

Plan

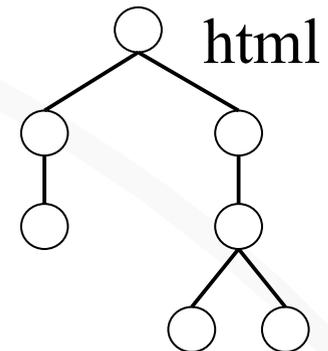
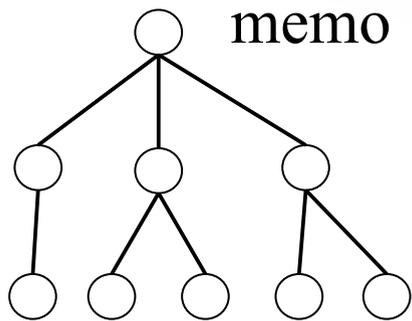
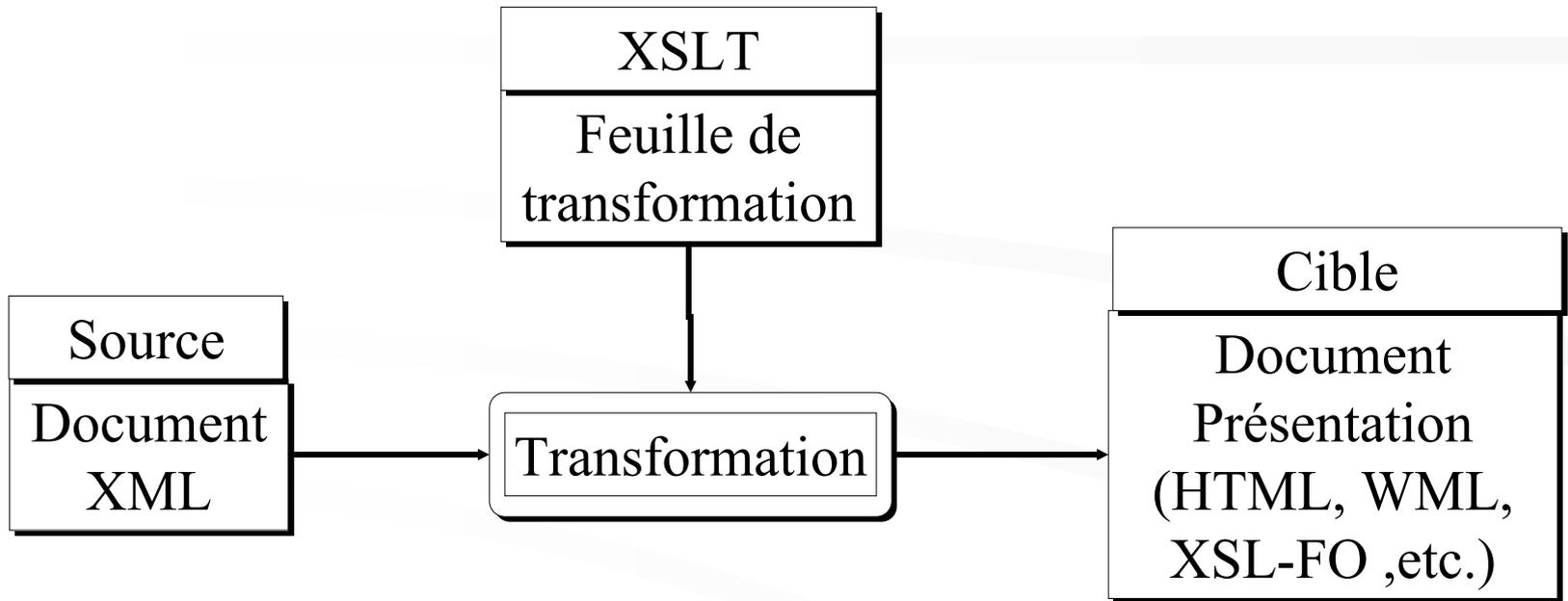
XSLT

- Concepts fondamentaux
- XSLT : transformation de documents

XSLT

- Conçu pour XML.
- Permet de transformer la structure du document ou des données
- Syntaxe : XML
- Structures de pages complexes, tables des matières, index, conversion de format

Principe de présentation étendu



Exemple

```
<memo security='team' >  
  <title>Rendez vous</title>  
  <to>J. Smith</to>  
  <from>B. Roberts</from>  
  
  ...  
</memo >
```

```
<html>  
  <body>  
    <h1> Rendez vous </h1>  
    <h3>From: B. Roberts To: J. Smith </h3>  
    <p> ... </p></body>  
</html>
```

