

XML et DTD

LIONEL MÉDINI
UFR INFORMATIQUE
UNIVERSITÉ CLAUDE BERNARD LYON 1

D'après le cours de Yannick Prié
2010-2011 – Master SIB
M1 – UE 3 / Bloc 2 – Cours 1

Objectifs de ce cours

- Etre capable de
 - Comprendre les documents XML
 - Comprendre les DTD
 - Construire des documents XML et des DTD

Plan

- Documents XML
 - Syntaxe XML et documents bien formés
- Types de documents XML
 - DTD et documents valides
 - Introduction à XML-Schema
- Le monde XML
 - Quelques normes liés à XML
 - Quelques DTD importantes

Un document XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE livre SYSTEM "D:\Enseignement\2004-
2005\SIB3.2\CM\exemple-intro.dtd">
<livre id="561" nbpages="190" titre="La compagnie des spectres">
  <auteur>
    <nom>Salvayre</nom>
    <prenom>Lydie</prenom>
  </auteur>
  <format type="poche">
    <mesure type="largeur" unite="cm">11</mesure>
    <mesure type="longueur" unite="cm">19</mesure>
    <mesure type="hauteur" unite="mm">10</mesure>
  </format>
</livre>
```

La DTD correspondante

```
<!ELEMENT livre (auteur, format)>
<!ATTLIST livre
  id CDATA #REQUIRED
  nbpages CDATA #REQUIRED
  titre CDATA #REQUIRED >
<!ELEMENT auteur (nom, prenom)>
<!ELEMENT format (mesure+)>
<!ATTLIST format
  type CDATA #REQUIRED >
<!ELEMENT mesure (#PCDATA)>
<!ATTLIST mesure
  type (hauteur | largeur | longueur) #REQUIRED
  unite (cm | mm | in) #REQUIRED >
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
```

Plan

- Documents XML
 - Syntaxe XML et documents bien formés
- Types de documents XML
 - DTD et documents valides
 - Introduction à XML-Schema
- Le monde XML
 - Quelques normes liés à XML
 - Quelques DTD importantes

Qu'y a-t-il dans un document XML ?

7

- Prologue
 - Déclaration XML
 - Déclarations de DTD
 - Instructions pour les processeurs XML
 - Instructions de traitement
 - Instructions pour applications externes
- Arbre des éléments
 - Eléments
 - Balises XML pour le marquage
 - Contenu
 - texte
 - autres éléments
 - Attributs des éléments
 - Information associées aux éléments
- Commentaires

Déclaration XML

8

- Syntaxe générale

```
<?xml version="1.0" [encoding = "encodage"] [standalone="yes|no *] ?>
```
- Est une des informations de traitement
- Indique
 - Conformité du document à une version de la norme XML
 - **version="1.0"**
 - Jeu de caractères utilisé dans le document
 - **encoding = "UTF-8"**
 - Présence ou non de références externes
 - **standalone="yes"**

Instructions de traitement

9

- Informations nécessaires à une application externe
- Format :
 - `<?NomApplication paramètres ?>`
- Exemples
 - Déclaration de feuille de style à utiliser

```
<?xml-stylesheet href="fichier.xml" type="text/xsl"?>
```
 - Déclaration XML de début de fichier

