

Université Abou-Bekr Belkaid –Tlemcen-
Faculté des Sciences
Département d'Informatique

Document Type Definition - DTD -

Année 2018 - 2019

Introduction

- XML fournit un moyen de vérifier la syntaxe d'un document grâce aux DTD (*Document Type Definition*).
- un document XML doit suivre scrupuleusement les conventions de notation XML et peut éventuellement faire référence à une DTD décrivant l'imbrication des éléments possibles.
- Un document XML *bien formé*, possédant une DTD et étant conforme à celle-ci est appelé *document valide*.

Qu'est-ce qu'une DTD?

- Une structure générique.
- Une DTD (Déclaration de Type de Document) permet de définir une structure type du document XML.
- Il s'agit d'une grammaire qui décrit la façon de construire des documents XML.
- Elles fournissent les règles que le parseur doit suivre pour la validation et l'interprétation correcte du document XML.

Pourquoi utiliser une DTD ?

- **Les documents valides présentent un double intérêt :**
 - ❖ Cela permet de n'avoir pas à réinventer des structures complexes disponibles et déjà largement utilisées : une DTD est réutilisable
 - ❖ Un document validé possède des propriétés intéressantes:
 - ✓ Il est bien formé
 - ✓ Il respecte la structure définie par la DTD
 - ✓ Toutes les références des entités peuvent être résolues.

Types de DTD

- La DTD est définie dans le document XML par la balise:

<!DOCTYPE

❖ **interne** au document XML: <!DOCTYPE élément_racine [...]>

❖ dans un fichier **externe**

✓ Privé : <!DOCTYPE élément_racine **SYSTEM** 'exemple.dtd'>

✓ Public : <!DOCTYPE html **PUBLIC** "-//W3C//**DTD XHTML 1.0 Strict**//**EN**"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

ou <!DOCTYPE doc **PUBLIC** "www.myweb.com/doc.dtd">

Structure de la DTD

- **La DTD définit la filiation des éléments**

- ❖ Quelle est la racine du document?

- ❖ Combien d'enfants ?

- ❖ Qui peut contenir du texte ?

- ❖ Qui a des attributs particuliers ?

- ❖ Quelles sont les valeurs possibles de ces attributs ?

Contenu d'une DTD

- **Une DTD peut contenir :**
 - ❖ Des déclarations d'éléments
 - ❖ Des déclarations de listes d'attributs
 - ❖ Des déclarations 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, ENTITY et NOTATION. La déclaration se termine par le caractère '>'.

Déclaration d'élément

- **<!ELEMENT nom_elt type_element>**

1. **Élément texte:**

#PCDATA (Parseable Character Data: données caractères parsées):

Correspond au texte contenu entre les balises de début et de fin de l'élément.

- ❖ Exemple

✓ **<! ELEMENT Nom (#PCDATA)>** --- DTD

✓ **<Nom>Victor Hugo</Nom>** --- XML

2. **Élément vide:**

EMPTY : L'élément ne contient pas de données spécifiques.

- ❖ Exemple

✓ **<!ELEMENT elt EMPTY>** --- DTD

✓ **<elt/>** --- XML

3. **Élément quelconque**

ANY : L'élément peut contenir tout type de donnée

<!ELEMENT elt ANY>

- Exemple

- ❖ <!ELEMENT tutorial (#PCDATA)>

- ❖ <tutorial>Ceci est un document XML</tutorial>

- ✓ Valide!

- ❖ <tutorial/>

- ✓ Valide

- ❖ <tutorial>

- <text> Ceci est un document XML </text>

- </tutorial>

- ✓ Non valide

Déclaration d'élément

4. Séquence d'éléments

l'élément est composé d'une **séquence** d'éléments (liste ordonnées d'éléments).

