M1–Master d'informatique – 2010/2011

BDA

mercredi 25 mai 2011

durée 3h — support de cours autorisé

Examen – session 1

Exercice 1 : On considère le fichier `XML` de nom `biblio.xml` suivant :

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

<!DOCTYPE biblio SYSTEM "biblio.dtd">
<biblio>
    <livre>
        <titre>Les Miserables</titre>
        <auteur>Victor Hugo</auteur>
        <nb_tomes>3</nb_tomes>
    </livre>
    <livre>
        <titre>L'Assomoir</titre>
        <auteur>Emile Zola</auteur>
    </livre>
    <livre lang="en">
        <titre>David Copperfield</titre>
        <auteur>Charles Dickens</auteur>
        <nb_tomes>3</nb_tomes>
    </livre>
</biblio>
```

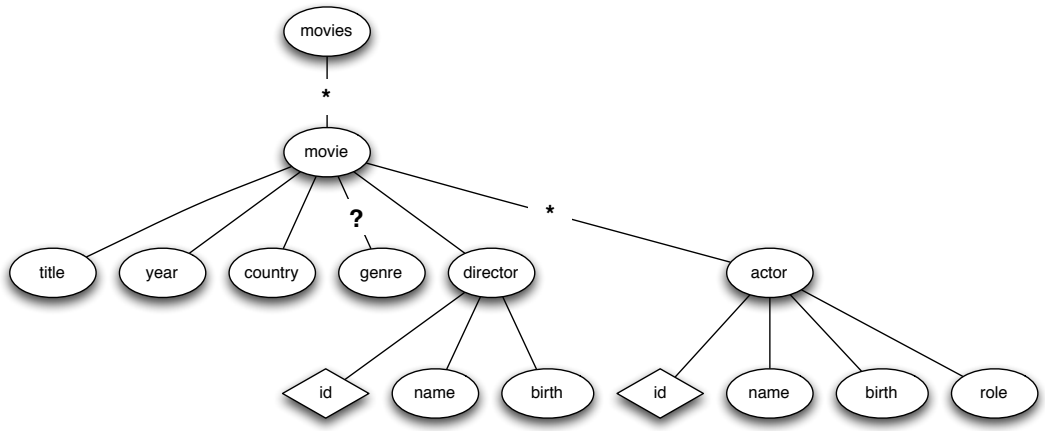
Question 1 : Définissez une `DTD` pour ce document. On précise qu'un élément `livre` doit être composé des trois éléments dans l'ordre : `titre`, `auteur` et `nb_tomes`, ce dernier élément étant optionnel. L'attribut `lang` de l'élément `livre` ne prend que les valeurs `en` ou `fr`, cette dernière valeur étant la valeur par défaut.

Question 2 : En supposant que votre `DTD` est définie dans un fichier de nom `biblio.dtd` situé dans le même répertoire que le fichier `biblio.xml`, donnez la ligne à ajouter dans ce fichier `biblio.xml` pour permettre son association avec la `DTD`.

Exercice 2 : On considère des fichiers xml de descriptions de films dont le schéma [1] est donné sous forme graphique où :

- un losange représente un attribut,
- un arc simple veut dire 1 sous-élément,
- un arc avec un point d'interrogation veut dire que le sous-élément est optionnel
- un arc avec une étoile veut dire 0,1 ou plusieurs occurrences du sous-élément,

1. Oui, c'est le même que pour le DS.



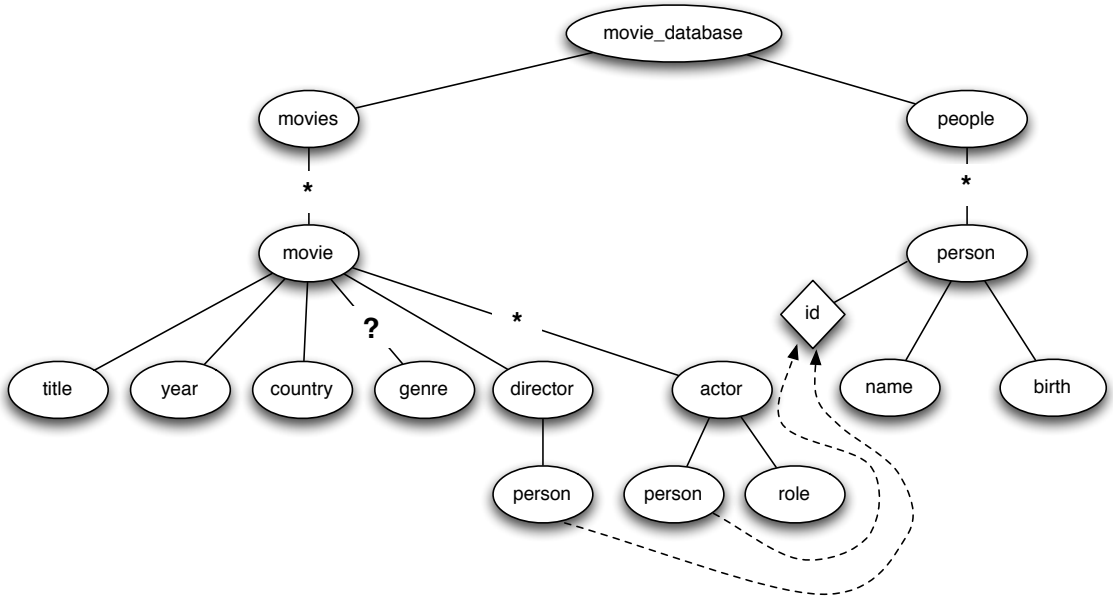
Un concepteur XML a qui on a demandé de définir le schéma `XML-Schema` à partir de cette représentation graphique propose :

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="movies">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="movie">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="title" type="xs:string"/>
              <xs:element name="year" type="xs:integer"/>
              <xs:element name="country" type="xs:NCName"/>
              <xs:element name="genre" type="xs:NCName"/>
              <xs:element name="director">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="name" type="xs:string"/>
                    <xs:element name="birth" type="xs:integer"/>
                  </xs:sequence>
                  <xs:attribute name="id"
                  use="required" type="xs:integer"/>
                </xs:complexType>
              </xs:element>
              <xs:element maxOccurs="unbounded" name="actor">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="name" type="xs:string"/>
                    <xs:element name="birth" type="xs:integer"/>
                    <xs:element name="role" type="xs:string"/>
