

## Documents XML valides : DTD et schémas XML

Andrei Popescu-Belis  
TIM / ETI, Université de Genève

*Cours n°3*

<http://moodle.unige.ch>

## Rappel

- Document XML
  - bien formé : respecte les règles de base d'XML [cours 2]
  - valide : respecte en plus les règles d'une application particulière, définies formellement par une DTD [cours 3]
- Feuille de style (XSLT)
  - contient les informations « graphiques » [cours 6]

2

## Création d'une application XML

- Définir des balises pour une application précise
    - fiches de bibliothèque
    - etc. [voir partie II du cours]
  - Communiquer à d'autres cette convention
  - Vérifier que des documents XML s'y conforment
- Exprimer des contraintes sur :
- les noms de balises autorisés
  - l'ordre autorisé des balises
  - quels éléments peuvent contenir quels éléments
  - quels éléments sont optionnels
  - les attributs autorisés, leur caractère obligatoire ou non, leur type

3

## La « grammaire des balises »

- DTD : *Document Type Definition*
  - notion provenant de SGML
  - très utilisée en XML
  - écrite dans un langage spécifique
- Document XML conforme à une DTD = valide

```
<?xml version="1.0" standalone="no"?>  
<!DOCTYPE racine SYSTEM "exemple.dtd">  
<racine>  
  ...  
</racine>
```

4

## Associer une DTD à un document XML

- Deuxième ligne du fichier, après la déclaration XML
- DTD externe : contenue dans un fichier .dtd

```
<!DOCTYPE livre SYSTEM "biblio.dtd">  
ou <!DOCTYPE livre SYSTEM "file:///H:/prive/biblio.dtd">  
ou <!DOCTYPE livre SYSTEM "http://... ">
```
- Autres options
  - DTD externe publique (connue)

```
<!DOCTYPE livre PUBLIC "nom_connu" "URL">
```
  - DTD interne au fichier XML

```
<!DOCTYPE ouvrage [ ...texte de la DTD... ]>
```

5

## Exemple de DTD

### 1. DTD externe, fichier biblio.dtd (format texte)

```
<!ELEMENT ouvrage (index, titre, auteur+)>  
<!ELEMENT index EMPTY>  
<!ATTLIST index valeur CDATA #REQUIRED>  
<!ELEMENT titre (#PCDATA)>  
<!ELEMENT auteur (nom, prenom+)>  
<!ELEMENT nom (#PCDATA)>  
<!ELEMENT prenom (#PCDATA)>
```

### 2. Déclaration DOCTYPE dans le fichier XML

```
<?xml version="1.0" standalone="no"?>  
<!DOCTYPE ouvrage SYSTEM "biblio.dtd">
```

6

## Structure d'une DTD : éléments

- Déclarations des **noms des éléments autorisés**

```
<!ELEMENT nom_balise (contenu)>
```

- Déclarations de l'**ordre des éléments**

– dans chaque (contenu)

7

## Exemple 1

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

- l'élément (balise) **prenom** contient des caractères de texte, mais pas d'autres éléments (balises)
- exemple de fichier XML autorisé:  

```
<prenom>Jean</prenom>
```
- exemple de fichier XML non autorisé:  

```
<prenom>  
  <premier>Jean</premier>  
  <deuxieme>Paul</deuxieme>  
</prenom>
```

8

## Exemple 2

```
<!ELEMENT titre (#PCDATA|chiffre)*>
```

- l'élément (balise) **titre** contient un mélange de caractères textuels et d'éléments **chiffre**
- c'est la seule façon de noter ce « mélange »
- fichier XML autorisé:  

```
<titre>  
  Il a parié <chiffre>7</chiffre> contre  
  <chiffre>1</chiffre> et il a gagné.  
</titre>
```

9

## Exemple 3

```
<!ELEMENT auteur (nom, prenom+, initiale?)>
```

- la balise (l'élément) **auteur** contient un élément **nom**, un ou plusieurs **prenom**, zéro ou une **initiale**, et rien d'autre
- exemple de fichier XML autorisé:  

```
<auteur>  
  <nom>Shaw</nom>  
  <prenom>George</prenom>  
  <prenom>Bernard</prenom>  
</auteur>
```

10

## Exemple 4

```
<!ELEMENT index EMPTY>
```

- signifie : la balise (élément) **index** est vide (pas de contenu), mais elle peut avoir des attributs

```
<!ELEMENT résumé ANY>
```

- signifie : la balise (élément) **résumé** peut contenir n'importe quoi (caractères, éléments)
- 'ANY' n'est pas très précis, plutôt à éviter

11

## Quelques symboles utilisés

```
<!ELEMENT nom (surnom|vrai_nom)>
```

= l'élément nom contient **soit** surnom **soit** vrai\_nom

```
<!ELEMENT exemple (exa, exb?, exc*, exd+)>
```

= l'élément exemple contient:

RIEN	=	un seul élément exa
?	=	zéro ou un éléments exb
*	=	zéro, un ou plusieurs éléments exc
+	=	un ou plusieurs éléments exd

Dans cet ordre, et sans autres caractères entre eux !

