

## TD 2 : DTD et espaces des noms

### Ex1. Espace des noms

Objectif: Cette section a pour objectif de vous familiariser avec la portée des espaces des noms dans un document XML

Q : Pour chaque portion de code, associer chaque élément et attribut à son espace des noms:

- ```

<lower:aaa xmlns:lower = "http://zvon.org/lowercase" >
  <lower:bbb xmlns:lower = "http://zvon.org/lowercase" >
    <lower:ccc xmlns:lower = "http://zvon.org/lowercase" />
  </lower:bbb>
  <upper:BBB xmlns:upper = "http://zvon.org/uppercase" >
    <upper:CCC xmlns:upper = "http://zvon.org/uppercase" />
  </upper:BBB>
  <xnumber:x111 xmlns:xnumber = "http://zvon.org/xnumber" >
    <xnumber:x222 xmlns:xnumber = "http://zvon.org/xnumber" />
  </xnumber:x111>
</lower:aaa>

```

Correction: Pour les déclaration d'espace de noms, des attributs réservés commençant par "xmlns" sont utilisés. Vous pouvez déclarer un espace de nom pour chaque élément utilisé, mais cette approche est fastidieuse et le code résultant est difficile à lire.

- ```

<lower:aaa xmlns:lower = "http://zvon.org/lowercase" >
  <lower:bbb >
    <lower:ccc />
  </lower:bbb>
  <upper:BBB xmlns:upper = "http://zvon.org/uppercase" >
    <upper:CCC />
  </upper:BBB>
  <xnumber:x111 xmlns:xnumber = "http://zvon.org/xnumber" >
    <xnumber:x222 />
  </xnumber:x111>
</lower:aaa>

```

Correction: Déclarer les espaces de noms comme dans l'exemple précédent serait peu pratique et source d'erreurs. Le standard prévoit plusieurs moyens d'accomplir cette tâche. La déclaration d'espace de noms pour un élément est aussi valide pour tous les éléments inclus dans celui-ci (enfants et descendants).

- ```

<lower:aaa   xmlns:lower   =   "http://zvon.org/lowercase"   xmlns:upper   =
"http://zvon.org/uppercase" xmlns:xnumber = "http://zvon.org/xnumber" >
  <lower:bbb >
    <lower:ccc />
  </lower:bbb>
  <upper:BBB >
    <upper:CCC />
  </upper:BBB>
  <xnumber:x111 >
    <xnumber:x222 />
  </xnumber:x111>
</lower:aaa>

```

Correction: Il est convenu de déclarer un espace de noms dans l'élément racine.

- ```

<lower:aaa xmlns:lower = "http://zvon.org/lowercase" xmlns:upper =
"http://zvon.org/lowercase" xmlns:xnumber = "http://zvon.org/lowercase" >
  <lower:bbb >
    <lower:ccc />
  </lower:bbb>
  <upper:BBB >
    <upper:CCC />
  </upper:BBB>
  <xnumber:x111 >
    <xnumber:x222 />
  </xnumber:x111>
</lower:aaa>

```

Correction: Ce n'est pas le préfixe qui identifie l'espace de noms mais la valeur de l'attribut xmlns. Dans cet exemple, tous les éléments appartiennent au même espace de noms bien qu'ils aient des préfixes différents.

- ```

<aaa >
  <lower:bbb xmlns:lower = "http://zvon.org/lowercase" >
    <lower:ccc />
  </lower:bbb>
  <lower:BBB xmlns:lower = "http://zvon.org/uppercase" >
    <lower:CCC />
  </lower:BBB>
  <lower:x111 xmlns:lower = "http://zvon.org/xnumber" >
    <lower:x222 />
  </lower:x111>
</aaa>

```

Correction: Alors que dans l'exemple précédent Exemple 5 , les éléments appartenait au même espace de noms avec des préfixes différents, dans le cas présent, ils ont chacun un espace de noms particulier bien qu'ils aient tous le même préfixe.

- ```

<aaa >
  <bbb xmlns = "http://zvon.org/lowercase" >
    <ccc />
  </bbb>
  <BBB xmlns = "http://zvon.org/uppercase" >
    <CCC />
  </BBB>
  <x111 xmlns = "http://zvon.org/xnumber" >
    <x222 />
  </x111>
</aaa>

```

Correction: Les espaces de noms n'ont pas besoin d'être déclarés explicitement avec des préfixes. L'attribut xmlns définit l'espace de noms par défaut d'un élément où il intervient et pour tous ses enfants et descendants.

- ```

<aaa xmlns:upper = "http://zvon.org/uppercase" xmlns:xnumber = "http://zvon.org/xnumber"
>
  <bbb xmlns = "http://zvon.org/lowercase" >
    <ccc />
    <upper:WWW />
    <xnumber:x666 />
  </bbb>
  <BBB xmlns = "http://zvon.org/uppercase" >
    <upper:WWW />