                  </xs:sequence>
                  <xs:attribute name="id"
                  use="required" type="xs:integer"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Question 1 : Ajoutez des types dans ce schéma afin de factoriser les définitions des éléments `director` et `actor`. Ne donnez que les lignes définissant ces types et les changements qui en découlent dans la définition de l'élément `actor`

Question 2 : Donnez les autres modifications à apporter au schéma pour que celui-ci corresponde à la représentation graphique donnée au début du sujet. N'écrire que les lignes à modifier.

Pour restructurer les fichiers décrivant les films avec une vision plus base de données, on considère le schéma suivant dans lequel les flèches en pointillé indiquent que le contenu des éléments `person` sous les éléments `director` et `actor` doit correspondre à une valeur d'attribut `id` :



Question 3 : Ce schéma peut-il être décrit par une DTD ? Si la réponse est oui, écrivez cette DTD, sinon expliquez pourquoi.

Question 4 : Sans tenir compte des contraintes de clés, écrivez les portions de `XML-Schema` donnant les définitions des éléments `director`, `actor` et `people` [2].

Question 5 : Donnez les portions de schéma XML permettant les définitions de clés primaires et étrangères pour assurer qu'un élément `person` sous un élément `director` ou `actor` corresponde bien à un élément `person` sous l'élément `people`. Vous préciserez dans quelle définition d'élément apparaît chacune des définitions de clé.

2. Il faudra donc définir aussi les éléments `person`.

Exercice 3 : On considère le fichier `XML`, de nom `liste.xml`, ci-dessous :

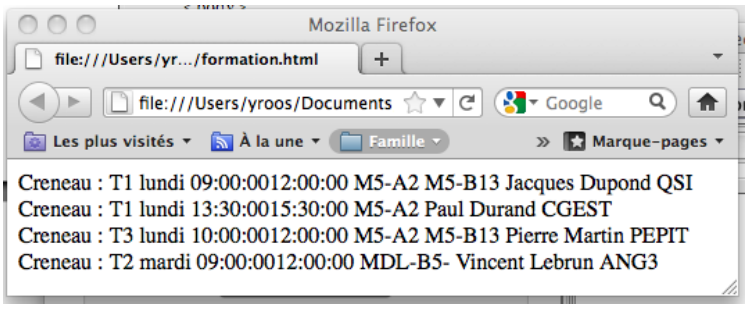
```
<?xml version="1.0" encoding="UTF-8"?>
<liste>
    <livre>
        <titre genre="jeu">Le Texas Hold'Em Poker online</titre>
        <auteur>Mark Stohan</auteur>
        <auteur>Robert Bluman</auteur>
        <parution>2006</parution>
    </livre>
    <livre>
        <titre genre="jeu">Sudoku Manga</titre>
        <auteur>Sudoku factory</auteur>
        <parution>2007</parution>
    </livre>
    <livre>
        <titre genre="jeu">Kakoku</titre>
        <auteur>Hizi Kagochi</auteur>
        <parution>2005</parution>
    </livre>
    <livre>
        <titre genre="photo">Manuel de la photo</titre>
        <auteur>Jackie Contiboeuf</auteur>
        <auteur>Alain Mocney</auteur>
        <parution>2006</parution>
    </livre>
</liste>
```

On considère également la feuille `XSLT` suivante :

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

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" indent="yes"/>
  <xsl:template match="/">
    <html>
      <body>
        <xsl:apply-templates/>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="titre">
    Titre : <xsl:value-of select="."/>
    <br/>
  </xsl:template>
</xsl:stylesheet>
```

