string" />
  </xsd:complexType>
```

1- Déclaration d'éléments complexes: éléments vides

Exemple 2: élément vide avec attributs

```

```

Un schema XML validant :

```
<xsd:element name="img">  
  <xsd:complexType>  
    <xsd:attribute name="src" type="xsd:anyURI" use="required"/>  
  </xsd:complexType>  
</xsd:element>
```

1- Déclaration d'éléments complexes: éléments vides

- Un attribut est optionnel par défaut.
- Pour que l'attribut soit obligatoire : **use="required"**

```
<xsd:element name="personne" type="personnelInfo" />
  <xsd:complexType name="personnelInfo">
    <xsd:attribute name="nom" type="xsd:string" use="required" />
    <xsd:attribute name="prenom" type="xsd:string" />
  </xsd:complexType>
```

- **nom** est un attribut obligatoire
- **prenom** est un attribut optionnel

2- Description des éléments ayant des sous-éléments

2.1- Une séquence ordonnée d'éléments:

- Dans la balise **xsd:complexType**, la liste des sous-éléments est décrite à l'intérieur d'une balise **xsd:sequence**.
- La balise **xsd:sequence** définit une liste ordonnée de sous-éléments.
- A l'intérieur de la balise **xsd:sequence**, chaque sous-élément est décrit par une balise **xsd:element** .

Exemple: auteur.xsd

```
<xsd:element name="auteur">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="nom" type="xsd:string" />
      <xsd:element name="prenom" type="xsd:string" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

2- Description des éléments ayant des sous-éléments

2.2- Une alternative d'éléments: **xsd:choice**

Exemple: soit l'adresse d'une personne ou son adresse électronique.

```
<xsd:complexType name="typePersonne">
  <xsd:sequence>
    <xsd:element name="nom" type="xsd:string" />
    <xsd:element name="prenom" type="xsd:string" />
    <xsd:element name="dateDeNaissance" type="xsd:date" />
    <xsd:choice>
      <xsd:element name="adresse" type="xsd:string" />
      <xsd:element name="adresseElectronique" type="xsd:string" />
    </xsd:choice>
  </xsd:sequence>
  <xsd:element name="telephone" type="numeroDeTelephone" />
</xsd:complexType>
```

2- Description des éléments ayant des sous-éléments

2.2- Une alternative d'éléments: **xsd:choice**

Exemple 2: l'équivalent de :

```
<!ELEMENT Librairie(Livre|Magazine|BD)*>
```

```
. <xsd:element name="Librairie">  
  <xsd:complexType>  
    <xsd:choice minOccurs="0" maxOccurs="unbounded">  
      <xsd:element name="Livre" type="xsd:string" />  
      <xsd:element name="Magazine" type="xsd:string" />  
      <xsd:element name="BD" type="xsd:string" />  
    </xsd:choice>  
  </xsd:complexType>  
</xsd:element>
```

2- Description des éléments ayant des sous-éléments

2.3- Une séquence non ordonnée d'éléments: **xsd:all**

Exemple: l'ordre n'est pas important du nom et du prénom.

```
<xsd:element name="personne">
  <xsd:complexType >
    <xsd:all>
      <xsd:element name="nom" type="xsd:string" />
      <xsd:element name="prenom" type="xsd:string" />
    </xsd:all>
  </xsd:complexType>
</xsd:element name>
```

3- Les éléments ayant que du texte et des attributs

Exemple1: Element à contenu simple

```
<xsd:element name="devise type="typeDevise" />
  <xsd:complexType name="typeDevise">
    <xsd:simpleContent>
      <xsd:extension base="xsd:float" >
        <xsd:attribute name="unite" type="xsd:string" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
```

L'élément **devise** peut prendre des valeurs réelles et a un attribut **unite**. Il s'agit d'un contenu simple (pas de sous éléments), extension du type float.

Exemple: <devise unite='dirham'>15.5</devise>

3- Les éléments ayant que du texte et des attributs

Exemple2:

```
<xsd:element name="personne" type="personnelInfo" />
<xsd:complexType name="personnelInfo">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string" >
      <xsd:attribute name="pays" type="xsd:string" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

Dans cet exemple, l'élément *personne* contient du **texte** et un **attribut** nommé *pays*.

Exemple: <personne pays="maroc">alaoui</personne>

4- Les éléments qui contiennent du texte et d'autres éléments (élément mixte)

```
<xsd:element name="personne" type="personnelInfo" />
<xsd:complexType name="personnelInfo" mixed="true">
  <xsd:sequence>
    <xsd:element name="nom" type="xsd:string" />
    <xsd:element name="prenom" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
```

Exemple XML correspondant :

