

LABD

février 2015

Transformations XSLT

Sur le portail, récupérez l'archive `tgz` disponible et décompressez-là dans votre espace de travail. Cette archive contient deux sous-répertoires contenant les fichiers nécessaires à chacun des deux exercices. Pour appliquer une transformation `xslt` sur un fichier `xml` vous pouvez utiliser le logiciel `editix`. Pour l'exercice 1, vous pouvez utiliser également la commande `xsltproc` disponible sous linux et mac os : par exemple, pour appliquer une transformation définie dans le fichier `trans.xsl` sur un fichier `data.xml`, il suffit de taper la commande `xsltproc trans.xl data.xml` qui donnera le résultat sur la sortie standard.

Exercice 1 : Vous trouvez dans le répertoire `exercice-1` un schéma `championnat.xsd` et une instance `championnat.xml` (qui doivent vous rappeler un exercice du TP 3).

Question 1 : Essayez d'appliquer la transformation `q1.xsl` sur le fichier `championnat.xml` : une erreur se produit. C'est donc qu'il y a des choses à modifier dans le fichier `q1.xsl`. Corrigez les erreurs et essayez de nouveau.

Question 2 : Définissez une transformation qui permet d'extraire le sous-arbre des clubs. On doit obtenir un fichier de contenu équivalent au fichier `q2.xml`.

Question 3 : Définir une transformation qui génère un document `html` contenant les informations sur les clubs sous forme de tableau comme dans le fichier `q3.html` dont la visualisation est donnée ci-dessous :



The screenshot shows a web browser window with the address bar displaying `file:///Users/yroos/Documents`. The browser's menu bar includes `Famille`, `Fil`, `Docs`, and `Lifi`. The main content area displays the title **Les clubs de Ligue 1** followed by the subtitle **saison 2014-2015**. Below this, there is a table with two columns: **ville** and **club**. The table lists the following clubs and their cities:

ville	club
Bastia	Sporting Club de Bastia
Bordeaux	Girondins de Bordeaux
Caen	Stade Malherbe Caen
Evian	Evian Thonon Gaillard Football Club
Guingamp	En Avant de Guingamp
Lens	Racing Club de Lens
Lille	Lille Olympique Sporting Club Lille Metropole
Lorient	Football Club Lorient

Question 4 : Définir une règle avec un paramètre n nombre entier, qui a pour résultat la journée de numéro n . Utiliser cette règle pour définir une transformation qui extrait le sous-arbre de la 18ème journée. On doit obtenir un contenu équivalent au fichier `q4.xml`.

Question 5 : Définir une transformation qui permet de construire un document `xml` qui présente les clubs et les résultats obtenus à domicile et en extérieur. On veut obtenir un document de la forme donnée dans le fichier `q5.xml`.

Question 6 : Reprendre la transformation précédente afin d’ajouter l’identifiant de club comme attribut de chaque club (comme c’est le cas dans le document d’origine) et le numéro de journée comme attribut de chaque rencontre. On veut obtenir un contenu équivalent au fichier `q6.xml`.

Exercice 2 : Les fichiers `xml` de cet exercice décrivent des informations sur des mouvements artistiques, des artistes et des oeuvres présentées dans des musées.

La manipulation de plusieurs fichiers en entrée se fait très simplement, en utilisant la fonction `document()` dans les chemins `xpath` utilisés dans les attributs `select` des règles de transformation. Par exemple `document('catalogue-1.xml')/catalogue/oeuvres` sélectionne le noeud élément `oeuvres` du document `catalogue-1.xml`. Plus généralement, l’argument de la fonction `document()` peut être une liste de chemins comme dans

`document(('catalogue-1.xml','catalogue-2.xml'))//oeuvre`

qui sélectionne les éléments `oeuvre` présents dans l’un des deux documents `catalogue-1.xml` ou `catalogue-2.xml`.

