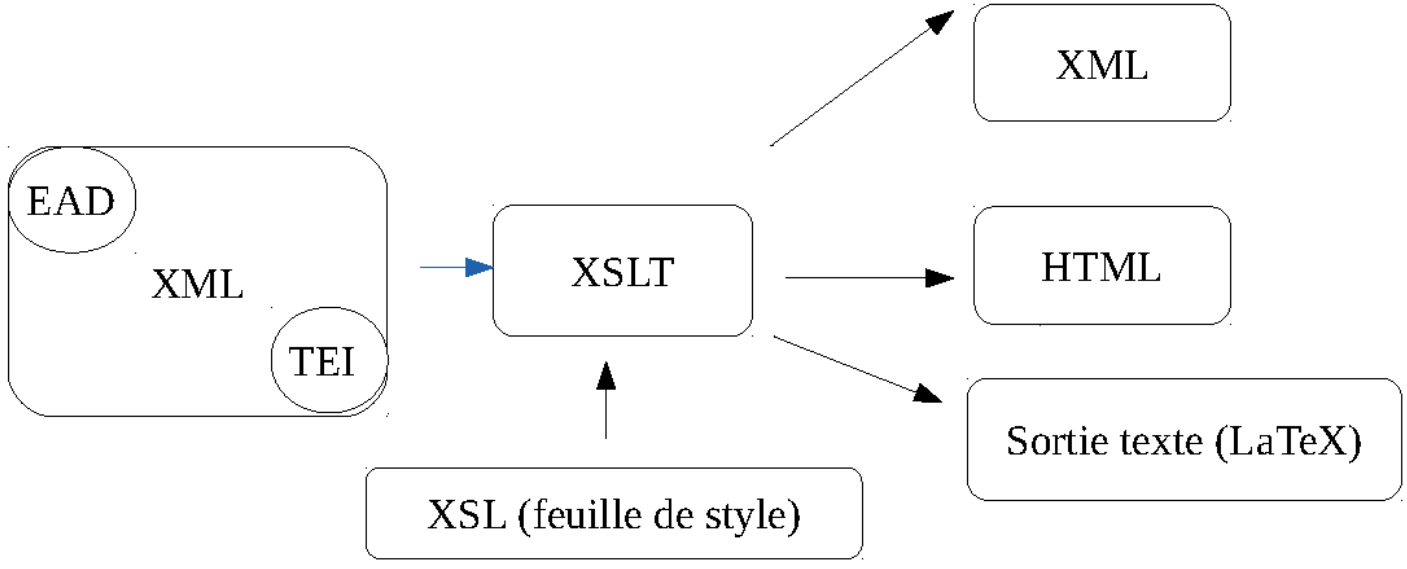
séance 2

Définitions et premières transformations

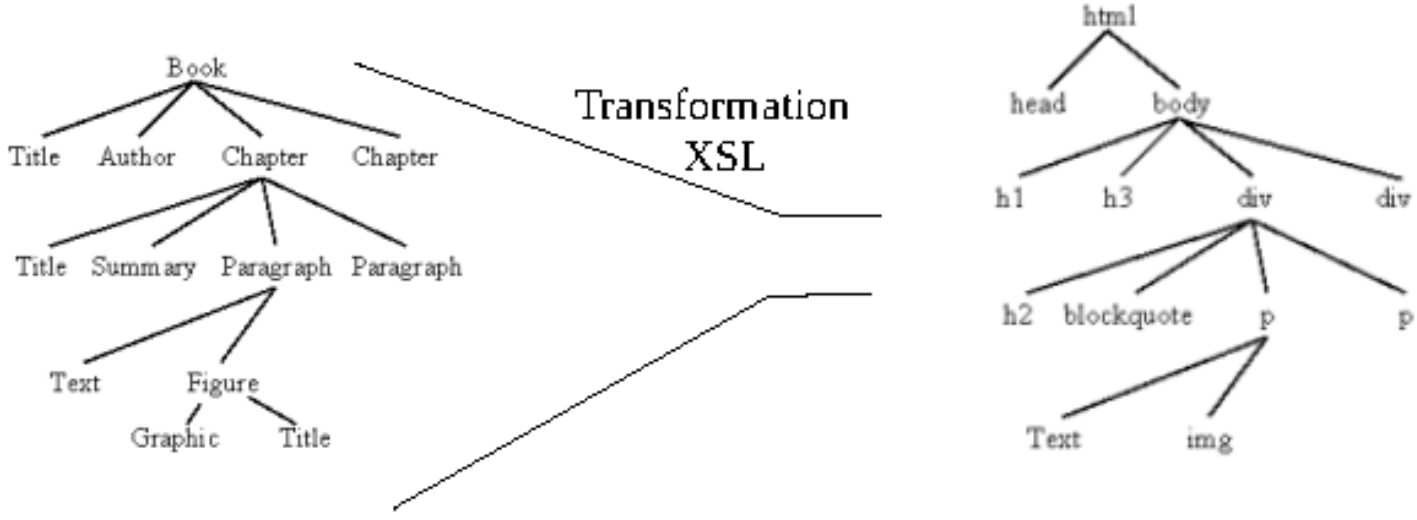
L'environnement XML



Définition

« XSLT (extensible Stylesheet Language Transformations) est un langage de programmation fonctionnel utilisé pour spécifier comment un document XML est transformé en un autre document qui peut, mais qui n'est pas nécessairement, un autre document XML. Un processeur XSLT lit un arbre XML en entrée et une feuille de style XSL et produit un arbre résultat en sortie. »

Elliotte Rusty Harold, W. Scott Means, Philippe Ensarguet [et al.], *XML en concentré*, Paris, O'Reilly, 2005, p. 519.