```
<personne>
  Je suis <nom>Rachad</nom> <prenom>Ali</prenom>.
</personne>
```

Déclaration d'éléments quelconques

- Comme pour les DTD, il peut y avoir **des éléments quelconques**:

```
<xsd:element name="personne" type="personneInfo" >
  <xsd:complexType name="personneInfo">
    <xsd:sequence>
      <xsd:element name="nom" type="xsd:string" />
      <xsd:element name="prenom" type="xsd:string" />
      <xsd:any />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

- Il est possible de faire de même pour les attributs:

```
<xsd:element name="personne" type="personnelInfo" />
<xsd:complexType name="personnelInfo">
  <xsd:anyAttribute />
</xsd:complexType>
```

Comment lier un fichier XML à un schéma ?

Un schéma est un document XML, on trouve dans son élément racine l'attribut

`xmlns:xsd="http://www.w3.org/2001/XMLSchema"`

Cela signifie que dans le document, tous les éléments commençant par *`xsd`* sont référencés à cette URL.

Si on a déposé un schéma à l'adresse:

`http://www.fsr.ac.ma/collection_schemas/biblio`

on peut l'appeler par :

`<xsd:biblio`

`xmlns="http://www.fsr.ac.ma/collection_schemas/biblio">`

Comment lier un fichier XML à un schéma ?

Dans le cas d'une référence locale (*correspondant à une DTD de type SYSTEM*), on fait référence au schéma dans le document XML en utilisant l'attribut **xmlns:noNamespaceSchemaLocation**,

par:

```
<biblio xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
        xsi:noNamespaceSchemaLocation="lien_vers_le_schema">
```

Par exemple,

```
<biblio xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
        xsi:noNamespaceSchemaLocation="biblio.xsd">
```

Utilisation du XML Schema dans un document XML

- Pour valider un document XML à l'aide d'un schéma XML, on doit ajouter au document XML ceci :

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

Utilisation d'un Schéma XML

exemple du fichier XML:

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

Types prédéfinis des éléments

- L'attribut *type* indique le type de l'élément.
- Série de types prédéfinis:
string, boolean, float, decimal, integer, nonPositiveInteger, negativeInteger, nonNegativeInteger, positiveInteger, timeDuration, timeInstant, time, timePeriod, date, month, year, century, uriReference, ID, IDREF, ENTITY, NOTATION, QName, Name....
- La cardinalité des éléments: *minOccurs, maxOccurs*.
maxOccurs="unbounded" : la valeur maximale n'a pas de limite.
Ces deux attributs ont une valeur égale 1 par défaut.

Types simples prédéfinis

Type simple

exemples

string

Un bienfait n'est jamais perdu

normalizedString

Un bienfait n'est jamais perdu

token

Un bienfait n'est jamais perdu

byte

-1, 126

unsignedByte

0, 126

base64Binary

GpM7

hexBinary

0FB7

integer

-126789, -1, 0, 1, 126789

positiveInteger

1, 126789

negativeInteger

-126789, -1

nonNegativeInteger

0, 1, 126789

nonPositiveInteger

-126789, -1, 0

int

1, 126789675

Exercice 1 XML Schéma

Voici une DTD: éléments sans attributs

```
<!ELEMENT Librairie (Livre)+>
```

```
  <!ELEMENT Livre (Titre, Auteur, Date, ISBN, Editeur)>
```

```
    <!ELEMENT Titre(#PCDATA)>
```

```
    <!ELEMENT Auteur(#PCDATA)>
```

```
    <!ELEMENT Date(#PCDATA)>
```

```
    <!ELEMENT ISBN(#PCDATA)>
```

```
    <!ELEMENT Editeur(#PCDATA)>
```

1- Donner un exemple de document valide.

2- Donner le document XML Schema équivalent.

Exemple de document XML valide

```
<?xml version="1.0" ?>
<Librairie xmlns="http://www.librairie.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.librairie.org librairie.xsd">
  <Livre>
    <Titre>Le guide du routard Galactique</Titre>
    <Auteur>Douglas Adams</Auteur>
    <Date>1979</Date>
    <ISBN>0345391802</ISBN>
    <Editeur>Ballantine Books</Editeur>
  </Livre>
</Librairie>
```

Le schéma XML correspondant

```
<xsd:element name="Librairie">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Livre" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Titre" type="xsd:string"/>
            <xsd:element name="Auteur" type="xsd:string"/>
            <xsd:element name="Date" type="xsd:string"/>
            <xsd:element name="ISBN" type="xsd:string"/>
            <xsd:element name="Editeur" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element>
```