

Introduction à XML [DVIR3]

Informatique et systèmes
finalité Réseaux &
télécommunication

2016-2017

XML

- 1.Introduction
- 2.Langage XML
- 3.Document XML
- 4.DTD : Déclaration de la structure du document
- 5.XML Schema
- 6.Adressage de fragments xml : Xpath
- 7.Contenu et présentation : CSS et XSL
- 8.Langage de transformation XSLT
- 9.Langage de formatage XSL-FO
- 10.PHP - XML

XML

2

- 1.Introduction
- 2.Langage XML
- 3.Document XML
- 4.DTD: Déclaration de la structure du document**
- 5.XML Schema
- 6.Adressage de fragments xml": Xpath
- 7.Contenu et présentation": CSS et XSL
- 8.Langage de transformation XSLT
- 9.Langage de formatage XSL-FO
- 10.PHP - XML

Des recommandations permettant de définir des modèles électroniques d'informations définies sous forme de métalangage

- les **DTD** d'XML
- les **Schemas**
- les **NameSpace** font partie de cette classe, en ce sens qu'ils assurent la coopération de modèles

DTD : Définition de type de document

24

Une DTD permet de définir de façon formelle la structure de référence d'un document indépendamment de son contenu

= un ensemble de règles de structuration qui précise ce qui est autorisé ou non dans un document

Pourquoi Validation?

25

- Une DTD est une interface entre le producteurs et les consommateurs des données :
- Le producteur peut contrôler la qualité des données produites
- Le consommateur peut séparer la vérification syntaxique des données (parseur) de la logique de l'application

Pour des *parsers* "validants", la notion de DTD dans XML permet de définir un modèle de données qui servira ensuite à valider, de façon électronique, toutes les instances supposées être conformes au modèle

Pour des *parsers* "non validants", la notion de DTD sert simplement à fournir un réservoir permettant de décrire où sont des fichiers qui doivent être inclus dans les documents XML au travers d'entités générales

exemple

27

```
<liste-messages-panne>
  <message>
    <description>FUIITE HUILE MOTEUR</description>
    <marque-vehicule>PEUGEOT</ marque-vehicule >
    <type>MAU000AKL683</ type >
    <numero-panne>HM003</numero-panne>
    <procedure-reparation href="pr.xml">PSA HM003-01 </procedure-reparation>
  </message>
  <message>
    ...
  </message>
</liste-messages-panne>
```


DTD : exemple

28

```
<?xml version='1.0' encoding= 'UTF-8' ?>
```

```
<!-- Exemple de DTD -->
```

```
<!ELEMENT liste-message-panne (message+) >
```

```
<!ELEMENT message (description, marque-vehicule, type, numero-panne, procedure-reparation) >
```

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

```
<!ELEMENT marque-vehicule (#PCDATA) >
```

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

```
<!ELEMENT numero-panne (#PCDATA) >
```

```
<!ELEMENT procedure-reparation (#PCDATA) >
```

```
<!ATTLIST procedure-reparation href CDATA #IMPLIED>
```

Un document XML

29

une structure physique :

entités externes (fichiers ou unités de stockage); entité document

une structure logique :

– prologue :

1. instructions d'évaluation : e.g.

`<?xml version='1.0' encoding='UTF-8' standalone='no'>`

2. déclaration du type de document (DTD) : `<!DOCTYPE ...>`

- document type définition : obtenu par le parseur à partir de la déclaration de type de document
- élément racine suivi d'autres éléments

Exemple de document XML avec DTD

30

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<!DOCTYPE hello [
```

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

```
]>
```

```
<hello>
```

Hello XML!

```
</hello> – épilogue
```



Autre exemple : officiel.xml

31

<officiel> *Ce document contient des informations sur des cinémas.*

<cinema>

<nom> St. André des Arts </nom>

<adresse>

<ville> Paris </ville>

<rue> rue St. André des Arts </rue>

<numero> 13 </numero>

</adresse>

<seance heure = '18:00' ref_film = '&13' />

<seance heure = '20:00' ref_film = '&14' />

</cinema> *Voici l'information sur le film 'Brazil' :*

<film film_id = '&13' actors = '&156 &158' >

<titre> Brazil </titre><année> 1986 </année>

</film>

....

</officiel>

explication

32

- l'élément officiel est la racine du document de “contenu mélangé”
- un *cinema* a un nom, une adresse et zéro ou plusieurs séances
- une *seance* contient un attribut `ref_film` qui permet de référencer d'autres éléments *dans le même document*. Remarque : les références ne sont pas typées.
- les *films* sont identifiés par la valeur de l'attribut `film_id`

Sa grammaire : officiel.dtd

33

