NFE204 - Bases documentaires et NoSQL

Principes de XSLT

Auteurs : Virginie Thion-Goasdoué, Philippe Rigaux

Équipe Vertigo
Laboratoire CEDRIC
Conservatoire National des Arts & Métiers, Paris, France

Publication de fichiers XML orientés documents

Plutôt pour des fichiers XML *orientés documents*.

- eXtensible HyperText Markup Language,
- Format XML pour la publication sur le web,
- Fichier XHTML = Fichier XML orienté document.

- Idéalement (bonne pratique), décrit la structure du document (titre, paragraphe, etc), la présentation est prise en charge par un fichier de feuille de style CSS.
- Dans la pratique, les balises de présentation disponibles dans XHTML sont souvent utilisées (`<font>`, `<color>`, etc) afin de ne pas avoir à écrire un fichier de feuille de style CSS associé.

XHTML et CSS

Une structure "seule" (sans information de présentation).

```
titre du doc : CV de Martin Durant

titre niveau 1 : Coordonnées

titre niveau 1 : Expérience professionnelles

titre niveau 1 : Diplomes
titre niveau 2 : 2008 - Licence pro CNAM
La licence pro cnam est un cursus ...
titre niveau 2 : 2000 - DUT génie industriel
titre niveau 2 : 1998 - BAC C

...
```

XHTML et CSS

A partir de la structure seule, les navigateurs Web sont capables (interpréteur (X)HTML intégré) d'inférer une présentation associée simple.
→ Renvoie un affichage basique du fichier XHTML en mettant les titres en gras, en affichant les listes avec des puces et en retrait, en soulignant les liens, etc.

Figure 1: Affichage basique à partir d'une structure

XHTML et CSS

Situation idéale

```
titre du doc : CV de Martin Durant

titre niveau 1 : Coordonnées

titre niveau 1 : Expérience professionnelles

titre niveau 1 : Diplomes
titre niveau 2 : 2008 - Licence pro CNAM
La licence pro cnam est un cursus ...
titre niveau 2 : 2000 - DUT génie industriel
titre niveau 2 : 1998 - BAC C

...
```

Fichier XHTML

le titre du doc est en lettres capitales
les titres niveau 1 sont en gras police Times 12pt, vert
les titres niveau 2 sont en gras police Times 10pt, bleu
le texte est du verdana 10pt, noir

le fond du document est beige

Fichier CSS associé

XHTML et CSS (suite)

Souvent, dans la pratique (pas de fichier CSS associé, informations de format incluses dans le fichier contenant les données).

```
fond du document : beige

titre du doc : En lettres capitales "CV de Martin Durant"

titre niveau 1 : En times, 12pt, vert, gras "Coordonnées"

titre niveau 1 : En times, 12pt, vert, gras "Expérience professionnell

titre niveau 1 : En times, 10pt, bleu, gras "Diplomes"
titre niveau 2 : En times, 10pt, bleu, gras "2008 - Licence pro CNAM"
En verdana, 10pt, noir "La licence pro cnam est un cursus ..."
titre niveau 2 : En times, 10pt, bleu, gras "2000 - DUT génie industri
titre niveau 2 : 1998 - En times, 10pt, bleu, gras "BAC S"
```

XHTML, Structure d'un document

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>CV de Martin Durant</title>
  </head>
  <body>
    Contenu structure du CV.
  </body>
</html>
```

Balise `<head></head>` : contient des méta-données du document p.e. author, copyright, generator, mots clef pour indexation par les moteurs de recherche, titre de la page XHTML, etc. et aussi la référence au fichier CSS associé.

Balise `<body></body>` : englobe le document en lui-même, sa partie visible.

Lien W3C : http ://www.w3.org/TR/2008/REC-xhtml-basic-20080729/

Quelques balises XHTML

Les titres

```
1  <html xmlns="http://www.w3.org/1999/xhtml" >
2    <head>
3      <title>CV de Martin Durant, Chef de projet informatique</title>
4    </head>
5    <body>
6      <h1>Martin Durant, Chef de projet informatique</h1>
7      Né le 1er janvier 1972
8      Marié, quatre enfants
9      Nationalité Française
10     <h1>Coordonnées</h1>
11     Adresse : 3 Allée du Lac, 91400 Orsay
12     Téléphone : 01 23 45 67 89
13     Messagerie : martin.durant@mamessagerie.fr
14     <h1>Expériences professionelles</h1>
15       <h2> Chef de projet junior chez Alcatel</h2>
16       J'ai occupé ce poste de 2004 à 2009. Intégré à l'équipe Réseaux et télécommunication, j'ai maintenu ...
17       <h2>Développeur JAVA chez TransSystèmes Pro</h2>
18       A la sortie de mon diplome de DUT, j'ai rejoins le goupe TransSystèmes Pro. J'y ai été développeur sur de
                modules ad-hoc sur le systèmes d'information mis en place en appui au service commercial de l'
                entreprise ...
19           <h3>Développement JAVA</h3>
20           En terme de développement JAVA, j'ai un particulièrement travaillé sur le développement de
                  frameworks ...
21           <h3>Gestion de projet</h3>
22           J'ai touché du doigt la gestion de projet à la fin de ...
23   </body>
24 </html>
```

Quelques balises XHTML (suite)

Paragraphes, saut de lignes forcés