Vous trouverez dans le répertoire `exercice-2` les fichiers suivants :

- `artistes.xml` qui contient des renseignements sur des artistes et des mouvements artistiques de schéma `artistes.xsd`.
- `catalogue-1.xml` et `catalogue-2.xml` qui sont des catalogues de musée, c’est-à-dire qu’ils contiennent toutes les oeuvres d’un musée, le nom de chaque musée étant contenu dans le document. Ces deux documents respectent le même schéma `catalogue.xsd`, et on peut imaginer qu’on possède plus que 2 documents de ce type.
- `catalogues.xml` qui contient la liste des fichiers qui sont des catalogues de musée (donc les deux documents précédents)

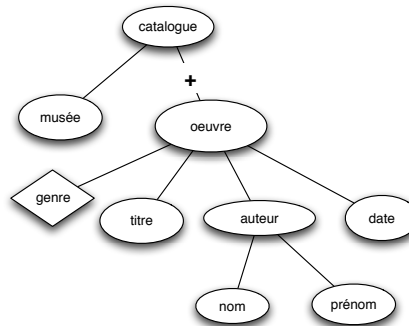
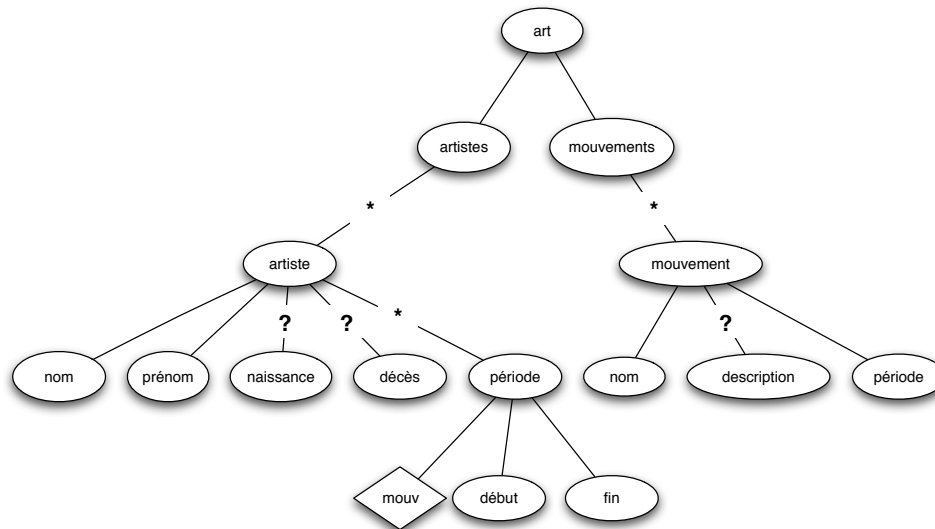
Une représentation graphique du schéma `artistes.xsd` est donnée figure 1 et une représentation graphique du schéma `catalogue.xsd` est donnée figure 2, où :

- 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,
- un arc avec un plus veut dire 1 ou plusieurs occurrences du sous-élément.

Les schémas de ces fichiers font partie du même espace de nom `http://labd/art`. On considérera que c’est l’espace de noms par défaut. De plus, il faut indiquer au processeur `xpath` l’espace de noms utilisé, afin de simplifier l’écriture des noms d’éléments dans les chemins `xpath`. Cette fonctionnalité n’est disponible qu’avec la version 2 de `xslt`, c’est pourquoi il est impossible d’utiliser `xsltproc` pour cet exercice. L’entête des fichiers `xslt` sera :

