Technologies XML et Web Sémantique

Définir le format d'un document: DTD

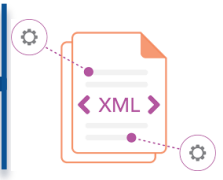
ENSA

جامعة شعيب الدكالي

Université Chouaïb Doukkali

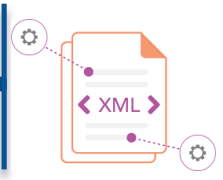
Présenté par: Pr. Fahd KALLOUBI

Introduction



- Il peut être parfois nécessaire de préciser les balises et attributs auxquels on a droit lors de la rédaction d'un documen XML
  - par exemple si l'on veut pouvoir partager le même type de document avec une communauté d'autres rédacteurs
- Deux solutions sont possibles :
  - Les schémas XML
  - Et les « Document Type Definition »
- Ce cours présente la vérification d'un document à l'aide de:
  - Les Document Type Definitions (DTD) venant de la norme SGML

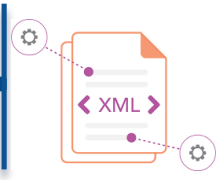
Introduction (rappel)



- Il y a deux niveaux de correction pour un document XML :
  - Un document XML bien formé (well formed) respecte les règles syntaxiques d'écriture XML : écriture des balises, imbrication des éléments, entités, etc. C'est le niveau de base de la validation.
  - Un document valide respecte des règles supplémentaires sur les noms, attributs et organisation des éléments.

La validation est cruciale pour une entreprise telle qu'une banque qui gère des transactions représentées en XML. S'il y a des erreurs dans les documents, cela peut compromettre l'entreprise.

Introduction: processus de validation



- D'abord, il faut disposer d'un fichier contenant des règles.
- Il existe plusieurs langages pour faire cela : DTD, XML Schemas, RelaxNG et Schematron.
  - Ces langages modélisent des règles de validité plus ou moins précises et d'une manière plus ou moins lisible.
- Chaque document XML est comparé à ce fichier de règles, à l'aide d'un outil de validation : xmlstarlet, xmllint, rnv. . .
- La validation indique que soit le document est valide, soit il contient des erreurs comme :
  - tel attribut de tel élément contient une valeur interdite par telle contrainte, il manque tel sous-élément dans tel élément, etc.

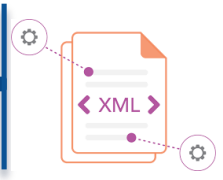
Document Type Definitions (DTD): présentation

Un Document Type Definitions est une liste de règles définies au début d'un document XML pour permettre sa validation avant sa lecture. Elle est déclarée par un élément spécial DOCTYPE juste après le prologue et avant la racine :

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Filiere ... >
<Filiere>
    <cours heure="200">XML</cours>
    <cours heure="140">Web sémantique</cours>
    <cours heure="220">Machine learning</cours>
</Filiere>
```

Les DTD sont issues de la norme SGML et n'ont pas la syntaxe XML.

Document Type Definitions (DTD): intégration



- Une DTD peut être :
  - interne, intégrée au document. C'est signalé par un couple [ ] :

```
<!DOCTYPE itineraire [
        ...
]>
```

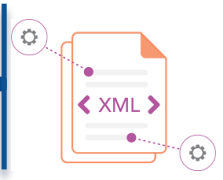
  - externe, dans un autre fichier, signalé par SYSTEM suivi de l'URL du fichier :

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

  - mixte, il y a à la fois un fichier et des définitions locales :

```
<!DOCTYPE itineraire SYSTEM "itineraire.dtd" [
        ...
]>
```

DTD: contenu d'une DTD



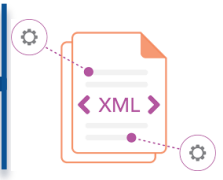
- Une DTD contient des règles comme celles-ci :

```
<!ELEMENT itineraire (etape+)>
<!ATTLIST itineraire nom CDATA #IMPLIED>
<!ELEMENT etape (#PCDATA)>
<!ATTLIST etape distance CDATA #REQUIRED>
```

Ce sont des règles de construction :

- des éléments (ELEMENT) : leur nom et le contenu autorisé,
- des attributs (ATTLIST) : nom et options.

DTD: racine du document



Le nom présent après le mot-clé DOCTYPE indique la racine du document. C'est un élément qui est défini dans la DTD.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Filiere [
    <!ELEMENT Filiere (cours+)>
    <!ATTLIST Filiere nom CDATA #IMPLIED>
    <!ELEMENT cours (#PCDATA)>
    <!ATTLIST cours heure CDATA #REQUIRED>
]>
<Filiere nom="2ITE">
    <cours heure="200">XML</cours>
    <cours heure="140">Web sémantique</cours>
    <cours heure="220">Machine learning</cours>
</Filiere>
```

DTD: définition d'un élément

La règle <!ELEMENT nom contenu> permet de définir un élément : son nom et ce qu'il peut y avoir entre ses balises ouvrante et fermante.

