Le langage XML

eXtensible Markup Language

et

DTD

Document Type Definition

Arguments

- l'occupation disque n'est plus forcément une issue actuellement pour des données de types texte => on privilégie donc le fait de structurer les informations (nom,valeur) plutôt que valeur en essayant de se rappeler à quoi correspond la valeur

- format unifié (texte) pour échange de contenu entre systèmes hétérogènes (voir pb encodage)

Historique

- W3C Recommendation : http://www.w3.org/XML/ (page d'accueil)
- Spécification du W3C : http://www.w3.org/TR/xml/
- Dérivé de SGML
- Version 1.0 (1ère édition 1998, 5ème édition, 2008)
- Version 1.1 (2004), un peu moins répandue (meilleure gestion des caractères UNICODE)

Actuellement, utilisé partout

- Flux RSS (news)
- SVG (images vectorielles)
- XSL (programmation sur des données XML)
- FO (->PDF)
- MathML
- XHTML (pages WEB) : fin avec HTML5
- ODF : Open Document Format (OpenOffice) / OOXML : Office Open XML (Norme)
- Ant, Maven (gestion de projet Java)
- XUL (langage de définition d'interface / Mozilla)
- XAML (Interface Windows)
- SOAP/WSDL/BPEL : Web Services (échange des données, définition d'interface, composition de WS)
- + des APIs pour pouvoir être intégré dans les applications (en Java dans le JDK : SAX/DOM/JAXB/JAXP/…)

e**X**tensible **M**arkup **L**anguage : Le concept

- Un fichier html est un fichier à balise qui contient de l'information et la façon d'afficher cette information à l'aide d'un browser (ie, netscape, mozilla, ...).
- *Le concept XML vise essentiellement*
  - *à séparer l'information de sa présentation*
  - *à pouvoir accéder facilement à toute ou partie de cette information*
  - *structurer l'information sous forme d'arbre*

Structure d'un document XML

- nœuds :
  - document
  - élément (balises) : définit la structure arborescente
    - nœuds fils
    - balise ouvrante, fermante, compactée
    - attribut (nom/valeur)
  - texte
  - commentaire
- instructions de traitement
  - entête
  - encodage : UTF-8 / ISO-8859-1 (Latin1 FR)
  - autres : traitement XSL, DTD, …
- espaces de noms (signature)

Structure d'un document XML

```
< ?xml version="1.0" encoding="UTF-8" ?>
<personne nom="toto" prenom="titi" >
    <adresse>2 bd Lucien Favre</adresse>
    <abonne value="true" />
</personne>
```

```
< ?xml version="1.0" encoding="UTF-8" ?>
<p:personne xmlns:p=http://www.tutu.com  nom="toto" prenom="titi" >
    <p:adresse>2 bd Lucien Favre</p:adresse>
     <p:abonne value="true" />
</p:personne>
```

Les éléments d'un arbre XML

- Un arbre XML est un ensemble de nœuds délimités par une balise ouvrante et la même balise fermante.
- Une balise ouvrante à un nom et contient de 0 à plusieurs attributs et est définie comme suit :
  <nomBalise att1= " … " … attn= " … " >.
- Une balise fermante se définit comme suit :
  </nomBalise>
- Un attribut a un nom et une valeur
- Un nœud contient de 0 à plusieurs nœuds et de 0 à plusieurs zones de textes (PCDATA) appelés nœuds textes.
- On peut définir des nœuds vides comme suit :
  <nomBalise listeAttributs/>. La liste des attributs peut être vide.