```
<?xml version='1.0' ?>
```

Éléments : règles de base

10

- Un nom d'élément
 - commence par une lettre ou souligné
 - contient des lettres, chiffres, et "-", ".", ":", "_"
 - peut posséder un préfixe
 - `prefixe:nom_element`
 - Ex.: `xsl:template`
- Les noms d'éléments dépendent de la casse
 - `<nom_element>` ≠ `<nom_Element>`
- Balises
 - de début : `<nom_element>`
 - de fin : `</nom_element>`
- Les éléments peuvent être vides
 - pas de contenu
 - `<element_vide />`
 - Ex: ``

Arbre des éléments

11

- Un seul élément racine qui contient tous les autres
- Pas d'intersections entre éléments
 - Mauvais : `<nom1><nom2>...</nom1></nom2>`
 - Bon : `<nom1><nom2>...</nom2></nom1>`
- Blancs ou retours chariot en général non significatifs
 - `<section><p> ... </p></section>`
 - `<section><p> ... </p></section>`
- Les éléments sont ordonnés

Caractères spéciaux

12

- Ces caractères ont une signification spéciale pour les outils XML
- Il faut les écrire différemment
 - `<` `<`
 - `>` `>`
 - `&` `&`
 - `'` `'`
 - `"` `"`

Attributs associés aux éléments : règles de base

13

- Dans les balises ouvrantes
 - `<el att1="valeur1" att2="valeur2">`
- Les noms d'attributs dépendent de la casse
 - `<el att1="valeur1" Att1="valeur2">`
- Valeurs d'attributs entourées
 - par des guillemets (") ou des apostrophes (')
- Les attributs sont non-ordonnés

Attributs

14

- Les valeurs peuvent être
 - des données textuelles
 - `value="N'importe quoi"`
 - des *tokens* (noms XML) simples
 - `value = "blue"`
 - des ensemble de *tokens*
 - `value = "red green blue"`
- Possibilité d'énumérer les valeurs possibles et de mettre des valeurs par défaut (voir DTD)

Attributs de type ID et IDREF(S)

15

- Permettent des relations non hiérarchiques entre éléments
 - ID : identificateur unique dans le document XML
 - IDREF : référence à un élément ayant un attribut de type ID
 - IDREFS : références à des éléments ayant un attribut de type ID
- Exemple