❖ Exemple:

✓ `<!ELEMENT elt (elt1, elt2, elt3)>` ---DTD---

✓ `<elt>`

`<elt1>(…)</elt1>`

`<elt2>(…)</elt2>`

`<elt3>(…)</elt3>`

`</elt>`

---XML---

Document valide

❖ `<elt>`

`<elt1>(…)</elt1>`

`<elt3>(…)</elt3>`

`</elt>`

Document non valide

(elt2 manquant)

❖ `<elt>`

`<elt1>(…)</elt1>`

`<elt3>(…)</elt3>`

`<elt2>(…)</elt2>`

`</elt>`

Document non valide

(l'ordre des éléments n'est pas respectés)

Déclaration d'élément

5. Choix d'éléments

donne le choix dans une liste de plusieurs éléments possibles.

❖ Exemple:

✓ `<!ELEMENT elt (elt1 | elt2 | elt3)>` ---DTD---

✓ `<elt>`

`<elt2>(...)</elt2>`

`</elt>`

---XML---

Document valide

❖ `<elt>`

`<elt2>(...)</elt2>`

`<elt3>(...)</elt3>`

`</elt>`

Document non valide

L'élément elt doit contenir un seul de ces éléments: elt1, elt2 ou elt3

Indicateurs d'occurrence

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?

❖ Exemple:

✓ `<!ELEMENT elt (elt1, elt2?, elt3+, elt4*)> ---DTD---`

✓ `<!ELEMENT elt (elt1* | elt2* | elt3*)>`

✓ `<elt>`

`<elt2>(...)</elt2>`

`<elt2>(...)</elt2>`

`</elt>`

✓ `<!ELEMENT elt (elt1 | elt2 | elt3)*>`

✓ `<elt>`

`<elt2>(...)</elt2>`

`<elt3>(...)</elt3>`

`<elt1>(...)</elt1>`

`</elt>`

Déclaration d'élément

6. Élément à contenu mixte

Un élément à contenu mixte peut contenir aussi bien du texte, que des éléments-enfants. Il se présente comme une liste de choix, avec des indicateurs d'occurrence bien choisis. Le texte contenu peut se trouver à n'importe quel endroit dans l'élément, et peut être une [section CDATA](#).

❖ Exemple:

✓ `<!ELEMENT citation (#PCDATA | auteur)*>` ---DTD

✓ `<citation>`

`<auteur>Shakespeare</auteur>Être ou ne pas être` ---XML

`</citation>`

Déclaration d'élément composé

- Exemple

- ❖ `<!ELEMENT XXX (AAA , BBB)>`

- `<!ELEMENT AAA (#PCDATA)>`

- `<!ELEMENT BBB (#PCDATA)>`

- ❖ `<XXX> <AAA>Début</AAA> <BBB>Fin</BBB> </XXX>`

- ✓ Valide

- ❖ `<XXX> <AAA/> <BBB/> </XXX>`

- ✓ Valide

- ❖ `<XXX> <AAA/> </XXX>`

- ✓ Non valide

- ❖ `<XXX> <BBB/> <AAA/> </XXX>`

- ✓ Non valide

- Exemple 2

- ❖ `<!ELEMENT XXX (AAA* , BBB)>`

- `<!ELEMENT AAA (#PCDATA)>`

- `<!ELEMENT BBB (#PCDATA)>`

- ❖ `<XXX> <AAA/> <BBB/> </XXX>`

- ✓ Valide

- ❖ `<XXX> <BBB/> </XXX>`

- ✓ Valide

- ❖ `<XXX> <AAA/> <AAA/> <AAA/> <AAA/> <AAA/> <AAA/> <AAA/>`

- `<AAA/> <BBB/> </XXX>`

- ✓ Valide

- ❖ `<XXX> </XXX>`

- ✓ Non valide

- ❖ `<XXX> <BBB/> <AAA/> </XXX>`

- ✓ Non valide : L'élément BBB doit suivre l'élément AAA

- ❖ `<XXX> <AAA/> <AAA/> <AAA/> <AAA/> <BBB/> <AAA/>`

- `<AAA/> </XXX>`

- ✓ Non valide

- Exemple 3

- ❖ <!ELEMENT XXX (AAA? , BBB+)>

- <!ELEMENT AAA (CCC? , DDD*)>

- <!ELEMENT BBB (CCC , DDD)>

- <!ELEMENT CCC (#PCDATA)>

- <!ELEMENT DDD (#PCDATA)>

- ❖ <XXX> <AAA> <CCC/><DDD/> </AAA> <BBB>
<CCC/><DDD/> </BBB> </XXX>

- ✓ Valide

- ❖ <XXX> <BBB> <CCC/><DDD/> </BBB> </XXX>

- ✓ Valide

- ❖ <XXX> <AAA> <CCC/><CCC/> <DDD/><DDD/>
</AAA> <BBB> <CCC/><DDD/> </BBB> </XXX>

Non valide : L'élément AAA peut contenir un élément CCC
au maximum


```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<!DOCTYPE biblio [
    <!ELEMENT biblio (livre)+>
    <!ELEMENT livre (titre, auteur, nb_tomes ?)>
    <!ELEMENT titre (#PCDATA)>
    <!ELEMENT auteur (#PCDATA)>
    <!ELEMENT nb_tomes (#PCDATA)>
    <!ATTLIST livre lang (en|fr) "fr">
]
```

```
>
<biblio>
  <livre lang="en">
    <titre>Les Misérables</titre>
    <auteur>Victor Hugo</auteur>
    <nb_tomes>3</nb_tomes>
  </livre>
  <livre>
    <titre>L'Assomoir</titre>
    <auteur>Émile Zola</auteur>
  </livre>
</biblio>
```

Exemple d'élément composé

<!ELEMENT personne (nom, prenom+, tel?, adresse)>

<!ELEMENT nom (#PCDATA) >

<!ELEMENT prenom (#PCDATA) >

<!ELEMENT tel (#PCDATA) >

<!ELEMENT Adresse (ANY) >

<personne>

<nom>Hugo</nom>

<prenom>Victor</prenom>

<prenom>Charles</prenom>

<tel>01</tel>

<adresse>

<rue/>

<ville>Paris</ville>

</adresse>

</personne>

Déclaration d'attributs

- **<! ATTLIST elt Attribut Type Mode >**
 - ❖ **elt** : spécifie l'élément auquel est attaché l'attribut
 - ❖ **Attribut**: est le nom de l'attribut déclaré
 - ❖ **Type**: définit le type de donnée de l'attribut choisi parmi:
 - ✓ **CDATA**
 - pour indiquer que la valeur de l'attribut est une séquence de caractères (y compris des entités caractères)
 - ✓ **Enumération**
 - Liste de valeurs limitée séparées par |
 - <! ATTLIST balise Attribut (Valeur1 | Valeur2 | ...) "valeurpardefaut">**
 - ✓ **ID et IDREF**
 - Clé et référence à clé
 - ❖ **Mode**: précise le caractère obligatoire ou non de l'attribut
 - ✓ **#REQUIRED, #IMPLIED, #FIXED** ou valeur par défaut

Exemple d'attributs

```
<!ELEMENT elt (...)>
```

```
<!ATTLIST elt attr CDATA "valeur">
```

```
<! ATTLIST personne num ID
```

```
age CDATA
```

```
genre (Masculin | Feminin ) >
```

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

```
<!ATTLIST auteur genre (Masculin | Feminin ) #REQUIRED
```

```
ville CDATA #IMPLIED>
```

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

```
<!ATTLIST editeur ville CDATA #FIXED "Paris">
```

Exemple de DTD

```
<!ELEMENT doc (livre* | article+) >
<!ELEMENT livre (titre, auteur+) >
<!ELEMENT article (titre, auteur*) >
<!ELEMENT titre(#PCDATA) >
<!ELEMENT auteur(nom, adresse) >
    <!ATTLIST auteur id ID #REQUIRED >
<!ELEMENT nom(prenom?, nomfamille) >
<!ELEMENT prenom (#PCDATA) >
<!ELEMENT nomfamille (#PCDATA) >
<!ELEMENT adresse (ANY) >
```

