

Séance 8 : ODD (2)

Tutoriel Oddbyexample (voir les vidéos de manipulation)

- Mise à jour des add-on dans Oxygen :
 - Options/Préférences/Association de type de document
 - Tout désactiver
 - Bien vérifier que la ligne "options globales" est cochée
- Add-ons
 - aide/installer les nouveaux add-ons
 - Ajouter <https://www.tei-c.org/release/oxygen/updateSite.oxygen>
 - Appliquer et accepter
 - Aide/Gérer les Add-ons/Installer
 - Tout activer et tout appliquer
 - Options/Préférences/Association de type de document
 - Tout activer, appliquer, accepter

- Mise en place du scénario
 - Configurer un scénario de transformation (CTRL+MAJ+C ou menu Document/Transformation/Configurer...)
 - Créer un nouveau scénario : XML transformation with XSLT;
 - Renseigner le chemin de l'XSL
 - `${frameworks}/tei/xml/tei/stylesheet/tools/oddbye
xample.xsl` (si cette solution ne fonctionnait pas, donner le chemin jusqu'au fichier .xsl fourni dans le dossier de la séance 8)

- Sélectionner processeur Saxon-PE ou HE 9
 - Options avancées, template (-it) : main;
 - Paramètres : corpus `${cfdu}` (i.e. répertoire courant, soit au dossier qui contient le ou les fichiers xml qui serviront de base à l'ODD)
- Configurer la sortie (onglet Sortie) : définir un nom et un emplacement pour la future ODD.
- Appliquer l'ODD sur le fichier Mon_reve_familierTEI.xml

Personnalisation du schéma

Pour personnaliser son schéma, il existe 4 manipulations principales :

- Ajouter des éléments ;
- Supprimer des éléments ;
- Changer des éléments ;
- Personnaliser les attributs et les valeurs d'attribut d'un élément.

Il existe deux types de modifications :

- Les modifications propres (« clean modification ») sont, dans le jargon de la TEI, une modification qui n'empêche pas la validité d'un document lorsqu'il est confronté aux modules de la TEI dans leur forme originelle. Soit, si vous procédez par réduction de la TEI, sans supprimer d'éléments obligatoires ou de parents d'éléments obligatoires.
- Une modification sale (« unclean modification ») est une modification qui ne permet pas cette validité. Invention d'un élément non TEI.

Par exemple : Dans le cas de l'attribut type de la classe d'attributs `att.textCritical`

- si je décide de restreindre cette liste à trois valeurs, et seulement à ces trois valeurs, est-ce une modification sale ? Non, car si je prends des documents dont l'attribut ne contient que ces trois valeurs, il reste valide face au modèle d'origine
- si j'ajoute un nouvel attribut `@subtype`, est-ce une modification propre ? Non, car je ne pourrais pas valider mon document face au modèle originel de la TEI.

N.B. : Les modifications "sales" ne sont pas mauvaises de façon inhérente, elles peuvent même être indispensables pour un projet donné, mais elles compliquent l'interopérabilité.

Suppression d'un élément

1- Suppression simple d'un élément avec @mode="delete"

- `<elementSpec ident="head" mode="delete"/>`

2- Suppression d'un élément ou une classe dans un module avec @except

- `<moduleRef key="textcrit" except="rdgGrp"/>`

3- Non insertion dans un module ou une classe avec @include

- `<moduleRef key="textcrit" include="app lem rdg"/>`

N. B. Dans la plupart des cas, la suppression d'un élément est une modification propre de la TEI, toutefois certains éléments disposent d'enfants obligatoires. Par exemple, `<fileDesc>` doit contenir à la fois `<titleStmt>` et `<sourceDesc>`. La suppression d'un élément enfant obligatoire dans le modèle de contenu rompt le modèle abstrait de la TEI

Manipulation 1

- Supprimer l'élément head dans l'ODD de Mon_reve_familierTEI.xml

N. B. À la fin de la manipulation, annulez cette suppression.

Modification d'un élément

Modification d'un élément avec @mode="change".

Cet attribut peut être appliqué sur les `<classSpec>` pour les classes, `<elementSpec>` pour les éléments, `<attDef>` pour les attributs, `<valList>` pour les listes de valeurs d'attributs.

```
<elementSpec ident="lg" mode="change">
  <attList>
    <attDef ident="type" mode="change">
      <valList mode="add" type="closed">
        <valItem ident="quatrain"/>
        <valItem ident="sizain"/>
        <valItem ident="sonnet"/>
        <valItem ident="tercet"/>
      </valList>
    </attDef>
  </attList>
</elementSpec>
```

Manipulation 2

- Modifier dans l'ODD l'élément `<body>` en lui ajoutant un attribut `@n` obligatoire grâce à l'attribut `@usage`, contraindre sa valeur à "sonnet".

Définir les modalités d'apparition d'un élément

La règle définissant une séquence apparaît directement en dessous de la balise ouvrante de l'**elementSpec** dans un élément **content**.

La séquence est contenue dans un élément **sequence** avec un attribut **preserveOrder** qui permet de spécifier si l'ordre de déclaration des éléments de la séquence est signifiant.

Chaque élément est appelé à l'aide d'un **elementRef** et d'un attribut **key** qui permet de donner le nom de l'élément. On peut également définir les modalités d'apparition des éléments de la séquence à l'aide des attributs **minOccurs** et **maxOccurs**.

Exemple

```
<elementSpec ident="div1" mode="change">
  <content>
    <sequence preserveOrder="true">
      <elementRef key="head" minOccurs="1"
        maxOccurs="1"/>
      <elementRef key="p" minOccurs="1"
        maxOccurs="unbounded"/>
      <elementRef key="div2" minOccurs="0"
        maxOccurs="unbounded"/>
    </sequence>
  </content>
</elementSpec>
```

NB : Pour autoriser du texte comme contenu, on peut ajouter dans la séquence : `<textNode/>`

Il est également possible de raffiner ses séquences avec l'élément

`<alternate>`

Exemple : définition du contenu de `<choice>`

```
<alternate>
  <sequence>
    <elementRef key="sic"/>
    <elementRef key="corr"/>
  </sequence>
  <sequence>
    <elementRef key="orig"/>
    <elementRef key="reg"/>
  </sequence>
  <sequence>
    <elementRef key="abbr"/>
    <elementRef key="expan"/>
  </sequence>
</alternate>
```

Manipulation 3

- Contraindre dans l'ODD, dans l'élément body la séquence suivante :
- `<head>`
- `<lg>`
- `<lg>`
- `<lg>`

Exercice

Reprendre le fichier XML de Lucain (séance 6),

1-Créer l'ODD de votre projet à l'aide de oddbyexample

2-Ajouter des règles à votre XML

- Rendre obligatoire dans l'apparat la présence d'un seul `<lem>` ;
- Rendre obligatoire la présence d'une ou plusieurs leçons dans `<app>` ;
- Générer votre schéma RelaxNG et l'associer à votre fichier XML.

3-Ajouter des contraintes d'usage d'attributs et de valeurs d'attribut