

XML

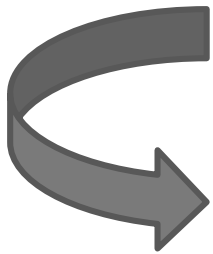
SÉANCE 3 : XML SCHEMA

2ÈME INFO

CHRISTOPHE DAMAS

DÉFICIENCES DES DTD

- N'est pas au format XML
- Ne supporte pas les espaces de nom
- Ne permet pas un typage fort des données



Alternative : XML schema

W3C Recommendation, 2 mai 2001

XML SCHEMA

Un fichier XML Schema (`monFichier.xsd`) est du XML:

- commence par une **déclaration XML**
- élément racine : **schema**

? schema qui valide les schemas ?

<http://www.w3.org/2001/XMLSchema>

XML SCHEMA

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<xs:schema xmlns:xs =  
"http://www.w3.org/2001/XMLSchema">
```

```
<!-- déclarations d'éléments, d'attributs et  
de types ici -->
```

```
</xs:schema>
```

?? XMLNS ??

XMLNS = XML NAMESPACE

NAMESPACE = ESPACE DE NOMS



En informatique, un espace de noms est une notion permettant de lever une ambiguïté sur des termes qui pourraient être homonymes sans cela.

Il est matérialisé par un préfixe identifiant de manière unique la signification d'un terme. Au sein d'un même espace de noms, il n'y a pas d'homonymes.

NAMESPACE – ESPACE DE NOM

Un espace de nom :

- Permet, grâce à un préfixe, l'usage d'un ensemble de balises définies
- Permet la distinction entre les éléments du langage et ceux définis par l'utilisateur.

Dans l'exemple `<xs:schema xmlns:xs =
"http://www.w3.org/2001/XMLSchema">`

`xs` est le préfixe qui permettra l'usage des éléments de l'espace de noms

`"http://www.w3.org/2001/XMLSchema"`

XML SCHEMA

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

Espace de noms par défaut :

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

XMLNS

Pour une référence **locale**,

- en utilisant l'attribut `noNamespaceSchemaLocation`, par exemple

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

Pour un schéma **public**,

- Attribut de l'élément racine, par exemple pour l'élément `persistence` ci-dessous:

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
  http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd"
  version="1.0">
```



```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
<!-- déclarations d'éléments, d'attributs et de types ici -->
```

```
</xs:schema>
```

Pour déclarer un élément :

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

Pour déclarer un attribut :

```
<xs:attribute name="" type="">
```

- **name** → nom de l'élément ou de l'attribut
- **type** → type de l'élément ou de l'attribut
sauf si **type anonyme!**

XS : ATTRIBUTE

- use : optional, required ou prohibited
- type nommé ou anonyme

→ Référence un type simple, càd simpleType

TYPE NOMMÉ

Exemples de type nommé:

```
<xs:simpleType name="pourcent">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="100"/>
  </xs:restriction>
</xs:simpleType>
```

Dans la déclaration des attributs :

```
<xs:attribute name = "resultat" type ="pourcent">
```

TYPE ANONYME

Directement dans la déclaration sans name:

```
<xs:attribute name="resultat">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

XS : ELEMENT

- `minOccurs` (=1 par défaut) et `maxOccurs` (=1 par défaut)
- `type` nommé ou anonyme
 - Référence un type simple, càd `simpleType`
 - Référence un type complexe, càd `complexType`

TYPE SIMPLE

- Ne contient pas d'élément ni d'attribut

- **Type existant : string, integer, ...**

```
<xs:attribute name="nom" type="xs:string"/>
```

- **simpleType avec restriction**

```
<xs:simpleType name="pourcent">  
  <xs:restriction base="xs:integer">  
    <xs:minInclusive value="0"/>  
    <xs:maxInclusive value="100"/>  
  </xs:restriction>  
</xs:simpleType>
```

TYPE SIMPLE

- **pattern**

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

- **enumeration**

```
<xs:simpleType name="visibilite">
  <xs:restriction base="xs:string">
    <xs:enumeration value="private"/>
    <xs:enumeration value="public"/>
    <xs:enumeration value="protected"/>
    <xs:enumeration value="default"/>
  </xs:restriction>
</xs:simpleType>
```

TYPE SIMPLE

- **list**

