

Séance 12 : ODD (3) – ODD avancée : séquence d'éléments (relaxNG) et schematron

II- Personnaliser son ODD, niveau 2

1-Définir les modalités d'apparition d'un élément

La règle définissant une séquence apparaît directement en dessus 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="div" mode="change">
  <content>
    <sequence preserveOrder="true">
      <elementRef key="head" minOccurs="1" maxOccurs="1"/>
      <elementRef key="p" minOccurs="1" 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>
```

2-contraindre des valeurs d'attributs, des enchaînements en fonction du contexte (schematron)

- La contrainte est introduite par un élément `<constraintSpec>`. Le langage utilisé est déclaré dans l'attribut **scheme="schematron"**, la règle est nommée à l'aide de l'attribut *ident*.
- La règle est contenue dans une balise `<constraint>`.

NB : Attention à bien déclarer le nom de domaine *schematron* dans le préambule : `xmlns:s="http://purl.oclc.org/dsdl/schematron"`

Exemple

```
<constraintSpec ident="reforkeyorname" scheme="schematron">
  <constraint>
    <s:assert test="@ref or @key or @name">One of the
      attributes 'name',
      'ref' or 'key' must be supplied</s:assert>
    </constraint>
  </constraintSpec>
```

s:assert permet la vérification de l'existence de la contrainte rédigée en Xpath.

Ajouter un contexte

```
<constraintSpec ident="subclauses"
  scheme="schematron">
  <constraint>
    <s:rule context="tei:div">
      <s:assert test="count( tei:div )!= 1">
        if it contains any subdivisions,
        a division must contain at least two of them
      </s:assert>
    </s:rule>
  </constraint>
</constraintSpec>
```

s:rule permet d'ajouter un contexte à l'application de **s:assert**.

Contraindre l'activation d'un élément ou d'un attribut en fonction d'un contexte donné

```
<constraintSpec ident="fromTo" scheme="schematron">  
  <constraint>  
    <s:rule context="tei:app[@type='structure']">  
      <s:assert test="@from and @to">  
        The beginning and the endpoint of the  
        lemma have to be identify/  
      </s:assert>  
    </s:rule>  
  </constraint>  
</constraintSpec>
```

Contraindre le type de contenu d'une valeur d'attribut :

```
<constraintSpec ident="fromType" scheme="isoschematron">
  <constraint>
    <s:rule context="tei:app[@from]">
      <s:assert test="matches(@from, '^#w\d+$')">
        @from='#w+nb'` </s:assert>
      </s:rule>
    </constraint>
  </constraintSpec>
```

Exercice

Reprendre le fichier XML de Lucain,

1-Ajouter des règles à votre XML

- Paramétrer l'ordre de la séquence des éléments de l'apparat de telle sorte à ce que le lemme soit déclaré avant les leçons.
- rendre obligatoire la présence d'un seul `<lem>` ;
- rendre obligatoire la présence d'une ou plusieurs leçons ;
- Écrire une règle schematron pour que les valeurs `@n` de `<l>` se suivent de telle sorte que : `number(@n) = number(preceding-sibling::tei:l[1]/@n) + 1` quand le vers n'est pas en première position (reprendre le cours de Xpath si nécessaire);
- Générer votre schéma RelaxNG (syntax XML) et l'associer à votre fichier XML.