```
1   <body>
2    <h1>Martin Durant, Chef de projet informatique</h1>
3     <p>Né le 1er janvier 1972</p>
4     <p>Marié, quatre enfants</p>
5     <p>Nationalité Française</p>
6    <h1>Coordonnées</h1>
```

`<p></p>` : paragraphe
`<br/>` : balise ouvrante-fermante pour saut de ligne forcé (là, on commence à introduire des balises de présentation dans XHTML i.e. XHTML ne contient pas juste des informations structurelles)

Quelques balises XHTML (suite)

Listes

```
1     <h1>Coordonnées</h1>
2     <ul>
3       <li>Adresse : 3 Allée du Lac, 91400 Orsay</li>
4       <li>Téléphone : 01 23 45 67 89</li>
5       <li>Messagerie : martin.durant@mamessagerie.fr</li>
6     </ul>
```

`<ul></ul>` : unordered list

`<ol></ol>` : ordered list avec `<li>item</li>`

`<dl></dl>` : definition list avec

`<dt>nom item</dt><dd>description item</dd>`

Quelques balises XHTML (suite)

Figure 2: Affichage d'un fichier XHTML par un navigateur

Quelques balises XHTML (suite)

Listes

```
1     <h1>Coordonnées</h1>
2     <dl>
3       <dt>Adresse</dt> <dd>3 Allée du Lac, 91400 Orsay</dd>
4       <dt>Téléphone</dt> <dd>01 23 45 67 89</dd>
5       <dt>Messagerie</dt> <dd>martin.durant@mamessagerie.fr</dd>
6     </dl>
```

Quelques balises XHTML (suite)

Figure 3: Affichage d'un fichier XHTML par un navigateur

Quelques balises XHTML (suite)

Liens

```
1  <h1>Expériences professionelles</h1>
2   <h2> Chef de projet junior chez Alcatel</h2>
3  J'ai occupé ce poste de 2004 à 2009. Intégré à l'équipe Réseaux et télécommunication, j'ai maintenu ...
4  <h2> Développeur JAVA chez <a href="www.transsystemes.org">TransSystèmes Pro</a></h2>
5  A la sortie de mon diplome de DUT, j'ai rejoins le goupe TransSystèmes Pro. J'y ai été développeur sur de
       modules ad-hoc sur le systèmes d'information mis en place en appui au service commercial de l'
       entreprise ...
```

Quelques balises XHTML (suite)

Fontes, italique, gras, etc

Comme pour la balise `<br/>`, on sort du cadre de la structure du document pour commencer à définir des options de présentation.

```
1   <h1>Coordonnées</h1>
2   <dl>
3     <dt>Adresse</dt> <dd>3 Allée du Lac, 91400 Orsay</dd>
4     <dt>Téléphone</dt> <dd>01 23 45 67 89 (<i>répondeur</i>)</dd>
5     <dt>Messagerie</dt> <dd>martin.durant@mamessagerie.fr </dd>
6   </dl>
7   <h1>Expériences professionelles</h1>
8   <h2> <font color="green">Chef de projet junior</font> chez Alcatel</h2>
9   J'ai occupé ce poste de 2004 à 2009. Intégré à l'équipe Réseaux et télécommunication, j'ai maintenu ...
10  <h2> Développeur JAVA chez <a href="www.transsystemes.org">TransSystèmes Pro</a></h2>
11  A la sortie de mon diplome de DUT, j'ai rejoins le goupe TransSystèmes Pro. J'y ai été développeur sur de
        modules ad-hoc sur le systèmes d'information mis en place en appui au service commercial de l'
        entreprise ...
```

Quelques balises XHTML (suite)

Figure 4: Affichage d'un fichier XHTML par un navigateur

Quelques balises XHTML (suite)

et des tableaux, des ancres, et encore beaucoup d'autres fonctionnalités, ...
Read The Friendly Manual...

Vous trouverez facilement pléthore d'informations sur XHTML [**?**].

Quelques balises XHTML (suite)

Comment ça marche ?

