

PXML

vendredi 10 mars 2011

durée 2h

support de cours autorisé

Interrogation écrite

Exercice 1 : Lecture sax d'un fichier xml

On considère le gestionnaire `sax` défini par la classe `InterroHandler` suivante :

```
1 import org.xml.sax.* ;
2 import org.xml.sax.helpers.* ;
3
4 public class InterroHandler extends DefaultHandler {
5     private int n ;
6
7     public void startDocument() {this.n = 0 ;}
8
9     public void endDocument() {System.out.println() ;}
10
11     public void startElement(String namespaceURI,
12                             String localName,
13                             String rawName,
14                             Attributes attributes) {
15         System.out.print(localName + "-" + this.n + "(") ;
16         this.n++ ;
17     }
18
19     public void endElement( String namespaceURI,
20                           String localName,
21                           String rawName) {
22         System.out.print(") ") ;
23         this.n++ ;
24     }
25 }
```

On considère aussi le fichier `a.xml` suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<a> <b/> <c/> <d> <c/> </d> </a>
```

Question 1 : (1,5 pt) Donner l'affichage obtenu en analysant le fichier `a.xml` à l'aide d'un gestionnaire `InterroHandler`.

```
a-0(b-1() c-3() d-5(c-6() ) )
```

Exercice 2 : DTD

On souhaite stocker dans des fichiers **xml** des archives de discussions entre plusieurs utilisateurs sur un forum. Un tel fichier doit contenir les informations suivantes :

- identifiant du message,
- référence à un autre message, dans le cas où c'est une réponse à un message précédent,
- titre
- identification du contributeur :
 - pseudo (identificateur),
 - avatar, c'est à dire une image le *personnalisant*, donnée par une URL,
 - localisation (éventuellement), donnée par un numéro de département
 - statut (*modérateur*, *habitué*, *régulier*, *nouveau*),
- contenu (sans ordre) :
 - un ou plusieurs éléments de texte,
 - zéro, un ou plusieurs smileys (identifiés par l'URL de l'image),
 - zéro, une ou plusieurs copies de code,
 - zéro, une ou plusieurs citations.

Question 1 : (2 pts) Écrire une DTD pour les documents d'archives de forum.

```
<!ELEMENT messages (message)*>

<!ELEMENT message (titre , contributeur>
  <!ATTLIST message id IDREF #REQUIRED ref IDREF #IMPLIED>

  <!ELEMENT contributeur (pseudo , avatar , (localisation)?)>
  <!ATTLIST contributeur statut (moderateur|habitué|régulier|nouveau →
    ↵ ) #REQUIRED>

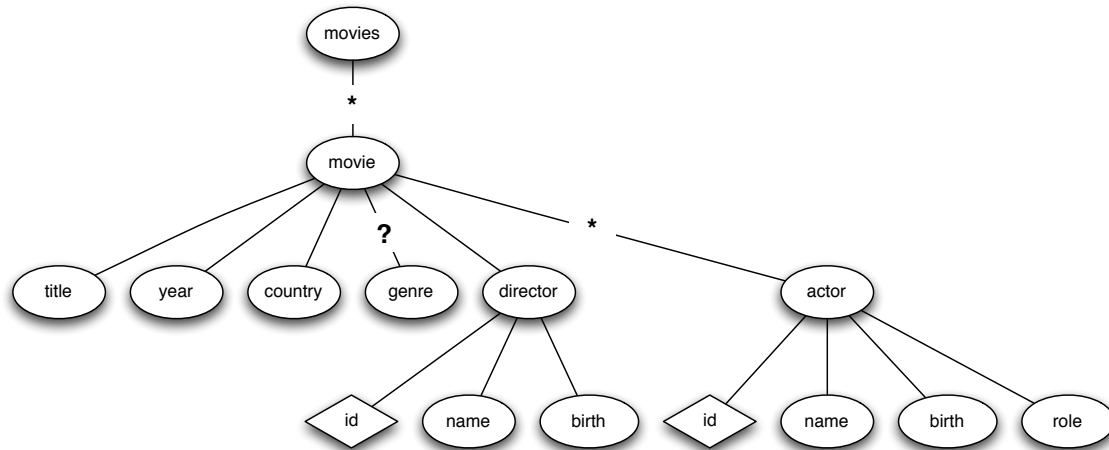
  <!ELEMENT contenu (#PCDATA|smiley|code|citation)*>