```
<société codes_services="A001 A003">
  <service code="A001">
    <employé code="E205" code_service="A001"> Jean Dupont </employé>
    <employé code="E206" code_service="A001"> Frédéric Marc </employé>
    <employé code="E207" code_service="A001"> Fabrice Detterre </employé>
    <employé code="H107" code_service="A003"> Angélique Millet </employé>
  </service>
  <service code="A003">
    <employé code="A115" code_service="A003"> Isabelle Mascot </employé>
  </service>
</société>
```
- Exercice
 - Construire l'arbre correspondant, dessiner les relations entre référence vers des identificateurs (idrefs) et identificateurs (id)

Commentaires

16

- Les commentaires ne sont pas considérés comme faisant partie du document XML
 - `<!-- Un commentaire -->`
- Pas de '--' dans un commentaire !
- Un commentaire ne peut pas se trouver dans une autre déclaration

Au bilan : dans un document XML

17

- Prologue
 - en-tête
 - déclaration de DTD (*pas encore vu*)
 - instructions de traitement
- Éléments
 - attributs
 - contenus
- Commentaires

Plan

18

- Documents XML
 - Syntaxe XML et documents bien formés
- Types de documents XML
 - DTD et documents valides
 - Introduction à XML-Schema

Traiter automatiquement un document XML

19

- **Parseur** (anglicisme d'après *parser*)
 - Outil qui lit un document XML et construit l'arbre des éléments en mémoire
- **Vérifier qu'un document répond bien à la syntaxe XML**
 - Document *bien formé*
 - Possibilité de l'utiliser en tant que tel
 - ex. : le présenter à l'utilisateur
- **Vérifier en plus qu'un document suit bien la grammaire définie (DTD / schéma)**
 - Document *valide*

Document Type Definition

20

- **Pour définir le type de document XML voulu**
 - grammaire pour décrire comment construire un document XML
- **Permet de**
 - valider un document XML (avec un parseur validant)
 - vérifier que tous les éléments sont présents et corrects
 - vérifier que les noms d'attributs et leurs valeurs sont corrects
 - transmettre cette connaissance à d'autres
 - ils pourront définir leurs propres documents XML suivant la même DTD
 - d'où possibilité de standardisation et d'échanges

DTD

21

- **Déclaration**
 - contenant la définition formelle de la structure autorisée,
 - décrit
 - quels noms sont utilisés pour les types d'éléments
 - comment ces types d'éléments s'organisent
 - ordre
 - hiérarchie
 - les attributs des éléments
 - des entités analysables ou non
 - des notations pour les types de données binaires
- **Liaison DTD / document XML**
 - soit la DTD est dans le document XML (inline)
 - soit le document XML réfère à la DTD avec une URI (la DTD est dans un fichier externe)

Langage de DTD

22

- **Un autre langage que XML**
 - autre syntaxe, mots-clés différents
 - origines dans SGML
- **Pour décrire des types de documents XML**
- **Remarque**
 - XML schema permet de faire la même chose, en XML

Déclarations

23

- **Instructions pour le processeur XML**
- **Format : <! ... > ou <! ... [<! ... >]>**
 - Document type - <!DOCTYPE ... >
 - Character data - <![CDATA[...]]>
 - Entities - <!ENTITY ... >
 - Notation - <!NOTATION ... >
 - Element - <!ELEMENT ... >
 - Attributes - <!ATTLIST ... >
 - <![INCLUDE [...]]> et <![IGNORE [...]]>

Déclaration Document Type

24

- Placée au début du document XML
- Identifie le nom de l'*élément racine* du document
 - <!DOCTYPE Racine>
- Permet aussi de rajouter des définitions d'entités et des DTD
 - <!DOCTYPE Racine [...] >
 - <Racine>
 - ...
 - </Racine>

déclaration
de la DTD

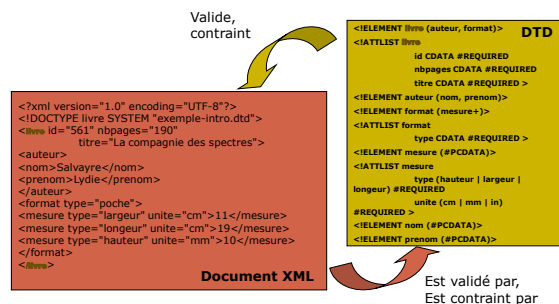
Déclaration *Character Data*

25

- Placée dans le document XML
- Permet de taper directement du texte qui ne doit pas être interprété par un outil XML
- Deux textes équivalents
 - `Press <<<ENTER>>>`
 - `<![CDATA[Press <<ENTER>>]]>`

DTD et document XML

26



Mettre en place une DTD

27

- Définir différents composants XML...
 - Entités, éléments, déclarations, instructions de traitements, listes d'attributs, etc.
- ... dans des DTD pour spécifier les règles permettant de valider des documents XML
 - Définir un modèle (type) de document de façon formelle
 - ✱ pour qu'un parser puisse valider les documents qui lui seront soumis
- Une DTD décrit
 - Quels noms peuvent être utilisés pour les types d'éléments