```
<!ELEMENT officiel (#PCDATA | cinema | film)*>
<!ELEMENT cinema (nom, adresse, (seance)*)>
<!ELEMENT nom (#PCDATA) >
<!ELEMENT adresse (ville, rue, (numero)?)>
<!ELEMENT seance EMPTY>
<!ATTLIST seance heure NMTOKEN #REQUIRED
              ref_film IDREF #REQUIRED>
<!ELEMENT film (titre, annee)>
<!ATTLIST film  film_id ID #REQUIRED
              actors IDREFS #IMPLIED>
<!ELEMENT titre (#PCDATA) >
....]>
```

Entités

34

- Les entités servent à donner un nom , raccourci à un contenu appelé "texte de remplacement"
- Ce texte remplacera les références d'entités qui utilisent ce nom dans le document XML

Types d'entités

35

- Entités générales et entités paramètres
- Entités internes et entités externes
- Entités analysables et entités non-analysables

Entités générales vs. entités paramètres

36

- générales : utilisables dans le contenu d'un document XML
- paramètres : utilisables dans un document DTD

Entités générales

37

La définition d'une entité dans la DTD

`<!ENTITY Name EntityDefinition >`

L'utilisation d'une entité dans le document : `&Name;`

Déclarations d'entités générales

38

`<!ENTITY nom_entité "valeur_entité" >`

Ex.: `<!ENTITY date_parution "31/10/2015" >`

...

`<p>`

` document paru le <i> &date_parution; </i>.`

`</p>`

Déclarations d'entités générales (2)

39

`<!ENTITY nom_entité "valeur_entité" >`

Ex.: `<!ENTITY date_parution "31/10/2015" >`

`<!ENTITY titre "Applications télécoms" >`

`<!ENTITY entete "&titre; paru le &date_parution;" >`

...

`<hr>`

` &entete; `

Déclarations d'entités paramètres

40

<!ENTITY % *nom_entité* "*valeur_entité*" >

Ex.: ...

<!ENTITY % residential_content "address, footage, rooms, baths">

<!ENTITY % rental_content "rent">

<!ENTITY % text "(#PCDATA|b|i|u) * " >

<!ELEMENT apartment (%residential_content;, %rental_content;)>

<!ENTITY % p %text; >

<!ENTITY % b %text; >

<!ENTITY % i %text; >

<!ENTITY % u %text; >

Entités internes vs. externes

41

Les entités internes font référence à un texte de remplacement défini dans la DTD du document.

```
<!ENTITY tecfaUnit "Unité de technologies de formation et apprentissage">
```

```
<!DOCTYPE [ <!ENTITY copyright
```

```
"Copyright 2002, As The World Corp. All rights reserved. For authorization  
contact legal@world.com."> ]>
```

Entités internes vs. externes

42

Les entités externes font référence à un texte de remplacement défini ailleurs à l'aide d'une URI, identificateur défini par une déclaration de notation.

```
<!ENTITY tecfaDesc SYSTEM  
"http://tecfa.unige.ch/..tecfa_description.xml">
```

