مدرسة علوم المعلومات
ⵜⵉⵎⵔⵔⴰⵜ ⵏ ⵜⵓⵙⵙⵏⵉⵡⵉⵏ ⵏ ⵜⵣⵔⴼⵜ
ECOLE DES SCIENCES
DE L'INFORMATION

Elément 02 :

Documents structurés

Pr . J. IDRAIS

Avant de commencer

- Note élément :
  - **A** : Travail particulier/comptes rendues TP
  - **CP** ; contrôle partiel
  - **DF** : Devoir final

$$\text{Note Elément} = [\ A \times 0.2\ ] + [\ CP \times 0.3\ ] + [\ DF \times 0.5\ ]$$

$$\text{Note Module} = \frac{\text{Note Elément 01} + \text{Note Elément 02}}{2}$$

Chapitre 2
Définition de Type de Document (DTD)

Définition de Type de Document (DTD)

Nécessité d'utiliser des modèles

Ambiguïté

Dans la plupart des cadres de travail, un document n'a pas de raison d'être s'il est seul de son espèce. Il fait en général partie d'une famille de documents analogues qui se conforment à un même modèle.

On voit donc apparaître deux types d'acteurs, qui, éventuellement peuvent se fondre dans la même personne :

- Le concepteur du modèle de document qui fixe les éléments et les attributs utilisés, ainsi que leur organisation,
- les auteurs des documents qui se conforment au modèle pour créer les documents.

Définition de Type de Document (DTD)

Différents types de modèles

Pour que le document XML soit compréhensible, les *modèles* doivent suivre des conventions précises.

Il existe donc des *langages* qui permettent de décrire, de manière précise, dans un document modèle, la structure que doivent vérifier les documents qui lui sont attachés.

Ce sont des **langages de schéma**.

- Ce chapitre présente le langage utilisé pour écrire des modèles nommés **Définition de Type de Document** ou **DTD**.
- Le chapitre suivant présentera le langage **XMLSchema**
- D'autres langages de schéma existent, comme par exemple **RelaxNG**

Définition de Type de Document (DTD)

Différents types de modèles

Un document XML bien formé a été défini comme un document qui respecte la syntaxe du langage XML.

On définit maintenant un **document valide** comme un document qui vérifie les propriétés suivantes :

- Il respecte la syntaxe XML : c'est un document **bien formé**.
- Il respecte la structure définie dans une **DTD** ou un **schéma**.

La notion de *document valide* implique donc l'existence d'un modèle auquel doit se conformer ce document.

Ce modèle peut être une DTD ou un **schéma XML**

Dans tous les cas il contient **une description exhaustive et non ambiguë** de la forme des documents.

Définition de Type de Document (DTD)

Principe de validation

« Tout ce qui n'est pas spécifié est interdit »

Tous les documents qui sont valides selon une DTD, forment une classe de documents, si bien que :

- Un document est une instance du modèle défini par une DTD
- En plus de vérifier si un document est bien formé, les analyseurs XML, ou parseurs XML, sont le plus souvent capables de vérifier si le document est valide selon une DTD.

Définition de Type de Document (DTD)

Définition

La DTD décrit la structure d'un document XML et peut être assimilée à une sorte de **grammaire** des documents XML et décrit :

- Les éléments types
  - Les noms de balises autorisés
  - L'ordre autorisé pour les balises
  - Quels éléments peuvent contenir quels éléments
  - Quels éléments sont optionnels
- Les attributs pour chaque élément
  - Noms des attributs autorisés, obligatoires / optionnels, leur type, valeur par défaut, etc.
- Les entités binaires (non analysables) ou textuelles (analysables) pouvant être incluses dans un document)

Définition de Type de Document (DTD)

Définition

Une DTD décrit un **vocabulaire pour un document XML**. Elle définit la structure d'une classe de documents XML en spécifiant les éléments et les règles d'utilisation.

Une DTD décrit de **manière précise** les éléments que peut contenir un document XML, dans quel ordre ils peuvent apparaître et quels sont leurs attributs possibles.