 - L'ordre dans lesquels ceux-ci peuvent apparaître
 - La hiérarchie documentaire
 - Les noms et les types des attributs d'éléments

Déclaration de DTD

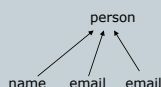
28

- La DTD est stockée
 - soit dans le fichier XML
 - soit dans un fichier extérieur
 - soit dans les deux
- Une DTD interne peut écraser ou ajouter des ENTITY ou des ATTLIST à des définitions de DTD externes
 - principe : "le dernier qui parle a raison"
- Une DTD est composée de déclarations
 - ELEMENT – définitions d'éléments
 - ATTLIST – définitions d'attributs
 - ENTITY – définitions d'entités
 - NOTATION – définitions de notations

Déclarations d'éléments

29

- Définir un élément et son contenu
 - `<!ELEMENT name (#PCDATA)>`
 - ⇒ `<name> ...du texte... </name>`
- Définir un élément vide (sans contenu)
 - `<!ELEMENT name EMPTY>`
 - ⇒ `<name/>`
- Définir des éléments fils spécifiques
 - `<!ELEMENT person (name, email*)>`
 - Éléments fils quelconques
 - ✱ `<!ELEMENT name ANY>`



Définir une hiérarchie (grammaire)

30

- Définir le contenu des éléments
 - `<!ELEMENT person (name, email*)>`
- ...et définir une hiérarchie d'éléments
 - `<!ELEMENT name (fname, surname)>`
 - `<!ELEMENT fname (#PCDATA)>`
 - `<!ELEMENT surname (#PCDATA)>`
 - `<!ELEMENT e-mail (#PCDATA)>`
- Organisation des sous-éléments
 - Connecteur de séquence ',': (A, B, C) [puis]
 - Connecteur de choix '|': (A | B | C) [ou]



Indicateurs de quantité

31

- Contraintes sur les éléments des DTD
 - A?** Possible [0..1]
 - A+** 1 fois et plus [1..*]
 - A*** 0 ou plus [0..*]
- Exemples
 - (A, B) +
 - ((A,B?) | C+) *

Déclaration d'attributs

32

- Les attributs sont associés aux types d'éléments
- Déclarés dans une déclaration ATTLIST liée à l'élément
 - <!ELEMENT nom_element ... >
 - <!ATTLIST nom_element ... >
 - Il faut ensuite définir
 - le nom de l'attribut
 - le type de l'attribut
 - sa valeur par défaut

Noms et types d'attributs

33

- Noms d'attributs
 - <!ATTLIST elem nom type par_defaut>
 - <!ATTLIST elem attribut_1 ...
 - attribut_2 ...
 - attribut_3 ... >
- Dix types possibles d'attributs
 - CDATA**
 - NMTOKEN**
 - NMTOKENS**
 - ENTITY**
 - ENTITIES**
 - ID**
 - IDREF**
 - IDREFS**
 - NOTATION**
 - name group**

Types d'attributs (1)

34

- **CDATA**
 - Chaîne de caractères
 - <!ATTLIST person name CDATA >
 - <person name = "Tom Jones">
 - **NMTOKEN**
 - Token unique
 - <!ATTLIST mug color NMTOKEN >
 - <mug color="red">
 - **NMTOKENS**
 - Tokens multiples
 - <!ATTLIST temp values NMTOKENS >
 - <temp values="12 15 34">
- Joue sur la manière dont le parseur interprète l'attribut :
- comme une chaîne de caractères quelconque
 - Comme une chaîne représentant un item d'une liste
 - comme un ensemble d'items séparés par des espaces

Types d'attributs (2)

35

- **ID**
 - Identificateur unique pour l'élément
 - <!ATTLIST person id ID >
 - <person id = "P09567">Toto</person>
 - <person id = "P09567">Tutu</person> ← Nonvalide
- **IDREF**
 - Référence à un ID
 - <!ATTLIST person father IDREF >
 - <person father="P09567">
- **IDREFS**
 - Référence à plusieurs ID
 - <!ATTLIST person children IDREFS >
 - <person children="P09567 P09677">

Types d'attributs (3)

36

- **Name group**
 - Liste restreinte de valeurs possibles
 - <!ATTLIST point coord (X|Y|Z) >
 - <point coord="X">
- **ENTITY** (substitut pour autre chose)
 - L'attribut est une référence d'entité
 - <!ATTLIST person photo ENTITY >
 - <person photo="MyPic">
- **ENTITIES**
 - Plusieurs références d'entités
 - <!ATTLIST album photos ENTITIES >
 - <album photos="pic1 pic2">

Types d'attributs (4)

37

- NOTATION

- Décrit des données non XML
- `<!NOTATION jpg SYSTEM "JPEG Image" >`
- `<!NOTATION gif PUBLIC "-//ISBN 0-7923-9432-1::Graphic Notation//NOTATION CompuServer Graphic Interchange Format//EN">`
- `<!ATTLIST image format NOTATION (jpg|gif) ...>`
- `<image format="gif">`

Valeurs d'attributs par défaut

38

- Quatre types

- **#REQUIRED** *doit être spécifié*
- **#IMPLIED** *peut être spécifié*
- **"default"** *valeur par défaut si non spécifié*
- **#FIXED** *seule valeur autorisée*

<ATTLIST	element	nom	type	par_defaut
<ATTLIST	citoyen	parents	IDREFS	#REQUIRED
		id	ID	#IMPLIED
		sex	(m f)	"f"
		adress	CDATA	#IMPLIED
		nat	CDATA	#FIXED "Fr"

Déclaration interne de DTD

39

- Dans la déclaration DOCTYPE

