

BDA

jeudi 23 février 2011

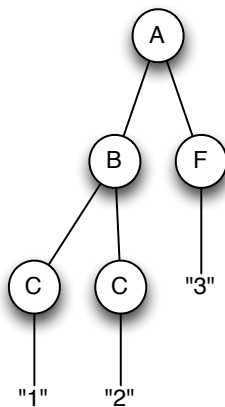
durée 1h20

support de cours autorisé

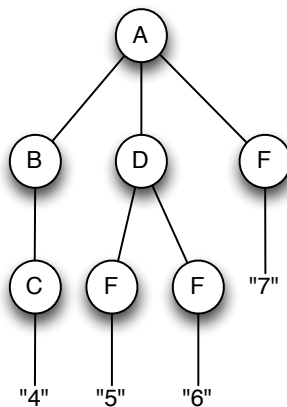
Interrogation écrite

Exercice 1 : DTD

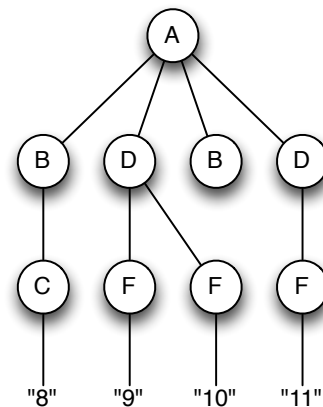
On considère les trois arbres xml suivants a_1 , a_2 et a_3 correspondant respectivement à des fichiers a1.xml, a2.xml et a3.xml :



a_1



a_2



a_3

On considère aussi les DTD suivantes :

```

<!ELEMENT A ((B|D)+,F?)>
<!ELEMENT B (C*)>
<!ELEMENT C (#PCDATA)>
<!ELEMENT D (F*)>
<!ELEMENT F (#PCDATA)>
    
```

d1.dtd

```

<!ELEMENT A ((B,D)+,F?)>
<!ELEMENT B (C*)>
<!ELEMENT C (#PCDATA)>
<!ELEMENT D (F*)>
<!ELEMENT F (#PCDATA)>
    
```

d2.dtd

```

<!ELEMENT A (B+,D+,F*)>
<!ELEMENT B (C*)>
<!ELEMENT C (#PCDATA)>
<!ELEMENT D (F*)>
<!ELEMENT F (#PCDATA)>
    
```

d3.dtd

Question 1 : Pour chacune de ces DTD, dire si elle valide ou non chacun des documents a1.xml, a2.xml et a3.xml.

Solution (3 pts)

d1.dtd : a1.xml, a2.xml; d2.dtd : a2.xml, a3.xml; d3.dtd : a2.xml

Exercice 2 : XML-Schema et XPath

On souhaite structurer dans un document XML les données d’emploi du temps d’une formation universitaire comme présenté dans le fichier `formation.xml` donné figure 1. La racine du document est un élément `formation` dont le contenu est la séquence formée d’un élément `creneaux` (contenant un nombre arbitraire de `creneau(x)` de l’emploi du temps) et d’un élément `matieres` (contenant un nombre arbitraire de description de `matiere(s)`).

Le schéma XML validant `formation.xml` est donné figure 2 par le fichier `formation.xsd` qui définit l’espace de noms "`http://fil.univ-lille1.fr/formation`".

Question 1 : Complétez la balise ouvrante de l’élément racine `formation` du fichier `formation.xml`

Solution (2 pts)

```
<formation
  xmlns="http://fil.univ-lille1.fr/formation"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://fil.univ-lille1.fr/formation formation. →
    ↪ xsd">
```

Question 2 : Complétez la définition de l’élément `semestre` du fichier `formation.xsd` sachant que le contenu d’un élément `semestre` est de la forme `Si` avec $i \in \{1, 2, 3, 4\}$.

Solution (1 pts)

```
<xsd:element name="semestre">
  <xsd:simpleType>
    <xsd:restriction base="xsd:token">
      <xsd:pattern value="S[1-4]" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

Un `creneau` de l’emploi du temps est caractérisé par la séquence d’informations suivantes :

1. quand a lieu le créneau (élément `semestre` ou élément `le` de type `date`),
2. le jour de la semaine du créneau (élément `jour` de type `jour-de-la-semaine`),
3. l’heure de début du créneau (élément `de` de type `time`),
4. l’heure de fin du créneau (élément `a` de type `time`),
5. une ou deux salles (éléments `salle` de type `token`),
6. un intervenant (éléments `intervenant` de type `string`),
7. une matière (élément `matiere` de type `token`).

Question 3 : Complétez la définition de l'élément **creneau** du fichier **formation.xsd**

Solution (2 pts)

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="le" type="xsd:date"/>
      <xsd:element ref="frm:semestre"/>
    </xsd:choice>
    <xsd:element name="jour" type="frm:jour-de-la-semaine"/>
    <xsd:element name="de" type="xsd:time"/>
    <xsd:element name="a" type="xsd:time"/>
    <xsd:element name="salle" type="xsd:token" minOccurs="1"
      maxOccurs="2"/>
    <xsd:element name="intervenant" type="xsd:string"/>
    <xsd:element name="matiere" type="xsd:token"/>
  </xsd:sequence>
</xsd:complexType>
```

Question 4 : Complétez la définition de l'élément **formation** du fichier **formation.xsd**

Solution (2 pts)

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="frm:creneaux" minOccurs="0"
      maxOccurs="unbounded"/>
    <xsd:element ref="frm:matieres" minOccurs="0"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Question 5 : Expliquez pourquoi ce schéma n'aurait pas pu être décrit par une DTD.