```
<!DOCTYPE [ <!ENTITY  copyright SYSTEM  
"http://www.worldspins.com/legal/copyright.xml">]>
```

Exemple d'utilisation

43

<PRESSRELEASE>

<HEAD>

Mini-globe revolutionizes keychain industry

</HEAD>

<LEAD>

Today As The World Spins introduces a new approach to key chains.

With the new MINI-GLOBE keys can be kept inside a chain, called for upon demand, and stored safely. Never more will consumers lose a key or stand at a door flipping through a stack of keys seeking the right one.

</LEAD>

<LEGAL> **©right;** </LEGAL>

</PRESSRELEASE>

Entités analysables vs. non-analysables

44

- Les entités analysables "*parsed entities*" contiennent des données textuelles, des éléments bien formés (éléments de balisage), des références à des entités caractères, des références à d'autres entités générales, analysables, des sections CDATA de données non structurées, des instructions de traitement, des commentaires, analysables par un parseur XML.

Entités non-analysables

45

- Les entités non-analysables sont notamment des fichiers audios, vidéos ou images.
- La déclaration d'une entité non-analysable s'effectue en spécifiant le type de données de l'entité par l'intermédiaire du mot clé *NDATA*.
Chacune de ces entités est associée à une notation, identifiée par une chaîne de caractères par l'intermédiaire de l'instruction *<!NOTATION...>*.

Entités non-analysables

46

- La **déclaration** dans la DTD d'une entité non-analysable s'effectue comme suit :

`<!ENTITY nom_entite SYSTEM "URI" NDATA notation>`

- Les **références** d'entités non-analysables sont utilisées dans le document XML de la manière suivante :

`&nom_entite;`

- **Déclaration dans la DTD :**

`<!ENTITY illustration SYSTEM "image.gif" NDATA GIF>`

- **Utilisation de l'entité dans le document XML :** Ce genre d'entité ne pouvant être analysée doit être déclarée dans un conteneur afin d'être correctement traité par l'analyseur XML courant

`<image src="illustration"/>` Le nom des entités non-analysables est fourni comme valeur d'un attribut de type ENTITY ou ENTITIES dans l'instruction `<!ATTLIST...>`.

Entités non-analysables

47

```
<?xml version="1.0" encoding="UTF-8">
<!DOCTYPE collection [
  <!ELEMENT collection (album+)>
  <!ELEMENT album EMPTY>
  <!ATTLIST album src ENTITY #REQUIRED>
  <!NOTATION jpg SYSTEM "image/jpg">
  <!ENTITY photo SYSTEM "../marriage.jpg" NDATA jpg> ]>

<collection>
  <album src="photo"/>
</collection>
```

Entités générales prédéfinies

48

- <!ENTITY lt "&#60;"> <
- <!ENTITY gt ">"> >
- <!ENTITY amp "&#38;"> &
- <!ENTITY apos "'"> "
- <!ENTITY quot """> '

Les notations

Désignent par un "nom de notation" :

- le format d'une entité externe non analysable
- le format d'un élément qui possède un attribut de type NOTATION
- l'application cible d'une instruction de traitement

Déclarations de notations

50

<!NOTATION *nom_de_la_notation* SYSTEM "*uri_externe*" >

<!NOTATION *nom_de_la_notation* PUBLIC "*id_public*" "*uri_externe*">

<!NOTATION *nom_de_la_notation* PUBLIC "*id_public*" >

Utilisation de notations

(entités non-XML)

51

- **D**éclaration du format (type = application) pour entités non-XML
- Référence à une entité de type notation seulement possible comme valeur d'attribut

```
<!DOCTYPE exemple [  
  <!NOTATION jpg SYSTEM '/usr/local/bin/xv' >  
  <!ENTITY myphoto SYSTEM './moi.jpg' NDATA jpg >  
  <!ELEMENT exemple (person) >  
  <!ELEMENT person (name) >  
  <!ELEMENT name (#PCDATA) >  
  <!ATTLIST person photo ENTITY #IMPLIED photoType NOTATION (jpg) #IMPLIED >  
  
<exemple>  
  <person photo= "myphoto" photoType="jpg" ><name>Gus</name></person>  
</exemple>
```


Déclarations d'éléments

52

<!ELEMENT nom_de_l'élément contenu_de_l'élément >

- **ANY** : n'importe quel ensemble d'éléments ou/et de caractères
- **EMPTY** : élt doit être vide
- **(#PCDATA)** : élt peut être vide ou contenir uniquement des données textuelles
- **(#PCDATA |élément1 |élément2 |...) *** : élt vide ou contient des données textuelles avec un ou plusieurs élt énumérés dans la liste
- **(contenu)** : élt doit contenir ce qui est défini entre ()
- **(contenu)?** : élt doit contenir 0 ou 1 fois ce qui est défini entre ()
- **(contenu)*** : élt doit contenir 0,1 ou n fois ce qui est défini entre ()
- **(contenu)+** : élt doit contenir 1 ou n fois ce qui est défini entre ()
- **élément1, élément2,...** : élt doit contenir la suite ordonnée des élt spécifiés
- **élément1| élément2 |,...** : élt doit contenir un des élt spécifiés

Utilisation d'un attribut

54

Pour décrire le contenu d'un élément

<entry>

<word language = "en"> cheese </word>

<word language = "fr"> fromage </word>

<word language = "ro"> branza </word>

<meaning> Un plat préparé ... </meaning>

</entry>

Utilisation d'un attribut

55

Pour exprimer une dimension ou un type