```
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE racine [
  <!-- ici la DTD -->
  <! ... >
  <! ... >
]>
<!-- ici le reste du fichier XML -->
<racine>
...
</racine>
```

Pas besoin de fichier supplémentaire

Déclaration de DTD externe privée

40

- Référence à la DTD externe par un chemin dans la déclaration DOCTYPE

```
<?xml version="1.0" standalone="no" ?> <!DOCTYPE
racine
  SYSTEM "dossiers/MyDoc.dtd" [
  <!-- déclarations supplémentaires -->
  <! ... >
  <! ... >
]>
<!-- ici le reste du fichier XML -->
```

DTD externe privée

- Les déclarations spécifiques au document restent définies de façon interne

Déclaration de DTD externe privée

41

- Référence à la DTD externe par une URL dans la déclaration DOCTYPE

```
<?xml version="1.0" standalone="no" ?> <!DOCTYPE
racine
  SYSTEM "http://serveur/chemin/MyDoc.dtd" [
  <!-- déclarations supplémentaires -->
  <! ... >
  <! ... >
]>
<!-- ici le reste du fichier XML -->
```

DTD externe privée, URL

- Les déclarations spécifiques au document restent définies de façon interne

Déclaration de DTD externe publique

42

- Utilisation du mot-clé PUBLIC

```
<!DOCTYPE racine
  PUBLIC "identifiant public" "url" >
```

- Exemple

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

Identificateurs publics

43

- PUBLIC** **"-//EBI//DTD My book//EN"** **"url"**
- **PUBLIC** keyword
 - **Identifier type** - Registered + / - / ISO
 - **Owner identifier**
 - **Public text class**
 - DTD, NOTATIONS, ENTITIES, TEXT
 - **Public text description**
 - **Public text language**
 - **url** non obligatoire mais conseillée

Exemples d'utilisation

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<INOTATION GIF89a PUBLIC "-//Compuserve//NOTATION
Graphics Interchange format 89a//EN" >
```

Les entités

44

- Sont des alias associant un nom à des « unités d'information »
- Les entités spécifiques au document sont décrites dans sa DTD interne
- Les entités plus générales sont décrites dans des DTD externes
- Chaque entité
 - est identifiée par un nom
 - est définie par une déclaration d'entité
 - est utilisée en appelant une référence d'entité



Utilisation des entités

45

- **Quand l'information**
 - Est utilisée dans plusieurs endroits
 - Ex. déclaration légale, caractère spécial
 - Est une partie d'un document qui doit être tronçonné pour rester gérable
 - Ex. livre = 1 fichier général + n chapitres soit n+1 fichiers en tout
 - Est conforme à un format de donnée différent de XML
 - Ex. : image.JPEG

Types d'entités

46

- **Entités internes**
 - générales
 - utilisées dans les documents XML
 - « paramètre »
 - utilisées dans les déclarations dans les DTD
- **Entités externes**
 - générales
 - paramètres
- **Entités analysables**
- **Entités non analysables**
- **Entités caractères**
 - déjà vues

Entités générales internes

47

```
<!ENTITY nom "chaîne de remplacement">
```

- **Entités analysables** utilisées uniquement dans le document
- **Référence** : &nom_entité;
- **Exemple**
 - Déclaration dans la DTD

```
<!ENTITY PCI "Permis de conduire informatique" >
```