<!ELEMENT titre (#PCDATA)>
<!ELEMENT pseudo (#PCDATA)>
<!ELEMENT avatar (#PCDATA)>
<!ELEMENT localisation (#PCDATA)>
<!ELEMENT smiley (#PCDATA)>
<!ELEMENT code (#PCDATA)>
<!ELEMENT citation (#PCDATA)>
```

Exercice 3 : XPath

On considère des fichiers `xml` de descriptions de films dont le schéma est donné sous forme graphique où :

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



Voici un exemple d'instance valide pour ce schéma :

```
<?xml version="1.0" encoding="UTF-8"?>
<movies xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="movies.xsd">
  <movie>
    <title>Match Point</title>
    <year>2005</year>
    <country>USA</country>
    <genre>Crime</genre>
    <director id="14">
      <name>Woody Allen</name>
      <birth>1935</birth>
    </director>
    <actor id="15">
      <name>Jonathan Rhys Meyers</name>
      <birth>1977</birth>
      <role>Chris Wilton</role>
    </actor>
    <actor id="16">
      <name>Scarlett Johansson</name>
      <birth>1984</birth>
      <role>Nola Rice</role>
    </actor>
  </movie>
</movies>
```

Question 1 : (0,5 pt) D'après la balise ouvrante de l'élément `movies` de l'instance `xml` ci-dessus, le schéma est donné sous la forme d'un XML-Schema décrit dans un fichier de nom `movies.xsd`. Expliquer pourquoi ce schéma pourrait être aussi décrit par une DTD.

Tous les éléments de même nom sont du même type.

Question 2 : (1,5 pt) Que retourne la requête `count(//actor[@id="15"]/name/text())` si celle-ci est évaluée sur l'instance `xml` ci-dessus ?

Même question avec la requête `count(//actor[//@id="15"]/name/text())`

la première requête retourne 0 car il n'y a pas d'attribut `id` à la racine, la seconde retourne 2 (le nombre total d'acteurs) car il y a bien dans le document un attribut `id` de valeur 15.

Question 3 : (1 pt par requête) Écrire les requêtes XPath permettant d'extraire les informations suivantes :

1. Les titres des films dans lesquels joue **Scarlett Johansson**.

```
//movie[actor/name="Scarlett Johansson"]/title/text()
```

2. Les rôles joués par **Michel Blanc**.

```
//actor[name="Michel Blanc"]/role/text()
```

3. Le nom du premier acteur dans la distribution de chaque film.

```
//movie/actor[1]/name/text()
```

4. Les noms des réalisateurs des films dans lesquels jouent à la fois **Michel Blanc** et **Thierry Lhermite**.

```
//movie[actor/name="Michel Blanc" and actor/name="Thierry Lhermite"]/director/name/text()
```

5. Les titres des films dans lesquels le réalisateur joue aussi un rôle comme acteur.

```
//movie[director/name=actor/name]/title/text()
```

6. Tous les noms d'acteurs en ne faisant figurer chaque nom qu'une seule fois.

```
//actor[not(preceding::actor/name=name)]/name/text()
```

7. Le nombre total d'acteurs du fichier interrogé.