Un DTD se présente sous la forme d'une liste de déclarations de quatre types :

- les déclarations d'éléments
- les déclarations de liste d'attributs
- les déclarations d'entités
- les déclarations de notations

Déclaration d'une DTD

Une **DTD** se présente sous la forme d'une liste de déclarations qui définissent complètement la structure que devront suivre les documents qui s'y réfèrent.

La **DTD doit être déclarée au début de du document XML, avant l'ouverture de l'*élément racine*.**

- La DTD peut faire partie du document XML, elle est dite **interne**,
- La DTD peut être un fichier à elle seule, elle est dite **externe**.
- DTD **mixte**

Les DTD externes peuvent être séparées en deux catégories :

- **privée (SYSTEM)** : est représentée par un fichier accessible uniquement en local
- **publique (PUBLIC)** : sera disponible pour tout le monde via une URI (Uniform Resource Identifier)

Déclaration d'une DTD

DTD interne

Une DTD contient des déclarations spécifiques au document XML et portant sur les éléments et les attributs et aussi sur les entités spécifiques ;

- **Une DTD interne n'a pas d'effet sur les autres documents.**

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

```
<?xml version="1.0" standalone="yes">
<!DOCTYPE racine [
<!-- DEFINITION DTD -->
]>
<racine>
...
</racine>
```

Déclaration d'une DTD

DTD externe

Une DTD externe contient des déclarations générales pouvant être aussi comme la DTD interne :

- des déclarations d'éléments ;
- des déclarations des attributs ;
- les déclarations d'entités dont chaque document peut avoir besoin.

Contrairement à la DTD interne, une DTD externe peut s'appliquer à plusieurs documents.

1. DTD externe privée (stockée dans un autre fichier autre que le document XML) ; elle est accessible avec un accès local)

```
<!DOCTYPE livre SYSTEM "biblio.dtd">
```

```
<!DOCTYPE racine SYSTEM "C://windows/user/desktop/racine.dtd">
```

2. DTD externe publique avec un accès via une URI

```
<!DOCTYPE livre PUBLIC "Identifiant_Public" "URI">
```

Déclaration d'une DTD

DTD mixte

Il peut arriver que pour certains des documents qui se conforment à une DTD, on veuille modifier certaines définitions sans toucher à la DTD.

Par exemple, l'auteur d'un document n'a pas de droit d'écriture sur la DTD.

Dans ce cas il pourra ajouter une **DTD interne** qui viendra **compléter ou modifier** la DTD **externe**.

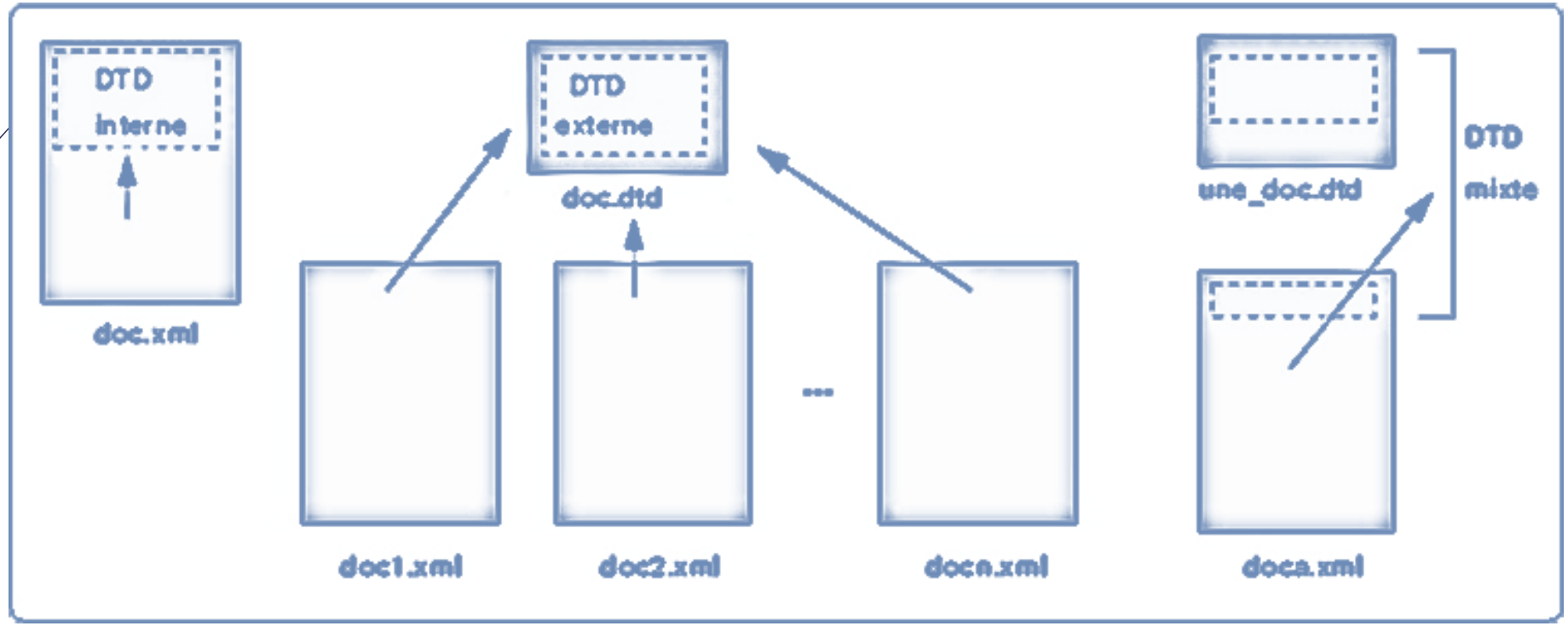
Le document devra alors être **valide** par rapport à **une DTD égale à la réunion des la DTD externe et de la DTD interne**.