 - Utilisation

```
<p>Le cours du PCI (&PCI;) se compose de...</p>
```

Entités générales externes

48

```
<!ENTITY nom SYSTEM "URI" >
```

- Permet de construire un document XML à partir de plusieurs autres documents
- **Référence** : &nom_entité;
- **Exemple**
 - Déclaration dans la DTD

```
<!ENTITY doc SYSTEM "http://toto.org/doc.xml" >
```

 - Utilisation

```
<aide>Voir cette URL : &doc; </aide>
```


Entités paramètres internes

49

- `<!ENTITY % nom "caractères de remplacement" >`
- Entités analysables uniquement utilisées dans les DTD
- Référence dans la DTD : `(%nom_entité;)` (parenthèses conseillées)
- Exemples
 - Déclarations DTD

```
<!ENTITY % tout "ANY" >
<!ENTITY % common "(para|list|table)">
```
 - Utilisations dans la DTD

```
<!ELEMENT paragraphe %tout; >
<!ELEMENT chapter ((%common;)*, section*)>
<!ELEMENT section (%common;)*>
```

Entités paramètres externes

50

- `<!ENTITY % nom SYSTEM "URI" >`
- Pour construire une DTD complexe à partir d'autres DTD complémentaires
- Référence dans la DTD : `%nom_entité;`
- Exemple
 - Déclaration dans la DTD

```
<!ENTITY % règles SYSTEM "http://toto.org/regles.dtd" >
```
 - Utilisation dans la DTD

```
%règles;
```

Entités analysables

51

- Le texte de remplacement fait partie intégrale du document
 - les données sont analysées correctement par le parser XML
- Déclaration dans la DTD comme ENTITY
- Utilisation avec `&nom;` ou `%nom;`

Entités non analysables

52

- `<!ENTITY % nom SYSTEM "URI" NDATA notation >`
- Pour déclarer un contenu non XML dans un document XML
 - fichier image, audio, etc.
- Référence : `&nom_entité;` uniquement comme attribut de type ENTITY
- Exemple
 - Déclaration DTD

```
<!NOTATION TIFF SYSTEM "format TIFF">
<!ENTITY photo SYSTEM "photo.tif" NDATA TIFF>
<!ELEMENT pic EMPTY>
<!ATTLIST pic name ENTITY #REQUIRED>
```
 - Utilisation dans le document XML

```
<pic name="photo" />
```

Déclarations de notations

53

- `<!NOTATION nom SYSTEM "URI" >`
- Pour
 - Identifier par un nom le format des entités non XML externes
 - Définir les formats des données et les applications qui permettent de les traiter
- Exemple

```
<!NOTATION GIF SYSTEM "GIF" >
<!NOTATION GIF89a PUBLIC "-//Compuserve//NOTATION
Graphics Interchange format 89a//EN" >
```

Construire une DTD

54

- Non trivial : il faut éviter de se tromper
 - Changer une DTD XML a des conséquences sur les documents qui la suivent
- Ressemble à la création d'un schéma de base de données
- Il faut considérer
 - Le problème de la granularité
 - La questions des attributs et des éléments
 - Les limitations inhérentes aux DTD

Identifier les données qui nécessitent d'être balisées

55

- Pour chaque unité d'information, déterminer
 - Peut-on lui donner un nom ?
 - Apparaît-elle tout le temps ?
 - Peut-il y en avoir plusieurs ?
 - Peut-on la décomposer en des unités plus petites ?
 - Y-a-t'il du contenu textuel qui ne change pas ?
 - Comment est-elle associée aux autres unités ?

Granularité

56

- **<PERSON>**
 <NAME>Jon Smith</NAME>
 </PERSON>
- **<PERSON>**
 <FORENAME>Jon</FORENAME>
 <SURNAME>Smith</SURNAME>
 </PERSON>

Éléments ou attributs ?