```

```

<xnumber:x666 />
<CCC />
</BBB>
<x111 xmlns = "http://zvon.org/xnumber" >
  <x222 />
  <upper:WWW />
  <xnumber:x666 />
</x111>
</aaa>

```

Correction: Même si un espace de noms par défaut est utilisé, des espace de noms pour certains éléments peuvent être explicitement déclarés.

- ```

<aaa xmlns = "http://zvon.org/lowercase" >
  <bbb >
    <ccc xmlns = "" >
      <ddd />
    </ccc>
  </bbb>
</aaa>

```

Correction: Si la valeur de l'espace de noms est une chaîne vide la déclaration d'espace de noms par défaut est annulée.

- ```

<lower:aaa xmlns:lower = "http://zvon.org/lowercase" xmlns:upper =
"http://zvon.org/uppercase" xmlns:xnumber = "http://zvon.org/xnumber" >
  <lower:bbb lower:zz = "11" >
    <lower:ccc upper:WW = "22" />
  </lower:bbb>
  <upper:BBB lower:sss = "****" xnumber:S111 = "???" />
  <xnumber:x111 />
</lower:aaa>

```

Correction: Des attributs peuvent être explicitement assignés à un espace de noms donné.

- ```

<lower:aaa xmlns:lower = "http://zvon.org/lowercase" xmlns:upper =
"http://zvon.org/uppercase" xmlns:xnumber = "http://zvon.org/xnumber" >
  <lower:bbb zz = "11" >
    <lower:ccc WW = "22" />
  </lower:bbb>
  <upper:BBB sss = "****" xnumber:S111 = "???" />
  <xnumber:x111 />
</lower:aaa>

```

Correction: Les attributs sans préfixe n'appartiennent à aucun espace de noms.

- ```

<aaa xmlns = "http://zvon.org/lowercase" xmlns:upper = "http://zvon.org/uppercase"
xmlns:xnumber = "http://zvon.org/xnumber" >
  <bbb zz = "11" >
    <ccc WW = "22" xmlns = "http://zvon.org/uppercase" />
  </bbb>
  <upper:BBB sss = "****" xnumber:S111 = "???" />
  <xnumber:x111 />
</aaa>

```

Correction: Les attributs non préfixés n'appartiennent à aucun espace de noms même si un espace de noms par défaut est défini pour les éléments auxquels ils sont assignés.

- ```

<aaa xmlns:lower = "http://zvon.org/lowercase" >
  <lower:BBB xmlns:lower = "http://zvon.org/uppercase" >
    <lower:x111 />
    <sss xmlns:lower = "http://zvon.org/xnumber" >
      <lower:x111 />

```

```
</sss>  
</lower:BBB>  
<lower:x111 />  
</aaa>
```

Correction: Une déclaration d'espace de noms est restreinte à la portée de l'élément dans lequel elle est faite.

## Ex2.Retour sur les DTDs

Objectif: Petits Rappels

Q : On souhaite stocker dans un fichier XML un petit Forum de discussion. Ce fichier XML contient des utilisateurs et un ensemble de messages. Chaque utilisateur est décrit par un nom, un prénom et une adresse e-mail. Chaque message est caractérisé par un expéditeur et un texte associé. Un message peut être lié à des messages antérieurs s'il en constitue une réponse. On vous demande d'écrire une DTD et un fichier XML permettant de stocker l'information pour reproduire un fil de discussion sans avoir à stocker d'information redondante. Vous vous poserez les questions suivantes:

- Quel moyen utiliser pour lier une réponse au message d'origine?

Correction: ID et IDREFS

- Quel moyen utiliser pour lier auteurs et messages?

Correction: ID et IDREF

- Quelles informations mettre sous forme d'attributs, quelles informations mettre sous forme d'éléments?