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**.

Déclaration d'une DTD

DTD { interne – externe – mixte}



Déclaration des éléments XML DTD

Un élément XML permet de structurer l'information (par exemple l'élément livre est un élément d'un document bibliothèque).

- Ainsi , La déclaration d'un élément est de la forme :

```
<!ELEMENT  nom_element  modele_de_contenu >
```

Cas de base

Les cas les plus simples sont les suivants.

expression du contenu	signification
EMPTY	contenu vide
ANY	contenu quelconque
(#PCDATA)	contenu textuel
(*elt*)	un seul élément
(elt_1, elt_2, ...,elt_n)	séquence d'éléments pris dans cet ordre
(elt_1 | elt_2 | ... | elt_n)	un des éléments au choix

Ils donnent une description "en extension" du contenu de l'élément.

Ils utilisent l'opérateur **,** pour marquer la séquence, et l'opérateur **|** pour marquer le choix

Déclaration des éléments XML DTD

Spécification d'un élément vide

Un élément vide ne contient **aucun texte, aucun autre élément**. Sa syntaxe est :

```
<!ELEMENT nom_element  EMPTY >
```

Spécification d'un élément libre

Un élément libre peut contenir **tout élément déclaré dans la DTD et du texte** sans aucune contrainte de structuration.

```
<!ELEMENT  nom_element  ANY >
```

Modèle de contenu des éléments

Le contenu d'un élément peut être contenir **uniquement** des données textuelles (Parsed Character DATA)

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

Déclaration des éléments XML DTD

Les **opérateurs *d'occurrence*** permettent de spécifier la répétition de sous-éléments ou de groupes :

opérateur	signification
?	0 ou 1 occurrence
*	0 ou plusieurs occurrences
+	1 ou plusieurs occurrences

```
<!ELEMENT nom_element (#PCDATA) >
<!ELEMENT nom_element (element1, element2 , element3) >
<!ELEMENT nom_element (element1| element2 | element3) >
<!ELEMENT nom_element (element1? ,element2+ ,element3*)? >
<!ELEMENT nom_element (#PCDATA | element2 ) >
```

Déclaration des éléments XML DTD

Contenu mixte des éléments

Certains éléments ont un contenu mixte et ils sont composés d'une alternance, dans un ordre quelconque, de fragments textuels et d'éléments.