Solution (1 pt)

Un élément **matiere** dans un élément **creneau** et un élément **matiere** dans l'élément **matieres** ne sont pas du même type.

Question 6 : On souhaite préciser la contrainte suivante : *Tout créneau doit se référer à une matière qui existe*. En précisant à quels endroits vous ajoutez les déclarations nécessaires, définissez une clé et une référence de clé assurant cette contrainte.

Solution (3 pts)

Dans la définition de l'élément **matieres** :

```
<xsd:key name="cleMatiere">
  <xsd:selector xpath="frm:matiere"/>
  <xsd:field xpath="frm:nom-court"/>
</xsd:key>
```

Dans la définition de l'élément **formation** :

```
<xsd:keyref name="referenceMatiere" refer="frm:cleMatiere">
  <xsd:selector xpath="frm:creneaux/frm:creneau"/>
  <xsd:field xpath="frm:matiere"/>
</xsd:keyref>
```

Question 7 : Pour chacune des requêtes XPath suivantes, dire ce qu'elle retourne quand elle est appliquée sur le fichier **formation.xml** tel qu'il apparaît figure 1.

1. `//creneau[//matiere="COMA"]/intervenant`
2. `//creneau[matiere="COMA"]/intervenant`
3. `//creneau[/matiere="COMA"]/intervenant`

Solution (2 pts)

1. Tous les intervenants (le prédicat ne dépend pas du nœud contexte et vaut toujours **true**)
2. `<intervenant>Vincent Huet</intervenant>`
3. Résultat vide (le prédicat ne dépend pas du nœud contexte et vaut toujours **false**)

Question 8 : Donnez des requêtes XPath permettant, quand on les applique sur le document **formation.xml**¹, d'extraire

1. tous les nœuds **creneau** du document
2. toutes les salles utilisées par la matière "BDA"
3. tous les noms d'intervenants sans doublon
4. le nom long des matières pour lesquelles aucun créneau n'est défini
5. le nom long des matières pour lesquelles il existe un créneau utilisant 2 salles

Solution (4 pts)

1. `//creneau`
2. `//creneau[matiere="BDA"]/salle`
3. `//intervenant[not(following::intervenant=.)]/text()`
4. `//matieres/matiere[not(//creneau/matiere=nom-court)]/nom-long/text()`
5. `//matieres/matiere[nom-court=//creneau[count(salle)=2]/matiere]/nom-long/text()`

1. On suppose ici, bien sûr, que ce fichier n'est pas obligatoirement celui de la figure 1 mais n'importe quel document validé par le schéma **formation.xsd**.

```

<?xml version="1.0" encoding="UTF-8"?>
<formation
  ...
>
  <creneaux>
    <creneau>
      <semestre>S1</semestre>
      <jour>lundi</jour>
      <de>09:00:00</de>
      <a>12:00:00</a>
      <salle>M5-A2</salle>
      <salle>M5-B13</salle>
      <intervenant>Thomas Haessle</intervenant>
      <matiere>QSI</matiere>
    </creneau>
    <creneau>
      <semestre>S2</semestre>
      <jour>lundi</jour>
      <de>10:00:00</de>
      <a>12:00:00</a>
      <salle>M5-A2</salle>
      <salle>M5-B13</salle>
      <intervenant>Laurence Duchien</intervenant>
      <matiere>PEPIT</matiere>
    </creneau>
    <creneau>
      <le>2011-09-19</le>
      <jour>lundi</jour>
      <de>14:00:00</de>
      <a>18:00:00</a>
      <salle>M3-DELATTRE</salle>
      <intervenant>Vincent Huet</intervenant>
      <matiere>COMA</matiere>
    </creneau>
  </creneaux>
  <matieres>
    <matiere>
      <nom-court>COMA</nom-court>
      <nom-long>COmmunication et MAnagement</nom-long>
    </matiere>
    <matiere>
      <nom-court>PEPIT</nom-court>
      <nom-long>Projet Et Pratiques Informatiques Transversales</nom →
        ↔ -long>
    </matiere>
    <matiere>
      <nom-court>QSI</nom-court>
      <nom-long>Qualite des Systemes d'Information</nom-long>
    </matiere>
  </matieres>
</formation>

```

FIGURE 1 – Le fichier formation.xml

```

<?xml version="1.0"?>

<xsd:schema
  targetNamespace="http://fil.univ-lille1.fr/formation"
  elementFormDefault="qualified"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:frm="http://fil.univ-lille1.fr/formation">

  <xsd:element name="semestre">
    ...
  </xsd:element>

  <xsd:simpleType name="jour-de-la-semaine">
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="lundi"/>
      ...
      <xsd:enumeration value="dimanche"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:element name="matieres">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="matiere" minOccurs="0" maxOccurs="→
          ↪ unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="nom-court" type="xsd:token"/>
              <xsd:element name="nom-long" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="creneaux">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="creneau" minOccurs="0" maxOccurs="→
          ↪ unbounded">
          ...
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="formation">
    ...
  </xsd:element>

</xsd:schema>

```

FIGURE 2 – Le fichier `formation.xsd`