```
<xs:simpleType name="listeEntiersType">  
  <xs:list itemType="xs:integer"/>  
</xs:simpleType >
```

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

```
<listeEntiers>12 4 9 -4 8</listeEntiers>
```

- **union**

```
<xs:simpleType name="zipUnionType">  
  <xs:union memberTypes="villeType cpType"/>  
</xs:simpleType>
```

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

```
<ville>1200</ville>
```

```
<ville>Gembloux</ville>
```


TYPE COMPLEXE

- Peut contenir des éléments et des attributs
- `complexType`
 - `simpleContent`
 - `sequence`
 - `choice`

TYPE COMPLEXE - SIMPLECONTENT

- **Par restriction**

```
<xs:complexType name="prix">
  <xs:simpleContent>
    <xs:restriction base="xs:positiveInteger">
      <xs:maxInclusive value="100"/>
      <xs:attribute name="devise" type="xs:string"/>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>
```

- **Par extension**

```
<xs:complexType name="prix">
  <xs:simpleContent>
    <xs:extension base="xs:positiveInteger">
      <xs:attribute name="devise" type="xs:string"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

SEQUENCE ET CHOIX

- **sequence**

```
<xs:complexType name="adresseType">
  <xs:sequence>
    <xs:element name="rue" type="xs:string"/>
    <xs:element name="numero" type="xs:positiveInteger"/>
    <xs:element name="codePostal" type="xs:positiveInteger"/>
    <xs:element name="ville" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

- **choice**

```
<xs:complexType name="personneType">
  <xs:choice>
    <xs:element name="professeur" type="xs:string"/>
    <xs:element name="etudiant" type="xs:string"/>
  </xs:choice>
</xs:complexType>
```

- MinOccurs et MaxOccurs également sur sequence et choice

ELEMENT VIDE

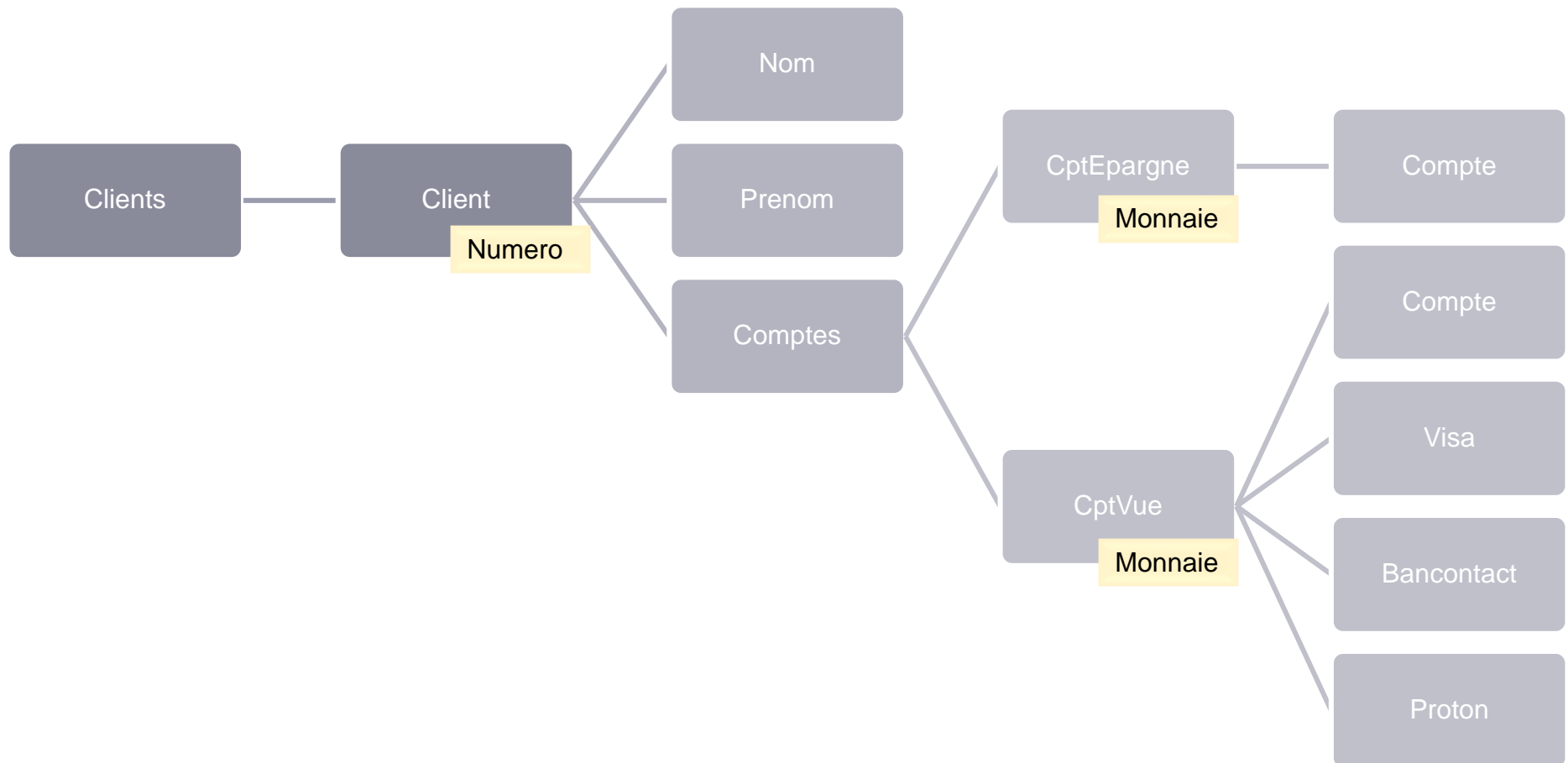
```
<xs:element name="prix">
  <xs:complexType>
    <xs:attribute name="montant" type="xs:float"/>
    <xs:attribute name="devise" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

```
<prix montant="25" devise="euro"/>
```

