

XML DTD

Marc Chevaldonné
marc.chevaldonne@u-clermont1.fr
<http://marc.chevaldonne.free.fr>

Année scolaire 2010 - 2011

Qu'est-ce qu'une DTD ?

- DTD = Document Type Definition : définition de type de document.
- définit des règles pour l'écriture de documents XML
- précise quels éléments, attributs, entités, ainsi que leur contenu, peuvent apparaître dans un document XML valide par rapport à cette DTD
- un document XML **valide** est un document XML qui respecte les règles imposées par une DTD
un document XML valide est donc bien formé
un document XML bien formé n'est pas nécessairement valide

Intérêts de la DTD

- Créer une famille de documents XML respectant les mêmes règles
Créer un langage XML
- Permettre et garantir l'utilisation par une application de n'importe quel document XML de cette famille
- Note : aujourd'hui, il existe une méthode plus puissante que la DTD pour définir un langage XML : le XML Schema.
La DTD est plus ancienne¹ et en conséquence encore présente.

¹ vient du SGML. La version XML est simplifiée

Exemple de DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT biographie (genre*, instrument*, histoire?, discographie?)>
  <!ATTLIST biographie nom CDATA #REQUIRED
    prénom CDATA #REQUIRED
    naissance CDATA #IMPLIED
    mort CDATA #IMPLIED>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT instrument (#PCDATA)>
<!ELEMENT histoire (#PCDATA | genre | instrument | date |
  groupe | musicien | disque | photographe | label)*>
<!ELEMENT date (#PCDATA)>
<!ELEMENT groupe (#PCDATA)>
<!ELEMENT musicien (((nom, prénom?) | (prénom?, nom)), instruments?)>
<!ELEMENT instruments (instrument+)>
<!ELEMENT disque (titre, musicien*)>
  <!ATTLIST disque parution NMTOKEN #IMPLIED>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT photographe (#PCDATA)>
<!ELEMENT discographie (disque+)>
  <!ATTLIST discographie complète (oui | non) 'non'>
<!ELEMENT label (#PCDATA)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prénom (#PCDATA)>
```

Écrire une DTD

- **Déclaration d'éléments**

nom XML



- syntaxe : `<!ELEMENT nom_élément (modèle_contenu)>`

décrit le contenu que peut avoir nom_élément (format de texte ou sous-éléments)

- élément texte : `<!ELEMENT disque (#PCDATA)>`

```
<disque>
  Olé
</disque>
```

```
<disque>
  <titre>Olé</titre>
</disque>
```

- élément vide : `<!ELEMENT image EMPTY>`

```
<image/>
<image src="photo.jpg" width="640" height="480"/>
<image src="photo.jpg" width="640" height="480"></image>
```

Écrire une DTD

- Déclaration d'éléments

- élément contenant des sous-éléments : `<!ELEMENT disque (titre, artiste)>`

DTD

```
<!ELEMENT disque (titre, artiste)>
```

document XML valide

```
<disque>
  <titre>Head Hunters</titre>
  <artiste>Herbie Hancock</artiste>
</disque>
```

document XML non valide

```
<disque>
  <del>artiste</del>Britney Spears</del>
  <del>titre</del>Blackout</del>
</disque>
```

Écrire une DTD

- Déclaration d'éléments

- indicateurs d'occurrences des sous-éléments (ou suffixes)

défait	1 fois
?	0 ou 1 fois
*	0 ou plusieurs fois
+	1 ou plusieurs fois

DTD

```
<!ELEMENT disque (titre, artiste+, année?)>
```

document XML valide

```
<disque>  
  <titre>Face To Face</titre>  
  <artiste>Eddy Louiss</artiste>  
  <artiste>Richard Galliano</artiste>  
  <année>2000</année>  
</disque>
```

document XML non valide

```
<disque>  
  <titre>Jump Back</titre>  
  <année>1971</année>  
  <année>1993</année>  
</disque>
```

Écrire une DTD

- Déclaration d'éléments

- choix

DTD

```
<!ELEMENT contact (courriel | téléphone | adresse)>
```

document XML valide

```
<contact>
  <courriel>truc@muche.fr</courriel>
</contact>
<contact>
  <téléphone>0473012345</téléphone>
</contact>
```

document XML non valide