La définition du contenu peut prendre différentes formes :

- EMPTY : signifie que l'élément doit être vide,
- ANY : signifie que l'élément peut contenir n'importe quels éléments et textes (rien ne sera testé),
- (#PCDATA) : signifie que l'élément ne contient que des textes,
- (définitions de sous-éléments) : spécifie les sous-éléments qui peuvent être employés, voir plus loin.

Utilisation

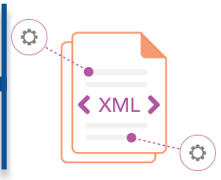
- Any et EMPTY : sans parenthèses
- (#PCDATA) : avec parenthèses

Les DTD ne sont pas exigeantes sur les types des données.

DTD: définition d'un élément (exemple)


```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Filiere [
    <!ELEMENT Filiere (professeur?, cours+, acheminement*)>
    <!ELEMENT professeur EMPTY>
    <!ELEMENT cours (#PCDATA)>
    <!ELEMENT acheminement ANY>
]>
<Filiere>
    <professeur/>
    <cours>xml</cours>
    <cours>web sémantique</cours>
    <acheminement>
        <cours>xml</cours><cours>web sémantique</cours>
    </acheminement>
</Filiere>
```

DTD: déclaration du contenu (1/4)



- Donnees textuelles parses : #PCDATA
  - Syntaxe: <!ELEMENT element (#PCDATA)>
  - Exemple: <!ELEMENT nom (#PCDATA)>
- Sous element
  - Syntaxe: <!ELEMENT element (souselement)>
  - Exemple: <!ELEMENT fax (numero)>
- Séquence de sous éléments (ordonnés)
  - Syntaxe: <!ELEMENT element (souselement1, souselement2, ...)>
  - Exemple: <!ELEMENT identite (prenom, nom)>

DTD: déclaration du contenu (2/4)

- Nombre d'occurrence de sous éléments
  - ? : l'élément ou groupe est optionnel
  - + : l'élément ou groupe apparait au moins 1 fois
  - * : l'élément ou groupe apparait de 0 a N fois
  - Exemple: <!ELEMENT identite (nom, nomjeunefille?, prenom+, surnom*)>
- Choix exclusif entre plusieurs sous éléments possibles
  - Syntaxe: <!ELEMENT element (souselement1 | souselement2 | ...)>
  - Exemple: <!ELEMENT situation (celibataire | marie | divorce)>

- Groupe (parenthèses) : permet de combiner les opérateurs précédents (ordonnés)
  - Exemple: - <!ELEMENT cercle (centre, (rayon | diametre))>

    - <!ELEMENT contact (nom, prenom?, adresse, (telfixe | telmobile | telbureau)*)>

DTD: déclaration du contenu (3/4)

Déclaration d'éléments composés:

Opérateur	Signification	Exemple
+	L'élément doit être présent au moins une fois	A+
*	L'élément peut être présent plusieurs fois (ou aucune)	A*
?	L'élément peut être optionnellement présent	A?
|	L'élément A ou B peuvent être présents (pas les deux)	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 opérations	(A,B)+

Attention à l'ordre

<!ELEMENT mont (devise, somme) > ≠ <!ELEMENT mont (somme, devise) >

DTD: déclaration du contenu (4/4)

- Contenu mixte : mélange de données textuelles et de sous éléments
  - La seule possibilité est de combiner #PCDATA, | et *
  - #PCDATA doit toujours être en tête

```
<!ELEMENT element (#PCDATA | souselement1 | ...)*>
<!ELEMENT paragraphe (#PCDATA | citation)*>
```

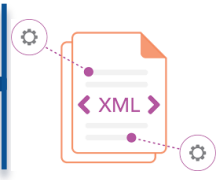
- élément vide : l'élément n'a pas de contenu

```
<!ELEMENT element EMPTY>
```

- contenu indetermine

```
<!ELEMENT element ANY>
```

DTD: déclaration d'attributs



- Structure d'une declaration d'attributs

<!ATTLIST element attribut$_1$ type déclaration par défaut
attribut$_2$ type déclaration par défaut
...
attribut$_n$ type déclaration par défaut
>

<!ATTLIST image src CDATA #REQUIRED
width CDATA #REQUIRED
height CDATA #REQUIRED
alt CDATA #IMPLIED
>

DTD: Types d'attributs (1/2)