DTD - BANQUE

```
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Prenom (#PCDATA)>
<!ELEMENT Compte (#PCDATA)>
<!ATTLIST Compte Monnaie (Eur|Cad|Usd ) 'Eur' >
<!ELEMENT Visa (#PCDATA)>
<!ELEMENT Bancontact (#PCDATA)>
<!ELEMENT Proton EMPTY>
<!ELEMENT CptVue (Compte , Visa? , Bancontact* , Proton?)>
<!ELEMENT CptEpargne (Compte)>
<!ELEMENT Comptes (CptEpargne | CptVue)+>
<!ELEMENT Client (Nom , Prenom , Comptes)>
<!ATTLIST Client Numero ID #REQUIRED>
<!ELEMENT Clients (Client+)>
```

BANQUE - EXEMPLE



SCHEMA – COMPTE1.XSD

```
<?xml version = "1.0" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name = "Nom" type="xs:string" />
  <xs:element name = "Visa" type="xs:positiveInteger" />
  <xs:element name = "Bancontact" type="xs:positiveInteger" />
  <xs:element name = "Proton"><xs:complexType/></xs:element>

  <xs:element name = "CptVue">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "Compte" />
        <xs:element ref = "Visa" minOccurs="0" />
        <xs:element ref = "Bancontact" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref = "Proton" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name = "CptEpargne">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "Compte" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Comptes">
    <xs:complexType>
      <xs:choice maxOccurs = "unbounded">
        <xs:element ref = "CptEpargne" />
        <xs:element ref = "CptVue" />
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

<xs:element name="Client">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Nom" />
      <xs:element ref="Prenom" />
      <xs:element ref="Comptes" />
    </xs:sequence>
    <xs:attribute name="Numero" use="required" type="xs:ID" />
  </xs:complexType>
</xs:element>
<xs:element name="Clients">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Client" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Compte">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="numeroSEPAType">
        <xs:attribute name="Monnaie" default="Eur" type="monnaieType" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:simpleType name="monnaieType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Usd" />
    <xs:enumeration value="Eur" />
    <xs:enumeration value="Cad" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="numeroSEPAType">
  <xs:restriction base="xs:string">
    <xs:pattern value="BE\d\d \d{4} \d{4} \d{4}" />
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```