```
<contact>
— <courriel>truc@muche.fr</courriel>
— <téléphone>0473012345</téléphone>
</contact>
```


Écrire une DTD

- Déclaration d'éléments
 - combinaisons (de séquences, de suffixes et de choix)

DTD

```
<!ELEMENT cercle (((centre, (rayon | diamètre)) | ((rayon | diamètre), centre))>  
<!ELEMENT centre ((x, y) | (y, x))>  
<!ELEMENT rayon (#PCDATA)>  
<!ELEMENT diamètre (#PCDATA)>  
<!ELEMENT x (#PCDATA)>  
<!ELEMENT y (#PCDATA)>
```

document XML valide

```
<cercle>  
  <rayon>1</rayon>  
  <centre>  
    <y>0.5</y>  
    <x>1.2</x>  
  </centre>  
</cercle>
```

Écrire une DTD

- Déclaration d'éléments

- élément à contenu mixte

DTD

```
<!ELEMENT recette (#PCDATA | ingrédient)*>  
<!ELEMENT ingrédient (#PCDATA)>
```

document XML valide

```
<recette>Ajouter ensuite le <ingrédient>sucre</  
ingrédient> et la <ingrédient>farine</ingrédient>, puis  
incorporer les <ingrédient>blancs d'oeufs</ingrédient>  
montés en neige bien ferme.</recette>
```

- élément ANY

- utile pendant la conception de la DTD
- mais à éviter dans une DTD terminée

DTD

```
<!ELEMENT elt ANY>
```

Écrire une DTD

- Déclaration d'attributs

DTD

```
<!ELEMENT elt (...)>  
<!ATTLIST elt  
    attr1 type1 declar_defaut1  
    attr2 type2 declar_defaut2 ... >
```

Écrire une DTD

- Déclaration d'attributs

DTD

```
<!ELEMENT elt (...)>  
<!ATTLIST elt  
    attr1 type1 declar_defaut1  
    attr2 type2 declar_defaut2 ... >
```

Noms XML



Écrire une DTD

- Déclaration d'attributs

DTD

```
<!ELEMENT elt (...)>  
<!ATTLIST elt  
  attr1 type1 declar_defaut1  
  attr2 type2 declar_defaut2 ... >
```

CDATA contenu textuel
NMTOKEN

NMTOKENS
énumération
ENTITY
ENTITIES
ID
IDREF
IDREFS
NOTATION

DTD

```
<!ELEMENT disque EMPTY>  
<!ATTLIST disque nb CDATA '1'>
```

XML valide

```
<disque nb="3"/>  
<disque></disque>
```

Écrire une DTD

- Déclaration d'attributs

DTD

```
<!ELEMENT elt (...)>  
<!ATTLIST elt  
  attr1 type1 declar_defaut1  
  attr2 type2 declar_defaut2 ... >
```

CDATA
NMTOKEN

unité lexicale nominale (nom XML dont le premier caractère est traité comme les autres)

NMTOKENS
énumération

Nom XML / NMTOKEN

_1998

txt

ENTITY

ENTITIES

ID

Miles_Davis

IDREF

~~Nom XML / NMTOKEN~~

IDREFS

1998

NOTATION

.txt

~~Nom XML / NMTOKEN~~

Miles Davis

Écrire une DTD

- Déclaration d'attributs

DTD

```
<!ELEMENT elt (...)>  
<!ATTLIST elt  
  attr1 type1 declar_defaut1  
  attr2 type2 declar_defaut2 ... >
```

CDATA
NMTOKEN

NMTOKENS liste d'unités lexicales nominales

énumération

ENTITY

ENTITIES

ID

IDREF

IDREFS

NOTATION

Écrire une DTD

- Déclaration d'attributs

DTD

```
<!ELEMENT elt (...)>  
<!ATTLIST elt  
  attr1 type1 declar_default1  
  attr2 type2 declar_default2 ... >
```

CDATA
NMTOKEN

NMTOKENS

énumération

ENTITY

ENTITIES

ID

IDREF

IDREFS

NOTATION

liste de valeurs possibles (unités lexicales nominales)

DTD

