2éme année Master Informatique

Option: **Génie Logiciel et Systèmes Distribués**

Université de Biskra

Année 2021/2022

XML : eXtensible Mark-up Language

Dr. Mohamed Lamine KERDOUDI

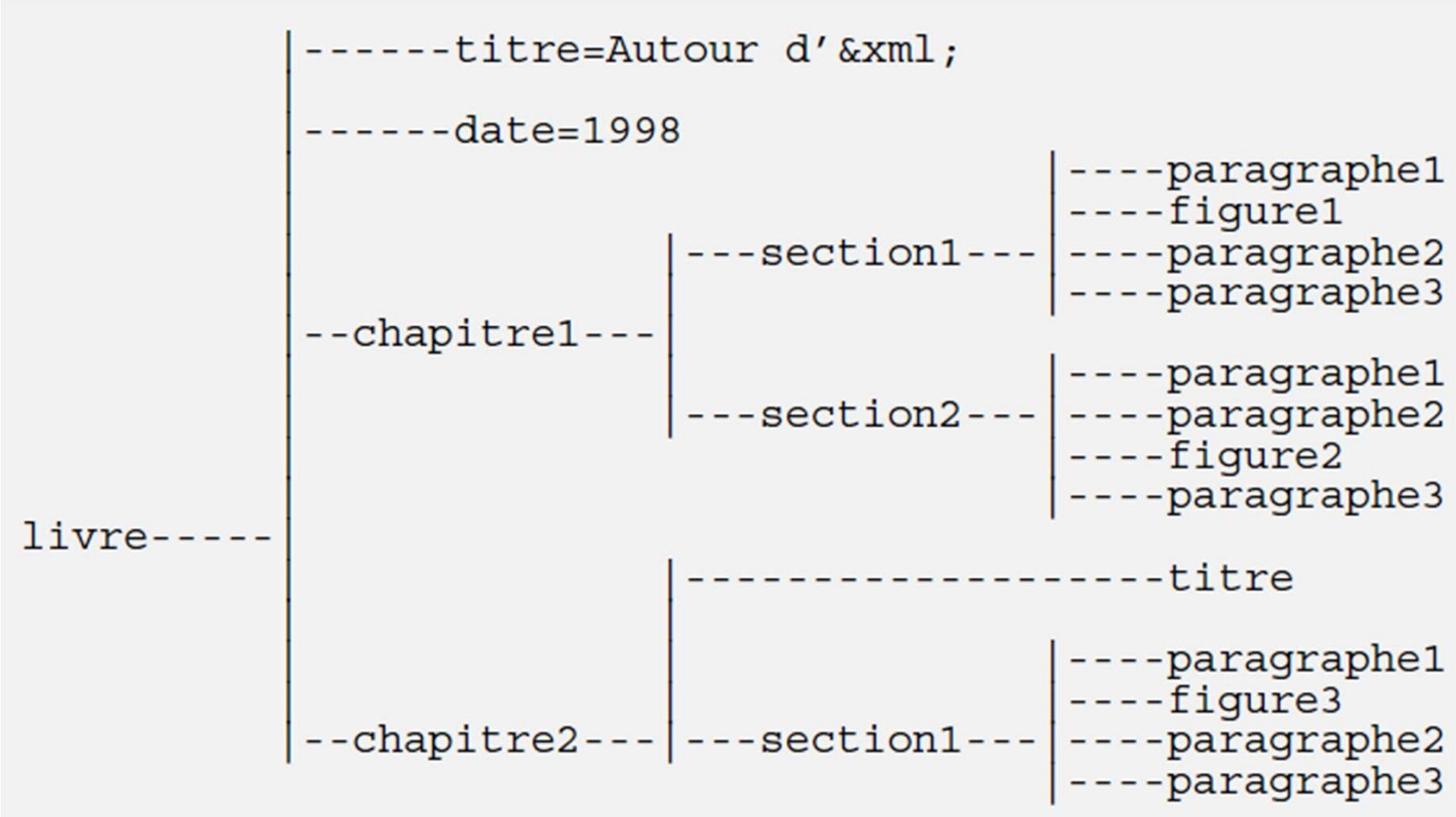
Email : **l.kerdoudi@univ-biskra.dz**

1. Introduction

- **Documents de l'entreprise**
  - Documents commerciaux
    - catalogues, fiches produits, factures, …
  - Documents techniques
    - spécifications techniques, manuels utilisateur, manuels d 'entretien, …
  - Documents qualités
    - suivi de fabrication de lots, rapports d'incident, …
- **Gestion électronique de documents**
  - Représentation ***uniforme*** *voir* ***normalisé***
  - Archivage
  - Recherche

1. Introduction

- **Document structuré**
  - Comporte des parties ayant une signification particulière



- **La GED utilise des documents structurés**
  - Je recherche tous les documents dont le titre contient XML !

