G. Hunault
Licence 2 Informatique

Angers, mars 2019

Technologie XML

1. Ecriture de DTD externe

On s'intéresse à des ventes d'articles à des clients suite à des commandes. On trouvera ci-dessous un exemple de fichier correspondant, nommé ventes.xml.

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<ventes>
   <commandes>
      <commande refclient="K105">
         <article ref="TA55"   quantité="5" />
         <article ref="VB7855" quantité="1" />
      </commande>
      <commande refclient="T843">
         <article ref="TA55"   quantité="10" />
         <article ref="IB3"    quantité="2" />
      </commande>
   </commandes>
   <clients>
      <client id="K105">
        <nom>DURAND</nom>
      </client>
      <client id="T843">
        <nom>DUPONT</nom>
      </client>
   </clients>
</ventes>
```

Donner pour ce fichier une grammaire DTD « *minimale et raisonnable* ».

2. Un peu de XPATH *via* xmllint

Donner les instructions de xmllint exécutées en mode *shell* qui permettent de répondre aux questions suivantes que l'on se pose sur un fichier de ventes structuré comme dans la section précédente.

Attention : il ne s'agit évidemment pas du fichier précédent mais d'un fichier structuré de la même façon. On ne doit donc pas fournir les réponses aux questions mais donner le code xmllint qui fournirait les réponses.

- quel est le nom du client dont l'attribut *id* est "BF12" ?

- combien y a-t-il de commandes dans le fichier ?

- combien d'articles différents ont été commandés dans la cinquième commande ?

- combien y a-t-il de commandes pour la référence client "BF12" ?

- combien y a-t-il de commandes avec plus de trois articles ?

- combien y a-t-il de commandes pour le client "ZOLA" ?

3. Articles

Donner le contenu XML d'un fichier articles.xml qui suit le schéma articles.xsd fourni en annexe avec les informations suivantes :

- l'article dont l'id est "TA55" a pour description "montre sport", son prix est de "45 euros", sa référence fournisseur est "FK867" ;

- l'article dont l'id est "IB3" a pour description "chaussures en cuir", son prix est de "81 euros", sa référence fournisseur est "GTA012".

4. Ventes et articles

On voudrait disposer d'un fichier va.xml constitué dynamiquement du fichier des ventes ventes.xml de la question 1 et du fichier des articles articles.xml de la question 3 à l'aide d'une DTD interne et de deux entités caractères. Donner le texte complet de ce fichier va.xml. On nommera <va> l'élément racine de ce fichier.

5. Transformations XSL (1)

On voudrait produire un fichier XML qui résume les commandes et les clients. Donner le contenu d'un fichier XSL qui effectue une transformation du fichier ventes.xml de la question 1 et qui produit le fichier texte suivant.

On essaiera de respecter le formatage des lignes, mais ce n'est pas grave s'il manque des espaces ou s'il y en a en trop.

```
Dans le fichier des ventes, il y a :

 - 2 commandes ;
 - 4 articles en commande ;
 - 2 clients.
```

Si vous n'arrivez pas à tout reproduire, ce n'est pas grave. Essayez de produire ce que vous pouvez. Il vaut mieux fournir un code XSL plus court mais correct et fonctionnel qu'un code long mais incorrect.

6. Transformations XSL (2)

Donner le contenu d'un fichier XSL qui effectue une transformation du fichier ventes.xml de la question 1 et qui produit le fichier XHTML suivant. On pourra utiliser les *templates* et autres sous-programmes de stdWeb2.xsl vus en cours et en TP.

