

Définition de Type de Document (DTD) XML schéma (XSD)

BOUALEM ABID & LEBCIR FAHIMA

ENS KOUBA 2015

MOTIVATIONS

Comment spécifier des contraintes plus précises et propres à notre langage XML

Un document XML est *valide* par rapport à une certaine DTD

DTD est comme une grammaire pour un langage → la possibilité de tester automatiquement son respect par un document

Avantages:

Faciliter l'échange et la mise en commun de documents produits par des rédacteurs différents ;

Aider les développeurs qui conçoivent des outils automatiques pour traiter les documents respectant la même DTD

LIER UN FICHER XML À UNE DTD

DTD INTERNE

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE cv [
.
.
.
]>
<cv>
.
.
.
</cv>
```

```
xmllint --dtdvalid madtd.dtd mondoc.xml --noout
```

LIER UN FICHER XML À UNE DTD

DTD EXTERNE

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no" ?>  
<!DOCTYPE cv SYSTEM "cv.dtd">  
<cv>  
.  
.  
.  
</cv>
```

LIER UN FICHER XML À UNE DTD

DTD MIXTE

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no" ?>  
<!DOCTYPE cv SYSTEM "cv.dtd" [  
.  
.  
.  
<cv>  
.  
.  
.  
</cv>
```

DÉFINIR LES ÉLÉMENTS ET LEURS CONTENUS

<!ELEMENT nom_element modèle_de_contenu>

Règles:

- le premier caractère est une lettre quelconque ou un _ (underscore ou tiret bas) ;
- les caractères suivants peuvent être des lettres, des chiffres, des tirets bas (_), des traits d'union (-) ou des points (.) ;
- il n'y a pas de limitation sur la longueur d'un nom XML.

DÉFINIR LES ÉLÉMENTS ET LEURS CONTENUS

1- CONTENU PUREMENT TEXTUEL

- **<!ELEMENT téléphone (#PCDATA)>**

2- SOUS-ÉLÉMENTS

- <!ELEMENT identité (prénom,nom)>
- <!ELEMENT identité (prénom+,surnom?,nom)>
- **? : 0 ou 1 fois ;**
- *** : 0, 1 ou plus ;**
- **+ : 1 ou plus.**

DÉFINIR LES ÉLÉMENTS ET LEURS CONTENUS

3- ALTERNATIVES

- <!ELEMENT expérience (stage|emploi)>

4- COMBINAISONS

- <!ELEMENT diplôme ((intitulé,année) | (année,compétences,stage?)+)>
- Dans ce cas un diplôme est :
 - soit l'intitulé du diplôme et l'année d'obtention ;
 - soit une suite d'année, avec les compétences acquises cette année là, éventuellement validées par un stage.

DÉFINIR LES ÉLÉMENTS ET LEURS CONTENUS

5- CONTENU MIXTE

- `<!ELEMENT parcours (#PCDATA | diplôme)*>`

6- CONTENU VIDE

- `<!ELEMENT br EMPTY>`

7- CONTENU QUELCONQUE

- `<!ELEMENT mavie ANY>`

DÉFINIR LES ÉLÉMENTS ET LEURS CONTENUS

Exemple:

```
<!ELEMENT liste_livres (livre+)>  
<!ELEMENT livre (titre,auteur+,éditeur,description?,prix)>  
<!ELEMENT titre (#PCDATA)>  
<!ELEMENT auteur (#PCDATA)>  
<!ELEMENT éditeur (#PCDATA)>  
<!ELEMENT description (#PCDATA)>  
<!ELEMENT prix (#PCDATA)>
```

DÉFINIR LES ÉLÉMENTS ET LEURS CONTENUS

DOCUMENT XML

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no" ?>
<!DOCTYPE liste_livres SYSTEM "livres.dtd">
```

```
<liste_livres>
<livre>
```

```
  <titre>Comprendre XSLT</titre>
  <auteur>Bernd Amann</auteur>
  <auteur>Philippe Rigaux</auteur>
  <éditeur>O'Reilly</éditeur>
  <description>
```

plus complexes du langage XSLT, et d'application des mécanismes de ce langage à des cas concrets d'utilisation. Il débute par un chapitre introductif en forme d'étude de cas qui propose, sur une application de type "Officiel des spectacles" adaptée au Web, une déclinaison des différents thèmes couverts.

```
  </description>
  <prix>33 euros</prix>
```

```
</livre>
<livre>
```

```
  <titre>Learning XML</titre>
  <auteur>Erik T. Ray</auteur>
  <éditeur>O'Reilly</éditeur>
  <prix>40 dollars</prix>
```

```
</livre>
</liste_livres>
```