1. Introduction

❑ Limites de HTML

- **HTML 4.0 et DHTML**
  - pas de validation de la structure
- **SGML : Standard Generalized Markup Language**
  - structure « sémantique » et type de document (DTD)
  - mais reste compliqué et non intégré aux browsers
- **XML**
  - principes de SGML mais simplification de la DTD

1. Introduction

❑ Qu'est-ce que XML?

- eXtensible Mark-up Language
- Défini par le W3C
- permet la définition de familles de langages de balises
- fait aussi référence à une famille de technologies

❑ Exemple

```
<fiche-identite>
    <nom>Parrain</nom>
   <prenom>Anne</prenom>
</fiche-identite>
```

1. Introduction

❑ A quoi sert XML?

- Représenter des données pour les manipuler, les échanger, les interroger.
- **Exemples de documents**
  - Documents de bureautique : OpenOffice
  - Documents texte : DocBook,. . .
  - Données informatiques : configurations. . .
  - Données échangées : XHTML, jabber, web services,. . .
  - Données stockées : bases de données XML
  - Beaucoup d'autres choses nouvelles, chaque jour ou presque !

Un Survol sur XML

- XML est utilisé pour décrire et stocker des documents et des données dans un format standard qui peut être facilement transporté par l'intermédiaire des protocoles standard d'Internet.

- En plus, il permet de séparer la structure d'un document, sa présentation et son contenu, de telle sorte ou le contenu d'un document XML peut être **présentés avec plusieurs manières**, utilisant des feuilles de style différentes.

- Un des avantages XML est le fait qu'il est un méta-langage qui permet de construire d'autres langages propres à des domaines d'activités.

XML : Concepts fondamentaux

1. Eléments et attributs d' XML

- Les documents **bien formés** de XML peuvent contenir des **éléments**, des **attributs**, et le **texte**.

a) Les éléments :

- un élément a toujours une balise d'ouverture et autre de fermeture : **<element></element>**
- Le nom de l'élément peut contenir des lettres et des nombre…
- Un élément ne peut pas commencer par un non alphabétique

b) Les attributs :

- Contiennent des valeurs qui sont **associées** à un **élément** et sont insérés dans la partie du balise ouverte d'un élément :

<element **attribute** ="**value**"> </element>

XML : Concepts fondamentaux

c) Les textes :

- sont localisés entre deux balises, généralement représentent la donnée associe à **l'élément** :

<element attribute =”value”> **Texte** </element>

❑ **Exemple**

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<note>
  <to>Ali</to>
  <from>Mohamed</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

XML : Concepts fondamentaux

- Le **prologue** des documents XML joue trois rôles importants:
  - Préciser qu'il s'agit d'un document XML
  - Identifier le jeu (codage) de caractères utilisé
  - Identifier la grammaire (DTD) utilisée

- Le **prologue** est **facultatif**, mais important car il précise des info. importantes pour les processeurs XML.

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="macss.css" type="text/css"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

- La déclaration XML
<?xml attribut="valeur" [attribut="valeur"] ?>

Définition de Type de document DTD :

- Une **définition d'un document XML** est un ensemble de **règles** que l'on impose au document.
  - Ces règles permettent de décrire la façon dont le document XML doit être construit.

- Le DTD précise la grammaire que doit suivre le document.

- Par exemple, ces règles peuvent être :
  - **imposer la présence d'un attribut ou d'une balise**,
  - imposer **l'ordre** d'apparition des balises dans le document
  - ou encore, imposer le **type** d'une donnée (nombre entier, chaîne, etc.).

- **Document valide** : **est un document bien formé conforme à une définition. (Respecter DTD).**

Définition de Type de document DTD :

- **Pourquoi écrire des définitions ?**
  - Associer une définition à un document oblige à une certaine rigueur dans l'écriture de vos données XML.
  - C'est d'autant plus important lorsque plusieurs personnes travaillent sur un même document.
  - La définition impose ainsi une écriture uniforme que tout le monde doit respecter.