- Il y a plusieurs types d'attributs
  - CDATA : n'importe quelle chaîne de caracteres
  - NMTOKEN : unité lexicale nominal = chaine de caractères obéissant aux règles d'un nom XML (element) excepte qu'il n'y a pas de restriction sur le premier caractère
  - NMTOKENS : série de NMTOKEN sépares par des blancs
  - énumération : un des NMTOKEN séparé par | dans une liste

    <!ATTLIST element attribut (NOMTOKEN1 | NMTOKEN2 | ...)>

    <!ATTLIST date mois (Janvier | Février | Mars | ...)>
  - ID : nom XML unique dans le document un élément ne peut avoir qu'un attribut de ce type

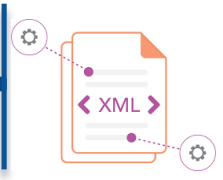
DTD: Types d'attributs (2/2)

- IDREF : fait référence a la valeur d'un attribut de type ID permet de créer des relations entre des éléments, exemple : des projets, des personnes) des personnes qui travaillent dans un ou plusieurs projets
- IDREFS : série de références a des ID séparées par des blancs
- ENTITY : contient le nom d'une entité non parsée
- ENTITIES : série de noms d'entités non parsés, separés par des blancs
- NOTATION : fait référence a une NOTATION déclarée dans la DTD

```
<!NOTATION png SYSTEM "image/png">
<!NOTATION jpg SYSTEM "image/jpeg">
<!ATTLIST image type NOTATION (png | jpg) #REQUIRED>
```

http://www.w3schools.com/xml/xml_dtd_attributes.asp

DTD: déclaration par défaut



- #IMPLIED : l'attribut est optionnel
- #REQUIRED : l'attribut est obligatoire
- #FIXED : la valeur de l'attribut est fixe et non modifiable

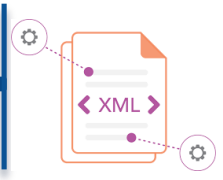
L'attribut est présent dans l'élément même si il est omis

```
<!ATTLIST element attribut type #FIXED
"valeur obligatoire">
```

- littéral : la valeur par défaut de l'attribut est spécifiée

```
<!ATTLIST element attribut type "valeur par default">
```

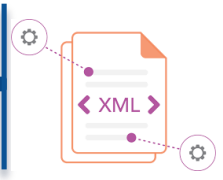
DTD: définition d'entités (rappel)



Les entités sont des symboles qui représentent des morceaux d'arbre XML ou des textes.

```
...
<!ENTITY copyright "© ENSA">
<!ENTITY depart "<etape>Point de départ</etape>">
<!ENTITY equipement SYSTEM "equipement.xml">
]>
<itineraire>
<auteur>&copyright;</auteur>
&equipement;
&depart;
</itineraire>
```

DTD: Entités paramètres



Il est possible de définir une entité utilisée dans la DTD elle-même.

La syntaxe est légèrement différente.

- Syntaxe:

<!ENTITY % entite parametre "texte de substitution">

- Exemple:

```
<!ENTITY % reference "(auteur, titre, date)">
<!ELEMENT livre (domaine, %reference;, ISBN, prix)>
<!ELEMENT email (%reference;, destinataires)>
```

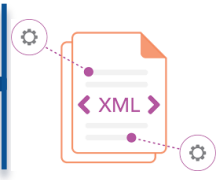
Notez le % entre ENTITY et son nom, également la référence d'entité s'écrit %nom; et pas &nom;

DTD: Exercice

Rédiger une DTD pour une bibliographie. Cette bibliographie :

- contient des livres et des articles ;
- les informations nécessaires pour un livre sont :
  - son titre général ;
  - les noms des auteurs ;
  - ses tomes et pour chaque tome, leur nombre de pages ;
  - des informations générales sur son édition comme par exemple le nom de l'éditeur, le lieu d'édition, le lieu d'impression, son numéro ISBN ;
- les informations nécessaires pour un article sont :
  - son titre ;
  - les noms des auteurs ;
  - ses références de publication : nom du journal, numéro des pages, année de publication et numéro du journal
- on réservera aussi un champ optionnel pour un avis personnel.