```
<picture>  
  <height dim= "cm"> 2400 </height>  
  <width dim= "in"> 96 </width>  
  <data encoding = "gif" compression = "zip">  
    M05-.+C$@02!G96YE<FEC ...  
  </data>  
</picture>
```

Quand utiliser un attribut ?

56

```
<person ssno= "123 45 6789">  
  <name> F. MacNiel </name>  
  <email>  
    fmacn@dcscs.barra.ac.sc  
  </email>  
  ...  
</person>
```

```
<person>  
  <ssno> 123 45 6789 </ssno>  
  <name> F. MacNiel </name>  
  <email>  
    fmacn@dcscs.barra.ac.sc  
  </email>  
  ...  
</person>
```

Les déclarations d'attributs

57

```
<!ATTLIST nom_élément déclaration_attribut1  
                déclaration_attribut2...>
```

où chaque déclaration d'attribut a la syntaxe suivante :

```
nom_attribut type_attribut valeur_par_défaut
```

```
<!ATTLIST procedure-reparation href CDATA #IMPLIED>
```

Types d'attributs (type):

1. String : CDATA : attribut doit contenir une chaîne de caractères
2. Enumerated : séquence de valeurs alternatives séparées par |
3. ID, IDREF, IDREFS : identification et références
4. ENTITY/ENTITIES : nom d'une entité non analysée déclarée ailleurs
5. NMTOKEN/NMTOKENS : chaîne de caractères sans blancs
6. NOTATION : une ou plusieurs notations (séparées par |)

1.a. String : CDATA

59

<!ELEMENT nom (#PCDATA) >

<!ATTLIST nom titre CDATA #REQUIRED>

<personne>

 <nom titre="M."> Dupond </nom>

</personne>

1.b. String : CDATA

60

```
<!ELEMENT periode EMPTY>
```

```
<!ATTLIST periode heures CDATA #REQUIRED  
                minutes CDATA #REQUIRED  
                secondes CDATA #REQUIRED >
```

```
<periode heures="10" minutes="22" secondes="0"/>
```

2. Enumerated

61

(nom1 |nom2 |nom3|...)

Ex.:

<!ELEMENT data ANY>

<!ATTLIST data public (yes|no) #REQUIRED>

<data public="yes">...</data>

<data public="no">... </data>

3.a. ID

62

L'attribut sert d'identificateur unique sans valeur par défaut.
Son nom ne peut pas commencer par un chiffre.

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

```
<!ATTLIST personne titre CDATA #REQUIRED  
                    matricule ID #REQUIRED >
```

```
<personne titre="M." matricule="DUP001"> Dupond </personne>
```

3.b. IDREF

63

La valeur de l'attribut fait référence à la valeur d'un attribut ID d'une autre balise existante obligatoirement du document.

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

```
<!ATTLIST service chef IDREF #REQUIRED>
```

```
<service chef="DUP001"> .... </service>
```

3.c. IDREFS

64

L'attribut est une liste d'identificateurs existants obligatoirement dans le document, c.-à.-d. de valeurs d'un ou de plusieurs attributs de type ID

```
<!ELEMENT lesSimpson EMPTY>
```

```
<!ATTLIST lesSimpson matricules IDREFS #REQUIRED>
```

```
<lesSimpson matricules="SIM001 SIM002 SIM003"/>
```

Example

65

```
<family>
  <person id = "jane" mother="mary" father="john">
    <name> Jane Doe </name>
  </person>
  <person id = "john" children="jane jack">
    <name> John Doe </name>
  </person>
  ...
</family>
```

Attributs ID et IDREF(S)

66

```
<!DOCTYPE family [  
  <!ELEMENT family (person)*>  
  <!ELEMENT person (name)>  
  <!ELEMENT name (#PCDATA)>  
  <!ATTLIST person  
    id      ID      #REQUIRED  
    mother  IDREF   #IMPLIED  
    father  IDREF   #IMPLIED  
    children IDREFS  #IMPLIED>  
>
```

Spécification alternative

67