Question 1 : La visualisation du fichier `HTML` produit par cette feuille de style correspont elle à la capture d'écran ci-dessous ? Expliquer pourquoi.



On considère maintenant l'autre feuille de style suivante :

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

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" indent="yes"/>
  <xsl:template match="/liste">
    <html>
      <body>
        <table>
          <xsl:for-each select="livre">
            <tr>
              <td>
                <xsl:value-of select="titre"/>
              </td>
              <td>
                <xsl:value-of select="parution"/>
              </td>
              <xsl:for-each select="auteur">
                <td>
                  <xsl:value-of select="."/>
                </td>
              </xsl:for-each>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Question 2 : Expliquer[3] ce que produit cette feuille `XSLT` quand elle est appliquée sur le fichier `liste.xml`.

3. En français, pas de code !

Question 3 : Donner une feuille `XSLT` équivalente (c'est à dire qui produit la même transformation) sans aucune instruction `xsl:for-each`.

Exercice 4 : On considère le fichier XML ci-dessous qui représente le stock d'un maraîcher :

```
<?xml version="1.0" encoding="UTF-8"?>
<produits>
  <fruit type="clementine" prix="290" calibre="1">
    <producteur>Production Bastia</producteur>
    <origine region="Corse">France</origine>
    <qtity>15</qtity>
    <note>Sans pepins, avec feuilles</note>
    <bio/>
  </fruit>
  <legume type="courgette" prix="300" calibre="2">
    <producteur>Madrid Hortelano</producteur>
    <origine>Espagne</origine>
    <qtity>100</qtity>
    <bio/>
  </legume>
  <legume type="chou fleur" prix="090" calibre="2">
    <producteur>Pontivy et Cie</producteur>
    <origine region="Bretagne">France</origine>
    <qtity>100</qtity>
  </legume>
  <legume type="salade" prix="075" calibre="3">
    <producteur>Marius Production</producteur>
    <origine region="Provence">France</origine>
    <qtity>35</qtity>
    <note>Batavia</note>
  </legume>
  <fruit type="melon" prix="150" calibre="1">
    <producteur>Marius Production</producteur>
    <origine region="Provence">France</origine>
    <qtity>50</qtity>
    <note>Melon brode</note>
    <bio/>
  </fruit>
</produits>
```

Question 1 : Donnez des requêtes `XPath` pour sélectionner les éléments suivants (ces requêtes doivent bien sûr fonctionner sur tout document de même nature que celui de l'exemple.)