Définition de Type de document DTD :

1. Eléments

❑ ***Syntaxe :*** Pour définir les règles portant sur les **balises**, on utilise le mot clef **ELEMENT**.

<!ELEMENT balise (contenu)>

- Une règle peut donc se découper en 3 mots clefs :

ELEMENT, balise et contenu.

❑ ***Retour sur la balise***

- Le mot-clef **balise** est à remplacer par le nom de la balise à laquelle vous souhaitez appliquer la règle.
- Par exemple: <nom>Bensalah</nom>

  On écrira alors :

  <!ELEMENT nom (contenu)>

Définition de Type de document DTD :

❑ ***Retour sur le contenu :*** Cet emplacement a pour vocation de décrire ce que doit contenir la balise : est-ce une autre balise ou est-ce une valeur ?

1. Cas d'une balise en contenant une autre

- Par exemple, regardons la règle suivante :

  <!ELEMENT personne (nom)>

  <!-- suite de la DTD -->

- Cette règle signifie que la balise<personne/>contient la balise<nom/>.

- Le document XML respectant cette règle ressemble donc à :

```
<personne>
    <nom> Hafidi </nom>
</personne>
```

Définition de Type de document DTD :

- Nous n'avons défini aucune règle pour la balise <nom/>.

Le document n'est, par conséquent, pas valide.

- En effet, dans une DTD, il est impératif de décrire tout le

document sans exception.

- Des balises qui n'apparaissent pas dans la DTD ne

peuvent pas être utilisées dans le document XML.

2. Cas d'une balise contenant une valeur

- Dans le cas où notre balise contient une **valeur simple**, on utilisera le mot clef #PCDATA
- Une **valeur simple** désigne par exemple une chaîne de caractères, un entier, un nombre décimal, un caractère, etc.

Définition de Type de document DTD :

- En se basant sur l'exemple précédent :

```
<personne>
    <nom> Hafidi </nom>
</personne>
```

- Nous pouvons maintenant compléter notre DTD en ajoutant une règle pour la balise <nom/>. Par exemple, la balise contienne une valeur simple, on écrira :

<!ELEMENT nom (#PCDATA)>

Au final, la DTD de notre document XML est donc la suivante :

<!ELEMENT personne (nom)>
<!ELEMENT nom (#PCDATA)>

3) Cas d'une balise vide

Il est également possible d'indiquer qu'une balise ne contient rien grâce au mot-clef **EMPTY**.

Définition de Type de document DTD :

Prenons les règles suivantes :

<!ELEMENT personne (nom)>

<!**ELEMENT** **nom** **EMPTY**>

Le document XML répondant à la définition DTD précédente est le suivant :

<**personne**>

  <**nom** ></**nom**>

</**personne**>

À noter : lors de l'utilisation du mot clef **EMPTY**, l'usage des parenthèses n'est pas obligatoire !

Définition de Type de document DTD :

4) Cas d'une balise pouvant tout contenir

- Une balise qui peut tout contenir, c'est à dire, une autre balise, une valeur simple ou tout simplement être vide. Dans ce cas, on utilise le mot-clef **ANY**.
- Prenons la règle suivante :

```
<!ELEMENT personne (nom)>
<!ELEMENT nom ANY>
```

- Les documents XML suivants sont bien valides :

```
<!-- valeur simple -->
<personne>
<nom> Hafidi </nom>
</personne>
<!-- vide -->
<personne>
 <nom > </nom>
</personne>
```

Définition de Type de document DTD :

- Bien que le mot-clef **ANY** existe, il est souvent déconseillé de l'utiliser afin de restreindre le plus possible la liberté de rédaction du document XML.
- Comme pour le mot-clef **EMPTY**, l'usage des parenthèses n'est pas obligatoire pour le mot-clef **ANY**!

Définition de Type de document DTD :

❖ Structurer le contenu des balises

- Par exemple, un répertoire contient généralement un nombre variable de personnes,

➔ Il faut donc permettre au document XML d'être valide quel que soit le nombre de personnes qu'il contient.

1. Séquence

- Une **séquence** permet de décrire l'enchaînement imposé des balises.
- Il suffit d'indiquer le nom des balises en les séparant par des ***virgules***.

