

DTD

Document Type Definition

Yves Bekkers

Mise à jour : 22 mars 2010

DTD - Y. Bekkers - IFSIC 1

Plan

1. Introduction
2. Validation par DTD
3. Déclaration d'élément
4. Déclaration d'attribut
 - Graphe par identifiant
5. Déclaration d'entité
6. Conclusion

DTD - Y. Bekkers - IFSIC 2

1 .Introduction

DTD - Y. Bekkers - IFSIC 3

Un schéma de données = un dialecte

- Un schéma spécifie un dialecte XML
- Pour les applications : une *grammaire*
 - vérifier la conformité d'un document bien formé vis à vis du dialecte considéré
- Pour les utilisateurs : une *spécification*
 - spécifier, documenter, s'échanger un dialecte

DTD - Y. Bekkers - IFSIC 4

Relations de conformité

« Est conforme à »

Programmation	Définition de classe	Instance d'objet
SGBD	Définition de table	Instance de table
XML	Schéma de données	Instance de document

DTD - Y. Bekkers - IFSIC 5

Validation de document

```

    graph LR
        XML[XML] --> A1[Analyse syntaxique niveau 1]
        A1 --> BF[Bien formé]
        BF --> A2[Analyse syntaxique niveau 2]
        S[Schéma] --> A2
        A2 --> VD[Valide selon une DTD]
    
```

DTD - Y. Bekkers - IFSIC 6

Principaux schémas de données

- Les DTDs (de SGML)
- Les schémas XML (W3C):
 - <http://www.w3.org/TR/xmlschema-0/>
- RELAX :
 - <http://www.xml.gr.jp/relax/>
- Tree Regular Expression (TREG) :
 - <http://www.thaiopensource.com/trex/>
- Relax-NG :
 - <http://www.oasis-open.org/committees/relax-ng/>
- Schematron :
 - <http://www.ascc.net/xml/resource/schematron/schematron.html>

DTD - Y. Bekkers - IFSIC

7

2. DTD Document Type Definition

L'héritage de SGML

DTD - Y. Bekkers - IFSIC

8

5 types d'éléments

- élément vide
`<!ELEMENT nom EMPTY>`
- élément avec contenu indifférent
`<!ELEMENT nom ANY>`
- élément avec du texte seulement comme contenu
`<!ELEMENT nom (#PCDATA)>`
- élément avec des éléments seuls comme contenu
`<!ELEMENT nom (nom1 | nom2?)>`
`<!ELEMENT nom (nom1 , (nom2 | nom3)*)>`
- élément mixte
`<!ELEMENT nom (#PCDATA | nom1 | nom2)*>`

DTD - Y. Bekkers - IFSIC

9

Une DTD c'est quoi ?

- DTD acronyme pour
 - *Document Type Definition*
- Une grammaire qui exprime les contraintes sur la structure d'un document
- Sa spécification et celle d'XML n'en font qu'un.
- DTD où et comment ?
 - une description interne au document lui-même
 - un document séparé, référencé par le document lui-même

DTD - Y. Bekkers - IFSIC

10

Un exemple de document avec DTD

```
<?xml version="1.0"?>
<!DOCTYPE personne [
  <!ELEMENT personne (#PCDATA)>
]>
<personne>Jacques Dupont</personne>
```

DTD

DTD - Y. Bekkers - IFSIC

11

Autres usages d'une DTD

- Définir des macros (entités)
- Modularité (éclatement physique d'une DTD)
- Donner des valeurs par défaut aux attributs
- Définir des contraintes d'unicité d'identification
 - attributs ID
- Définir des contraintes de référence
 - attributs IDREF

DTD - Y. Bekkers - IFSIC

12

Une DTD c'est où

- Dans l'entête du document XML
- Trois méthodes
 - DTD interne dans le document
`<!DOCTYPE nom [... déclarations ...]>`
 - DTD externe dans un fichier (pour la partager)
`<!DOCTYPE nom SYSTEM "Fichier/URL">`
`<!DOCTYPE nom PUBLIC "identifiant" "Fichier">`
 - DTD Mixte (interne/externe)
`<!DOCTYPE élémentRacine SYSTEM "URL" [
... déclarations ...
>`

DTD - Y. Bekkers - IFSIC

13

DTD par identifiant public

- Problème des DTD externes
 - Il faut la charger (éventuellement à travers le WEB)
- Solution
 - Donner un identifiant public unique à chacune de vos DTD.
 - Les applications peuvent alors chercher une version locale grâce à l'identifiant unique
 - seulement si cela s'avère impossible, utiliser l'URI pour télécharger la DTD à partir d'un serveur externe.