1. les producteurs de fruits.
2. les légumes produits en Espagne.
3. les origines des clémentines de calibre 1 issues de l'agriculture biologique.
4. les producteurs bretons.

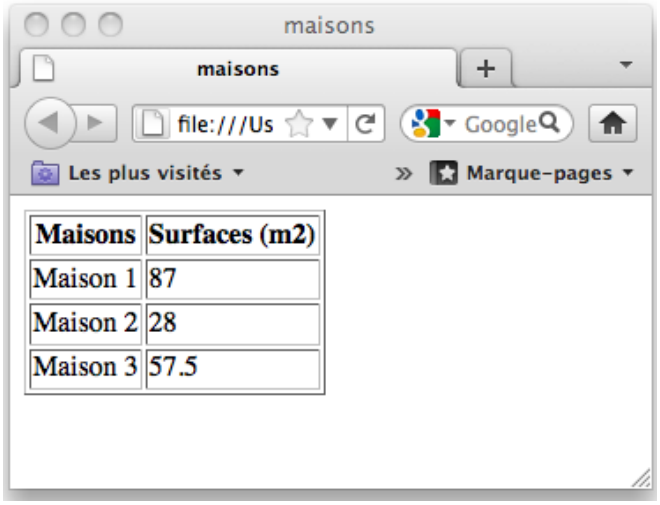
Question 2 : On souhaite réorganiser le fichier `XML` pour placer d'abord les légumes, puis les fruits. L'ensemble des articles est encore placé dans un élément `produits`. Proposez une requête `XQuery` ou une transformation `XSLT`, au choix, réalisant cette transformation.

Exercice 5 :

Voici de nouveau le fichier de description de logements que vous connaissez-bien :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<maisons>
  <maison id="1">
    <RDC>
      <cuisine surface-m2="12">Evier Inox. Mobilier encastre</cuisine>
      <WC>Lavabo.</WC>
      <sejour>Cheminee en pierre. Baie vitree</sejour>
      <bureau surface-m2="14">Bibliotheque</bureau>
      <garage/>
    </RDC>
    <etage>
      <terrasse/>
      <chambre surface-m2="28" fenetre="3">
        <alcove surface-m2="8"/>
      </chambre>
      <chambre surface-m2="18"/>
      <salledeBain surface-m2="15">
        Douche, baignoire, lavabo
      </salledeBain>
    </etage>
  </maison>
  <maison id="2">
    <RDC>
      <cuisine surface-m2="12">en ruine</cuisine>
      <garage/>
    </RDC>
    <etage>
      <mirador surface-m2="1">
        Vue sur la mer
      </mirador>
      <salledeBain surface-m2="15">Douche</salledeBain>
    </etage>
  </maison>
  <maison id="3">
    <RDC>
      <sejour surface-m2="40"/>
    </RDC>
    <etage>
      <chambre surface-m2="17.5">Exposition plein sud</chambre>
    </etage>
  </maison>
</maisons>
```

Question 1 : Écrire un programme `XQuery` qui, à partir de ce fichier `XML`, calcule, pour chaque maison, sa superficie totale. La sortie du programme sera un fichier `HTML` dont la visualisation correspond à la capture écran suivante :


maisons
maisons
file:///Us
Google
Les plus visités
Marque-pages
Maisons
Surfaces (m2)
Maison 1
87
Maison 2
28
Maison 3
57.5