```
1  <html xmlns="http://www.w3.org/1999/xhtml" >
2    <head>
3      <title>CV de Martin Durant, Chef de projet informatique</title>
4    </head>
5    <body>
6      <h1>Martin Durant, Chef de projet informatique</h1>
7      Né le 1er janvier 1972
8      Marié, quatre enfants
9      Nationalité Française
10     <h1>Coordonnées</h1>
11     Adresse : 3 Allée du Lac, 91400 Orsay
12     Téléphone : 01 23 45 67 89
13     Messagerie : martin.durant@mamessagerie.fr
14     <h1>Expériences professionelles</h1>
15       <h2> Chef de projet junior chez Alcatel</h2>
16       J'ai occupé ce poste de 2004 à 2009. Intégré à l'équipe Réseaux et télécommunication, j'ai maintenu ...
17       <h2>Développeur JAVA chez TransSystèmes Pro</h2>
18       A la sortie de mon diplome de DUT, j'ai rejoins le goupe TransSystèmes Pro. J'y ai été développeur sur de
                modules ad-hoc sur le systèmes d'information mis en place en appui au service commercial de l'
                entreprise ...
19           <h3>Développement JAVA</h3>
20           En terme de développement JAVA, j'ai un particulièrement travaillé sur le développement de
                  frameworks ...
21           <h3>Gestion de projet</h3>
22           J'ai touché du doigt la gestion de projet à la fin de ...
23   </body>
24 </html>
```

Quelques balises XHTML (suite)

Qu'affiche le navigateur si on remplace la ligne `<html xmlns="http://www.w3.org/1999/xhtml">` par la ligne `<html>`

Quelques balises XHTML (suite)

Figure 5: Affichage d'un fichier XHTML par un navigateur

CSS

Principe : permet d'appliquer des propriétés de présentation évoluée aux balises XML via des règles.

- Cascading Style Sheets ;
- Langage non XML pour la définition de présentation de document XML (souvent XHTML) ;
- Trois niveaux (versions) 1, 2 et 3. Niveau 2 le plus utilisé (niveau 3 pas encore entièrement supporté par les navigateurs) ;

[?]

CSS (suite)

Règle CSS : `selecteur { proriete : valeur }`

selecteur champ d'application de la règle
par exemple la balise `h1`

propriete la propriété de la balise à instancier
par exemple la couleur `color`

valeur valeur de la propriété
par exemple `green`

```
h1 {color : green}
```

CSS (suite)

De façon plus générale

```
sel1, ..., selk {
p1 : v1;
... ;
pn : vn
}
```

```
h1, h2, h3 {
color : green;
font: normal small "Trebuchet MS", Arial, Helvetica, sans-seri
text-decoration : underline
}
```

Il existe plus d'une centaine de propriétés CSS...

CSS (suite)

Un fichier CSS est composé d'un ensemble de règles CSS

Un fichier CSS est lié au fichier X(HT)ML à présenter par la balise :

```
<link rel="stylesheet" type="text/css" href="path_to_style/style.css"/
```

déclarée dans le fichier X(HT)ML

CSS (suite)

Si on veut pouvoir définir des styles différents pour une même balise, utiliser l'attribut `class`. *On retrouvera ce principe avec l'attribut* `mode` *de XSLT*

Par exemple :

```
1  h1.commun {
2  color : green;
3  font: normal small "Trebuchet MS", Arial, Helvetica, sans-serif;
4  text-decoration : underline
5  }
6
7  h1.coord {
8  color : red;
9  font: bold small "Trebuchet MS", Arial, Helvetica, sans-serif;
10 text-decoration : underline
11 }
12
13 h2, h3 {
14 color : green;
15 font: normal small "Trebuchet MS", Arial, Helvetica, sans-serif;
16 text-decoration : underline
17 }
```

Listing 1 – Fichier CSS simple (styleCV.css)

CSS (suite)

Fichier XHTML :