Définition de Type de document DTD :

<!ELEMENT balise (balise2, balise3, balise4, balise5, etc.)>

❑ **Exemple**:

<!ELEMENT personne (nom, prenom, age)>

<!ELEMENT nom (#PCDATA)>

<!ELEMENT prenom (#PCDATA)>

<!ELEMENT age (#PCDATA)>

- Cette définition impose que la balise <**personne** /> contienne obligatoirement les balises<**nom** />,<**prenom** /> et<**age** /> **dans cet ordre**.

Définition de Type de document DTD :

- Regardons alors la validité des documents XML qui suivent:

```
<!-- valide -->
<personne>
        <nom>Hafidi</nom>
         <prenom> Houssem Eddine </prenom>
         <age>24</age>
</personne>
<!-- invalide -->
<!-- les balises ne sont pas dans le bon ordre -->
<personne>
        <prenom> Houssem Eddine </prenom>
         <nom>Hafidi</nom>
         <age>24</age>
</personne>
```

Définition de Type de document DTD :

- Regardons alors la validité des documents XML qui suivent:

```
<!-- invalide -->
 <!-- il manque une balise -->
<personne>
        <prenom> Houssem Eddine </prenom>
        <age>24</age>
</personne>
<!-- invalide -->
<!-- il y a une balise en trop, qui plus est non déclarée -->
<personne>
       <nom>Hafidi</nom>
       <prenom> Houssem Eddine </prenom>
        <age>24</age>
        <date>12/12/2012</date>
</personne>
```

Définition de Type de document DTD :

2. *Liste de choix*

- Une **liste de choix** permet de dire qu'une balise contient **l'une des balises décrites**.
- Il suffit d'indiquer le nom des balises en les séparant par une **barre verticale**.

<!ELEMENT balise (**balise2 | balise3 | balise4 | etc.**)>

❑ **Exemple**:

<!ELEMENT personne (**nom | prenom**)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>

- Cette définition impose que la balise <personne /> contienne obligatoirement la balise <nom /> **ou** la balise <prenom />.

Définition de Type de document DTD :

- Regardons alors la validité des documents XML ci-dessous:

```
<!-- valide -->
<personne>
   <nom>Hafidi</nom>
</personne>
    <!-- valide -->
<personne>
    <prenom> Houssem Eddine </prenom>
</personne>
```

Définition de Type de document DTD :

<!-- invalide -->

<!-- les 2 balises prenom et nom ne peuvent pas être présentes en même temps. -->

```
<personne>
  <prenom> Houssem Eddine </prenom>
  <nom> Hafidi</nom>
</personne>
```

<!-- invalide -->

<!-- il manque une balise -->

```
<personne >
 </personne>
```

Définition de Type de document DTD :

3. *Balise optionnelle*

- Une balise peut être **optionnelle**.
- Pour indiquer qu'une balise est optionnelle, on fait suivre son nom par un **point d'interrogation**.
- <!ELEMENT balise (balise2, **balise3 ?**, balise4)>

❑ **Exemple** :

<**!ELEMENT** personne (nom, **prenom?**)>

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

<**!ELEMENT** prenom (#PCDATA)>

- Cette définition impose que la balise <personne /> contienne **obligatoirement** la balise <nom /> puis **éventuellement** <prenom />.

Définition de Type de document DTD :

- Regardons alors la validité de ces documents XML :

```
<!-- valide -->
   <personne>
      <nom> Hafidi</nom>
   </personne>
<!-- valide -->
   <personne>
      <nom> Hafidi</nom>
      <prenom> Houssem Eddine </prenom>
   </personne>
 <!-- invalide -->
<!-- l'ordre des balises n'est pas respecté -->
   <personne>
      <prenom> Houssem Eddine </prenom>
      <nom> Hafidi</nom>
   </personne>
```

Définition de Type de document DTD :

4. *Balise répétée optionnelle*

- Une balise peut être **répétée plusieurs fois** optionnellement.
- Pour indiquer une telle balise, on fait suivre son nom par une **étoile**.

<**!ELEMENT** balise (balise2, **balise3***, balise4)>

- **Exemple** :

<**!ELEMENT** repertoire (**personne***)>

<**!ELEMENT** personne (nom, prenom)>

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

<**!ELEMENT** prenom (#PCDATA)>

- Cette définition impose que la balise <repertoire /> contienne entre **0 et une infinité de fois** la balise <personne />.
- La balise<personne />, quant à elle, doit obligatoirement contenir les balises <nom /> et <prenom /> dans cet ordre.