```
count(//actor[not(preceding::actor/name=name)]/name)
```

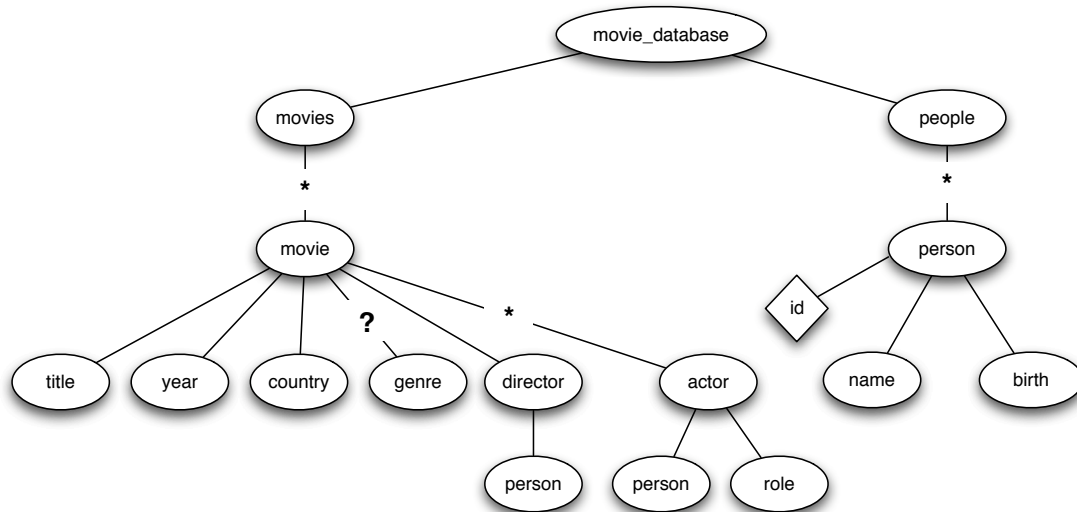
8. Les noms des réalisateurs qui n'ont fait qu'un seul film.

```
//director[not(preceding::director/name=name) and not(following::director/name=name)]/name/text()
```

Question 4 : (0 pt) On rappelle qu'on peut faire l'union des ensembles résultats de plusieurs expressions XPath à l'aide de l'opérateur `|`. Utiliser cet opérateur d'union pour construire une expression XPath pouvant servir à tester si un nœud (contexte) est un attribut ou pas.

```
count(.|../@*) = count(../@*)
```

Pour restructurer les fichiers décrivant les films avec une vision plus base de données, on considère le schéma suivant :



Question 5 : (0,5 pt) Ce schéma peut-il être décrit par une DTD ? Si la réponse est oui, écrire cette DTD, sinon expliquer pourquoi.

Non car plusieurs éléments **person** ne sont pas du même type.

Exercice 4 : XML-Schema

On considère le fichier `xml` suivant :

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

<compagnie xmlns="http://exo-schema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://exo-schema
    http://www.fil.univ-lille1.fr/compagnie.xsd">
  <nom>UFR IEEA</nom>
  <adresse>
    <rue>Cite scientifique</rue>
    <CP>59655</CP>
    <ville>Villeneuve d'Ascq cedex</ville>
    <pays>France</pays>
  </adresse>
  <personnel>
    <personne numen="59678">
      <nom>Routier</nom>
      <prenom>Jean-Christophe</prenom>
      <mail>jean-christophe.routier@univ-lille1.fr</mail>
      <fonction>Professeur</fonction>
    </personne>
    <personne numen="59864">
      <nom>Roos</nom>
      <prenom>Yves</prenom>
      <mail>roos.yves@univ-lille1.fr</mail>
      <fonction>Maitre-de-Conferences</fonction>
    </personne>
    ....
  </personnel>
  <services>
    <service>
      <nom>Informatique</nom>
      <batiment>M3</batiment>
      <responsable>59678</responsable>
    </service>
    ....
  </services>
</compagnie>
```

Question 1 : (0,5 pt) Donner le (un) contenu (cohérent) de la balise ouvrante `schema`, racine du document `compagnie.xsd`.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://exo-schema"
  xsi:schemaLocation="http://www.w3.org/2001/XMLSchema http://www.w3 →
    .org/2001/XMLSchema.xsd"
  targetNamespace="http://exo-schema"
  elementFormDefault="qualified">
```

Question 2 : (0,5 pt) Donner la définition du type de l'élément **fonction** du schéma **compagnie.xsd**, sachant que les seules valeurs autorisées pour cet élément sont **Professeur** ou **Maitre-de-Conferences**.