```
<!DOCTYPE family [  
  <!ELEMENT family (person)*>  
  <!ELEMENT person (mother?, father?, children*,name)>  
  <!ATTLIST person id ID #REQUIRED>  
  <!ELEMENT name (#PCDATA)>  
  <!ELEMENT mother EMPTY>  
  <!ATTLIST mother idref IDREF #REQUIRED>  
  <!ELEMENT father EMPTY>  
  <!ATTLIST father idref IDREF #REQUIRED>  
  <!ELEMENT children EMPTY>  
  <!ATTLIST children idrefs IDREFS #REQUIRED>  

```


4. a. ENTITY

68

- L'attribut doit être le nom d'une entité externe non-analysable "unparsed entity" définie par une déclaration d'entité dans la DTD.
- L'entité pointe vers une ressource externe (fichier) à l'aide d'une URI (image attachée au document)

<!ELEMENT image EMPTY>

<!ATTLIST image src ENTITY #REQUIRED>

4. b. ENTITIES

69

- L'attribut doit être une liste d' entités externes non-analysables
"unparsed entity" définie par des déclarations d'entité dans la DTD.

5. NMTOKEN / *NMTOKENS*

70

- L'attribut(s) doit être (*une liste de*) un nom qui respecte une syntaxe particulière (ensemble de lettres, de chiffres et de caractères '.', '-', '_', ':', ou d'autres caractères spéciaux Unicode).

6. NOTATION

71

- NOTATION (notation1 | notation2 | notation3 | ...)
- L'attribut fait référence à une des notations mentionnées, qui doivent être définies par une déclaration de notation dans la DTD.
- Cet attribut servira à l'application pour interpréter l'élément auquel il est joint.

<!ELEMENT graphe EMPTY>

<!ATTLIST graphe format NOTATION (autocad | indesign) #REQUIRED>

Valeurs par défaut d'un attribut

72

- **#REQUIRED** : pas de valeur par défaut mais attr. obligatoire
- **#IMPLIED** : pas de valeur par défaut et attr. non obligatoire
- **"valeur" ou 'valeur'** : valeur par défaut

<!ELEMENT layer ANY>

<!ATTLIST layer visibility (show| inherit|hide) "hide">

- **#FIXED "valeur"» ou 'valeur'** : valeur constante de l'attribut

Appel à une DTD externe

73

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<!DOCTYPE élément_racine_du_document SYSTEM "uri_de_la_DTD">
```

```
<!DOCTYPE liste-messages-panne SYSTEM "liste_pannes_dtd.dtd">
```

```
<!DOCTYPE book SYSTEM "http://www.site.com/dtds/book.dtd">
```

Appel à une DTD publique

74

Qui est rangée à un autre endroit, il faut ajouter le mot-clé **PUBLIC** à l'identificateur public. Celui-ci est utilisé par le parseur XML pour tenter de retrouver le contenu de la DTD.

Si la DTD n'est pas un standard ISO, ajouter dash (—)

S'il échoue, il doit utiliser l'URI.

```
<!DOCTYPE book PUBLIC "-//CompanyXYZ//DTD book//EN"  
"http://www.site.com/dtds/book.dtd">
```

Appel à une DTD interne

75

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<!DOCTYPE élément_racine_du_document    [règles_de_la_DTD...] >
```

```
<!DOCTYPE liste-messages-panne [  
    <!ELEMENT liste-messages-panne (message+)>  
    <!ELEMENT message (description, marque-vehicule, type, numero-  
panne, procedure-reparation)>  
    <!ELEMENT description (#PCDATA)>  
    ...  
]>  
<liste-messages-panne>  
    <message>.... </message>  
    <message>.... </message>  
</liste-messages-panne>
```


- DTD = grammaire (sans contexte)
- DTD = modèle de données
 - document XML = arbre **ordonné**
 - types atomiques limités (#PCDATA, ID, IDREF, ...)
 - références non typées (IDREF)
 - pas de types complexes
 - définition de tags globale

Les DTD sont trop "pauvres" pour représenter des modèles de données plus riches :

- la relation élément/sous-élément correspond essentiellement à la relation *part-of*
- un seul type atomique: #PCDATA
- une DTD n'est pas un document XML

Ajouter une DTD externe au document .xml mettant en évidence la structure des cours à l'ESI :

- Section
- Quadrimestre
- UE
- AA