```
<!ELEMENT cours EMPTY>  
<!ATTLIST cours jour (lundi | mardi |  
mercredi | jeudi | vendredi) #REQUIRED>
```

XML valide

```
<cours jour="jeudi"/>
```


Écrire une DTD

- Déclaration d'attributs

DTD

```
<!ELEMENT elt (...)>  
<!ATTLIST elt  
  attr1 type1 declar_defaut1  
  attr2 type2 declar_defaut2 ... >
```

CDATA
NMTOKEN

NMTOKENS
énumération

ENTITY entité non analysée (cf. plus loin)

ENTITIES liste d'entités non analysées

ID

IDREF

IDREFS

NOTATION

Écrire une DTD

- Déclaration d'attributs

DTD

```
<!ELEMENT elt (...)>  
<!ATTLIST elt  
  attr1 type1 declar_defaut1  
  attr2 type2 declar_defaut2 ... >
```

CDATA
NMTOKEN

NMTOKENS
énumération
ENTITY
ENTITIES

ID identifiant unique
IDREF référence à un identifiant unique
IDREFS liste de références à un identifiant unique
NOTATION

Écrire une DTD

- Déclaration d'attributs

DTD

```
<!ELEMENT elt (...)>  
<!ATTLIST elt  
  attr1 type1 declar_defaut1  
  attr2 type2 declar_defaut2 ... >
```

CDATA
NMTOKEN

NMTOKENS
énumération
ENTITY
ENTITIES
ID

IDREF
IDREFS

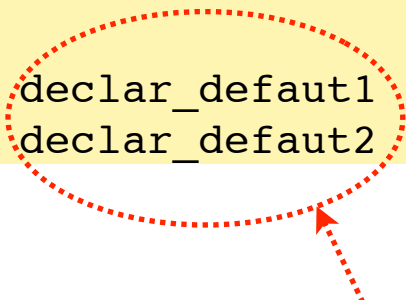
NOTATION contient le nom d'une notation déclarée dans la DTD

Écrire une DTD

- Déclaration d'attributs

DTD

```
<!ELEMENT elt (...)>  
<!ATTLIST elt  
  attr1 type1 declar_default1  
  attr2 type2 declar_default2 ... >
```



#IMPLIED	optionnel
#REQUIRED	obligatoire
#FIXED 'valeur'	valeur fixe non modifiable
'valeur'	valeur par défaut

Écrire une DTD

- Lier une DTD et un fichier XML
 - DTD interne
 - DTD externe
 - DTD mixte

Écrire une DTD

- Lier une DTD et un fichier XML
 - DTD interne

document
XML valide

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE discothèque [
  <!ELEMENT discothèque (disque+)>
  <!ELEMENT disque ((titre, artiste+) | (artiste+, titre))>
  <!ELEMENT titre (#PCDATA)>
  <!ELEMENT artiste ((nom, prénom) | (prénom, nom))>
  <!ELEMENT nom (#PCDATA)>
  <!ELEMENT prénom (#PCDATA)> ]>
<discothèque>
  <disque>
    <titre>Transpacifik</titre>
    <artiste><prénom>Bojan</prénom><nom>Zulfikarpašić</nom></artiste>
  </disque>
  <disque>
    <artiste><prénom>Michel</prénom><nom>Portal</nom></artiste>
    <artiste><prénom>Richard</prénom><nom>Galliano</nom></artiste>
    <titre>Blow Up</titre>
  </disque>
</discothèque>
```

Écrire une DTD

- Lier une DTD et un fichier XML
 - DTD externe

DTD externe

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT discothèque (disque+)>
<!ELEMENT disque ((titre, artiste+) | (artiste+, titre))>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT artiste ((nom, prénom) | (prénom, nom))>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prénom (#PCDATA)>
```

document
XML valide

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE discothèque SYSTEM "http://.../DTDexterne.dtd">
<discothèque>
  <disque>
    <artiste><prénom>Michel</prénom><nom>Portal</nom></artiste>
    <artiste><prénom>Richard</prénom><nom>Galliano</nom></artiste>
    <titre>Blow Up</titre>
  </disque>
</discothèque>
```

