

CR TDM02 XML

Noémie PRIN et Maria-Bianca ZUGRAVU

Points importants CM

- **(EMPTY|ANY)** peut être vide ou contenir n'importe quel élément de la dtd.
- XML est une forme extensible de HTML: **définition de balises** (en HTML les balises et les sémantiques associées sont prédéfinies)
- Les fichiers xml peuvent être utilisés avec les protocoles HTTP, URL
- Un document XML est une structure logique **arborescente**.

On utilise un élément(**balise**) lorsque:

- le contenu comporte plusieurs mots
- l'ordre est important (**il n'y a pas d'ordre sur les attributs**)

On utilise un **attribut** lorsque:

- l'info qu'on veut définir modifie la balise :

```
<liste type="numerotee"> ... </liste>
```
- si on veut contrôler les valeurs
- l'info est un id

Déclaration d'un attribut:

```
<!ATTLIST element attribut type value>
```

value :

- une valeur par défaut spécifiée entre guillemets:

```
<personne individual_id="e10001">
```
- **#REQUIRED** : instantiation de l'attribut obligatoire. ne fonctionne pas dans toutes les configurations
- **#IMPLIED** : valeur d'attribut optionnelle, instantiation non obligatoire
- **#FIXED 'val'** : valeur constante

type:

- **CDATA** : chaîne de caractère
- **(a|b)** : valeurs possibles
- **ID** : 2 valeurs d'attributs d'un élément utilisé 2 fois ne peuvent être les mêmes
- **IDREF** : si référence à un seul ID d'un autre attribut

- **IDREFS** : référence à plusieurs ID d'attribut (**chaque valeur est séparée par un espace**)
- **ENTITY**
- **NOTATION**

Cardinalités :

- **+** : 1 -> inf
- **-** : 0 ou inf
- **?** : 0 ou 1
- **e1 | e2**

```
<!ELEMENT nomElement (element1+, element28,element3?)>
```

Un élément peut contenir:

- uniquement des données: **PCDATA**

```
<!ELEMENT nomElement (#PCDATA)>
```
- uniquement d'autre éléments:

```
<!ELEMENT nomElement (element1,element2,...)>
```
- des données et d'autre éléments:

```
<!ELEMENT type(#PCDATA|element1|element2)>
```

Exemple DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT personne (identite, activite*)>
<!ELEMENT identite (prenom, nom)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT activite (#PCDATA)>
```

Exemple arbre XML

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE arbre SYSTEM "../personne.dtd" >
<arbre>
  <personne individual_id="e10001" parent_id="e10002 e10003">
    <prenom>Bart</prenom><nom>Simpson</nom>
  </personne>
  <personne individual_id="e10002">
    <prenom>Homer</prenom><nom>Simpson</nom>
  </personne>
  <personne individual_id="e10003">
    <prenom>Marge</prenom><nom marital="oui">Simpson</nom><nom>Bouvier</nom>
  </personne>
```

</arbre>

Entité : alias pour un groupe de données

Composition d'une DTD dans une DTD:

```
<!ENTITY %secondeDTD SYSTEM "secondeDTD.dtd">
...
%secondeDTD;
```

Notation:

- identifie par un nom le format des entités non analysées par le parseur XML
- définit le format des données et les applications qui permet de les traiter

```
<!NOTATION nomNotation SYSTEM|PUBLIC "notation">
<!NOTATION GIF SYSTEM "GIF">
```

Structure d'un document XML

- **Prologue** :

- déclaration XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

standalone : la présence ou non de références externes

- déclaration de type:

```
<!DOCTYPE bibliographie SYSTEM "bibliographie.dtd">
```

- instructions de traitement:

```
<?xml-stylesheet type="text/css" href="bibliographie.css" ?>
```

Instructions "facultatives". Leur contenu est transmis à une application pour traitement.

```
<cible arg1="val1" arg2="val2" ... >
```

- *cible*: nom d'une application
- arguments passés à l'application cible

- **Arbre des éléments** :

- éléments (balises)
- attributs
- entités

DTD

Une DTD sert à définir la grammaire d'un document XML. Donc, elle facilite la validation de document et l'échange de fichiers.