Feuille de transformation

```
<?xml version= « 1.0 »?>
<xsl:stylesheet version="1.0"
  xmlns:xsl=« http://www.w3.org/1999/XSL/Transform»
  xmlns=« http://www.w3.org/REC-xhtml » >

  <xsl:template match=«memo»>
    <html>
      <body>
        <h1> <xsl:value-of select=«title»/> </h1>
        <h3>From: <xsl:value-of select=«from»/>
          To: <xsl:value-of select=«to»/> </h3>
        <p> ... </p></body>
      </html>
    </xsl:template>
  </xsl:stylesheet>
```

Structure générale

```
<?xml version= « 1.0 »?>  
<xsl:stylesheet version="1.0"  
  xmlns:xsl=« http://www.w3.org/1999/XSL/Transform»  
  xmlns=« http://www.w3.org/REC-xhtml » >  
  
  <xsl:template match=«memo»> <!-- Règle memo -->  
    <!-- Corps de la règle memo -->  
  </xsl:template>  
  
  <xsl:template match=«body»> <!-- Règle body -->  
    <!-- Corps de la règle body -->  
  </xsl:template>  
  
  ...  
  
<xsl:stylesheet>
```

Déclaration d'une règle

- Syntaxe:

```
<template match=« pattern »>  
  <!-- Corps de la règle -->  
</template>
```
- Ordre de déclaration pas important
- Pattern (ou sélecteur):
 - « memo » : s'applique aux éléments memo
 - « memo/to » : s'applique aux éléments to qui sont fils de l'élément memo

Exécution d'une règle

- Exécution séquentielle des éléments composant la règle
- Éléments non XSL : copier dans la cible
- Éléments XSL (instruction) : exécuter selon leur sémantique

xsl:value-of

Permet de générer du contenu XML à partir d'une expression

`xsl:value-of select=« title »`

-> Rendez vous

`xsl:value-of select=« ./@security»`

-> team

Passage d'une règle à une autre

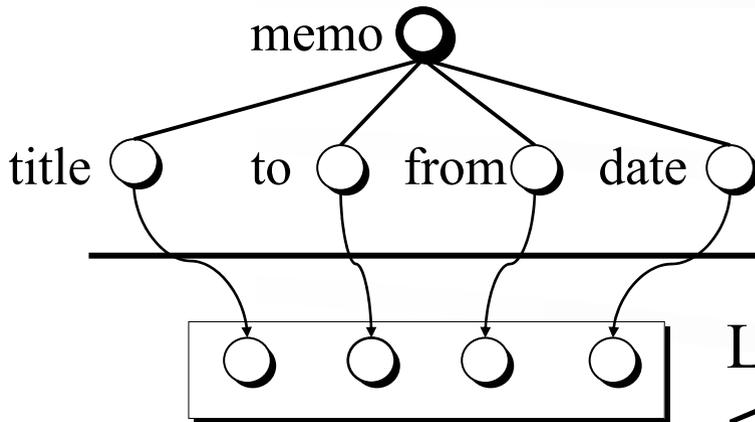
`<xsl:apply-templates/>`: sélectionne les fils du nœud courant

```
<xsl:template
  match=« memo »>
  <xsl:apply-templates/>
</xsl:template>
```

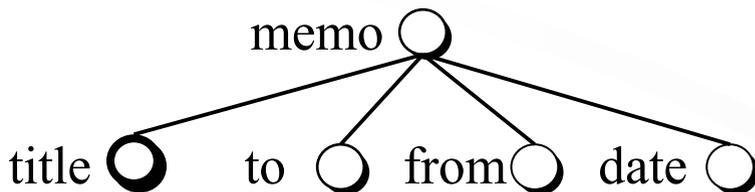
```
<xsl:template match=« title »>
  <h1>
    <xsl:value-of select=« . »/>
  </h1>
</xsl:template>
```

1. Sélection des nœuds fils du nœuds courant
2. Recherche une règle pour chacun de ces noeuds
3. Si règle trouvée alors l'instancier : le nœud instancié devient le nœud courant
4. Exécution de la règle

Exemple



○ Nœud courant



2: Recherche d'une règle dans la feuille de transformation qui correspond à title