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://labd/art"
  xpath-default-namespace="http://labd/art" >
```

Question 1 : Écrire une transformation `xslt` qui, à partir de `artistes.xml` et `catalogue-1.xml`, réalise un seul document `nb-oeuvres.xml` qui permet d’insérer pour chaque artiste le nombre de

FIGURE 1 – Le schéma `catalogue.xsd`FIGURE 2 – Le schéma `artistes.xsd`

ses oeuvres présentes dans le catalogue (dans un élément de nom `<nb-oeuvres>` par exemple).

Question 2 : À partir de `artistes.xml` et `catalogue-1.xml`, réaliser un seul document que vous appellerez `fusion.xml` qui permet d'insérer pour chaque artiste le sous-arbre de ses oeuvres triées par ordre chronologique. Le schéma du document produit est donné figure 3.

Question 3 : Écrire une transformation `xslt` qui construit un fichier `artistes-avec-clefs.xml` à partir de `artistes.xml` en ajoutant une clef primaire comme attribut de chaque artiste.

Question 4 : Écrire une transformation qui, à partir de `artistes-avec-clefs` et `catalogue-1.xml` produit un fichier contenant les artistes, les mouvements et les oeuvres, avec des clefs étrangères entre les oeuvres et leurs auteurs. Le schéma du document produit est donné figure 4

Question 5 : Reprendre la transformation précédente afin de fusionner `artistes-avec-clefs` avec tous les catalogues (pas seulement `catalogue-1.xml`). La transformation utilisera `catalogues.xml` pour connaître les catalogues à fusionner avec le document des artistes.

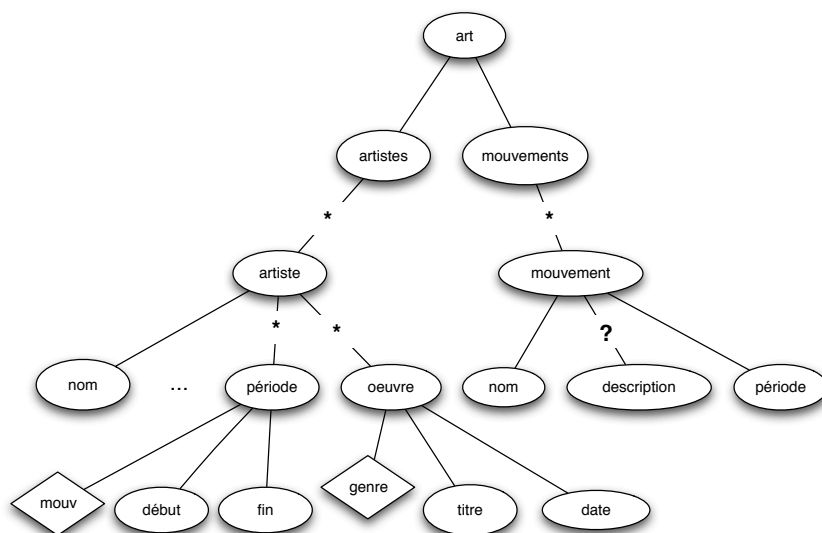


FIGURE 3 – Le schéma du document à produire pour la question 2

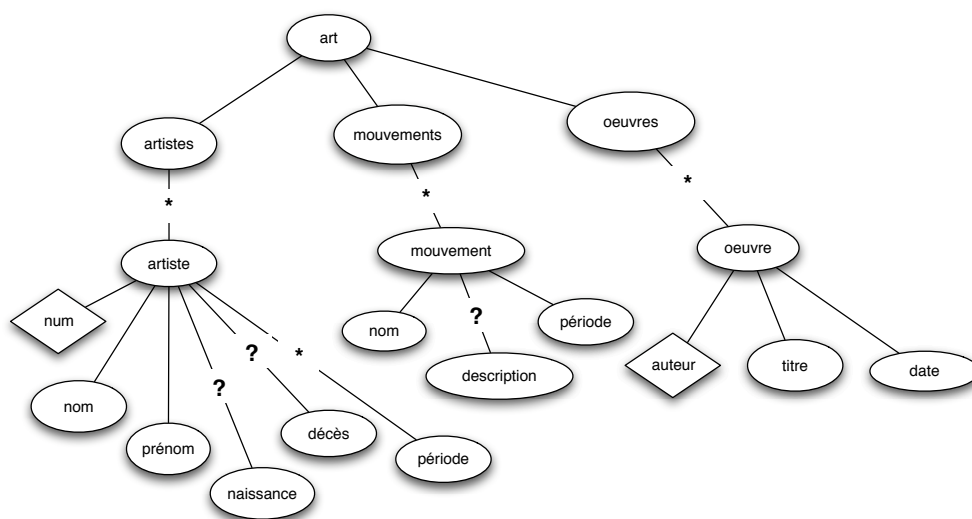


FIGURE 4 – Le schéma du document à produire pour la question 4