Ontologies et Web Sémantique
Master 2

Chapitre I:XML

I.3-vlaidation d'un document XML

Document Type Definition DTD

Mr Bougherara.S

Plan du cours

- Introduction
- ***Validation d'un doc XML***
- ***Validation par DTD***

Introduction

Le contenu d'un document XML provient de plusieurs sources :

- Réseau :échange de données
- Saisie manuelle
- Automatique : d'un bd ou autre document

Erreurs possibles:

- Absence d'un champs
- Sequence
- …..

Introduction

Problème : une erreur engendre un mal fonctionnement du système

Solution : validation de document XML

Validation d'un doc XML

La validation a pour objectif de garantir qu'un documetn XML est bien structuré sans aucune incoherence.

•La validation renforce la qualité des échanges en contraignant l'émetteur et le consommateur de données à vérifier la cohérence des données structurées en XML.

•Par cohérence, il faut entendre à la fois le vocabulaire (éléments, attributs et espaces de noms) mais également, chose aussi importante, l'ordre et les quantités.

•En fin de compte,la validation revient à établir un visa sur le document XML.

Validation par DTD

- Une DTD (*Document Type Definition) est une forme de grammaire décrivant la structure d'un document XML*

- Une DTD *decrive les elements constituant un document XML:*
  - *Nom*
  - *Type*
  - *Cardinalité*
  - *Eléments fils*
  - *…..*

- Un document DTD est un document à base d'instruction

Contenu de la DTD

- Une DTD (*Document Type Definition) est une forme de grammaire décrivant la structure d'un document XML*

- Une DTD *decrive les elements constituant un document XML:*
  - *Nom*
  - *Type*
  - *Cardinalité*
  - *Eléments fils*
  - *…..*

- Un document DTD est un document à base d'instruction

Contenu de la DTD

•Une DTD est constituée de déclarations d'éléments, d'attributs et d'entités.

•Chacune de ces déclarations commence par la chaîne '<!' suivi d'un mot clé qui indique le type de déclaration.

• Les mots clés possibles sont ELEMENT, ATTLIST et ENTITY. La déclaration se termine par le caractère '>' .

Contenu de la DTD

1-commentaire

•Une DTD peut contenir des commentaires qui utilisent la syntaxe des commentaires XML délimités par les chaînes de caractères '<!--' et '-->'.

•Ceux-ci sont placés au même niveau que les déclarations d'éléments, d'attributs et d'entités.

• Ils ne peuvent pas apparaître à l'intérieur d'une déclaration.

Contenu de la DTD

2- Déclaration d'élément

Les déclarations d'éléments constituent le coeur des DTD car elles définissent la structure des documents valides.Elles spécifient quels doivent être les enfants de chaque élément et l'ordre de ces enfants.

La déclaration d'un élément est nécessaire pour qu'il puisse apparaître dans un document. Cette déclaration précise le nom et le type de l'élément. Le nom de l'élément doit être un nom XML et le type détermine.

Contenu de la DTD

2- Déclaration d'élément

le contenu valide de l'élément qui peut etre :

1. contenu textuel uniquement constitué de texte ,ou bien
2. contenu pur : uniquement constitué d'autres éléments, ou bien
3. contenu mixte qui mélange éléments et texte

Contenu de la DTD

2- Déclaration d'élément textuel

Un élément peut uniquement contenir du texte, Ce texte est formé de caractères, d'entités qui seront remplacées au moment du traitement et de sections littérales

Une déclaration d'élément prend la forme suivante :

<!ELEMENT *element* (#PCDATA)>

examples :

<!ELEMENT *nom* (#PCDATA)>
<!ELEMENT *adresse* (#PCDATA)>

Contenu de la DTD

2- Déclaration d'éléments purs

Le contenu d'un élément est pur lorsqu'il ne contient aucun texte et qu'il est uniquement constitué d'éléments qui sont ses enfants. Ces enfants peuvent, à leur tour, avoir des contenus textuels, purs ou mixtes.

Exemples :personne constitué de (nom,prenom,adresse)

Contenu de la DTD

2- Déclaration d'éléments purs

La déclaration d'un élément de contenu pur détermine quels peuvent être ses enfants et dans quel ordre ils doivent apparaître.

Une déclaration d'élément prend la forme suivante :
<!ELEMENT *element regexp*> où :

- *element :* specifie lee nom de l'élément

- *regexp :* décrit les suites autorisées d'enfants dans le contenu de l'élément , Cette expression rationnelle est construite à partir des noms d'éléments en utilisant les opérateurs ',', '|', '?', '*' , '+' et '(' et ')' pour former des groupes

Contenu de la DTD

2- Déclaration d'éléments purs

Regex

Opérateur	Signification
,	Mise en séquence
|	Choix
?	0 ou 1 occurrence
*	Répétition d'un nombre quelconque d'occurrences
+	Répétition d'un nombre non nul d'occurrences

Exemples

<!ELEMENT personne(nom, pren, adresse)>

<!ELEMENT plan (introduction**?,chapitre+,conclusion?)>**

<!ELEMENT chapitre (auteur***,paragraphe+)>**

<!ELEMENT livre (auteur**?,chapitre)+>**

Contenu de la DTD

2- Déclaration d'éléments purs Exemples

•<!ELEMENT elem (elem1, elem2, elem3)>

L'élément *elem* doit contenir un élément *elem1*, un élément *elem2* puis un élément *elem3* dans cet ordre.