Définition de Type de document DTD :

- Regardons alors la validité des documents XML :

```
<!-- valide -->
<repertoire>
        <personne>
            <nom> Hafidi </nom>
            <prenom> Houssem Eddine </prenom>
        </personne>
        <personne>
            <nom>Sebti</nom>
            <prenom>Riad</prenom>
        </personne>
</repertoire>
```

Définition de Type de document DTD :

```
<!-- valide -->
<repertoire>
    <personne>
        <nom>Hafidi </nom>
        <prenom> Houssem Eddine </prenom>
     </personne>
</repertoire>
<!-- valide -->
<repertoire >
</repertoire>
```

Définition de Type de document DTD :

<!-- invalide -->

<!-- il manque la balise prenom dans la seconde balise personne-->

```
<repertoire>
  <personne>
        <nom>Hafidi</nom>
        <prenom> Houssem Eddine </prenom>
  </personne>
  <personne>
        <nom>Sebti</nom>
  </personne>
</repertoire>
```

Définition de Type de document DTD :

5. Balise répétée

- Une balise peut être **répétée plusieurs fois**.
- Pour cela, on fait suivre le nom de la balise par un **plus**.

<**!ELEMENT** balise (balise2, **balise3+**, balise4)>

- Prenons l'exemple suivant :

<!**ELEMENT** repertoire (**personne+**)>

<!**ELEMENT** personne (nom, prenom)>

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

<!**ELEMENT** prenom (#PCDATA)>

- Cette définition impose que la balise <repertoire /> contienne **au minimum une fois** la balise<personne />.
- La balise<personne />quant à elle doit obligatoirement contenir les balises<nom />et<prenom />dans cet ordre.

Définition de Type de document DTD :

- Regardons alors la validité des documents XML :

```
<!-- valide -->
<repertoire>
  <personne>
      <nom>Hafidi</nom>
      <prenom> Houssem Eddine </prenom>
   </personne>
   <personne>
      <nom>Sebti</nom>
      <prenom>Riad</prenom>
   </personne>
</repertoire>
```

Définition de Type de document DTD :

- Regardons alors la validité des documents XML :

```
<!-- valide -->
<repertoire>
   <personne>
      <nom>Hafidi</nom>
       <prenom> Houssem Eddine </prenom>
   </personne>
</repertoire>
<!-- invalide -->
<!-- la balise personne doit être présente au moins une fois-->
<repertoire >
</repertoire>
```

Définition de Type de document DTD :

❖ Déclaration d'une DTD

- La *DTD* qui sert de modèle a un document doit être déclarée au début de celui-ci, avant l'ouverture de l'*élément racine*.
- Trois modalités sont possibles qui conduisent, selon le cas, à une *DTD interne*, *DTD externe* ou *DTD mixte*.

1. DTD interne

La DTD est intégralement incluse dans le document qu'elle décrit, selon la syntaxe suivante :

```
<!DOCTYPE nom_racine [ liste_déclarations ] >
```

- Si plusieurs documents se réfèrent à la même DTD, et que celle-ci est interne, il est clair que la DTD devra être copiée dans chaque document, ce qui entraîne des risques d'erreurs et des manipulations assez lourdes.

Définition de Type de document DTD :

- **Exemple**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE personne [
 <!--début de la DTD interne -->
<!ELEMENT personne (prenom, nom)>
 <!ELEMENT prenom (#PCDATA)>
 <!ELEMENT nom (#PCDATA)>
 <!--fin de la DTD interne --> ] >
<!--début du document XML-->
<personne>
   <prenom>Ahmed</prenom>
    <nom>Almi</nom>
 </personne>
 <!--fin du document XML-->
```

Définition de Type de document DTD :

2. DTD externe

- La DTD est mémorisée dans un fichier texte spécifique, différent de celui du document qui s'y réfère.
- Pour que le lien soit établi, la DTD est déclarée dans l'en-tête du document par le biais de son *URI (*chaîne de caractères identifiant une ressource sur un réseau) selon la syntaxe suivante :

<!DOCTYPE *nom_racine* SYSTEM *URI_dtd* >

- L'URI est une chaîne de caractères qui peut représenter :

➢ L'identification absolue du fichier contenant la DTD, par exemple :