```
<!DOCTYPE html ...
[...]
<head><title>Clients pour les ventes</title></head>
<body class="beige_jpg">
<blockquote>
<h1>Tableau trié des clients</h1>
  <table>
    <tr><th>Id</th><th>Nom</th></tr>
    <tr><td>T843</td><td>DUPONT</td></tr>
    <tr><td>K105</td><td>DURAND</td></tr>
</table>
</blockquote>
</body>
</html>
```

7. Discussion

Essayez de répondre à la question suivante :

> *Peut-on s'attendre à ce que dans les prochaines années les navigateurs implémentent les transformations XSL version 2.0 ?*

Votre réponse devra mettre en évidence votre culture naissante ainsi que votre recul et votre esprit de synthèse en matière de modélisation et de traitement d'informations structurées.

Votre réponse devra faire 10 lignes au minimum, sans limite de maximum. On utilisera au moins 3 mots de 4 syllabes ou plus pour transmettre « un contenu rédactionnel fort ».

ANNEXE

Schéma **articles.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="articles">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="article" />
      </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="article">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="description" />
        <xs:element ref="prix"/>
        <xs:element ref="fournisseur" />
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:ID"/>
    </xs:complexType>
</xs:element>

<xs:element name="description" type="xs:string"/>

<xs:element name="prix" type="xs:string"/>

<xs:element name="fournisseur">
    <xs:complexType>
      <xs:attribute name="ref" use="required" type="xs:ID"/>
    </xs:complexType>
</xs:element>

</xs:schema>
```

ESQUISSE DE SOLUTION

1. Ecriture de DTD externe

```
<?xml encoding="UTF-8"?>

<!ELEMENT ventes    (commandes,clients) >

<!ELEMENT commandes (commande)+         >
<!ELEMENT clients   (client)+           >
<!ELEMENT commande  (article)+          >

<!ELEMENT client    (nom)               >
<!ELEMENT article   EMPTY               >
<!ELEMENT nom       (#PCDATA)           >

<!ATTLIST commande refclient NMTOKEN #REQUIRED>
<!ATTLIST client   id        NMTOKEN #REQUIRED>
<!ATTLIST article  quantité CDATA   #REQUIRED>
<!ATTLIST article  ref       NMTOKEN #REQUIRED>
```

2. Un peu de XPATH *via* xmllint

```
ls //client[@id="BF12"]/nom

xpath count(//commande)

xpath count(//commande[5]/article)

xpath count(//commande[@refclient="BF12"])

xpath count(//article[4])

xpath count(//commande[@refclient=//client[nom="ZOLA"]/@id])
```

3. Articles

```
<articles>

    <article id="TA55">
       <description>montre sport</description>
       <prix>45 euros</prix>
       <fournisseur ref="FK867" />
    </article>

    <article id="IB3">
       <description>chaussures en cuir</description>
       <prix>81 euros</prix>
       <fournisseur ref="GTA012" />
    </article>

</articles>
```

4. Ventes et articles

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

<!DOCTYPE va [
  <!ENTITY ventes   SYSTEM "ventes.xml"   >
  <!ENTITY articles SYSTEM "articles.xml" >
]>

<va>
   &ventes;
   &articles;
</va>
```

5. Transformations XSL (1)

```
<xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform' version="1.0">
<xsl:output method="text" encoding="UTF-8" />

<xsl:template match="/">Dans le fichier des ventes, il y a :<xsl:text>&#10;</xsl:text>
 - <xsl:value-of select="count(//commande)" /> commandes ;
 - <xsl:value-of select="count(//article)" /> articles en commande ;
 - <xsl:value-of select="count(//client)" /> clients.
</xsl:template>

</xsl:stylesheet>
```

6. Transformations XSL (2)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform' version="1.0">
<xsl:import href="stdWeb2.xsl" />
<xsl:output method="xml" encoding="UTF-8" />

<xsl:template match="/">
     <xsl:call-template name="debutPage">
          <xsl:with-param name="leTitre">Clients pour les ventes</xsl:with-param>
          <xsl:with-param name="h1redite">no</xsl:with-param>
     </xsl:call-template>

     <h1>Tableau trié des clients</h1><xsl:text>&#10;</xsl:text>
     <table><xsl:call-template name="sdl" />
       <tr><th>Id</th><th>Nom</th></tr><xsl:call-template name="sdl" />
        <xsl:apply-templates select="//client">
          <xsl:sort select="nom" />
        </xsl:apply-templates>
     </table>

     <xsl:call-template name="sdl" />
     <xsl:call-template name="finPage" />
</xsl:template>

<xsl:template match="client">
 <tr>
    <td><xsl:value-of select="@id" /></td>
    <td><xsl:value-of select="nom" /></td>
 </tr>
</xsl:template>

</xsl:stylesheet>
```