```
1  <html xmlns="http://www.w3.org/1999/xhtml">
2  <link rel="stylesheet" type="text/css" href="./styleCV.css" />
3    <head>
4      <title>CV de Martin Durant</title>
5    </head>
6    <body>
7      <h1 class="commun">Martin Durant, Chef de projet informatique</h1>
8      <p>Né le 1er janvier 1972</p>
9      <p>Marié, quatre enfants</p>
10     <p>Nationalité Française</p>
11     <h1 class="coord">Coordonnées</h1>
12     <dl>
13       <dt>Adresse</dt> <dd>3 Allée du Lac, 91400 Orsay</dd>
14       <dt>Téléphone</dt> <dd>01 23 45 67 89 (<i>répondeur</i>)</dd>
15       <dt>Messagerie</dt> <dd>martin.durant@mamessagerie.fr </dd>
16     </dl>
17     <h1 class="commun">Expériences professionelles</h1>
18     <h2> <font color="green">Chef de projet junior</font> chez Alcatel</h2>
19     J'ai occupé ce poste de 2004 à 2009. Intégré à l'équipe Réseaux et télécommunication, j'ai maintenu ...
20     <h2>
```

Listing 2 – Fichier XHTML (cv.xhtml)

CSS (suite)

Interprété par Firefox :

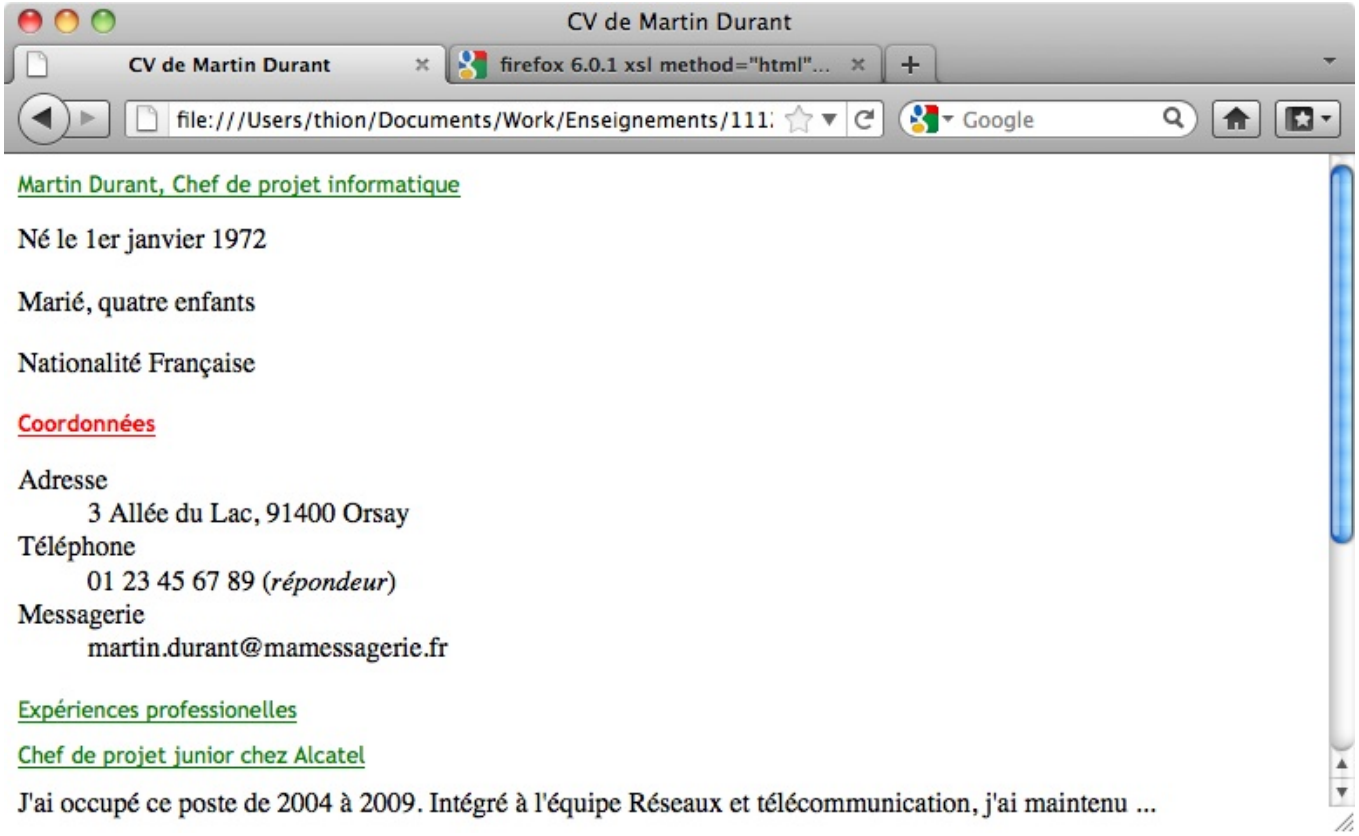


Figure 6: Affichage d'un fichier XHTML par un navigateur

CSS (suite)

Une feuille de style peut devenir assez compliquée (des règles peuvent rentrer en conflit, une notion de priorité est définie, etc). Le langage des feuilles de style CSS est un langage assez puissant qui, s'il est maitrisé, permet d'élaborer des présentations sophistiquées.

CSS (suite)

