

## TD/TP 4 de PXML

janvier 2012

**Exercice 1 :** Définir les types simples suivants en les dérivant à partir des types simples prédéfinis les plus pertinents<sup>1</sup> :

- une heure comprise entre 2h30 (du matin) et 16h50 ;
- un nombre réel en précision simple, supérieur ou égal à -3476.4 et strictement inférieur à 5 ;
- une chaîne de quatre caractères ;
- une chaîne de caractères qui ne peut être égale qu'à "jpg", "gif" ou "png" ;
- un type de numéro ISBN : c'est un entier à 13 chiffres comme 9782744072369<sup>2</sup>.

**Exercice 2 :** Définir les types complexes suivants :

- Un type d'élément pouvant contenir du texte et un attribut optionnel `quantité` qui doit être un nombre entier.
- Modifier le type précédent afin que `quantité` possède comme valeur par défaut 1.
- Un type d'élément pouvant contenir la suite des éléments `elt1`, `elt2`, `elt3` et `elt4` ; l'élément `elt1` étant facultatif et présent au maximum une fois, `elt2` pouvant être présent deux ou trois fois, `elt3` et `elt4` étant présent une et une seule fois.
- Modifier le type précédent de manière à ce qu'on ait le choix entre `elt3` et `elt4`.

**Exercice 3 :**

On considère le fichier XML suivant :

```
<stock nom="OccazEtNeufs">
  <occases>
    <item prix="2000">
      <marque>Renault</marque>
      <modele>Clio II</modele>
      <annee>1998</annee>
    </item>
    <item prix="10000">
      <marque>BMW</marque>
      <modele>Serie 5</modele>
      <annee>2002</annee>
    </item>
    <item prix="4800">
      <marque>Peugeot</marque>
      <modele>307</modele>
      <annee>2001</annee>
    </item>
  </occases>
  <neufs>
    <item prix="18000">
      <marque>Renault</marque>
      <modele>Clio IV</modele>
```

1. il peut être utile de consulter la documentation <http://www.w3.org/TR/xmlschema-2/#built-in-datatypes>

2. On ne demande pas de vérifier que le numéro est valide ; par exemple le dernier chiffre est un chiffre de contrôle calculé à partir des 12 premiers. Le plus curieux peuvent consulter <http://fr.wikipedia.org/wiki/ISBN>

```
</item>
<item prix="29900">
  <marque>BMW</marque>
  <modele>Serie 5</modele>
</item>
</neufs>
</stock>
```

Question 1 : Définissez un XML-Schema à partir de cette instance XML en privilégiant la concision et l'*élégance*<sup>3</sup> de votre solution.

#### Exercice 4 :

En plein championnat de football, on veut mémoriser les scores et les dates du championnat dans un fichier XML.

- Dans le championnat de France, il y a 20 clubs. On mémoriser la liste des clubs, indépendamment du calendrier du championnat. Chaque club possède un nom, et une liste d'entraîneurs et de joueurs dont on veut connaître la date d'arrivée et la date de départ.
- Le championnat est divisé en journées, et chaque journée comporte 10 rencontres. On mémoriser la date de chaque journée, mais pas la date de chaque match (parfois un match se joue en avance ou en retard par rapport à la date officielle). Il y a au total 38 journées.
- Les clubs se rencontrent tous 2 fois, une fois à domicile et une fois à l'extérieur. Les 19 premières journées, chaque club rencontre tous les autres (on parle de matches aller). Les 19 journées suivantes, chaque club rencontre aussi tous les autres mais pas sur le même terrain (on parle de matches retour). On mémoriser tous les scores des matches du championnat.

Question 1 : Définissez un schéma XML qui permet de représenter le championnat de France de football. Vous définirez une clef primaire pour les clubs, et chaque rencontre fera référence à cette clef pour les 2 clubs qui y participent. La qualité d'un schéma est aussi liée à sa lisibilité. Evitez les imbrications trop profondes (schéma de type *poupées russes*).

---

3. qualité complètement subjective, mais bon...