Écrire une DTD

- Lier une DTD et un fichier XML
 - DTD mixte

DTD externe

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT disque ((titre, artiste+) | (artiste+, titre))>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT artiste ((nom, prénom) | (prénom, nom))>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prénom (#PCDATA)>
```

document
XML valide

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE discothèque SYSTEM "http://.../DTDexterne.dtd" [
  <!ELEMENT discothèque (disque+)> ]>
<discothèque>
  <disque>
    <artiste><prénom>Michel</prénom><nom>Portal</nom></artiste>
    <artiste><prénom>Richard</prénom><nom>Galliano</nom></artiste>
    <titre>Blow Up</titre>
  </disque>
</discothèque>
```


Écrire une DTD

- Déclaration d'entités (≅ raccourcis ou références)
 - entités internes
 - entités externes
 - entités externes analysées (ou XML)
 - entités externes non analysées (ou non XML)
 - entités paramètres
 - entités paramètres internes
 - entités paramètres externes
- inclusion conditionnelle

Écrire une DTD

- Déclaration d'entités (≅ raccourcis ou références)

- entités internes

DTD

```
<!ENTITY symbole "valeur">
```

référence

```
&symbole;
```

- exemple :

DTD

```
<!ENTITY mail "machin@bidule.fr">
```

XML valide

```
<contact><mail>&mail;</mail></contact>
```

interprété par le parseur comme :

```
<contact><mail>machin@bidule.fr</mail></contact>
```

Écrire une DTD

- Déclaration d'entités (≅ raccourcis ou références)
 - entités externes analysées : représente un morceau d'XML

DTD

```
<!ENTITY doc_xml SYSTEM "url_doc_xml">
```

référence

```
&doc_xml;
```

le morceau peut être mal formé mais le résultat final doit être bien formé

entité externe XML (mal formée, il manque une racine)

```
<artiste>Charlie Parker</artiste>  
<artiste>Thelonious Monk</artiste>
```

DTD

```
<!ENTITY file SYSTEM "/musique/jazz.xml">
```

XML valide

```
<discothèque>  
  <artiste>Sonny Rollins</artiste>  
  &file;  
</discothèque>
```

interprété par le parseur comme :

```
<discothèque>  
  <artiste>Sonny Rollins</artiste>  
  <artiste>Charlie Parker</artiste>  
  <artiste>Thelonious Monk</artiste>  
</discothèque>
```

Écrire une DTD

- Déclaration d'entités (≅ raccourcis ou références)
 - entités externes non analysées : représente n'importe quoi

- déclarer le type

```
<!NOTATION jpeg SYSTEM "image/jpeg">
```

- déclarer l'entité

```
<!ENTITY bird SYSTEM "http://  
lavoixdecassandre.hautetfort.com/  
images/medium_Charlie_  
Parker.jpg" NDATA jpeg>
```

- déclarer un attribut typé

```
<!ELEMENT artiste (#PCDATA)>  
  <!ATTLIST artiste image ENTITY #IMPLIED>
```

- référencer l'entité comme attribut

```
<artiste image="bird">  
  Charlie Parker</artiste>
```

Écrire une DTD

- Déclaration d'entités (≡ raccourcis ou références)
 - entités paramètres internes : permet de factoriser et paramétrer les DTD

DTD

```
<!ENTITY % nom_ent "valeur_ent">
```

référence

```
%nom_ent;
```

les références doivent obligatoirement se trouver dans une DTD externe

- exemple :

DTD

```
<!ENTITY % individu "((nom, prénom) | (prénom, nom)), naissance?, mort?">  
<!ELEMENT musicien (%individu;, instrument+)>  
<!ELEMENT compositeur (%individu;, composition+)>
```

XML valide

```
<musicien>  
  <nom>Parker</nom><prénom>Charlie</prénom>  
  <naissance>1920</naissance><mort>1955</mort>  
  <instrument>saxophone alto</instrument>  
</musicien>  
<compositeur>  
  <prénom>Ornette</prénom><nom>Coleman</nom>  
  <naissance>1930</naissance>  
  <composition>Lonely Woman</composition>  
</compositeur>
```

Écrire une DTD

- Déclaration d'entités (≅ raccourcis ou références)
 - entités paramètres externes : permet des morceaux de DTD externes