	XHTML seul (structure et présentation "mélangées")	XHTML pour la structure et CSS pour la présentation
Facilité de définition	Simple à écrire	Plus compliqué à écrire
Langage de définition de la présentation	Verbeux	Compact
Informations concernant la présentation	"mélangées" au contenu, répétées pour chaque tag	Indépendantes non répétées
Maintenance	Difficile si plusieurs fichiers à uniformiser (voir même un seul !)	Un grand plus par exemple si tous les fichiers d'un site web pointent vers une même feuille de style CSS car il suffit de mettre à jour le fichier CSS pour changer l'apparence de toutes les pages du site
Profil administrateur	"lambda"	Demande une connaissance de CSS

XSLT

XSL : Extensible Stylesheet Language.

Composé de deux dialectes de XML :

- XSL-FO (XSL Formatting Objects) – dialecte XML permettant de décrire un rendu attendu sur une page de texte. Langage technique plutôt destiné à des typographes. À partir d'un document XSL-FO, un formateur XSL-FO peut créer un document final, par exemple un document pdf.
- XSLT (XSL Transformations) – dialecte XML permettant de définir des transformations pour transformer un document XML en autre un document XML. Ce document XML peut (par exemple) être :
  - XHTML (pour afficher "élégamment" le contenu d'un fichier XML) ;
  - XSL-FO si l'objectif est un rendu très perfectionné ou papier.

XSLT (suite)

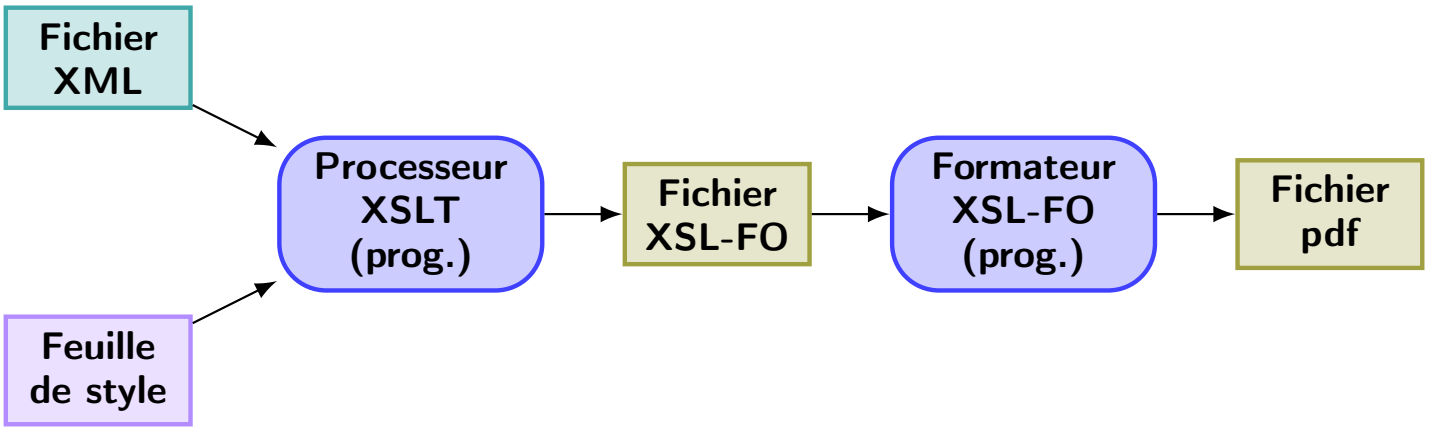


Figure 7: Fonctionnement de XSLT/XSL-FO