```
<!DOCTYPE html SYSTEM
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

➢ Une adresse relative si document et DTD sont sur le même site :

```
<!DOCTYPE guide SYSTEM "../../Dtd/rando/rando.dtd">
```

Définition de Type de document DTD :

- Un nom de fichier si **DTD** et **document sont dans le même répertoire**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE personne SYSTEM "personnes.dtd">
<!--début du document-->
<personne>
    <prenom>Ahmed</prenom>
    <nom>Almi</nom>
</personne>
<!--fin du document-->
```

- Avec le contenu du fichier **personnes.dtd** :

```
<!ELEMENT personne (prenom, nom)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT nom (#PCDATA)>
```

Définition de Type de document DTD :

3. DTD mixte

- On peut modifier certaines définitions sans toucher à la DTD.
- Lorsque l'auteur d'un document n'a pas de droit d'écriture sur la DTD.

➔ Il pourra ajouter une DTD interne qui viendra compléter ou modifier la DTD externe.

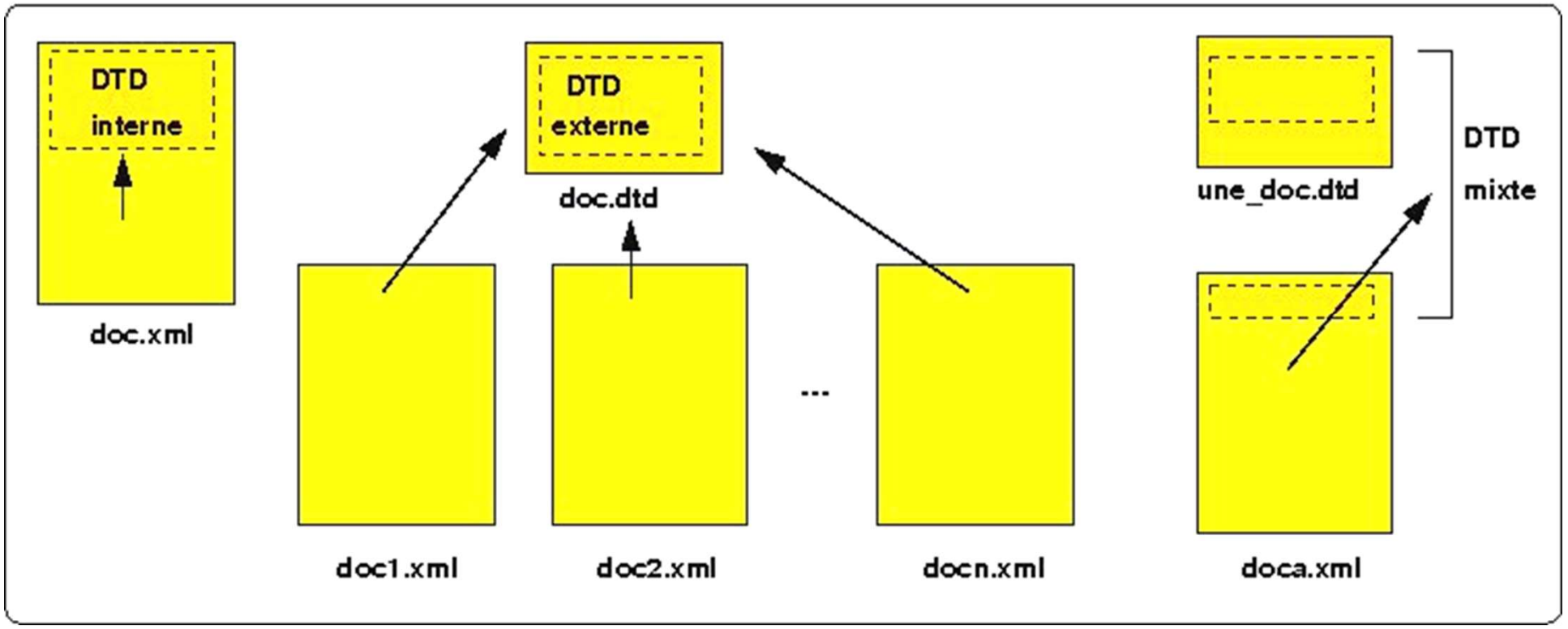
- Le document devra alors être valide par rapport à une DTD égale à la réunion des la DTD externe et de la DTD interne.
- La partie interne de la DTD **surchage** la partie externe.
- On peut par exemple changer le statut ou la valeur d'un attribut, mais **il n'est pas possible de redéfinir un élément**.

➔ Tout élément doit être déclaré dans la partie interne ou dans la partie externe mais pas dans les deux

<!DOCTYPE *nom_racine* SYSTEM *URI_dtd* [*liste_déclarations de la DTD interne*]>

Définition de Type de document DTD :

Schéma récapitulatif et contenu d'une DTD



Définition de Type de document DTD :

❖ En résumé

- Un **document valide** est un document bien formé conforme à une définition.