DÉFINIR LES ATTRIBUTS

<!ATTLIST identité prénom CDATA #REQUIRED

nom CDATA #REQUIRED

surnom CDATA #IMPLIED>

Types d'attributs:

- **CDATA**

- **NMTOKEN**

- <!ATTLIST VILLE NOM CDATA #REQUIRED
CODE NMTOKEN #REQUIRED>

- **NMTOKENS**

- **ÉNUMÉRATION:**

- <!ATTLIST DATE JOUR (LUNDI|MARDI|MERCREDI|JEUDI|VENDREDI|SAMEDI|DIMANCHE) #REQUIRED
NUM NMTOKEN #REQUIRED
MOIS NMTOKEN #REQUIRED
ANNÉE NMTOKEN #REQUIRED>

DÉFINIR LES ATTRIBUTS

ID

- <!ATTLIST diplôme intitulé CDATA #REQUIRED

code ID #REQUIRED>

IDREF

- <!ATTLIST stage entreprise CDATA #REQUIRED
diplôme IDREF #REQUIRED>

IDREFS

ENTITY: une entité définie dans la DTD

ENTITYS: UNE SUITE DE ENTITY, SÉPARÉES PAR DES ESPACES.

DÉFINIR LES ENTITÉS

Certaines entités sont déjà définies en XML:

- < (<),
 - > (>),
 - & (&),
 - " ("),
 - et' (')
-

ENTITÉS GÉNÉRALES:

- <!ENTITY ac "anticonstitutionnellement">
→ Permet l'utilisation de ∾ dans le document XML associé à la DTD

Exemple:

```
<!ENTITY piedpage '<hr />
```

```
<p>Copyright 2005, dernière mise à jour octobre 2005.
```

```
</p>'
```

```
>
```

DÉFINIR LES ENTITÉS

ENTITÉS GÉNÉRALES EXTERNES

- `<!ENTITY piedpage SYSTEM "pp.xml">`
-

ENTITÉS GÉNÉRALES EXTERNES NON PARSÉES

- `<!NOTATION jpeg SYSTEM "image/jpeg">`
`<!ENTITY maphoto SYSTEM "toto.jpg" NDATA jpeg>`

ENTITÉS PARAMÈTRES

- `<!ELEMENT identité (nom,prénom,naissance)>`
`<!ELEMENT enseignant (nom,prénom)>`
- `<!ENTITY % elements_personne "nom,prénom">`
- `<!ELEMENT identité (%elements_personne;;naissance)>`
- `<!ELEMENT enseignant (%elements_personne)>`

DTD MODULAIRES

```
<![IGNORE[  
<!ELEMENT personne (nom,prénom)>  
<!ELEMENT nom (#PCDATA)>  
<!ELEMENT prénom (#PCDATA)>  
]]>
```

```
<![INCLUDE[  
<!ELEMENT entreprise (dénomination,taille)>  
<!ELEMENT dénomination (#PCDATA)>  
<!ELEMENT taille (#PCDATA)>  
]]>
```

DTD MODULAIRES

```
<!ENTITY % bloc_personnes "INCLUDE">  
<!ENTITY % bloc_entreprises "IGNORE">
```

```
<![%bloc_personnes;  
<!ELEMENT personne (nom,prénom)>  
<!ELEMENT nom (#PCDATA)>  
<!ELEMENT prénom (#PCDATA)>  
]]>
```

```
<![%bloc_entreprises[  
<!ELEMENT entreprise (dénomination,taille)>  
<!ELEMENT dénomination (#PCDATA)>  
<!ELEMENT taille (#PCDATA)>  
]]>
```

