Devoir XML / XSLT

Frédérik Bilhaut
Université de Caen
Département d'informatique

Les fichiers fournis pour réaliser le devoir sont à récupérer ici :
http://www.info.unicaen.fr/~fbilhaut/ens/radi/devoir.tgz

Lorsqu'ils traitent un document, certains systèmes de traitement automatique de la langue marquent des segments dans le texte tout en leur associant des informations calculées au cours de l'analyse. Le document obtenu est dit *annoté*, et il existe bien-sûr de nombreuses variantes de ce procédé.

Un modèle couramment utilisé pour représenter les annotations elles-mêmes est celui des *structures de traits*. Une telle structure consiste en une liste de couples attribut/valeur, où chaque valeur peut être soit une chaîne de caractères, soit, récursivement, une autre structure de traits. En voici un exemple (pour information, cette structure représente pour nous la valeur sémantique de l'expression « depuis le milieu des années 1980 ») :

$$\begin{bmatrix} type : période \\ début : \begin{bmatrix} type : décennie \\ pivot : 1980 \\ delta : milieu \end{bmatrix} \end{bmatrix}$$

Dans cet exemple, le trait « type » a pour valeur la chaîne « période », et le trait « début » a pour valeur une structure de traits contenant à son tour les traits « type », « pivot », et « delta » (*on ne se préoccupera pas ici de la signification exacte de ces différents traits*).

On voit immédiatement que par leur nature arborescente, les structures de traits se prêtent tout naturellement à une représentation XML. Une solution simple est de faire correspondre à chaque trait un élément dont le contenu sera soit un fragment textuel (c'est une feuille), soit une nouvelle structure de traits (c'est un noeud). Sous ce format (que nous appellerons ici XFS pour « XML Feature Sets »), on obtiendrait pour l'exemple précédent :

```
<type>période</type>
<début>
  <type>décennie</type>
  <pivot>1980</pivot>
  <delta>milieu</delta>
</début>
```

Le fichier `annotations.lss` fourni contient des annotations produites sous ce format par un système d'analyse automatique. Le schéma utilise son propre espace de noms[1] (ici avec le préfixe **`lss`**). La racine du document est un élément **`semantics`**. Elle contient une liste d'éléments **`sem`** correspondant chacun à une annotation identifiée par les attributs **`type`** et **`id`**, chaque couple (type;id) étant unique. À l'intérieur de ces éléments, on s'intéressera uniquement aux éléments **`text`** et **`value`**. Le premier contient le texte du segment textuel annoté, et le second contient une structure de traits sous la forme exemplifiée ci-dessus (à noter que l'on n'utilise plus l'espace de noms **`lss`** dans la structure elle-même). Par exemple :

1 http://www.linguastream.org/LSS/2.0

```
<lss:sem type="xxx" id="yyy">
  <lss:value>
     <a>b</a>
     <c>
        <d>e</d>
        <f>g</f>
     </c>
  </lss:value>
</lss:sem>
```

L'objectif du devoir est de construire une feuille de transformation XSLT capable de transformer ces structures de traits au format MathML (évoqué en cours à plusieurs reprises), afin d'en produire une représentation sous forme de matrices (similaires à celle présentée au début de ce document). Il ne s'agit bien-sûr pas d'apprendre ce langage dans son intégralité, mais seulement les éléments nécessaires à la représentation de matrices. L'exemple qui suit suffit à donner une bonne idée des quelques balises MathML à mettre en œuvre (cela ressemble beaucoup aux tables HTML). On se rapportera au besoin à la spécification officielle[2] pour obtenir plus d'informations.

Voici une représentation MathML possible de la structure précédente (la représentation graphique attendue est présentée à droite) :

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mfenced open="[" close="]">
     <mtable columnalign="left">
        <mtr>
           <mtd><mi>a</mi> : b</mtd>
        </mtr>
        <mtr>
           <mtd><mi>c</mi> :
              <mfenced open="[" close="]">
                 <mtable columnalign="left">
                    <mtr><mtd><mi>d</mi> : e</mtd></mtr>
                    <mtr><mtd><mi>f</mi> : g</mtd></mtr>
                 </mtable>
              </mfenced>
           </mtd>
        </mtr>
     </mtable>
  </mfenced>
</math>
```

$$\begin{bmatrix} a:b \\ c:\begin{bmatrix} d:e \\ f:g \end{bmatrix} \end{bmatrix}$$

Travail à réaliser :

1°) Écrire une feuille XSLT permettant d'extraire une seule structure de traits du fichier `annotations.lss` (c'est à dire un noeud **`lss:sem`**). La structure à sélectionner sera identifiée par deux paramètres **`type`** et **`id`**.

2°) Implémenter une seconde feuille XSLT permettant de transformer *une* structure de traits au format XFS en formule MathML. Cette feuille devra être capable de fonctionner sur n'importe quel exemple présent dans le fichier `annotations.lss`.

Pour visualiser vos résultats, vous devrez utiliser un outil capable d'afficher du MathML. Les versions récentes de Mozilla[3] en sont capables (il existe même une page permettant de tester cette fonctionnalité en

2 http://www.w3.org/TR/REC-MathML/

3 http://www.mozilla.org/projects/mathml/

ligne[4]), pour peu que les fontes nécessaires soient installées[5]. Vous devriez par exemple pouvoir ouvrir directement dans Mozilla le fichier `exemple.xml` fourni.

À noter que vous pouvez installer une version de Mozilla (ou FireFox) en local sur votre compte. Si des problèmes se posent, vous pouvez également utiliser Amaya[6] ou tout autre outil capable d'afficher du MathML[7].

Questions subsidiaires :

3°) En utilisant correctement les espaces de noms[8], et en s'appuyant sur les exemples donnés en cours, créer à la main un document composite XHTML+MathML. Vous pourrez par exemple placer les différentes structures de traits à côté du texte annoté (un exemple est donné dans le fichier `exemple.xhtml`, que Mozilla est capable d'afficher correctement). Écrire une feuille XSLT permettant d'arriver automatiquement à ce résultat à partir d'un fichier tel que `annotations.lss`.

4°) Discuter la pertinence du format XFS. Quels en sont les avantages et inconvénients ? Quelle alternative pourrait-on proposer ? Chercher sur Internet d'autres vocabulaires XML visant la représentation des structures de traits.

Modalités :

Ce devoir doit impérativement être réalisé individuellement. Il devra être remis au plus tard le **8 novembre**, sous la forme d'un rapport argumenté, incluant les feuilles XSLT dans leur intégralité, et faisant apparaître clairement les résultats obtenus.

Contact :

Frédérik Bilhaut
bureau S3-385
fbilhaut@info.unicaen.fr
http://www.info.unicaen.fr/~fbilhaut

4 http://www.mozilla.org/projects/mathml/demo/tester.html
5 http://www.mozilla.org/projects/mathml/fonts/
6 http://www.w3.org/Amaya/
7 http://www.w3.org/Math/implementations.html
8 XHTML : http://www.w3.org/1999/xhtml
MathML : http://www.w3.org/1998/Math/MathML