XML : Un exemple de fichier XML

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<etudiant  num="1" prenom="Maude" nom="Eme" annee="ing2">
    <ou adresse="1 rue du bo" ville="Paris" cp="75000" />
    <matiere  indexe= "I13"  libelle ="info">
        <appreciation>tu peux mieux faire</appreciation>
        <note epreuve="ds n° 1" valeur="10.5" />
        <note epreuve="ds n° 2" valeur="11" />
        <note epreuve="ds n° 3" valeur="12" />
    </matiere>
    <matiere  indexe= "C15"  libelle="comptabilité générale">
        <appreciation>belle prestation</appreciation>
        <note epreuve="ds" valeur="15" />
    </matiere>
</etudiant>
```

XML : **D**ocument **T**ype **D**efinition

XML permet d'utiliser un fichier afin de vérifier qu'un document XML est conforme à une syntaxe donnée

La norme XML définit ainsi une définition de document type appelée DTD (*Document Type Definition*),

une grammaire permettant de vérifier la conformité du document XML

La norme XML n'impose pas l'utilisation d'une DTD pour un document XML, mais elle impose par contre le respect exact des règles de base de la norme XML.

XML : **D**ocument **T**ype **D**efinition

- Deux fichiers d'étudiants avec leur adresse et leurs notes dans les matières diffèrent dans leurs valeurs (nom, prénom, adresse et notes) mais pas dans leur sémantique

- On a deux façons de décrire cette sémantique
  - À travers une DTD : Document Type Definition.
  - A travers un schéma : description sous forme de classes. Un fichier XML associé à ce schéma est une instance.

- Nous nous intéresserons à une représentation sous forme de *DTD*.

XML : Un exemple de DTD

Une DTD peut être définie de 2 façons :

- sous **forme interne**, c'est-à-dire en incluant la grammaire au sein même du document

- sous **forme externe**, soit en appelant un fichier contenant la grammaire à partir d'un fichier local ou bien en y accédant par son URL

Déclaration d'un élément

Pour pouvoir créer un document XML il est utile dans un premier temps de définir les éléments pouvant être utilisés, ou plus exactement les informations que l'on désire utiliser.

Pour définir un élément on utilisera la syntaxe suivante :

<! ELEMENT Nom Modèle >

XML : Modèle

Le paramètre modèle représente :

- soit un type de donnée prédéfini,
- soit une règle d'utilisation de l'élément.

Type prédéfini Description	
ANY	L'élément peut contenir tout type de données en caractères et les autres éléments de votre choix définis dans votre DTD
EMPTY	L'élément ne contient pas de données spécifiques HTML classique par exemple ou <img />
#PCDATA	L'élément doit contenir une chaîne de caractères