DTD VERSUS XML SCHÉMA

DTD

types pauvres, peu de contraintes sur les contenus

nombre d'apparitions d'un élément à choisir entre 0 et 1

pas de gestion des espaces de noms

pas un format XML

XSD

utilisation et définition de types, contraintes sur les contenus

possibilité de définir précisément le nombre d'apparitions d'un élément

espaces de noms supportés

format XML, parsable facilement

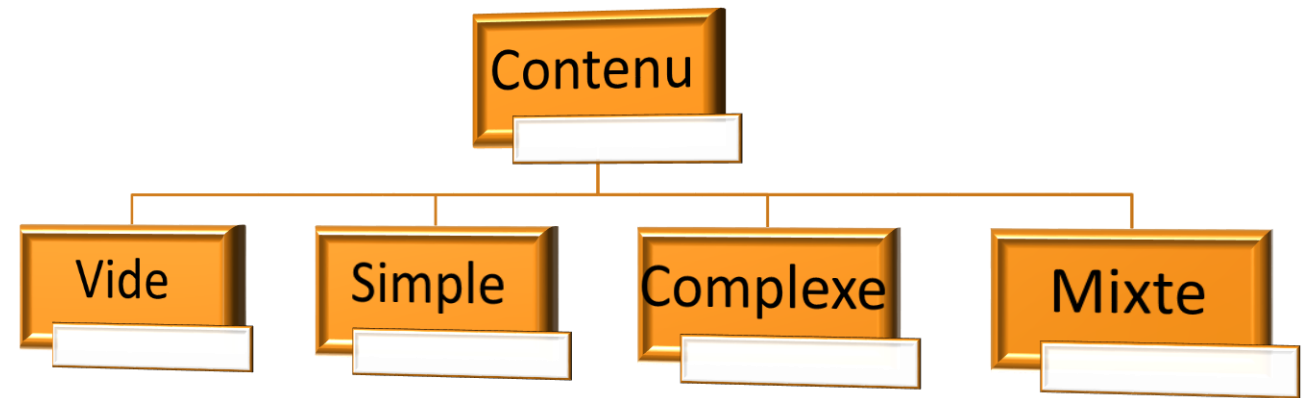
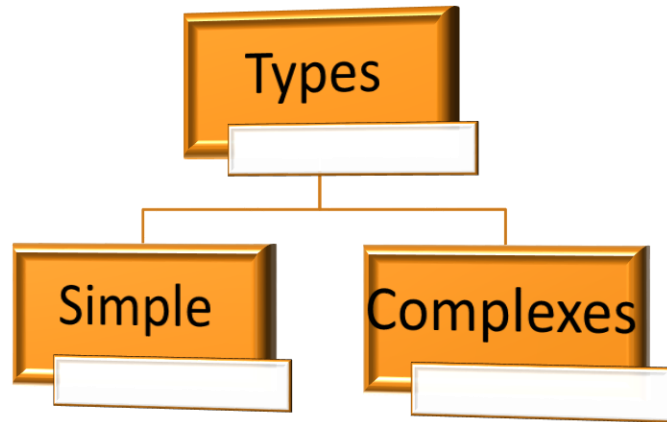
PRATIQUE

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

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

```
xmllint --schema monschema.xsd undoc.xml
```

MODÈLES DE CONTENU ET TYPES



LES TYPES SIMPLES PRÉDÉFINIS

- String
 - decimal, integer, positiveInteger, real
 - ~~date, dateTime, duration~~
 - ID, IDREF, ENTITY, NMTOKEN, etc. (venant des DTD)
-

DÉFINIR UN NOUVEAU TYPE SIMPLE

Liste de valeurs d'un même type simple

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

Union de types simples

- `<xs:simpleType name="contact">`
 `<xs:union memberTypes="adresse telephone" />`
`</xs:simpleType>`

Restriction d'un type simple

- **Énumération pour les chaînes de caractères, nombres et dates**

```
<xs:simpleType name="jourSemaine">
```

```
<xs:restriction base="xs:string">
```

```
  <xs:enumeration value="lundi"/> <xs:enumeration value="mardi" /> <xs:enumeration value="mercredi"
```

```
  <xs:enumeration value="jeudi" /> <xs:enumeration value="vendredi" /> <xs:enumeration value="samedi"
```

```
  <xs:enumeration value="dimanche" />
```

```
</xs:restriction>
```

```
</xs:simpleType>
```

DEFINIR UN NOUVEAU TYPE SIMPLE

Restriction d'un type simple (suite)

longueur fixe, minimale ou maximale des chaînes de caractères

- ```
<xs:simpleType name="telephone">
 <xs:restriction base="xs:string">
 <xs:length value="10" />
 </xs:restriction>
</xs:simpleType>
```
- ```
<xs:simpleType name="motdepasse">  
  <xs:restriction base="xs:string">  
    <xs:minLength value="8" /> <xs:maxLength value="20" />  
  </xs:restriction>  
</xs:simpleType>
```