DTD - Y. Bekkers - IFSIC

14

DTD externe (exemple)

```
<!ELEMENT carteDeVisite  
  (prénom, nom, organisme, adresse)>  
<!ELEMENT prénom    (#PCDATA) >  
<!ELEMENT nom        (#PCDATA) >  
<!ELEMENT organisme  (#PCDATA) >  
<!ELEMENT adresse    (numéro, voie, ville)>  
<!ELEMENT numéro     (#PCDATA) >  
...
```

DTD - Y. Bekkers - IFSIC

15

DTD externe (suite)

```
...  
<!ELEMENT voie      (#PCDATA) >  
<!ATTLIST voie  
  type (rue | avenue | impasse | cours  
  | square | boulevard | chemin | allée  
  | quai | route | passage | place  
  | rondPoint ) 'rue' >  
<!ELEMENT ville     (#PCDATA) >  
<!ATTLIST ville  
  codepostal CDATA #IMPLIED >
```

DTD - Y. Bekkers - IFSIC

16

3. Déclaration d'élément

- Syntaxe
`<!ELEMENT nom modèleDeContenu >`
 - *nom* : nom de l'élément
 - *modèleDeContenu* : expression définissant le contenu autorisé dans l'élément

DTD - Y. Bekkers - IFSIC

17

Déclaration d'élément

- Syntaxe
`<!ELEMENT nom modèleDeContenu >`
 - *nom* : nom de l'élément
 - *modèleDeContenu* : expression définissant le contenu autorisé dans l'élément

DTD - Y. Bekkers - IFSIC

18

Opérateurs d'expressions régulières

Sémantique	Opérateur
Enchaînement	... , ...
Choix
Zéro ou 1	...?
Zéro ou plus	...*
Un ou plus	...+
Groupe	(...)
un et un seul	nom de l'élément

DTD - Y. Bekkers - IFSIC

19

Opérateur d'expressions régulières

```
<!-- choix parmi deux -->
<!ELEMENT mobile (train | avion)>
<!-- séquence d'éléments -->
<!ELEMENT adresse (numéro, voie, ville)>
<!-- Au moins un élément -->
<!ELEMENT carnetDAdresse (carteDeVisite+)>
<!-- séquence d'éléments dont le dernier peut
apparaître 0 ou plusieurs fois -->
<!ELEMENT carteDeVisite (prénom, nom, organisme,
adresse, tel*)>
```

DTD - Y. Bekkers - IFSIC

20

Espaces non significatifs

- Une DTD renseigne les compilateurs sur les espaces non significatifs contenus dans un document XML
- Exemple :

```
<personne>␣
→ <nom>Durant</nom>␣
→ <prénom>Paul</prénom>␣
</personne>
```

élément sans
contenu textuel

DTD - Y. Bekkers - IFSIC

21

4. Déclaration d'attribut

DTD - Y. Bekkers - IFSIC

22

Déclaration d'attributs

- Associer une liste d'attributs à un élément
- Schéma de la déclaration

```
<!ATTLIST nomElément
  nomAttribut type Contrainte
  ...
  nomAttribut type Contrainte
```

} Liste de triplets

- Exemple

```
<!ATTLIST ville
  nom CDATA #IMPLIED } deux
  id ID #REQUIRED } attributs
```

DTD - Y. Bekkers - IFSIC

23

Nom d'attribut

- Syntaxe
 - Un nom d'attribut a la même syntaxe qu'un identificateur XML
- Exemples
 - codePersonne, a12, type, nom-personne