57

- Comment les données doivent-elles être encapsulées ?
 - `<book>`
 <title>The Forty-nine Steps</title>
 ...
 - `</book>`
 - `<book title="The Forty-nine Steps">`
 ...
 - `</book>`
- Tout dépend de ce que l'on veut faire...
- Il existe des avis tranchés...

Éléments ou attributs ? (2)

58

- Séparer le contenu des métadonnées
 - Données qui doivent être imprimées comme du texte comme contenu
 - Métadonnées comme attributs
- Règles générales
 - Si on enlève toutes les balises, le document doit encore être lisible et utilisable
 - S'il y a doute, utiliser un attribut

Limites des DTD

59

- XML est seulement une syntaxe
- XML ne porte pas de sémantique
- Uniquement description de structure
- Pas de types
- Un des moyens de pallier certains problèmes
 - XML-schema

Problèmes des DTD

60

- Une syntaxe de description non-XML, héritée de SGML
 - Oblige à apprendre un langage supplémentaire
 - Ne permet pas de manipuler les DTD avec des outils XML
- Pas assez de contraintes sur les données manipulées
 - Toute donnée est une chaîne de caractères
 - Impossible de
 - » spécifier des types simples
 - ex. entiers, dates, etc.
 - » spécifier des cardinalités simples
 - ex. « un ARTICLE aura entre 1 et 4 MOTS-CLE »
 - » spécifier des contraintes simples
 - ex. entier positif

XML-Schema

61

- Autre manière de spécifier des types de documents XML
- Le schéma est exprimé en XML
- Possibilité de spécifier plus de contraintes sur les données
- Possibilités avancées d'extension des schémas
- On élargit l'approche de gestion documentaire à celle plus générale de gestion de données

Comparaison DTD/Schémas

62

Caractéristique	DTD	Schémas
Syntaxe	Notation EBNF + pseudo-XML	XML 1.0
Outils	Outils SGML existants (chers et complexes)	Tous les outils XML existants et à venir
Supports DOM/SAX	Non	Oui (comme pour les fichiers XML).
Modèles de contenu	- Listes : ordonnées ou de choix - Cardinalité : 0, 1 ou plusieurs occurrences - Pas d'éléments nommés ou de groupes d'attributs.	- Listes : ordonnées et de choix (détails de contenus mixtes) - cardinalité : spécification d'un nombre exact d'occurrences possible - groupes de modèles nommés
Typage des données	Faible (chaînes, jetons nominaux, ID...)	Fort (nombres, chaînes, date/heure, booléen, structures...)
Héritage	Non	Oui
Extensibilité	Non (pas sans modification de la recommandation XML)	Oui (puisque fondés sur l'extensibilité de XML)
Contraintes légales	Compatibilité avec SGML	Aucune (simplement des « emprunts » aux DTD, comme pour les types de données)
Nombre de vocabulaires supportés	Une seule DTD par document	Autant que nécessaire (grâce aux espaces de noms)
Dynamisme	Aucune : les DTD sont en lecture seule	Peuvent être modifiés dynamiquement