Correction: Le fichier forum.xml:

```
<?xml version="1.0" ?>  
<!DOCTYPE forum SYSTEM "forum.dtd" >  
<forum>  
<personnes>  
<personne ref="p1">  
<nom>Thomas</nom>  
<prenom>vincent</prenom>  
<adresse>vincent.thomas@loria.fr</adresse>  
</personne>  
<personne ref="p2">  
<nom>Lhoste</nom>  
<prenom>Regis</prenom>  
<adresse>lhoste@loria.fr</adresse>  
</personne>  
</personnes>  
<messages>  
<message ref="m1">  
<expediteur ref="p1" />  
<texte>Bonjour</texte>  
</message>  
<message ref="m2">  
<expediteur ref="p2" />  
<texte>Hello</texte>  
</message>  
<message ref="m3">  
<expediteur ref="p2" />  
<texte>Bonjour, Ca va bien ?</texte>  
<anterieur ref="m1" />  
</message>  
<message ref="m4">
```

```

<expediteur ref="p1" />
<texte>Ca va !!</texte>
<anterieur ref="m3" />
</message>
</messages>
</forum>
Et le fichier forum.dtd
<!ELEMENT forum ( personnes, messages) >
<!ELEMENT personnes ( personne*) >
<!ELEMENT messages ( message*) >
<!ELEMENT personne ( nom, prenom, adresse) >
<!ATTLIST personne ref ID #REQUIRED>
<!ELEMENT nom (#PCDATA) >
<!ELEMENT prenom (#PCDATA) >
<!ELEMENT adresse (#PCDATA) >
<!ELEMENT message (expediteur, texte, anterieur?) >
<!ATTLIST message ref ID #REQUIRED>
<!ELEMENT expediteur EMPTY>
<!ATTLIST expediteur ref IDREF #REQUIRED >
<!ELEMENT texte (#PCDATA)>
<!ELEMENT anterieur EMPTY>
<!ATTLIST anterieur ref IDREF #REQUIRED >

```

Q : Lier des DTDs à un fichier XML. Précision sur le cours:

- En général, il y a une seule déclaration de DTD par document. Pour utiliser plusieurs DTD, faire des appels d'entités ou au stratagème suivant:

```

<!DOCTYPE acme PUBLIC "-//ACME//MyDTD//EN"
"http://www.acme.com/entities/MyDTD.dtd">
<!DOCTYPE acme SYSTEM "http://www.acme.com/entities/MyDTD.dtd">

```

Les outils tentent de résoudre les identificateurs PUBLIC. S'ils n'y arrivent pas, ils accèdent au fichier SYSTEM.

- Les DTD peuvent être : externes, internes ou les 2 : dans ce cas, le sous-ensemble interne est lu en premier.

Que se passe-t-il ici:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE zoo SYSTEM "zoo.dtd" [
  <!ENTITY % notations SYSTEM "notations.ent">
  %notations;
  <!ENTITY % svg_structure SYSTEM "svg.dtd" >
  %svg_structure;
  <!ENTITY flipper.jpg SYSTEM "flipper.jpg" NDATA jpg>
  <!ENTITY oum SYSTEM "oum.xml"> ]>
<zoo>
  <aquarium>
    <dauphin id="jhgtr13" photo="flipper.jpg">
      <nom>Flipper</nom>
  ...

```

Sachant que le fichier "notation.ent" est le suivant:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!NOTATION jpg SYSTEM "images/jpeg">
<!NOTATION gif SYSTEM "images/gif">

```

Q : Rédiger la définition (DTD) d'un document de type « Bon de commande » en utilisant le texte ci-dessous :

Un bon de commande est émis par une organisation identifiée par un nom, une adresse, un numéro de téléphone et éventuellement un numéro de télécopie. Il s'adresse à une autre organisation définie de la même manière. Il comporte une date, un montant total ainsi que une ou plusieurs lignes de commande. Chaque ligne comporte une référence de produit, un intitulé, une quantité, un prix unitaire.

Correction: <?xml version="1.0" encoding="ISO-8859-1" ?>

<!ELEMENT bon\_commande (emetteur, receveur, date, montant, commande+)>

<!ELEMENT emetteur (organisation)>

<!ELEMENT receveur (organisation)>

<!ELEMENT organisation (nom, adresse, telephone, fax ?)>

<!ELEMENT nom (#PCDATA)>

<!ELEMENT adresse (#PCDATA)>

<!ELEMENT telephone (#PCDATA)>

<!ELEMENT fax (#PCDATA)>

<!ELEMENT date (#PCDATA)>

<!ELEMENT montant (#PCDATA)>

<!ELEMENT commande (intitule, quantite, prix\_unitaire)>

<!ATTLIST commande ref\_produit ID #REQUIRED>

<!ELEMENT ref\_produit (#PCDATA)>

<!ELEMENT intitule (#PCDATA)>

<!ELEMENT quantite (#PCDATA)>

<!ELEMENT prix\_unitaire (#PCDATA)>

ID correspond à des identifiants. Il s'agit d'un attribut qui désigne de manière unique un élément. Cette valeur peut être référencée par un attribut de type IDREF (qui vérifie alors si l'ID est bien présente dans le document) (cf exercice 7) Un problème de référence vient d'un fichier bien formé mais non valide