Exemple DTD interne avec attributs de type ID et IDREF

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [
    <!ELEMENT DOCUMENT(PERSONNE*)>
    <!ELEMENT PERSONNE (#PCDATA)>
        <!ATTLIST PERSONNE PNUM ID #REQUIRED>
        <!ATTLIST PERSONNE MERE IDREF #IMPLIED>
        <!ATTLIST PERSONNE PERE IDREF #IMPLIED>
    ]>
<DOCUMENT>
    <PERSONNE PNUM = "P1">Marie</PERSONNE>
    <PERSONNE PNUM = "P2">Jean</PERSONNE>
    <PERSONNE PNUM = "P3" MERE="P1" PERE="P2">Pierre
    </PERSONNE>
    <PERSONNE PNUM = "P4" MERE="P1" PERE="P2">Julie
    </PERSONNE>
</DOCUMENT>
```

Les entités

- **Remplacement** d'une chaîne de caractères par un **symbole**, puis l'utilisation du symbole à la place de cette chaîne.

`<!ENTITY nom "chaîne">`

- Elles peuvent être utilisées aussi bien dans la DTD que dans le document XML.
- Il existe deux catégories d'entités:
 - ❖ Entités générales
 - ❖ Entités paramètres

Entités générales

- Les *entités générales* (sauf les entités prédéfinies) sont définies dans la DTD et utilisées dans les **documents XML correspondants**.
- Une *entité générale* est toujours invoquée sous la forme **&symbole;** au sein d'un document là où devrait apparaître le *texte de remplacement* associé.

Entités générales

- **Les Entités de caractères**

- ❖ Il existe des entités prédéfinies pour certains caractères spéciaux:

&	&
<	<
>	>
"	"
'	'

- ❖ Il existe des entités numériques du type

- ✓ $&\#n$; (où n est une valeur décimale)

- ✓ $&\#xh$; (où h est une valeur hexadécimale)

Exemple: $&\#960$; pour le symbole π

Entités générales

- **Les entités internes**

Elles sont utilisées pour ne pas avoir à taper de longues chaînes de caractères plusieurs fois dans un document.

- Elle sont déclarées dans les DTD internes.

- ❖ Note :

Si la chaîne de caractères doit être modifiée, on la modifie seulement une fois.

- ❖ Exemple:

```
<!ENTITY tic "les technologies de l'information et de la  
communication">
```

```
<ex>Le séminaire porte sur &tic;</ex>
```

Entités générales

- **Les entités externes**

Dans ce cas le texte de remplacement est défini dans un autre fichier.

```
<!ENTITY nom_entité SYSTEM "uri_remplacement">
```

- ❖ Exemple:

```
<!ENTITY chap1 SYSTEM "chapitre1.xml">
```

```
<!ENTITY chap2 SYSTEM "chapitre2.xml">
```

```
<livre>
```

```
&chap1;
```

```
&chap2;
```

```
</livre>
```

Entités paramètres

- Les *entités paramètres* sont définies dans la DTD et utilisées dans la **DTD** elle-même.
- Elles permettent:
 - ❖ d'éviter des répétitions dans les définitions des éléments ou des attributs,
 - ❖ d'intégrer des fragments de DTD mémorisés dans un fichier, ce qui est utile pour fractionner les grosses DTD.
- Ces entités paramètres doivent être définies dans une DTD externe (elles ne sont pas prises en considération si elles sont définies en interne).
 - ❖ Exemple :

```
<!ENTITY %    genres    "homme|femme">  
<!ATTLIST  auteur  genre    (%genres ;)    #REQUIRED>
```

Entités paramètres

- Un autre exemple :

```
<!ENTITY % commun "niveau, couleur">
```

```
<!ELEMENT rectangle (%commun;, sommet+)>
```

```
<!ELEMENT triangle (%commun;, sommet+)>
```

```
<!ELEMENT disque (%commun;, centre, rayon)>
```

- Il est équivalent au fragment de DTD suivant :

```
<!ELEMENT rectangle (niveau, couleur, sommet+)>
```

```
<!ELEMENT triangle (niveau, couleur, sommet+)>
```

```
<!ELEMENT disque (niveau, couleur, centre, rayon)>
```

Merci pour votre attention