Exercice 1 : Proposez plusieurs documents XML valides suivant la DTD suivante

```
<!ENTITY % opt_fields "year?, volume?, pages?, month?, url?,  
abstract?, note?";  
<!ENTITY % req_fields "author, title";  
<!ENTITY % key_atts "key ID #REQUIRED";  
  
<!ELEMENT bibtext-file  
(article|inproceedings|book|techreport|phdthesis|unpublished|misc  
)>  
  
<!ELEMENT key (#PCDATA)>  
<!ELEMENT author (name+)>  
<!ELEMENT name (#PCDATA)>  
<!ELEMENT title (#PCDATA)>  
<!ELEMENT booktitle (short, long?)>  
<!ELEMENT short (#PCDATA)>  
<!ELEMENT long (#PCDATA)>  
<!ELEMENT year (#PCDATA)>  
<!ELEMENT volume (#PCDATA)>  
<!ELEMENT pages EMPTY>  
<!ATTLIST pages  
first NMTOKEN #REQUIRED  
last NMTOKEN #REQUIRED>  
<!ELEMENT month (#PCDATA)>  
<!ATTLIST month  
mtype (short | long) "short">  
<!ELEMENT url EMPTY>  
<!ATTLIST url  
ftype (ps|pdf|html) #REQUIRED  
href CDATA #REQUIRED>  
<!ELEMENT abstract (#PCDATA)>  
  
(suite)  
<!ELEMENT number (#PCDATA)>  
<!ELEMENT note (#PCDATA)>  
<!ELEMENT publisher (#PCDATA)>  
<!ELEMENT institution (#PCDATA)>  
<!ELEMENT school (#PCDATA)>  
<!ELEMENT howpublished (#PCDATA)>  
<!ELEMENT address (#PCDATA)>  
<!ELEMENT inproceedings (%req_fields;, booktitle,  
%opt_fields;)>  
<!ATTLIST inproceedings %key_atts;>  
<!ELEMENT article (%req_fields;, journal,  
%opt_fields;)>  
<!ATTLIST article %key_atts;>  
<!ELEMENT book (%req_fields;, publisher,  
%opt_fields;)>  
<!ATTLIST book %key_atts;>  
<!ELEMENT techreport (%req_fields;, institution?,  
number?, %opt_fields;)>  
<!ATTLIST techreport %key_atts;>  
<!ELEMENT phdthesis (%req_fields;, school?,  
%opt_fields;)>  
<!ATTLIST phdthesis %key_atts;>  
<!ELEMENT unpublished (%req_fields;, %opt_fields;)>  
<!ATTLIST unpublished %key_atts;>  
<!ELEMENT misc (%req_fields;, howpublished,  
%opt_fields;)>  
<!ATTLIST misc %key_atts;>
```

```
(suite)  
<!ELEMENT abstract (#PCDATA)>  
<!ELEMENT number (#PCDATA)>  
<!ELEMENT note (#PCDATA)>  
<!ELEMENT publisher (#PCDATA)>  
<!ELEMENT institution (#PCDATA)>  
<!ELEMENT school (#PCDATA)>  
<!ELEMENT howpublished (#PCDATA)>  
<!ELEMENT address (#PCDATA)>  
<!ELEMENT inproceedings (%req_fields;, booktitle, %opt_fields;)>  
<!ATTLIST inproceedings %key_atts;>  
<!ELEMENT article (%req_fields;, journal, %opt_fields;)>  
<!ATTLIST article %key_atts;>  
<!ELEMENT book (%req_fields;, publisher, %opt_fields;)>  
<!ATTLIST book %key_atts;>  
<!ELEMENT techreport (%req_fields;, institution?, number?, %opt_fields;)>  
<!ATTLIST techreport %key_atts;>  
<!ELEMENT phdthesis (%req_fields;, school?, %opt_fields;)>  
<!ATTLIST phdthesis %key_atts;>  
<!ELEMENT unpublished (%req_fields;, %opt_fields;)>  
<!ATTLIST unpublished %key_atts;>  
<!ELEMENT misc (%req_fields;, howpublished, %opt_fields;)>  
<!ATTLIST misc %key_atts;>
```

64

Exercice 2

65

- Proposez deux DTD permettant de valider le document XML suivant

Un examen a entre 3 et 4 questions et chaque question a une ou plusieurs parties. Une partie se découpe en textes mélangés avec d'autres parties. Le code est uniquement alphanumérique, sans espace, la valeur de mois doit être une chaîne de caractère valide.

```
<examen code="coursXML">  
<titre>Outils et documents XML</titre>  
<date mois="jan" annee="2006"/>  
<questions>  
<question> <partie>Première partie<partie>Une sous-partie</partie>  
/partie</question>  
<question> <partie>Deuxième partie</partie> </question>  
<question> <partie> </question>  
</questions>  
</examen>
```

Exercice 1

66

- On veut représenter en XML des données concernant les étudiants en SIB et leurs enseignements
- Proposez une DTD