XSLT : d'un fichier XML à un autre fichier XML

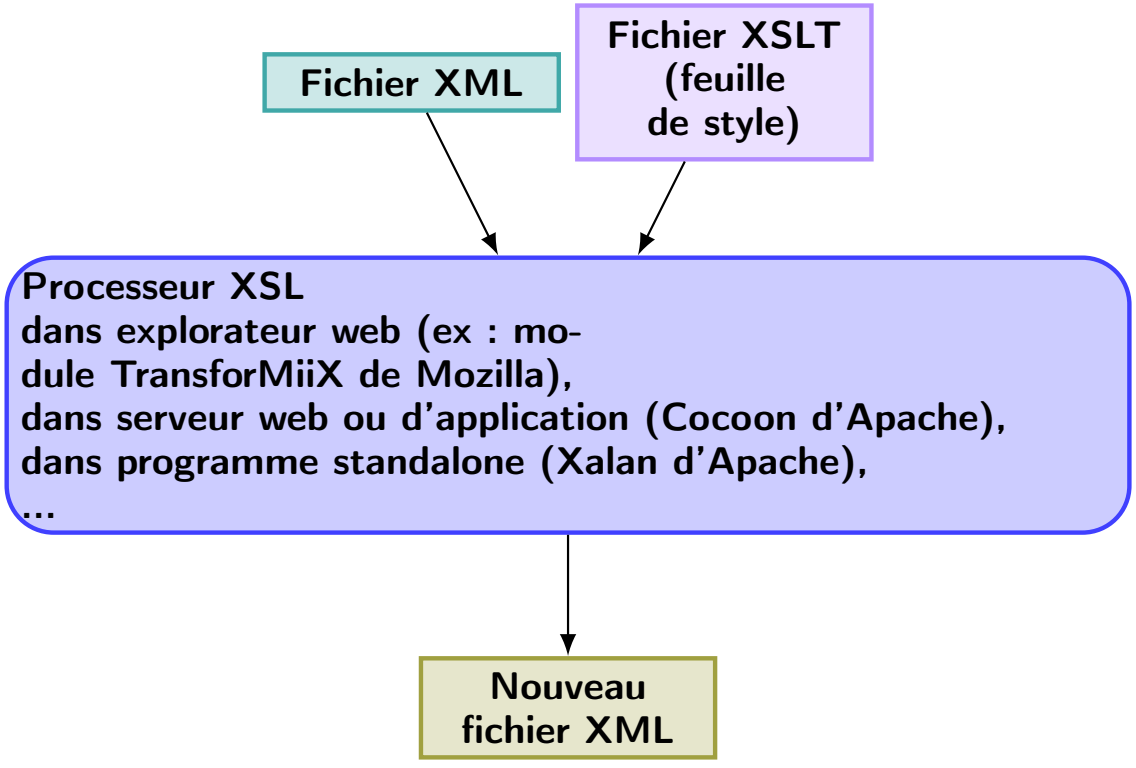


Figure 8: D'un fichier XML à un autre fichier XML

XSLT : d'un fichier XML à un autre fichier XML (suite)

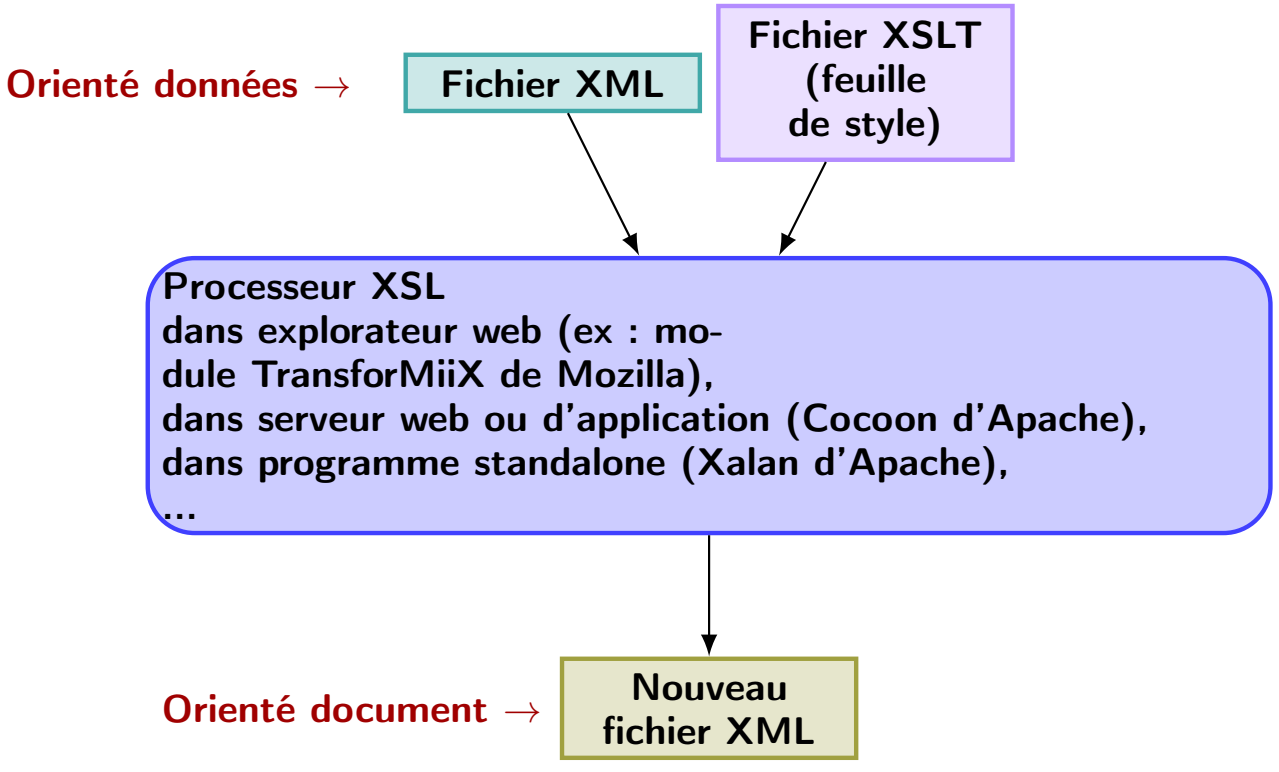


Figure 9: Publication d'un fichier XML orienté données à l'aide de XSLT

XSLT : exemple

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <diagrammesUML>
3    <diagramme UML1='yes' UML2='yes'>
4      <name>Diagramme de classes</name>
5      <description>Modélise les objets du système étudié et leurs relations.</
           description>
6      <objets>
7        Acteurs, cas d'utilisation
8      </objets>
9    </diagramme>
10   <diagramme UML1='yes' UML2='yes'>
11     <name>Diagramme de cas dutilisation</name>
12     <description>Modélise l'ensemble des actions réalisées par le système
           étudié d'un utilisateur extérieur au système</description>
13     <objets>Classes, associations</objets>
14   </diagramme>
15   <diagramme UML1='yes' UML2='yes'>
16     <name>Diagramme de séquence</name>
17     <description>Modélise des interactions entre éléments du système</
           description>
18     <objets>Rôle, ligne de vie, messages.</objets>
19   </diagramme>
20   <diagramme UML1='yes' UML2='yes'>
21     <name>Diagramme d'activité</name>
22     <description>Modélise une activité</description>
23     <objets>Noeuds, activité, action, partitions.</objets>
```