DTD

```
<!ENTITY % dtd_ext SYSTEM "dtd_externe.dtd">
```

référence

```
%dtd_ext;
```

- Inclusion conditionnelle

IGNORE met en commentaire un morceau de déclaration

```
<![ IGNORE [  
    <!ELEMENT qqchse (#PCDATA)>  
]]>
```

INCLUDE indique que les déclarations sont utilisées

```
<![ INCLUDE [  
    <!ELEMENT qqchse (#PCDATA)>  
]]>
```

Écrire une DTD

- Déclaration d'entités (≅ raccourcis ou références)
 - exemple d'utilisation d'entités paramètres externes et d'inclusion conditionnelle

DTD

```
<!ENTITY % détails "INCLUDE">
<!ENTITY % zik "((nom, prénom) | (prénom, nom)),
naissance?, mort?">
<!ELEMENT musique (musicien | compositeur)*>
<![%détails;[
  <!ELEMENT musicien (%zik, instrument+)>
  <!ELEMENT compositeur (%zik, composition+)>
  <!ELEMENT instrument (#PCDATA)>
  <!ELEMENT composition (#PCDATA)>
  <!ELEMENT nom (#PCDATA)>
  <!ELEMENT prénom (#PCDATA)>
  <!ELEMENT naissance (#PCDATA)>
  <!ELEMENT mort (#PCDATA)>
]]>
```

Document XML valide

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE musique SYSTEM "externe.dtd" [
  <!ENTITY % détails "IGNORE">
  <!ELEMENT musicien (#PCDATA)
  <!ELEMENT compositeur (#PCDATA) ]>
<musique>
  <musicien>Charlie Parker</musicien>
  <compositeur>O. Coleman</compositeur>
  <musicien>Sonny Rollins</musicien>
</musique>
```

Écrire une DTD

- Déclaration d'entités (≅ raccourcis ou références)
 - exemple d'utilisation d'entités paramètres externes et d'inclusion conditionnelle

DTD

```
<!ENTITY % détails "INCLUDE">
<!ENTITY % zik "((nom, prénom) | (prénom, nom)),
naissance?, mort?">
<!ELEMENT musique (musicien | compositeur)*>
<![%détails;[
  <!ELEMENT musicien (%zik, instrument+)>
  <!ELEMENT compositeur (%zik, composition+)>
  <!ELEMENT instrument (#PCDATA)>
  <!ELEMENT composition (#PCDATA)>
  <!ELEMENT nom (#PCDATA)>
  <!ELEMENT prénom (#PCDATA)>
  <!ELEMENT naissance (#PCDATA)>
  <!ELEMENT mort (#PCDATA)>
]]>
```

entité paramètre interne

Document XML valide

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE musique SYSTEM "externe.dtd" [
  <!ENTITY % détails "IGNORE">
  <!ELEMENT musicien (#PCDATA)
  <!ELEMENT compositeur (#PCDATA) ]>
<musique>
  <musicien>Charlie Parker</musicien>
  <compositeur>O. Coleman</compositeur>
  <musicien>Sonny Rollins</musicien>
</musique>
```


Écrire une DTD

- Déclaration d'entités (≡ raccourcis ou références)
 - exemple d'utilisation d'entités paramètres externes et d'inclusion conditionnelle

DTD

```
<!ENTITY % détails "INCLUDE">
<!ENTITY % zik "((nom, prénom) | (prénom, nom)),
naissance?, mort?">
<!ELEMENT musique (musicien | compositeur)*>
<![%détails;[
  <!ELEMENT musicien (%zik; instrument+)>
  <!ELEMENT compositeur (%zik; composition+)>
  <!ELEMENT instrument (#PCDATA)>
  <!ELEMENT composition (#PCDATA)>
  <!ELEMENT nom (#PCDATA)>
  <!ELEMENT prénom (#PCDATA)>
  <!ELEMENT naissance (#PCDATA)>
  <!ELEMENT mort (#PCDATA)>
]]>
```

entité paramètre interne

et ses références

Document XML valide

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE musique SYSTEM "externe.dtd" [
  <!ENTITY % détails "IGNORE">
  <!ELEMENT musicien (#PCDATA)
  <!ELEMENT compositeur (#PCDATA) ]>
<musique>
  <musicien>Charlie Parker</musicien>
  <compositeur>O. Coleman</compositeur>
  <musicien>Sonny Rollins</musicien>
</musique>
```

Écrire une DTD

- Déclaration d'entités (≅ raccourcis ou références)
 - exemple d'utilisation d'entités paramètres externes et d'inclusion conditionnelle

DTD

