CIHAM (UMR 5648), CNRS

XSLT : application

Biblissima+, scripts et manuscrits

Ariane Pinche
ariane.pinche@cnrs.fr

Biblissima+, scripts et manuscrits, 5-7 mai 2025

Table of Contents

Objectif de l'exercice

- Améliorer l'encodage TEI d'un poème structuré (voir fichier *Mon_reve_familierTEI*).
- Passer d'un balisage non conforme à un balisage conforme aux bonnes pratiques.
- Utiliser les balises dédiées à la poésie : `<lg>`, `<l>`, attributs `@type`, etc.
- Numéroter automatiquement le poème.

Consignes

1. Mettre en place une feuille de transformation XML vers XML
2. Reproduire la structure générale du poème
3. Copier le teiHeader
4. Transformer chaque `<p>` contenant une strophe en `<lg>` avec l'attribut `@type` adapté.
5. Si plusieurs hiérarchies de strophes sont présentes, les distinguer dans plusieurs `<lg>` successifs.
6. Transformer les `<seg>` qui encadrent chaque vers en `<l>`.
7. Numéroter automatiquement, en utilisant la fonction `xsl:number`, les vers à l'aide de l'attribut `@n` dans chaque `<l>`.

Table of Contents

Objectif de l'exercice

- Écrire une feuille XSLT pour générer un index de `<persName>` à partir d'un corpus TEI.
- Grouper les noms par attribut `@ref`.
- Compter les occurrences.
- Lister les fichiers dans lesquels ils apparaissent.

Étape 1

1. Préparer sa feuille de transformation à partir du premier fichier XML de la collection proposée.
2. Mettre en place une feuille de transformation XML vers XML.
3. Voici la structure XML attendue à la sortie :

```
<index>
  <entry>
    <name></name>
    <count></count>
    <sources>
      <source></source>
      <source></source>
    </sources>
  </entry>
</index>
```

Étape 2

1. Créer une règle racine au niveau

```
<xsl:template match="/">
  <index>
    <!-- contenu à générer -->
  </index>
</xsl:template>
```

Étape 3

1. Récupérer les noms de personnes et les groupes en fonction de @ref;
2. Remplir les champs de l'`entry`
   - Récupérer le texte de `persName`
   - Compter le nombre d'occurrences de la même personne
   - Trier les occurrences en fonction de leur nombre d'apparition
   - Donner le nom du document source à partir du @xml:id de `TEI`

Étape 4 : Définir le chemin vers une collection

- Donner le chemin vers la collection de fichiers à traiter

```
<xsl:param name="directory"
  select="'chemin/vers/le/dossier'"/>
```

- Récupérer tous les fichiers depuis le répertoire spécifié

```
<xsl:variable name="allFiles"
  select="uri-collection($directory)"/>
```

  La fonction uri-collection() en XSLT (et XPath 3.0) sert à récupérer une liste d'URI (Uniform Resource Identifiers), c'est-à-dire des adresses de fichiers, à partir d'un répertoire ou d'une collection définie.

- Filtrer la collection pour n'inclure que les fichiers xml

```
<xsl:variable name="xmlFiles"
  select="$allFiles[ends-with(., '.xml')]"/>
```

Étape 5 : Adapter le pointeur de la boucle pour

- Donner le chemin vers la collection de fichiers à traiter

```
<xsl:for-each-group
  select="document($xmlFiles)//body//persName"
  group-by="@ref">
```

La fonction `document()` permet de charger le contenu d'un ou plusieurs documents XML externes pendant la transformation.