- DTD *externe local* : *SYSTEM*
- DTD *externe publique* : *PUBLIC*
- dans le fichier .dtd il n'y a pas de **< !DOCTYPE >** ; il sert à définir les relations parent-fils, les cardinalités, l'enchaînement;

Erreurs typiques :

- indentation
- balises mal placées, fermetures oubliées
- conformité avec la dtd
- à vérifier : encodage, mettre utf-8

Exemple :

```
<?xml version="1.0" encoding="utf-8"?>
<e1>
  <e2 id="id1">...</e2>
  <e2 id="id2">...</e2>
  ...
  <e3 ref="id1"/>
  ...
  <e4 refs="id1 id2"/>
</e1>
```

CSS

- Remarque : Pour placer la ponctuation, on est obligé de les placer après/avant chaque type plutôt que d'utiliser un sélecteur global (comme AllOf) et d'ajouter une exception.

Remarques pratiques

- Un document possède une racine et une seule!
- Si un document n'a pas de DTD on doit préciser ça dans sa déclaration XML: **standalone**
- Un document sans DTD peut être affiché mais ne peut pas être interprété.
- Un document est dit **valide** s'il est conforme à sa DTD.
- Un document est dit **bien formé** s'il respecte la syntaxe XML(même s'il n'a pas de DTD)

- **CDATA dans XML** indique à l'analyseur de ne pas tenir compte du balisage. Utile lorsque l'on inclut du Javascript à XML.
- Le prologue d'une DTD est identique au celui pour un XML, excepté standalone et DOCTYPE qui n'ont pas de sens. **Une DTD n'a pas nécessairement de prologue.**

Scripts

Serveur bibliographique (1) : XML et CSS

Enoncé: On définit une bibliographie comme un ensemble de publications. Une publication est composée d'un titre, d'une liste d'auteurs (auteurs), d'un ensemble de pages (au moins une ; ex : 200-210) et d'une année de publication. Une publication est caractérisée par un type (soit article, soit conférence). Un (auteurs) est un ensemble d'auteurs.

bibliographie.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
  <!ELEMENT bibliographie (publication+)>
  <!ELEMENT publication (titre,auteurs+,pages+,annee)>
  <!ATTLIST publication type (ARTICLE|CONFERENCE) #REQUIRED>
  <!ELEMENT titre (#PCDATA)>
  <!ELEMENT auteurs (auteur+)>
  <!ELEMENT auteur (#PCDATA)>
  <!ELEMENT pages (#PCDATA)>
  <!ELEMENT annee (#PCDATA)>
```

bibliographie.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE bibliographie SYSTEM "bibliographie.dtd">
<?xml-stylesheet type="text/css" href="bibliographie2.css" ?>
<bibliographie>
  <publication type="CONFERENCE">
    <titre>Noémie and Bianca are awesome</titre>
    <auteurs>
      <auteur>Bianca</auteur>
      <auteur>Noémie</auteur>
    </auteurs>
    <pages>500</pages>
    <annee>2019</annee>
  </publication>
```

```

<publication type="ARTICLE">
  <titre>Noémie and Bianca are still awesome</titre>
  <auteurs>
    <auteur>Malandain</auteur>
    <auteur>Delestre</auteur>
  </auteurs>
  <pages>600</pages>
  <annee>2019</annee>
</publication>
</bibliographie>

```

bibliographie.css

```

* {}
publication {
  display: block;
  background-color: pink;
  padding-left: 10px;
  width: 500px;
  font-family: arial ;
  color: black;
  font-size: 12pt ;
}
titre, auteurs, auteur, annee, pages {
  display: inline;
}
auteur:after, pages:after {
  content: ", "; /*des virgules apparaissent entre chaque élément*/
}
titre:before {
  content: "\"";
}
titre {
  font-style: italic; /*le titre apparait entre guillemets et en italiques,*/
}
titre:after {
  content: "\"";
}
pages:before {
  content: "pp. ";
}
annee:after {
  content: "."; /*un point est ajouté à la fin de chaque publication*/
}
publication {

```

```
margin-bottom: 10px;  
}
```