DTD - Y. Bekkers - IFSIC

24

Types d'attributs

Description	Type
Texte	CDATA
Type énuméré	(v1 v2 ...)
Définition d'identifiant unique	ID
Référence à identifiant	IDREF
un mot sans espace	NMTOKEN
nom d'entités externes	ENTITY
nom de notations	NOTATION

DTD - Y. Bekkers - IFSIC

25

Type énuméré

```
...
<!ELEMENT voie      (#PCDATA) >
<!ATTLIST voie
    type (rue | avenue | impasse |
         cours
         | square | boulevard | chemin |
         allée
         | quai | route | passage | place
         | rondPoint ) 'rue' >
```

DTD - Y. Bekkers - IFSIC

26

valeurs ID, IDREF, ENTITY, NMTOKEN

- Syntaxe
 - doivent respecter la syntaxe des noms d'éléments
- Contraintes
 - Un attribut ID doit identifier de manière unique un élément au sein d'un document considéré (*contrainte d'unicité*)
 - un attribut IDREF est contraint à prendre la valeur d'un attribut ID existant dans le document (*contrainte de référence*)

DTD - Y. Bekkers - IFSIC

27

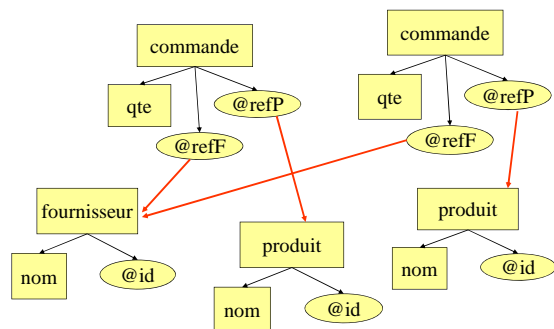
Type d'attributs pour construire un graphe de données

- Attribut de type ID :
 - Contrainte d'unicité (*clé primaire en SGBD*)
 - L'élément est identifié de manière unique (portée : le document)
- Attribut de type IDREF, IDREFS :
 - Contrainte d'intégrité (*clé étrangère en SGBD*)
 - Référence un élément identifié (ID)

DTD - Y. Bekkers - IFSIC

28

Exemple de graphe



DTD - Y. Bekkers - IFSIC

29

Construire un graphe exemple de document

```
<grossite>
  <fournisseur id="f1">
    <nom>machin</nom>
  </fournisseur>
  <fournisseur id="f2">
    <nom>sansos</nom>
  </fournisseur>
  <produit id="p1">
    <nom>carotte</nom>
  </produit>
  <produit id="p2">
    <nom>pomme</nom>
  </produit>
  <commande id="c1">
    <client>truc</client>
    <ligne qte="2"
      refF="f1"
      refP="p1"/>
    <ligne qte="10"
      refF="f2"
      refP="p2"/>
  </commande>
</grossite>
```

DTD - Y. Bekkers - IFSIC

30

Construire un graphe exemple de DTD

```
<!ATTLIST fournisseur
  id ID #REQUIRED
>
<!ATTLIST produit
  id ID #REQUIRED
>
<!ATTLIST ligne
  ref IDREF #REQUIRED
  refP IDREF #REQUIRED
  qte CDATA #REQUIRED
>
```

DTD - Y. Bekkers - IFSIC

31

Listes de valeurs

Type simple	Type liste
ID	IDS
IDREF	IDREFS
NMTOKEN	NMTOKENS
ENTITY	ENTITIES

Exemple : ``

DTD - Y. Bekkers - IFSIC

32

Contraintes d'occurrence

Type de contrainte	Expression de la contrainte
Val par défaut d'un type énuméré	'val'
Val par défaut	#DEFAULT val
Obligatoire	#REQUIRED
Non obligatoire	#IMPLIED
Valeur constante	#FIXED val

DTD - Y. Bekkers - IFSIC

33

Déclarations d'attributs - Exemples

```
<!ATTLIST voie
  type (rue | boulevard | place) 'rue'
>
<!ATTLIST ville
  codepostal CDATA #IMPLIED
  nom CDATA #IMPLIED
  id ID #REQUIRED
>
```

DTD - Y. Bekkers - IFSIC

34

5. Déclaration d'entités

Un héritage de SGML

DTD - Y. Bekkers - IFSIC

35

Rappel - Entités caractères

- Notation qui permet de désigner un caractère unicode par son code

	Référence	Caractère
En décimal	<code>&#238;</code>	î
	<code>&#x2200;</code>	∅
En hexadécimal	<code>&#x0152;</code>	Œ
	<code>&#xA9;</code>	©

- Utilisation "le gîte et le couvert"

`<titre>le gîte et le couvert</titre>`

DTD - Y. Bekkers - IFSIC

36

Entité – qu'est-ce que c'est ?