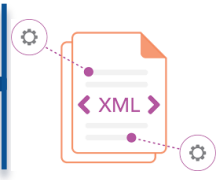
DTD: Exercice




```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!ELEMENT biblio (livre,article)*>
        <!ELEMENT livre (titre, auteur+, tome*, edition, avis?)>
                <!ELEMENT titre (#PCDATA)>
                <!ELEMENT auteur (#PCDATA)>
                <!ELEMENT tome (nb_pages)>
                        <!ELEMENT nb_pages (#PCDATA)>
                <!ELEMENT edition (editeur, lieu_edition, lieu_impression, isbn)>
                        <!ELEMENT editeur (#PCDATA)>
                        <!ELEMENT lieu_edition (#PCDATA)>
                        <!ELEMENT lieu_impression (#PCDATA)>
                        <!ELEMENT isbn (#PCDATA)>
                <!ELEMENT avis (#PCDATA)>
        <!ELEMENT article (titre, auteur+, journal)>
                <!ELEMENT journal (nom_journal, page, num_journal, annee)>
                        <!ELEMENT nom_journal (#PCDATA)>
                        <!ELEMENT page (#PCDATA)>
                        <!ELEMENT num_journal (#PCDATA)>
                        <!ELEMENT annee (#PCDATA)>
```

DTD: Limitations

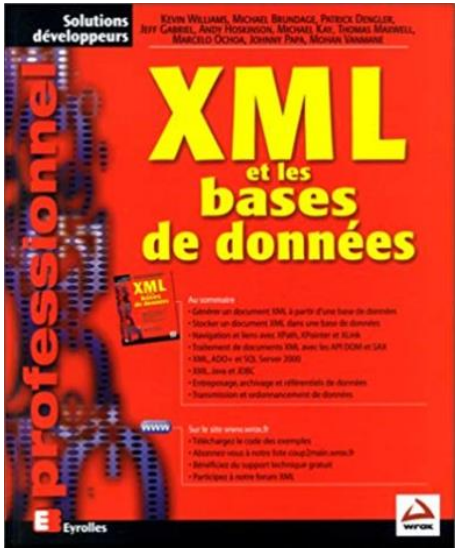


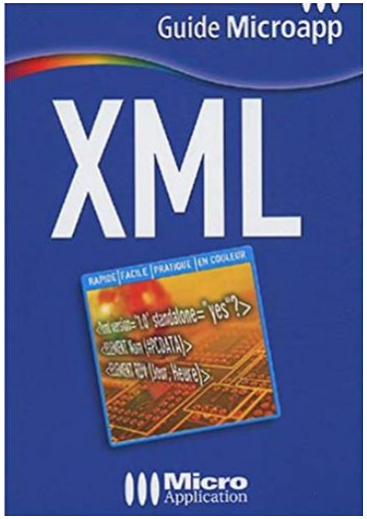
- Premièrement, les DTD ne sont pas au format XML. Cela signifie qu'il est nécessaire d'utiliser un outil spécial pour manipuler un tel fichier, différent de celui utilisé pour l'édition du fichier XML.

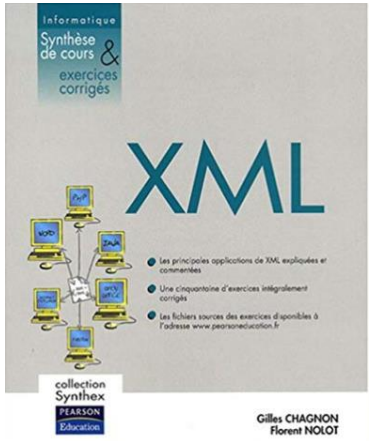
- Deuxièmement, les DTD ne supportent pas les « espaces de nom ». En pratique, cela implique qu'il n'est pas possible d'importer des définitions de balises définies par ailleurs dans un fichier XML défini par une DTD.

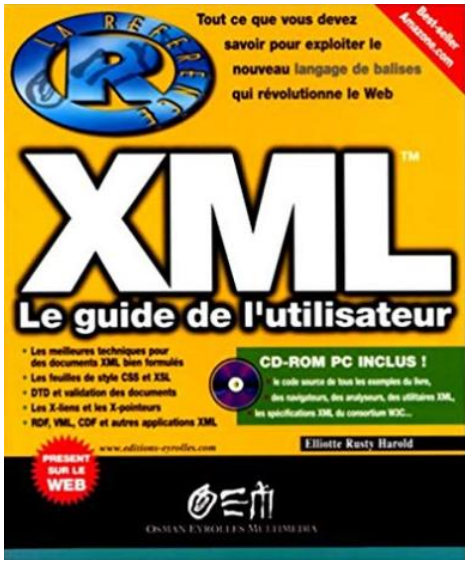
- Troisièmement, le « typage » des données (c'est-à-dire la possibilité de spécifier par exemple qu'un attribut ne doit être qu'un nombre entier) est extrêmement limité.

Littérature – quelques sources









- XML in a Nutshell, O'Reilly (Elliotte Rusty Harold & W. ScottMeans)
- Introduction a XML, O'Reilly (Erik T. Ray)
- XML langage et applications, Eyrolles (Alain Michard)