Book
Title
Author
Chapter
Chapter
Title
Summary
Paragraph
Paragraph
Text
Figure
Graphic
Title
Transformation
XSL
html
head
body
h1
h3
div
div
h2
blockquote
p
p
Text
img

Application de la règle n°1 qui porte sur XML A – Le motif demande la transformation en rouge de l’élément.

Principes généraux de fonctionnement

« Par défaut, un processeur XSLT lit le document XML d'entrée de haut en bas, commençant à l'élément racine et descendant dans l'arborescence en suivant l'ordre d'apparition des éléments. Les règles modèles sont activées dans l'ordre de rencontre des éléments. Ceci signifie qu'une règle modèle pour un élément sera activée avant les règles modèles correspondant à ses sous-éléments. »

Elliotte Rusty Harold, W. Scott Means, Philippe Ensarguet [et al.], *XML en concentré*, Paris, O'Reilly, 2005, p. 164.

Observer

Dans Oxygen :

1. Ouvrir le fichier Mon_reve_familierTEI.xml et un nouveau fichier xsl
2. Appliquer une XSLT vide sur le document Mon_reve_familierTEI.xml
3. Appliquer la règle suivante :

```
<xsl:template match="/">
        <xsl:apply-templates/>
</xsl:template>
```

4. Appliquer la règle suivante :

```
`<xsl:template match="text()"/>`
```

5. Appliquer la règle suivante :

```
<xsl:template match="/">
  <xsl:copy-of select="."/>
<xsl:template/>
```

Sélectionnner des éléments XML

1. Comment sélectionner et copier l'élément `<lg>` ?;
2. Sélectionner et copier uniquement les `<lg>` dont la valeur de @type est 'quatrain';
3. Sélectionner et copier tous les troisièmes vers;
4. Sélectionner et copier le troisième `<l>` du deuxième tercet;
5. Copier les `<lg>` qui sont premiers dans l'arbre;
6. Copier les `<lg>` qui sont moins que deuxième dans l'arbre;
7. Copier les `<lg>` qui sont plus que deuxième dans l'arbre.

Pour les plus aguerris : sélectionner uniquement le troisième vers du poème sans passer par `<lg>[1]`.

Mettre en place une règle

- Pour donner des instructions XSL, on utilise des templates/règles grâce à l'élément :

```
<xsl:template>
```

- On utilise l'attribut match pour sélectionner un élément de l'arbre XML :

```
<xsl:template match="mon_element_xml">
```

On peut ajouter dans la règle du texte,

```
<xsl:template match="mon_element_xml">
    Ici, il y avait mon élément
</xsl:template>
```

mais aussi des balises, ou un motif qui permet de récupérer certains éléments de l'arbre XML soit tels quels, soit transformés ou triés par des fonctions XSL ou Xpath.

```
<xsl:template match="mon_element_xml">
    <xsl:text>Ici,
      il y avait mon élément</xsl:text>
</xsl:template>
```

Éléments et attributs XSL

Éléments

Méthode n°1 :

```
<xsl:template match="mon_element_xml">
   <p>Ici, il y avait mon élément</p>
</xsl:template>
```

Méthode n°2 :

```
<xsl:template match="mon_element_xml">
<xsl:element name="p">Ici,
  il y avait mon élément</xsl:element>
</xsl:template>
```

Attributs

Méthode n°1 :

```
<xsl:template match="mon_element_xml">
  <p type="valeur_attribut">Ici,
    il y avait mon élément</p>
</xsl:template>
```

ou

```
<xsl:template match="mon_element_xml">
        <p type="{./chemin_Xpath}">Ici,
          il y avait mon élément</p>
</xsl:template>
```

Méthode n°2 :

```
<xsl:template match="mon_element_xml">
   <xsl:element name="p">
        <xsl:attribute name="type">
             <xsl:text>valeur_attribut</xsl:text>
        </xsl:attribute>
        <xsl:text>Ici,
          il y avait mon élément</xsl:text>
    </xsl:element>
</xsl:template>
```

Copy et copy-of

Copy

« L'élément XSL copy copie le nœud courant du document source vers le document de sortie. Il ne copie que le nœud lui-même. Cependant il ne copie pas ses enfants et ses attributs. »

Elliotte Rusty Harold, W. Scott Means, Philippe Ensarguet[et al.], *XML en concentré*, Paris, O'Reilly, 2005, p. 525.

```
<xsl:template match="mon_element_xml">
        <xsl:copy/>
</xsl:template>
```

Copy-of

« L'instruction xsl:copy-of insère le fragment d'arbre résultat identifié par l'attribut select dans le document de sortie. Cette instruction copie les nœuds spécifiques sélectionnés par l'expression et tous leurs enfants, attributs, espaces de noms et descendants. C'est en cela qu'il diffère de xsl:copy.»

Elliotte Rusty Harold, W. Scott Means, Philippe Ensarguet[et al.], *XML en concentré*, Paris, O'Reilly, 2005, p. 525.

```
<xsl:template match="mon_element_xml">
        <xsl:copy-of/>
</xsl:template>
```

Value-of

« L'élément xsl:value-of calcule la valeur textuelle d'une expression Xpath et l'insère dans l'arbre résultat. »

Elliotte Rusty Harold, W. Scott Means, Philippe Ensarguet[et al.], *XML en concentré*, Paris, O'Reilly, 2005, p. 543.

```
<xsl:template match="mon_element_xml">
        <xsl:value-of select="chemin_Xpath"/>
</xsl:template>
```

Apply-templates

Cette règle indique que les règles définies dans l'XSL doivent être appliquées aux éléments enfants de l'élément sélectionné par la règle.

```
<xsl:template match="mon_element_xml">
     <xsl:element name="p">
          <xsl:apply-templates/>
     </xsl:element>
</xsl:template>
```