- Sorte de *d'abréviations* (ou de macro) qui associe
 - un nom d'entité
 - à un contenu d'entité qui est
 - Un simple texte ou un fragment de document XML
- Définition


```
<!ENTITY dtd "Document Type Definition">
<!ENTITY chap1 SYSTEM "chapitre1.xml">
```
- Utilisation on pose une référence
 - dans les contenus d'élément ou dans les valeurs d'attributs


```
&dtd; ou &chap1;
```
 - la référence est remplacée par le contenu de l'entité

DTD - Y. Bekkers - IFSIC

37

Entités -syntaxe

- Schéma de la définition


```
<!ENTITY nom [SYSTEM] "valeur">
```
- Syntaxe des références


```
&nom;
```
- La valeur associée peut contenir des balises :

```
<!ENTITY ifsic "Institut de Formation Supérieur ..." >
<!ENTITY piedDePage '<hr size="1"/>' >
```

DTD - Y. Bekkers - IFSIC

38

Trois usages

- Créer une abréviation


```
<!ENTITY dtd "Document Type Definition">
```
- Créer un lien vers une source de données externe (construction modulaire)


```
<!ENTITY chap1 SYSTEM "chapitre1.xml">
```
- Exprimer la transcriptions de signes spéciaux.

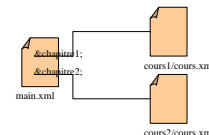

```
<!ENTITY euro "&#x20AC;">
```

DTD - Y. Bekkers - IFSIC

39

Construction modulaire

```
<?xml version="1.0"?>
<!DOCTYPE livre [
  <!ELEMENT livre (html*)>
  <!ENTITY chapitre1 SYSTEM "cours1/cours.xml">
  <!ENTITY chapitre2 SYSTEM "cours2/cours.xml">
]>
<livre>
&chapitre1;
&chapitre2;
</livre>
```



DTD - Y. Bekkers - IFSIC

40

Caractères spéciaux

- 5 entités prédéfinies

Référence	glyphe	Nom
&	&	ampersand
<	<	plus petit
>	>	plus grand
'	'	apostrophe
"	"	double quote

- Utilisation : "A >5"


```
<if>A &gt; 5</if>
```

DTD - Y. Bekkers - IFSIC

41

Deux sortes d'entités

- **Entités générales** pour insérer du texte
 - dans la DTD
 - <!ENTITY dtd "Definition type document">
 - dans le document XML, en dehors de la DTD
 - Référence : &dtd;
- **Entités paramètres** pour insérer du texte
 - Dans la DTD seulement
 - <!ENTITY % contenuAdresse "ville,rue">
 - Référence : %contenuAdresse;

DTD - Y. Bekkers - IFSIC

42

Une notion d'héritage (1)

Héritage de contenus d'élément

- Déclaration

```
<!ENTITY % pos "X,Y">
```

- Utilisation

```
<!ELEMENT carre (%pos;,lg)>  
<!ELEMENT rectangle (%pos;,hauteur,largeur)>  
<!ELEMENT cercle (%pos;,diametre)>
```

DTD - Y. Bekkers - IFSIC

43

Une notion d'héritage (2)

Héritage d'attributs

- Déclaration

```
<!ENTITY % pos  
"X CDATA #REQUIRED  
Y CDATA #REQUIRED">
```

- Utilisation

```
<!ELEMENT carre (EMPTY)>  
<!ATTLIST carre %pos; lg CDATA #REQUIRED>  
<!ELEMENT cercle (EMPTY)>  
<!ATTLIST cercle %pos; diam CDATA #REQUIRED>
```

DTD - Y. Bekkers - IFSIC

44

Ce que vous avez vu aujourd'hui

- Introduction
- Validation par DTD
- Déclaration d'élément
- Déclaration d'attribut
 - Graphe par identifiant
- Déclaration d'entité

DTD - Y. Bekkers - IFSIC

45

6. Conclusion

DTD - Y. Bekkers - IFSIC

46

La force des DTDs

- Un moyen compact de spécifier un dialecte XML
- 20 ans d'expérience d'utilisation des DTDs dans les milieux SGML
 - Des centaines de langages normalisés par leur DTD (HTML en premier...)