DÉFINIR UN NOUVEAU TYPE SIMPLE

Restriction d'un type simple (suite)

bornes sur les entiers et les dates

- `<xs:simpleType name="temperature">`
 - `<xs:restriction base="xs:integer">`
 - `<xs:minInclusive value="-15" />`
 - `<xs:maxInclusive value="+35" />`
 - `</xs:restriction>`
- `</xs:simpleType>`

nombre de chiffres sur les nombre

- `<xs:simpleType name="codepostal">`
 - `<xs:restriction base="xs:integer">`
 - `<xs:totalDigits value="5" />`
 - `</xs:restriction>`
- `</xs:simpleType>`
- `<xs:simpleType name="prix">`
 - `<xs:restriction base="xs:decimal">`
 - `<xs:fractionDigits value="2" />`
 - `</xs:restriction>`
- `</xs:simpleType>`

DÉFINIR UN NOUVEAU TYPE SIMPLE

Restriction d'un type simple (suite)

expressions régulières sur tous les types simples

- `<xs:simpleType name="mail">`
 - `<xs:restriction base="xs:string">`
 - `<xs:pattern value=".+@.+" />`
 - `</xs:restriction>`
 - `</xs:simpleType>`

DÉFINIR UN ATTRIBUT

```
<xs:attribute name="score" type="xs:integer" use="required" default="0" />
```

DÉFINIR UN ÉLÉMENT DE TYPE SIMPLE

- `<xs:element name="nom" minOccurs="1" maxOccurs="1" type="xs:string" />`
- `<xs:element name="prénom" minOccurs="1" maxOccurs="unbounded" type="xs:string" />`
- `<xs:element name="surnom" minOccurs="0" maxOccurs="1" type="xs:string" />`

DÉFINIR UN ÉLÉMENT DE TYPE COMPLEXE

CONTENU VIDE AVEC ATTRIBUT

- `<xs:element name="br">`
 - `<xs:complexType>`
 - `<xs:attribute name="class" type="string" />`
 - `</xs:complexType>`
 - `</xs:element>`
-

CONTENU SIMPLE AVEC ATTRIBUT

- `<xs:element name="title">`
- `<xs:complexType>`
- `<xs:simpleContent>`
- `<xs:extension base="string">`
- `<xs:attribute name="lang" type="string" />`
- `</xs:extension>`
- `</xs:simpleContent>`
- `</xs:complexType>`
- `</xs:element>`

DÉFINIR UN ÉLÉMENT DE TYPE COMPLEXE

CONTENU COMPLEXE

- `<xs:element name="auteur">`
- `<xs:complexType>`
- `<xs:sequence>`
- `<xs:element name="nom" minOccurs="1" maxOccurs="1" type="xs:string" />`
- `<xs:element name="prénom" minOccurs="1" maxOccurs="unbounded" type="xs:string" />`
- `<xs:element name="surnom" minOccurs="0" maxOccurs="1" type="xs:string" />`
- `</xs:sequence>`
- `</xs:complexType>`
- `</xs:element>`

CONTENU MIXTE

- `<xs:element name="p">`
- `<xs:complexType mixed="true">`
- `<xs:choice minOccurs="0" maxOccurs="unbounded">`
- `<xs:element name="em" type="xs:string" />`
- `<xs:element name="strong" type="xs:string" />`
- `</xs:choice>`
- `</xs:complexType>`
- `</xs:element>`

CONTRAINTES D'UNICITÉ ET DE CLEF

• UNICITÉ

- `<xs:element name="bibliotheque">`
- `<xs:complexType>`
- ...
- `</xs:complexType>`
- `<xs:unique name="uniquelivre">`
- `<xs:selector xpath="book" />`
- `<xs:field xpath="isbn" />`
- `</xs:unique>`
- `</xs:element>`

CLÉS

- `<XS:ELEMENT NAME="BIBLIOTHEQUE">`
- `<XS:COMPLEXTYPE>`
- ...
- `</XS:COMPLEXTYPE>`
- `<XS:KEY NAME="CLEFLIVRE">`
- `<XS:SELECTOR XPATH="LIVRE" />`
- `<XS:FIELD XPATH="ISBN" />`
- `</XS:KEY>`
- `</XS:ELEMENT>`

CONTRAINTES D'UNICITÉ ET DE CLEF

RÉFÉRENCE

- `<xs:keyref name="refcleflivre" refer="cleflivre">`
- `<xs:selector xpath="citation"/>`
- `<xs:field xpath="ref" />`
- `</xs:keyref>`