```
<xsd:simpleType name="type-fonction">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Professeur"/>
    <xsd:enumeration value="Maitre-de-Conferences"/>
  </xsd:restriction>
</xsd:simpleType>
```

Question 3 : (0,5 pt) Donner la définition du type de l'élément **mail** du schéma **compagnie.xsd**, sachant que les seules valeurs autorisées pour cet élément sont de la forme **string.string@string.string**.

```
<xsd:simpleType name="type-mail">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value=".+[\.]+\@+[\.]+"/>
  </xsd:restriction>
</xsd:simpleType>
```

Question 4 : (0,5 pt) Donner la définition du type de l'attribut **numen** du schéma **compagnie.xsd**, sachant qu'un **numen** est une valeur entière comprise entre 10000 et 99999.

```
<xsd:simpleType name="type-numen">
  <xsd:restriction base="xsd:nonNegativeInteger">
    <xsd:minInclusive value="10000"/>
    <xsd:maxInclusive value="99999"/>
  </xsd:restriction>
</xsd:simpleType>
```

Question 5 : (1 pt) Donner la définition de l'élément **personne** du schéma **compagnie.xsd**, sachant que l'attribut **numen** est obligatoire.

```
<xsd:element name="personne">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="nom" type="xsd:string"/>
      <xsd:element name="prenom" type="xsd:string"/>
      <xsd:element name="mail" type="type-mail"/>
      <xsd:element name="fonction" type="type-fonction"/>
    </xsd:sequence>
    <xsd:attribute name="numen" type="type-numen" use="required"/>
  </xsd:complexType>
</xsd:element>
```

Question 6 : (1,5 pt) On souhaite définir comme clé primaire, pour toute personne, la valeur de son attribut `numen`. En précisant dans quelle définition d'élément du fichier `compagnie.xsd` faire apparaître cette définition de clé, donner les instructions `xml-schema` correspondantes.

```
<!-- dans l'element personnel -->
<xsd:key name="clef-personne">
  <xsd:selector xpath="personne"/>
  <xsd:field xpath="@numen"/>
</xsd:key>
```

Question 7 : (1,5 pt) On souhaite ajouter comme contrainte que le responsable d'un service doit être un membre du personnel. En précisant dans quelle définition d'élément du fichier `compagnie.xsd` faire apparaître cette contrainte, donner les instructions `xml-schema` correspondantes.

```
<!-- dans l'element compagnie -->
<xsd:keyref name="contrainte-responsable" refer="clef-personne">
  <xsd:selector xpath="services/service"/>
  <xsd:field xpath="responsable"/>
</xsd:keyref>
```

Exercice 5 : Espaces de noms

On considère le fichier `xml` suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<exercice xmlns:pre="http://nom-1">
  <pre:niveau xmlns:pre="http://nom-4" xmlns="http://nom-1">
    <out:garage xmlns:out="http://nom-3">
      <pre:alcove/>
      <entree xmlns:pre="http://nom-5">
        <pre:cuisine/>
      </entree>
    </out:garage>
  </pre:niveau>
  <def xmlns="http://nom-2">
    <pre:figue>
      <levain/>
    </pre:figue>
  </def>
</exercice>
```

Question 1 : (1,5 pt) Donner l'espace de nom de chacun des éléments `exercice`, `niveau`, `garage`, `alcove`, `entree`, `cuisine`, `def`, `figue` et `levain`.

```
exercice: aucun
niveau: http://nom-4
garage: http://nom-3
alcove: http://nom-4
entree: http://nom-1
cuisine: http://nom-5
```



```
def: http://nom-2  
figue: http://nom-1  
levain: http://nom-2
```