DTD - Y. Bekkers - IFSIC

47

Limites dans la spécification des contraintes de type

- Les DTDs ne permettent pas de *typer* les chaînes contenues dans les éléments et dans les attributs
 - On aimerait exprimer qu'un contenu d'élément ou d'attribut est un *entier positif*, ou un *entier dans l'intervalle [0..99]*
 - Quelque chose que l'on sait faire dans les SGBDRs depuis longtemps
 - Tout ce que l'on sait exprimer c'est qu'un contenu est une *chaîne* !
- La spécification de la cardinalité d'un élément est pauvre : *?,+,* (0 ou 1, 1 ou plus, 0 ou plus)*
 - On aimerait contraindre la cardinalité d'un élément sur un intervalle quelconque (entre 3 et 7 par exemple)

DTD - Y. Bekkers - IFSIC

48

DTD et espace de noms

- Les DTDs et les espaces de noms ne font pas bon ménage.

DTD - Y. Bekkers - IFSIC

49

Autres problèmes des DTDs

- N'utilise pas la syntaxe XML et n'offre pas de version en XML
- Sa spécification n'est pas séparée de celle d'XML (difficile de la faire évoluer ...)
- Très peu de support pour la modularité et la réutilisation des descriptions
- Pas de notion d'héritage, d'extension
- Documentation pas assez structurée
- Portée globale pour les éléments et les attributs
- Des valeurs par défaut pour les attributs mais pas pour les éléments

DTD - Y. Bekkers - IFSIC

50

Quelle suggestion ?

- Il y a de la marge pour la définition d'une nouvelle technologie de modélisation plus précise ...
 - C'est la brèche ouverte par les *Schémas XML* et les nombreuses autres propositions équivalentes

DTD - Y. Bekkers - IFSIC

51

Les DTDs restent utiles

- Quand on a besoin d'une spécification compacte d'un dialecte XML
- Quand la chose importante du dialecte XML considéré est la structure arborescente de ses éléments et non le typage de ses feuilles
- Quand les outils dont on dispose supportent mieux les DTDs que les schémas

DTD - Y. Bekkers - IFSIC

52

Outils de validation

DTD - Y. Bekkers - IFSIC

53

Outils de validation

- Les compilateurs en général
- Les éditeurs XML (qui permettent une saisie contextuelle assistée grâce aux DTDs)
 - XMLSpy
 - Eclipse avec des plug-ins *éditeurs XML*
 - WTP
 - XMLBuddy
 - Nombreux autres éditeurs

DTD - Y. Bekkers - IFSIC

54

Utiliser Eclipse pour saisir et valider des documents XML

DTD - Y. Bekkers - IFSIC

55

Plugins utilisés à l'IFSIC pour XML

- Éditeur XML de **WTP** (gratuit)
- Transformation de document XML
`de.eclipseproject.transclipse version 0.2.0`
 - Transformateurs XSLT externes possibles
- Interrogation XPath
`org.xpathexplorer.xpe version 1.20030402.1343`
- Visualisation d'un document SVG
`com.dlsc.batik.viewer version 1.0.0`

DTD - Y. Bekkers - IFSIC

56

Services offerts par l'éditeur XML de WTP

- Éditeur syntaxique guidé par les DTD ou les schémas XML
 - Aide contextuelle en ligne
- Vérification de validité (`Validate`)
- Formatage de documents (`Format`)
- Deux vues du document : `Design`, `Source`
- Création d'un document XML par
File>New>Others...>xml>xml

DTD - Y. Bekkers - IFSIC

57