La déclaration d'un élément à **contenu mixte** se fait toujours selon la syntaxe suivante, le mot clé **#PCDATA étant nécessairement en première position** :

```
<!ELEMENT nom_element (#PCDATA | element2 | element3 | element1 ) >
```

Déclaration des éléments XML Exemple

Comment écrire les règles suivantes sous forme d'une DTD :

Une **personne** est composé, dans cet ordre, de :

- un élément *nom*
- un ou plusieurs éléments *prénom*
- zéro, un ou plusieurs éléments *téléphone*
- zéro ou un élément *adresse*

les quatre éléments ***nom***, ***prénom***, ***téléphone*** et ***adresse*** sont des éléments textuels.

Déclaration des éléments XML Exemple

Expliquer le code DTD suivant :

```
---------------------------------------------------------------------------
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE exemple [
    <!ELEMENT exemple (e1, e2, e3, e4, e5)>
    <!ELEMENT e1 (a, b)+ >
    <!ELEMENT e2 (a | b)+ >
    <!ELEMENT e3 (a+ | b+) >
    <!ELEMENT e4 ((a|b), c)>
    <!ELEMENT e5 ((a|b)+, c)>
    <!ELEMENT a (#PCDATA)>
    <!ELEMENT b (#PCDATA)>
    <!ELEMENT c (#PCDATA)>
]>
---------------------------------------------------------------------
```

Déclaration des attributs XML DTD

Déclaration des attributs

Une déclaration d'attributs associé à un élément XML cible, précise :

- Les noms d'attributs autorisés,
- Le type de chaque attribut,
- La valeur par défaut de l'attribut.

- La déclaration d'attributs peut être séparée ou sous forme de liste avec le mot-clé de déclaration : **ATTLIST**.

```
<!ATTLIST element_cible  nom_attribut type_attribut valeur_défaut .. >
```

Déclaration des attributs XML DTD

Déclaration des attributs

```
<!ATTLIST element_cible  nom_attribut type_attribut valeur_défaut .. >
```

Nom d'attribut

- Le nom d'un attribut doit être un nom XML :
- le premier caractère est une lettre quelconque ou un _ (underscore ou tiret bas) ;
- les caractères suivants peuvent être des lettres, des chiffres, des tirets bas (_), des traits d'union (-) ou des points (.) ;
- il n'y a pas de limitation sur la longueur d'un nom XML.

Déclaration des attributs XML DTD

Déclaration des attributs

```
<!ATTLIST element_cible  nom_attribut type_attribut valeur_défaut .. >
```

Types autorisés

- Les types autorisés pour les attributs sont décrits dans le tableau suivant :

Déclaration des attributs XML DTD

Déclaration des attributs

```
<!ATTLIST element_cible  nom_attribut type_attribut valeur_défaut .. >
```

type	définition
CDATA	n'importe quelle chaîne de caractère ne contenant pas <, >, &, ' ou "
(val_1 | val_2 | ... | val_n)	énumération des valeurs autorisées pour l'attribut
ID	un nom XML qui doit être unique dans le document car il sert d'identifiant à l'élément
IDREF	un nom XML qui doit exister dans le document en tant que valeur d'un attribut ID car il désigne un élément cible
IDREFS	une liste de noms XML, séparés par des espaces, qui doivent exister dans le document en tant que valeur d'un attribut ID car ils désignent des éléments cibles
NMTOKEN	une unité lexicale nominale
NMTOKENS	liste d'unités lexicales nominales séparées par des espaces
ENTITY	le nom d'une entité externe non XML (voir plus loin)
ENTITIES	liste de noms d'entités externes non XML séparées par des espaces
NOTATION	le nom d'une notation déclarée dans la DTD (voir plus loin)

Déclaration des attributs XML DTD

Déclaration des attributs

```
<!ATTLIST element_cible  nom_attribut type_attribut valeur_défaut .. >
```

Valeur d'attribut

La valeur d'un attribut peut être par facultative ou spécifiée par un mot clé de remplacement :

- **#REQUIRED** : valeur d'attribut doit être spécifiée
- **#IMPLIED** : valeur d'attribut peut rester non spécifiée
- **#FIXED** "val" : valeur de l'attribut fixée à "**val**" et non modifiable par l'utilisateur

Déclaration des attributs XML DTD

Déclaration des attributs

```
<!ATTLIST element_cible  nom_attribut type_attribut valeur_défaut .. >
```