Listing 3 – Extrait d'un simple fichier XML (listeUML.xml)

Feuille de style XSLT

Une feuille de style (*stylesheet*) XSLT est un fichier XML (puisque XSLT est un dialecte XML).

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3
4    <xsl:template match="/">
5      <html xmlns="http://www.w3.org/1999/xhtml">
6          <body>
7            <h1>Hello world !</h1>
8          </body>
9       </html>
10   </xsl:template>
11
12   </xsl:stylesheet>
```

Listing 4 – Un simple fichier XSL (listeUMLHelloWorld.xsl) contenant une seule règle de transformation

Lignes 4 à 12 : Règle de transformation XSL (*template rule*)

Test

1. Ligne 2 : vous me reconnaissez ?
2. À quel(s) élément(s) est appliqué le namespace XSLT ?

Feuille de style XSLT (suie)

Ensemble de règles de transformation (*template rules*).

- Une règle est composée de :
  - une expression XPath appelée *pattern* identifiant les éléments que la règle traite
  - un fragment de code XML (bien formé) appelé *corps* créé lorsque la règle est instanciée.
- La règle est instanciée lorsque des éléments correspondant au pattern sont trouvés dans le fichier XML à transformer.

Syntaxiquement, une règle est décrite par un élément `xsl:template` contenant le *corps* entre son tag ouvrant et son tag fermant, et ayant un attribut `match` dont la valeur est le pattern.

Test

Étant donnée cette description, écrivez la forme d'une règle ayant pour pattern `req_XPath` et pour corps `Well-formed_XML_fragment`.

Feuille de style XSLT (suie) (suite)

Intuition du mécanisme de base de l'exécution

Initialement, un processeur XSLT cherche à appliquer, à la racine du fichier XML à transformer (nœud contexte initial), un template ayant le pattern /.

Pour chaque règle, étant donné un nœud *contexte* dans le document à transformer, le processeur XSLT sélectionne un ensemble de nœuds $\mathcal{N}$ résultant de l'évaluation de l'expression XPath de son pattern. Pour chaque nœud $n \in \mathcal{N}$, le processeur XSLT instancie le corps de la règle, cela l'amenant éventuellement à choisir d'autres règles à appliquer avec le nœud contexte n.

Feuille de style XSLT (suie) (suite)

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/
       Transform">
3
4    <xsl:template match="/">
5      <html xmlns="http://www.w3.org/1999/xhtml">
6        <body>
7          <h1>Hello world !</h1>
8        </body>
9      </html>
10   </xsl:template>
11
12   </xsl:stylesheet>
```

Test

- Cette feuille de style contient une règle. Quel est son pattern ? Quel est son corps ?
- À quel(s) élément(s) sont appliqués les différents espaces de noms apparaissant dans cette feuille de style ?

Associer une feuille de style à un fichier XML

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <?xml-stylesheet type="text/xsl" href="listeUMLHelloWorld.xsl" ?>
3  <diagrammesUML>
4    <diagramme UML1='yes' UML2='yes'>
5      <name>Diagramme de classes</name>
6      <description>Modélise les objets du système étudié et leurs relations.</
           description>
7      <objets>
8        Acteurs, cas d'utilisation
9      </objets>
10   </diagramme>
11   <diagramme UML1='yes' UML2='yes'>
12     <name>Diagramme de cas dutilisation</name>
13     <description>Modélise l'ensemble des actions réalisées par le système
           étudié d'un utilistaeur extérieur au système</description>
14     <objets>Classes, associations</objets>
15   </diagramme>
16   <diagramme UML1='yes' UML2='yes'>
17     <name>Diagramme de séquence</name>
18     <description>Modélise des interactions entre éléments du système</
           description>
19     <objets>Rôle, ligne de vie, messages.</objets>
20   </diagramme>
```

Listing 5 – Extrait d'un simple fichier XML (listeUML_xslcallHelloWorld.xml)

Référence à la feuille de style en ligne 2.

Transformation d'un fichier XML avec XSLT

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <html xmlns="http://www.w3.org/1999/xhtml">
3    <body>
4      <h1>Hello world !</h1>
5    </body>
6  </html>
```