3: Title devient nœud courant et la règle est exécutée

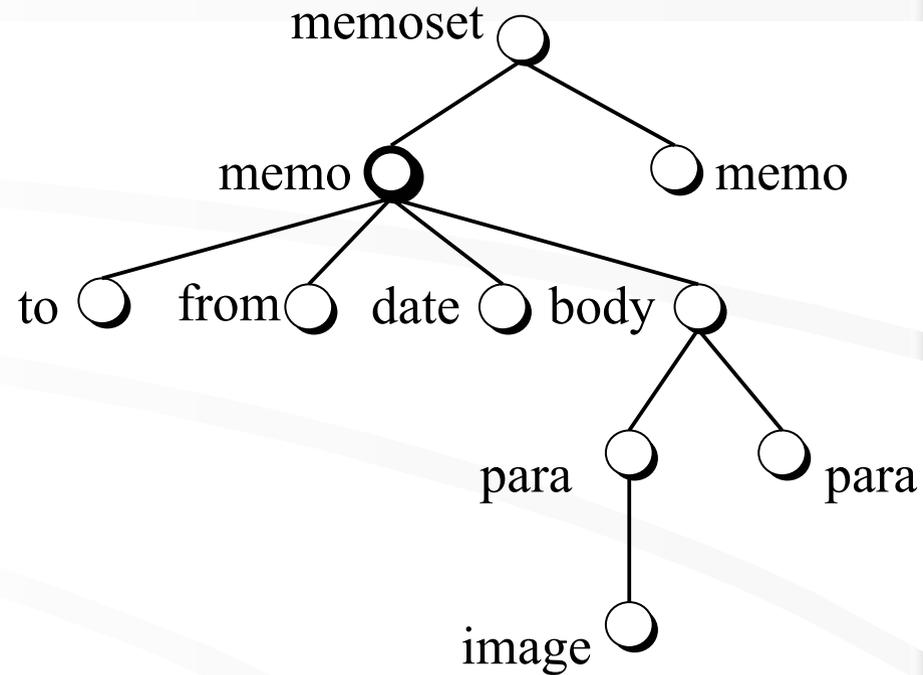
État initiale

- Nœud courant : racine de du document

Vers des sélections plus complexes

(1)

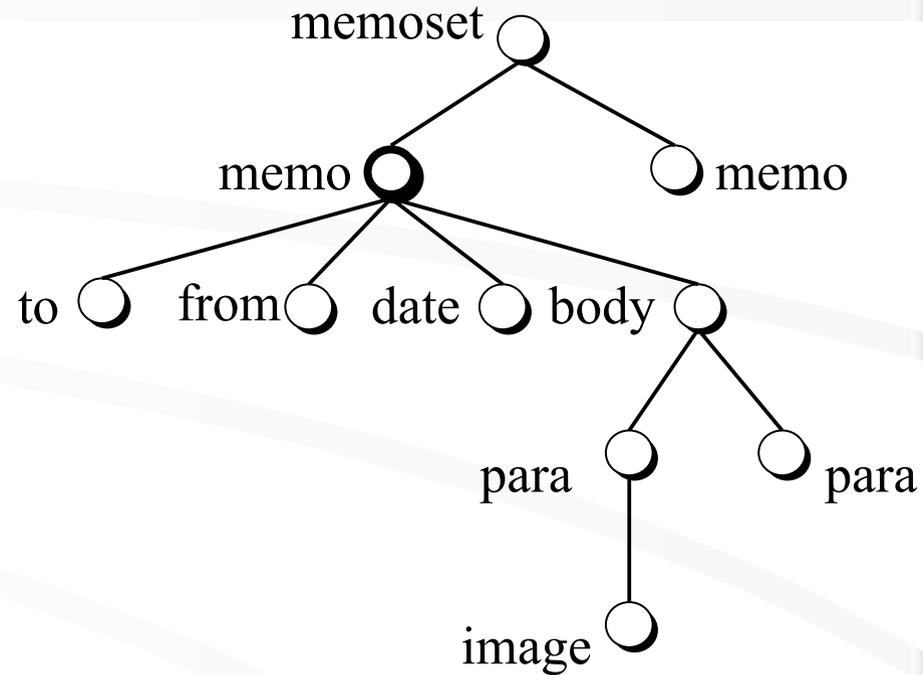
- `/memo[1]/to`
Sélectionne le destinataire du premier mémo
- `descendant::para`
Sélectionne tous les para descendant du nœud courant
- **Les autres instructions XSLT**



Vers des sélections plus complexes

(2)

- following-sibling::*
Le prochain frère par rapport au nœud courant
- Body/descendant::image
Toutes les images descendant de body
- /memoset/memo[descendant::image]
Tous les mémos ayant une image



Les autres instructions XSLT

```
<xsl:if test=« expression »/>
```

```
<xsl:if test=« from='dessgi' »>
```

```
.... Instructions ....
```

```
</xsl:if>
```