Exemple :

```
<!ATTLIST compte pays CDATA #FIXED "MAROC">
<!ATTLIST compte nom CDATA #REQUIRED>
<!ATTLIST compte prenom CDATA #REQUIRED>
<!ATTLIST compte age CDATA #IMPLIED>
<!ATTLIST compte profile (guest | admin | user)>
```

Déclaration des attributs XML DTD

Déclaration des attributs

```
<!ATTLIST element_cible  nom_attribut type_attribut valeur_défaut .. >
```

Il est possible de définir plusieurs attributs d'un même éléments dans la même définition

```
<!ELEMENT Compte EMPTY>
<!ATTLIST Compte  pays CDATA #FIXED "MAROC«
                  nom CDATA #REQUIRED
                  prenom CDATA #REQUIRED
                  age CDATA #IMPLIED
                  profile (guest | admin | user)>
```

```
<?xml version="1.0" ?>
<!DOCTYPE ecole[
<!ATTLIST ecole codes_departement IDREFS #IMPLIED>
<!ELEMENT ecole (departement+)>
<!ELEMENT departement(prof+)>
<!ATTLIST departement code ID #REQUIRED>
<!ELEMENT prof (#PCDATA)>
<!ATTLIST prof code ID #REQUIRED code_departement IDREF #REQUIRED> ]>
<ecole codes_departement="A001 A003">
    <departement code="A001">
        <prof code="E205" code_departement ="A001"> EL MOUDENE</prof>
        <prof code="E206" code_departement ="A001"> EL ABASSI </prof>
        <prof code="E207" code_departement ="A001"> FARHAOUI </prof>
        <prof code="H107" code_departement ="A003"> FADEL </prof>
    </departement>
    <departement code="A003">
        <prof code="A115" code_departement ="A003"> IDRAIS </prof>
    </departement >
</ecole>
```

Déclaration des entités XML DTD

Déclaration des entités XML

Définition : Entité permet d'associer un nom à un contenu (alias).

Chaque entité est :

- identifiée par un nom
- définie par une déclaration d'entité
- utilisée en appelant une référence d'entité

caractère	<	>	'	"	&
entité	<	>	'	"	&

Syntaxe de déclaration : `<!ENTITY nom_entité "valeur de remplacement">`

La référence d'entité se note : **&nom_entité;**

Exemple : `<!ENTITY DC "documents structurés" >`

Utilisation : <description>Le cours de (&DC;) se compose de...</description>

Déclaration des entités XML DTD

Déclaration des entités paramètres XML

Ce type d'entité permet d'éviter de répéter les mêmes informations.

Déclaration : `<!ENTITY % nom "caractères de remplacement" >`

Utilisation : Référence dans la DTD (parenthèses conseillées) **(%nom_entité;)**

Exemple :

Déclarations DTD

```
<!ENTITY % a "ANY" >
<!ENTITY % u "para|list|table">
```

Utilisations dans la DTD

```
<!ELEMENT paragraphe %a; >
<!ELEMENT chapitre ((%u;)*, section*)>
<!ELEMENT section (%u;)*>
```

Déclaration des notations XML DTD

Notations

Les NOTATIONS sont utilisées pour identifier un élément avec un attribut de type NOTATION dont **la valeur est une donnée non XMl** qui n'est pas analysables par un processeur XML .

Syntaxe :

```
<!NOTATION name SYSTEM "URI">
<!NOTATION name PUBLIC "public_ID">
<!NOTATION name PUBLIC "public_ID" "URI">
```

Dans la DTD :

```
<!ELEMENT document EMPTY>
<!ATTLIST document source ENTITY #REQUIRED>
<!ENTITY mon_CV SYSTEM « cv_2022.pdf" NDATA PDF>
<!NOTATION PDF SYSTEM « doc/pdf" >
```

Dans le document XML :

```
<document source="mon_CV"/>
```

Travail à préparer

Volontaires pour les présentations suivantes :

1. XML et la GED : utilisations et cas pratiques
2. Manipulation des documents XML par les languges de programmation (python, java, C#, PHP, …Etc)
3. HTML, DOM & Javascript

---

2 binômes pour chaque sujet

Deadline : **semaine prochaine**