•<!ELEMENT elem (elem1 | elem2 | elem3)>

L'élément *elem* doit contenir un seul des éléments *elem1*, *elem2* ou *elem3*.

•<!ELEMENT elem (elem1, elem2?, elem3)>

L'élément *elem* doit contenir un élément *elem1*, un ou zéro élément *elem2* puis un élément *elem3* dans cet ordre.

•<!ELEMENT elem (elem1, elem2*, elem3)>

L'élément *elem* doit contenir un élément *elem1*, une suite éventuellement vide d'éléments *elem2* et un élément *elem3* dans cet ordre.

Contenu de la DTD

2- Déclaration d'éléments purs

Exemples

- <!ELEMENT elem (elem1, (elem2 | elem4), elem3)>

L'élément *elem* doit contenir un élément *elem1*, un élément *elem2* ou un élément *elem4* puis un élément *elem3* dans cet ordre.

- <!ELEMENT elem (elem1, elem2, elem3)*>

L'élément *elem* doit contenir une suite d'éléments *elem1*, *elem2*, *elem3*, *elem1*, *elem2*, … jusqu'à un élément *elem3*.

- <!ELEMENT elem (elem1 | elem2 | elem3)*>

L'élément *elem* doit contenir une suite quelconque d'éléments *elem1*, *elem2* ou *elem3*.

- <!ELEMENT elem (elem1 | elem2 | elem3)+>

L'élément *elem* doit contenir une suite non vide d'éléments *elem1* , *elem2* ou *elem3*.

Contenu de la DTD

2- Déclaration d'éléments à Contenu mixed

•élément peut uniquement contenir du texte et d’autres elements.

•Une déclaration d'élément prend la forme suivante :
•<!ELEMENT *element* (#PCDATA | *element1* | ... | *elementN*)*>

•Il n'y a aucun contrôle sur le nombre d'occurrences de chacun des éléments et sur leur ordre d'apparition dans le contenu de l'élément ainsi déclaré et mot clé #PCDATA doit apparaître en premier avant tous les noms des éléments.

Contenu de la DTD

2- Déclaration d'éléments à Contenu vide

•le contenu de l'élément est nécessairement vide Cet élément peut uniquement avoir des attributs.
• Les éléments vides sont souvent utilisés pour des liens entre éléments.

•Une déclaration d'élément prend la forme suivante :<!ELEMENT *element* EMPTY>

•Exemple : **<!ELEMENT ref EMPTY>**

Contenu de la DTD

2- Déclaration d'éléments à Contenu libre

•n'impose aucune contrainte sur le contenu de l'élément. néamoins ,ce contenudoit, bien entendu, être bien formé et les éléments contenus doivent également être déclarés.

•Ce type de déclarations permet de déclarer des éléments dans une DTD en cours de mise au point afin de procéder à des essais de validation.

•<!ELEMENT *element* ANY>

3-Définition des attributs

Les attributs sont précisés dans l'instruction ATTLIST. Cette dernière, on doit préciser le nom de l'élément sur lequel s'applique le ou les attributs.

Syntaxe de declaration des attributs d'un element :

<!ATTLIST element …--détails attribut->

Syntaxe de **détails attribut** :

nom TYPE OBLIGATION VALEUR_PAR_DEFAUT

<!ATTLIST chapitre
titre **CDATA #REQUIRED**
auteur **CDATA #IMPLIED>**

Types d'attribut

•CDATA : du texte (*Character Data) ;*
• ID : identifiant unique (combinaison de chiffres et de lettres) ;
• IDREF : une référence vers un ID ;
• IDREFS : liste de références vers des ID (séparation par un blanc) ;

• NMTOKEN : un mot (donc pas de blanc) ;
• NMTOKENS : une liste de mots (séparation par un blanc) ;
• Une énumération de valeurs : chaque valeur est séparée par le caractère |.

Obligation d'attribut

L'OBLIGATION ne concerne pas les énumérations qui sont suivies d'une valeur par défaut.

Dans les autres cas, on l'exprime ainsi :

- #REQUIRED : attribut obligatoire.
- #IMPLIED : attribut optionnel.
- #FIXED : attribut toujours présent avec une valeur. Cela peut servir, par exemple, à imposer la présence d'un espace de noms.

Définition des attributs exempls

<!ATTLIST chapitre
titre **CDATA #REQUIRED**
auteur **CDATA #IMPLIED>**

<!ATTLIST crayon
couleur **(rouge|vert|bleu) >**

Définition d'entités

•Les entités sont déclarées par l'instruction ENTITY.
•l'entité associe un nom à une valeur.
Ce nom est employé dans le document XML comme une forme d'alias ou de raccourci vers la valeur suivant la syntaxe &nom.
•La valeur d'une entité peut être interne ou externe.
•les mot clés SYSTÈME et PUBLIC indique que l'entité est interne ou bien externe

Syntaxe :<!ENTITY nom "VALEUR">

<!**ENTITY nom SYSTEM "unTexte.txt">**

<!**ENTITY nom PUBLIC "unTexte.txt">**