

Structures types de documents, DTD

1st November 2005

1 Introduction

1.1 But d'une DTD

But d'une DTD Permet de définir une règle pour la structure du document XML en précisant

- Eléments autorisés (nature et règle d'occurrence)
- Attributs autorisés (valeurs, occurrence, nature)
- Enchaînements (imbrications, séquences, choix) (Grammaire)
- Déclarations d'entités et notations

1.2 Lier DTD et fichier XML

Lier DTD et fichier XML

- DTD interne (préciser standalone):

```
<!DOCTYPE compte [  
    contenu de la DTD ] >
```

- DTD externe:

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

- DTD mixte:

```
<!DOCTYPE compte SYSTEM "compte.dtd" [  
    contenu supplémentaire ...] >
```

- DTD publiques: DocBook, TEI, XHTML, HTML, etc... sur Internet

```
<!DOCTYPE monDocument SYSTEM "http://www.docbook.org/xml/simple/1.1CR2/sdocbook.dtd">
```

Intérêt: réutilisabilité des DTDs, modularité

Commentaire:

Le processus de validation permet de vérifier qu'un document Xml est conforme avec une déclaration de type donnée

Ce processus est à distinguer de la vérification de forme qui contrôle qu'un document suit les règles de Syntaxe

1.3 Document Bien Formé / Valide

Document Bien Formé / Valide

- Document bien formé
 - un seul élément racine
 - balises correctement imbriquées
 - nom des balises bien formé
 - valeurs des attributs entre quotes
- Document valide :
 - la DTD est respectée
 - toutes les entités sont résolues

Un Parser XML validant (xmllint sous linux) permettra de vérifier la correspondance entre le document XML et sa DTD. Il repérera les éléments et attributs manquants, les noms d'éléments ou d'attributs erronés, les mauvaises imbrications, les mauvais contenus, les mauvais ordonnancements et les mauvaises valeurs d'attributs. De plus, il remplace automatiquement les valeurs définies comme entités internes.

Commentaire:

Xmllint est un programme sous linux qui permet de manipuler les fichiers xml.

Il permet de valider si un document xml respecte la SyntaxeXml ou s'il est valide selon une Dtd.

Pour utiliser xmllint, ouvrez un terminal et placez-vous dans le répertoire

où se trouve les documents xml à manipuler.

Utilisez l'option --valid pour vérifier la validité. Options principales

- * --noout-- supprime la sortie
- * --noent-- supprime les appels d'entités en les remplaçant par leur valeur
- * --valid permet de valider le document (la dtd est indiquée dans le document par un DOCTYPE
- * --dtdvalid url permet de valider le document selon la dtd indiquée à l'url.

2 Mise en Oeuvre

2.1 DTD : Les noms bien formés

DTD : Les noms bien formés Pour la formation des noms d'éléments, d'attributs, entités, ...

- Premier caractère est une lettre ou soulignement
- Caractères suivants sont des lettres, chiffres, soulignement ou point
- La séquence des 3 lettres xml est bannie, sous toutes combinaisons
- Pas d'espaces

Commentaire:

2.2 DTD : Déclaration d'éléments simples

DTD : Déclaration d'éléments simples

```
<!ELEMENT nom-de-la-balise modèle-de-contenu >
```

- élément vide:

```
<!ELEMENT soc EMPTY>
```

- élément libre:

```
<!ELEMENT soc ANY>
```

- élément texte:

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

Commentaire:

PCDATA: parsed Character Data

Commentaire:

2.3 DTD : Déclaration d'éléments composés

DTD : Déclaration d'éléments composés

- éléments fils:

– séquence:

```
<!ELEMENT gugus (nom,prenom,age)>
```

– alternative:

```
<!ELEMENT soc (nom|cigle)>
```

– 0 à n occurrences:

```
<!ELEMENT agenda (rdv)*>
```

– 1 à n occurrences:

```
<!ELEMENT voiture (roue)+>
```

– 0 ou 1 occurrence:

```
<!ELEMENT informaticien (copine?)>
```

```
<!ELEMENT personne (nom, prenom?, email+) >
```

- élément mixte

```
<!ELEMENT adresse (#PCDATA|cp|ville)*>
```

Interdit de mêler EMPTY à toute autre construction

2.4 Déclaration de la liste d'attributs

Déclaration de la liste d'attributs

```
<!ATTLIST lien
  href    CDATA #REQUIRED
  visited CDATA #IMPLIED >
```