```
<!ENTITY % détails "INCLUDE">
<!ENTITY % zik "((nom, prénom) | (prénom, nom)),
naissance?, mort?">
<!ELEMENT musique (musicien | compositeur)*>
<![%détails;[
  <!ELEMENT musicien (%zik;, instrument+)>
  <!ELEMENT compositeur (%zik;, composition+)>
  <!ELEMENT instrument (#PCDATA)>
  <!ELEMENT composition (#PCDATA)>
  <!ELEMENT nom (#PCDATA)>
  <!ELEMENT prénom (#PCDATA)>
  <!ELEMENT naissance (#PCDATA)>
  <!ELEMENT mort (#PCDATA)>
]]>
```

entité paramètre externe

Document XML valide

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE musique SYSTEM "externe.dtd" [
  <!ENTITY % détails "IGNORE">
  <!ELEMENT musicien (#PCDATA)
  <!ELEMENT compositeur (#PCDATA) ]>
<musique>
  <musicien>Charlie Parker</musicien>
  <compositeur>O. Coleman</compositeur>
  <musicien>Sonny Rollins</musicien>
</musique>
```

Écrire une DTD

- Déclaration d'entités (≡ raccourcis ou références)
 - exemple d'utilisation d'entités paramètres externes et d'inclusion conditionnelle

DTD

```
<!ENTITY % détails "INCLUDE">
<!ENTITY % zik "((nom, prénom) | (prénom, nom)),
naissance?, mort?">
<!ELEMENT musique (musicien | compositeur)*>
<![%détails;[
  <!ELEMENT musicien (%zik;, instrument+)>
  <!ELEMENT compositeur (%zik;, composition+)>
  <!ELEMENT instrument (#PCDATA)>
  <!ELEMENT composition (#PCDATA)>
  <!ELEMENT nom (#PCDATA)>
  <!ELEMENT prénom (#PCDATA)>
  <!ELEMENT naissance (#PCDATA)>
  <!ELEMENT mort (#PCDATA)>
]]>
```

entité paramètre externe

la déclaration dans la DTD interne est prioritaire : le morceau de DTD externe est donc ignoré

Document XML valide

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE musique SYSTEM "externe.dtd" [
  <!ENTITY % détails "IGNORE">
  <!ELEMENT musicien (#PCDATA)
  <!ELEMENT compositeur (#PCDATA) ]>
<musique>
  <musicien>Charlie Parker</musicien>
  <compositeur>O. Coleman</compositeur>
  <musicien>Sonny Rollins</musicien>
</musique>
```

Écrire une DTD

- Déclaration d'entités (≅ raccourcis ou références)
 - exemple d'utilisation d'entités paramètres externes et d'inclusion conditionnelle

DTD

```
<!ENTITY % détails "INCLUDE">
<!ENTITY % zik "( (nom, prénom) | (prénom, nom) ),
naissance?, mort?">
<!ELEMENT musique (musicien | compositeur)*>
<![%détails;|
  <!ELEMENT musicien (%zik;, instrument+)>
  <!ELEMENT compositeur (%zik;, composition+)>
  <!ELEMENT instrument (#PCDATA)>
  <!ELEMENT composition (#PCDATA)>
  <!ELEMENT nom (#PCDATA)>
  <!ELEMENT prénom (#PCDATA)>
  <!ELEMENT naissance (#PCDATA)>
  <!ELEMENT mort (#PCDATA)>
]]>
```

entité paramètre externe

et sa référence permettant de créer une inclusion conditionnelle

dont la valeur par défaut est : INCLUDE

Document XML valide

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE musique SYSTEM "externe.dtd" [
  <!ENTITY % détails "IGNORE">
  <!ELEMENT musicien (#PCDATA)
  <!ELEMENT compositeur (#PCDATA) ]>
<musique>
  <musicien>Charlie Parker</musicien>
  <compositeur>O. Coleman</compositeur>
  <musicien>Sonny Rollins</musicien>
</musique>
```