12

## Structure d'une DTD : attributs

- Déclarations des **attributs de chaque élément** (noms, types, attribut obligatoire ou non)

```
<!ATTLIST nom_balise attribut-1 type-1
                attribut-2 type-2
                attribut-3 type-3>
```

- L'ordre des déclarations n'est pas important
  - mais les attributs viennent après l'élément correspondant

13

## Déclaration des attributs

```
<!ATTLIST nom-élément
                nom-attribut type nature>
```

- La séquence `nom-attribut type nature` est répétable
- Le **type** peut être :
  - pour des chaînes de caractères : `CDATA`
  - pour une liste de valeurs possibles : `(Lundi|mardi)`
  - et d'autres types ... (voir feuille aide-mémoire)
- La **nature** peut être :
  - `"une-valeur-par-défaut"`
  - `#REQUIRED`
  - `#IMPLIED`
  - `#FIXED "une-valeur-fixée-pour-toujours"`

14

## Exemple de DTD (bis)

Fichier `biblio.dtd` (format texte)

```
<!ELEMENT ouvrage (index, titre, auteur+)>
<!ELEMENT index EMPTY>
<!ATTLIST index valeur CDATA #REQUIRED>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT auteur (nom, prenom+)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
```

15

## XML Schema Definition (XSD)

- « Grammaire des balises » plus moderne
  - écrite en XML (contrairement aux DTD)
  - permet de spécifier des types pour les éléments
    - (DTD : on peut seulement contraindre les attributs)
    - hiérarchie de types et de sous-types
      - dates, nombre, chaînes, etc.
      - on peut spécifier ce qu'un élément peut/doit contenir
- Brièvement
  - XSD : plus précis, plus flexible, plus « informatique »
  - DTD : plus facile à comprendre et à rédiger ☺

16

## Vérification et validation

- Bien distinguer BF et validité
  - **bonne formation** : respecter la syntaxe XML
  - **validité** : respecter une DTD précisée (ou un schéma XML)
- Outils
  - Logiciels, boîtes à outils, éditeurs XML
    - éditeurs *Treebeard*, *XMLSpy* → fonction *valider*
  - En ligne sur Internet : *RUWF* ou *STG*, p.ex.
    - copier/coller le fichier avec une DTD interne (courte)
    - ou bien: mettre le document *et la DTD* sur sa page web
  - Les navigateurs web ne le font pas !

17

## DTD dans XMLSpy

- Menu 'XML'
  - diverses commandes pour la création de fichiers XML
  - 'Check well-formedness' (ou F7) ou bouton marqué
  - 'Validate' ou F8 ou bouton marqué
- Menu 'DTD/Schema'
  1. Associer une DTD à un document : 'Assign DTD'
    - noter l'évolution des 'Helpers' (fenêtres de gauche) lorsqu'on associe une DTD à un document XML en cours de rédaction
    - ne pas oublier de faire 'Update entry helpers' en changeant de DTD
  2. Générer une DTD minimale qui correspond à un document
    - la DTD la plus simple qui permet de valider un document XML donné
    - utile pour définir des DTD à partir d'un prototype de document XML
    - magique ? peut-être, mais ne peut pas deviner toutes vos intentions...

18

## Application pratique

### Exercices sur la bonne formation et la validité des fiches bibliographiques

## Exercices (1/2)

- Téléchargez les fichiers livre31.xml, livre32.xml
  - (ils ressemblent à ceux du dernier cours)
  - sont-ils bien formés ?
  - sont-ils valides ? (*question piège*)
- Associez à ces fichiers la DTD livre.dtd (téléchargée aussi)
  - soit en écrivant l'instruction DOCTYPE suivant le modèle du cours
  - soit via le menu 'DTD/XML Schema' de XMLSpy
  - les fichiers sont-ils maintenant valides ? Corrigez-les si nécessaire
- En utilisant la fonction DTD>Generate DTD de XMLSpy
  - créez une DTD nommée livre1.dtd à partir de livre31.xml
  - créez une DTD nommée livre2.dtd à partir de livre32.xml
  - est-ce qu'elles sont identiques ? Sont-elles identiques à livre.dtd ?

20

## Exercices (2/2)

- Associez la DTD livre2.dtd à livre31.xml
    - est-ce que livre31.xml est encore valide ?
    - même question avec la DTD livre1.dtd et livre32.xml
  - Créez une DTD livre3.dtd pour laquelle livre31.xml et livre32.xml soient valides en même temps
    - ressemble-t-elle à livre.dtd ?
  - Créez un nouveau fichier livre33.xml semblable aux autres et collez la DTD livre3.dtd en tant que DTD interne
    - variez le plus possible la forme de livre33.xml tout en vérifiant sa validité
    - où sont les limites de ce que vous pouvez faire varier ?
      - pour les dépasser... modifiez la DTD
- ➔ **Devoir : déposez le fichier livre32.xml ou livre33.xml sur Moodle avant lundi prochain 13.11.06 à 12h00**

21