- Un **document conforme à une définition** est un document qui respecte toutes les règles qui lui sont imposées dans les fameuses définitions.
- Il est possible d'écrire de nombreuses règles grâce aux **DTD**.
- Il existe les **DTD internes** et les **DTD externes**.
- Le mot clef **ELEMENT** permet d'écrire les règles relatives aux **balises XML**.

Définition de Type de document DTD :

❖ Elément vs Attribut

- Utiliser **ELEMENT** pour :
  - ➢ Un contenu relativement gros
  - ➢ Un contenu important
  - ➢ Le contenu fait partie de l'information du document
- Utiliser **Attribut** pour
  - ➢ Un contrôle sur les valeurs
  - ➢ Le contenu est un identifiant

Définition de Type de document DTD :

2. DTD : les attributs et les entités

- Définir des **règles à nos attributs**.

❖ ***Syntaxe***

- Pour indiquer que notre règle porte sur un **attribut**, on utilise le mot clef **ATTLIST**.

On utilise alors la syntaxe suivante :

<!ATTLIST balise attribut type mode>

- Une règle peut donc se diviser en 5 mots clefs :

ATTLIST balise attribut type mode.

Définition de Type de document DTD :

❑ Retour sur la balise et l'attribut

- Ecrire le nom de la balise et de l'attribut concerné

❑ **Exemple**: <personne sexe="masculin" />

- On écrira alors :

<!**ATTLIST personne sexe type mode**>

❑ Retour sur le type

- Il représente le type de l'attribut

Définition de Type de document DTD :

1) Cas d'un attribut ayant pour type la liste des valeurs possibles

Les différentes valeurs possibles pour l'attribut sont **séparées** par une **barre verticale |**

<!ATTLIST balise attribut (valeur 1 | valeur 2 | valeur 3 | etc.) mode>

- Reprenons une nouvelle fois la balise<**personne** />

<!**ATTLIST personne sexe (masculin|féminin) mode**>

❑ **Exemples de documents XML possibles :**