- une ATTLIST par élément
- définition :

nom type modificateur

2.5 Attributs

Attributs

- Type
 - CDATA: texte
 - ID: label identifiant, il ne peut y avoir deux ID identiques dans un document XML
 - IDREF et IDREFS: référence vers une(des) ID (utilisée dans le document)
Commentaire:
on ne peut faire référence qu'à une ID déjà déclarée dans le document
 - Enumération:
`<!ATTLIST personne genre (homme|femme) "homme">`
 - NMTOKEN et NMTOKENS
Commentaire:
Indique que la valeur de l'attribut est une chaîne de caractères pouvant contenir une ou plusieurs lettres, des chiffres, des points (.), des tirets hauts (-) ou bas (_), et un double point (:).
 - `<!NOTATION flash SYSTEM "/usr/bin/flash.exe">`
déclaration de notation associant le format nommé flash à l'application spécifiée
- Modificateur
 - #REQUIRED
: attribut obligatoire et pas de valeur par défaut
 - #IMPLIED
: attribut facultatif et pas de valeur par défaut
 - une valeur par défaut
 - #FIXED
suivi d'une valeur (entre guillemets)

Commentaire:

2.6 Entités

Entités

Commentaire:

UN document XML peut être réparti au sein de plusieurs unités de stockage. Celles ci peuvent être logiques (on résume une expression par une version abrégée que l'on remplace au passage). Elles peuvent aussi être physiques (un autre fichier)

Une entité est une Storage Unit d'un document XML

- Entité analysée (parsed entity) : un morceau de texte d'un document XML
- Entité non analysée (unparsed entity) : une ressource externe référencée par un document XML (une image...)
- Entité générale : utilisée dans le corps d'un document XML
- Entité paramètre : utilisée seulement dans la DTD

Commentaire:

Dans une DTD, une entité dont la définition est complète et qui ne fait pas appel à un élément extérieur est dite -interne- sinon, elle est -externe-
Les entités internes sont toujours parsées, les externes peuvent ne pas l'être.

2.7 Entités comme Macros dans les documents XML

Entités comme Macros dans les documents XML Entité générale interne

- définition:

```
<!ENTITY nom "texte de remplacement">
```

- utilisation:

```
&nom;
```

- exemple:

```
<?xml version="1.0">
<!DOCTYPE texte [
  <!ELEMENT texte (#PCDATA)>
  <!ENTITY auteur "machin truc">
]>
<texte>&auteur;</texte>
```

Commentaire:

Permet de représenter des caractères spéciaux (ou réservés à XML).

Dans l'exemple, auteur sera remplacer par machin truc, on peut le voir sous xmllint

2.8 Entités comme Macros dans les DTDs

Entités comme Macros dans les DTDs Entité Paramètre

- définition:

```
<!ENTITY % nom "texte de remplacement">
```

- utilisation:

```
%nom;
```

seulement dans les DTD

- exemple:

```
<!ENTITY % url.att "href CDATA #REQUIRED">
<!ELEMENT bookmark (#PCDATA)>
<!ATTLIST bookmark %url.att;>
```

2.9 Entités pour la modularité des DTDs

Entités pour la modularité des DTDs Les entités paramètres permettent de faire des inclusions conditionnelles si elles sont combinées avec l'instruction IGNORE.

- L'instruction IGNORE écarte une partie de la Dtd :

```
<![IGNORE[ <!ELEMENT autres_activites (#PCDATA)> ] ]>
```

- Pour l'inclure, on remplace INGOORE par INCLUDE.

```
<![INCLUDE[<!ELEMENT autres_activites (#PCDATA)>] ]>
```

- Pour réaliser une Dtd modulaire, il suffit d'utiliser un appel d'entité.

```
<!ENTITY % autres "INCLUDE">
...
<[%autres;[ <!ELEMENT autres_activites (#PCDATA)> ] ]>
```

2.10 Entités pour l'inclusion

Entités pour l'inclusion Entité externe (paramètre)

- définition:

```
<!ENTITY nom SYSTEM "URI">
```

- utilisation:

```
&nom;
```

- principe: le fichier référencé par l'URI est inclus dans le fichier référençant en cas de validation

Commentaire:

```
***Attention coquille dans le poly, il s'agit d'entité paramètre  
Les entités externes, déclarées à partir d'un nom d'entité et d'une URI,  
permettent de référencer une valeur se situant dans un autre fichier.  
Cette valeur n'est pas nécessairement exprimée au format XML,  
mais elle doit respecter les règles propres à la partie  
du document dans laquelle elle va être insérée.
```

2.11 Entités pour l'inclusion : Exemple

Entités pour l'inclusion : Exemple fichier doc1.xml:

```
<?xml version="1.0"?>  
<texte>et hop</texte> >
```

et

```
<?xml version="1.0"?>  
<!DOCTYPE recueil [  
  <!ELEMENT recueil (texte)*>  
  <!ELEMENT texte (#PCDATA)>  
  <!ENTITY contenu SYSTEM "doc1.xml" >  
>  
<recueil>&contenu;</recueil>
```