```
<!ELEMENT note ANY>          :   <note>
                                     description de la matiere
                                     <matiere> informatique </matiere>
                                 </note>
<!ELEMENT br EMPTY>          :   <br />
<!ELEMENT matiere (#PCDATA)> :   <matiere> informatique </matiere>
```


Les éléments d'un arbre XML

- Un arbre XML est un ensemble de nœuds délimités par une balise ouvrante et la même balise fermante.
- Une balise ouvrante à un nom et contient de 0 à plusieurs attributs et est définie comme suit :
  <nomBalise att1= " … " … attn= " … " >.
- Une balise fermante se définit comme suit :
  </nomBalise>
- Un attribut a un nom et une valeur
- Un nœud contient de 0 à plusieurs nœuds et de 0 à plusieurs zones de textes (PCDATA) appelés nœuds textes.
- On peut définir des nœuds vides comme suit :
  <nomBalise listeAttributs/>. La liste des attributs peut être vide.

Le mot clé ELEMENT et les symboles associés

- Dans une DTD, on utilise le mot clé ELEMENT pour définir la structure d'un nœud.
- Pour décrire un nœud, on décrit ses fils de premier niveau en précisant leurs ordres d'apparition et leurs multiplicités.
- Pour la multiplicité, on utilise les symboles +, * et ?
  - ? : un seul ou pas du tout (cardinalité (0 ou 1)
  - + : un ou plusieurs
  - * : zéro ou plusieurs
- On utilise le symbole "," pour définir une séquence de type de nœuds.
- On utilise le symbole "|" pour définir un choix de nœud.
- On utilise le symbole #PCDATA pour définir un nœud texte.

XML : Résumé

Opérateur	Signification	Exemple
+	L'élément doit être présent au minimum une fois	A+
*	L'élément peut être présent plusieurs fois (ou aucune)	A*
?	L'élément peut être optionnellement présent	A?
|	L'élément A ou l'élément B peuvent être présents	A|B
,	L'élément A doit être présent et suivi de l'élément B	A,B
()	Les parenthèses permettent de regrouper des éléments afin de leur appliquer les autres opérateurs	(A,B)+

Le mot clé ATTLIST et les symboles associés

- Dans une DTD, on utilise le mot clé ATTLIST pour définir les attributs d'un nœud.
- Avec ce mot clé, on doit préciser de quel nœud il s'agit et la liste des attributs.

Déclaration des attributs

- Il est possible d'ajouter des propriétés à un élément particulier en lui affectant un attribut, c'est-à-dire une paire clé/valeur.

Ainsi avec XML la syntaxe pour définir un attribut est la suivante :

<! ATTLIST Elément Attribut Type >

Déclaration des attributs

Type représente le type de donnée de l'attribut, il en existe trois :

- **littéral**: il permet d'affecter une chaîne de caractères à un attribut. Pour déclarer un tel type il faut utiliser le mot clé *CDATA*

- **l'énumération**: cela permet de définir une liste de valeurs possibles pour un attribut donné, afin de limiter le choix de l'utilisateur. La syntaxe de ce type d'attribut est :

<! ATTLIST Elément Attribut (Valeur1 | Valeur2 | ...) >

Pour définir une valeur par défaut il suffit de faire suivre l'énumération par la valeur désirée entre guillemets :

<! ATTLIST Elément Attribut (Valeur1 | Valeur2) "valeur par défaut" >

- **atomique**: il permet de définir un identifiant unique pour chaque élément grâce au mot clé *ID*.

Déclaration des attributs

Chacun de ces types d'attributs peut être suivi d'un mot clé particulier permettant de spécifier le niveau de nécessité de l'attribut :

- **#IMPLIED** signifie que l'attribut est optionnel, c'est-à-dire non obligatoire
- **#REQUIRED** signifie que l'attribut est obligatoire
- **#FIXED** signifie que l'attribut sera affecté d'une valeur par défaut s'il n'est pas défini. Il doit être immédiatement suivi de la valeur entre guillemets

```
<!ATTLIST destinataire
ecole CDATA #FIXED  "EISTI "
>
```

XML valide:
< destinataire ecole= " EISTI" />

XML non valide:
< destinataire ecole= " CNAM" />

XML : Un exemple de DTD (1)

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!ELEMENT etudiant (ou, matiere*)>
<!ATTLIST etudiant
      num ID #REQUIRED
       nom CDATA #REQUIRED
       prenom CDATA #REQUIRED
       annee (ing1|ing2|ing3) "autre" #REQUIRED
       age CDATA #IMPLIED
       >
<!ELEMENT ou  (#PCDATA)>
<!ATTLIST ou
     adresse CDATA #REQUIRED
     ville CDATA #REQUIRED
     cp CDATA #REQUIRED
     >
```


XML : Un exemple de DTD (2)

```
<!ELEMENT matiere(appreciation,note*) >
<!ATTLIST matiere
    indexe  ID #REQUIRED
    libelle CDATA #REQUIRED
    >
<!ELEMENT appreciation (#PCDATA| attention)+ >
<!ELEMENT attention (#PCDATA)>
<!ATTLIST note
    epreuve CDATA #REQUIRED
    valeur CDATA #IMPLIED
    >
```


XML : Un exemple de fichier XML

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE etudiant SYSTEM "d:/etudiant.dtd">
<etudiant  num="1" prenom="Maude" nom="Eme" annee="ing2">
    <ou adresse="1 rue du bo" ville="Paris" cp="75000" />
    <matiere  indexe= "I13"  libelle ="info">
        <appreciation>tu peux mieux faire</appreciation>
        <note epreuve="ds n° 1" valeur="10.5" />
        <note epreuve="ds n° 2" valeur="11" />
        <note epreuve="ds n° 3" valeur="12" />
    </matiere>
    <matiere  indexe= "C15"  libelle="comptabilité générale">
        <appreciation>belle prestation</appreciation>
        <note epreuve="ds" valeur="15" />
    </matiere>
</etudiant>
```