```
<!-- valide -->
<personne sexe="masculin" ></personne>
<!-- valide -->
<personne sexe="féminin" ></personne>
<!-- invalide -->
<personne sexe="autre" ></personne>
```

2) Cas d'un attribut ayant pour type du texte non "parsé"

- "**texte non "parsé"**" : la possibilité de mettre ce que l'on veut comme valeur : un nombre, une lettre, une chaîne de caractères, etc.
- Il s'agit de données qui ne seront pas analysées par le "parseur" au moment de la validation et/ou l'exploitation de votre document XML
- On utilise le mot clef **CDATA**.

<!ATTLIST balise attribut CDATA mode>

- Reprenons une nouvelle fois la balise<**personne** />

<!ATTLIST **personne sexe CDATA mode**>

❑ **Exemples de documents XML possibles :**

```
<!-- valide -->
<personne sexe="masculin" ></personne>
<!-- valide -->
<personne sexe="féminin" ></personne>
<!-- valide -->
<personne sexe="autre" ></personne>
<!-- valide -->
<personne sexe="12" ></personne>
```

3) Cas d'un attribut ayant pour type un identifiant unique

- Prenons l'exemple d'une course. Dans le classement de la course, il y aura un **unique** vainqueur, un **unique** second et un **unique** troisième.
- On utilise le mot clef **ID** comme **ID**entifiant

<!ATTLIST balise attribut ID mode>

❑ **Exemple** : <!**ATTLIST personne position ID mode**>

- **Voici quelques exemples de documents XML :**

<!-- valide -->

<personne position="**POS-1**" ></personne>

<personne position="**POS-2**" ></personne>

<personne position="**POS-3**" ></personne>

<!-- invalide -->

<personne position="**POS-1**" ></personne>

<personne position=**"POS-1**" ></personne>

<personne position="**POS-2**" ></personne>

4) Cas d'un attribut ayant pour type une référence à un identifiant unique

- Cela permet de ne pas écrire 100 fois les mêmes informations.
- Par exemple, votre document XML peut vous servir à représenter des liens de parenté entre des personnes
- Pour faire référence à un identifiant unique, on utilise le mot clef **IDREF.**

❑ **Prenons par exemple la règle suivante :**

<!ATTLIST father id ID mode >

<!ATTLIST child id ID mode

father IDREF mode >

- Cette règle signifie que la balise **child** a **2 attributs** :
  1. **id** qui est l'identifiant unique de la personne
  2. **father** qui fait référence une autre personne.

❑ **Exemple** : Illustrons immédiatement avec un exemple XML :

```
<!-- valide -->
<father id="PER-1" > </father>
<child id="PER-2" father="PER-1" > </child>
<!-- invalide -->
<!-- l'identifiant PER-0 n'apparaît nulle part -->
<father id="PER-1" > </father>
<child id="PER-2" father="PER-0" > </child>
```

❑ Dans cet exemple, la personne **PER-2** a pour père la personne **PER-1**.

❑ Ainsi, on voit bien le lien entre ces 2 personnes.

Définition de Type de document DTD :

❑ Retour sur le mode

- Cet emplacement permet de donner une information supplémentaire sur l'attribut
- Exemple une indication sur son obligation ou sa valeur.

1) Cas d'un attribut obligatoire

- Si attribut est obligatoire, on utilise le mot clef **#REQUIRED**.
- **Exemple** le sexe d'une personne soit obligatoire,

<!**ATTLIST** **personne** sexe **(masculin|féminin)** **#REQUIRED**>

❑ Exemples

```
<!-- valide -->
<personne sexe="masculin" ></personne>
<!-- valide -->
<personne sexe="féminin" ></personne>
<!-- invalide -->
<personne></personne>
```

Définition de Type de document DTD :

2) Cas d'un attribut optionnel

- Si attribut **n'est pas obligatoire**, on utilise le mot clef **#IMPLIED**.
- **Exemple** le sexe d'une personne n'est pas obligatoire,

<!**ATTLIST** **personne** sexe **CDATA** **#IMPLIED** >

❑ **Exemples**

<!-- valide -->
<personne sexe="masculin" ></personne>
<!-- valide -->
<personne sexe="féminin" ></personne>
<!-- valide -->
<personne></personne>
<!-- valide -->
<personne sexe="15" ></personne>

Définition de Type de document DTD :

3) Cas d'une valeur par défaut

- Il suffit d'écrire cette valeur **"valeur"** dans la règle.
- **Exemple** le sexe d'une personne est un homme par défaut

<!**ATTLIST** **personne** sexe **CDATA** **"masculin">**

❑ **Exemples**

<!-- valide -->
<personne sexe="masculin" ></personne>
<!-- valide -->
<personne sexe="féminin" ></personne>
<!-- valide -->
<!-- l'attribut sexe vaut "masculin" -->
<personne ></personne>

Définition de Type de document DTD :

4) Cas d'une constante

- Il est possible de fixer la valeur d'un attribut
- On utilise le mot clef **#FIXED** suivi de la valeur.
- **Exemple de règle**
- Indiquer que la devise doit avoir pour seule valeur possible l'**Euro**.

<!ATTLIST **objet** devise CDATA **#FIXED** "**Euro**">

Exemples

<!-- valide -->

<objet devise="**Euro**" ></objet>

<!-- invalide -->

<objet devise="**Dollar**" ></objet>

<!-- valide -->

<objet ></objet>

Définition de Type de document DTD :

❖ En résumé

- Le mot clef ATTLIST permet d'écrire les règles relatives aux attributs d'une balise.
- On peut remarquer que les DTD ont quelques défauts:

1. Un nouveau format

- les DTD ne sont pas au format XML.
- Nous avons dû apprendre un nouveau langage avec sa propre syntaxe et ses propres règles.

2. Le typage de données

- Les DTD ne permettent pas de typer des données.
- Impossible de préciser si l'on souhaite que ça soit un nombre entier, un nombre décimal, une date, une chaîne, etc.

Questions ??