Listing 6 – Fichier XHTML (xslcallHelloWorld.xhtml) généré(*) par application de la feuille de style au fichier XML

(*) Ici, généré par Xalan [?] (fichier indenté manuellement pour lisibilité)

Transformation d'un fichier XML avec XSLT (suite)

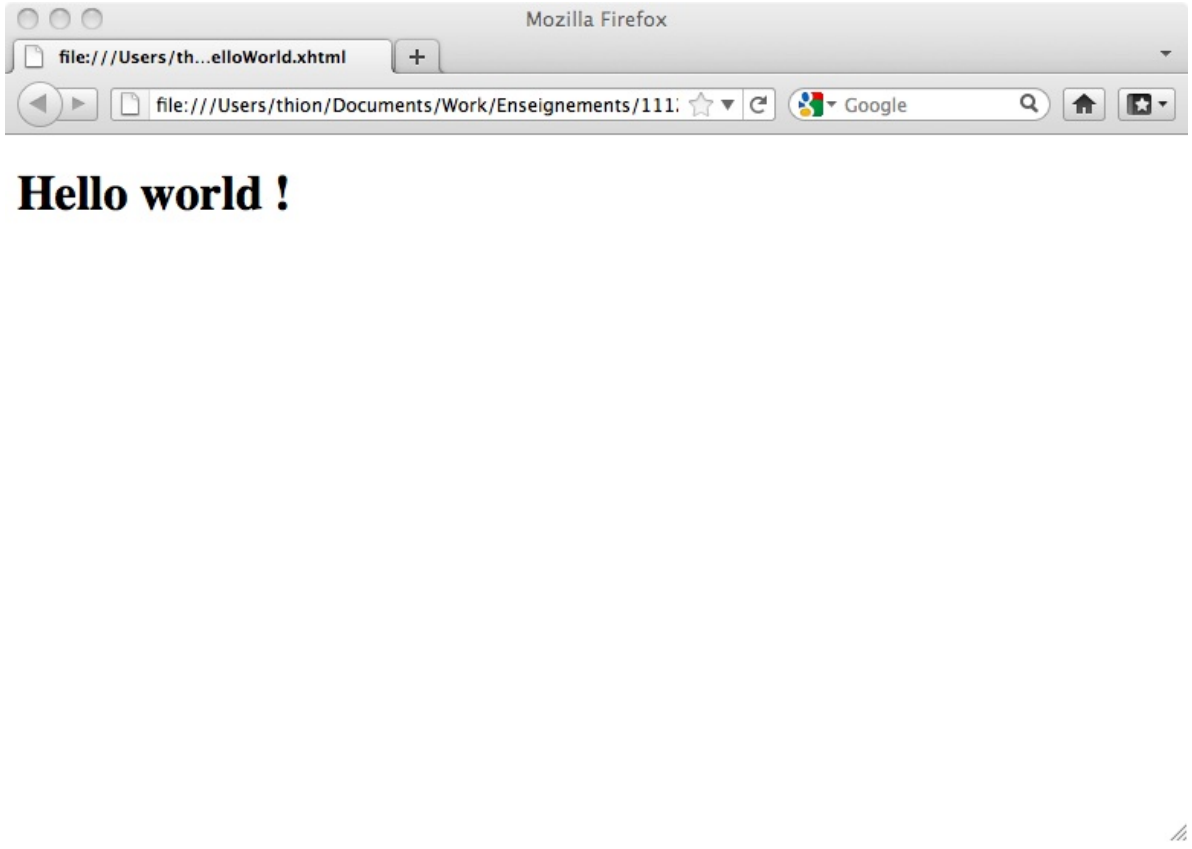


Figure 10: Ouverture du fichier Listing **??** avec Firefox (v. 6.0.1)

Transformation d'un fichier XML avec XSLT (suite)

Voici une autre feuille de style :

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/
       Transform">
3
4    <xsl:template match="/diagrammesUML">
5      <html xmlns="http://www.w3.org/1999/xhtml">
6        <body>
7          <h1>Hello world !</h1>
8        </body>
9      </html>
10   </xsl:template>
11
12 </xsl:stylesheet>
```

Test

1. Quel type de fichier cherche-t-on à générer par cette transformation ?
2. Quel est le contenu du fichier obtenu par transformation du fichier XML du Listing 3 par cette feuille de style ?
3. Aurait-on pu générer un fichier SVG contenant le texte "Hello World" à la place de générer un fichier XHTML ?

Transformation d'un fichier XML avec XSLT (suite)

Voici une variante de la feuille de style précédente.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0"
3                  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4                  xmlns="http://www.w3.org/1999/xhtml">
5
6  <xsl:template match="/">
7        <html>
8          <body>
9            <h1>Hello world !</h1>
10         </body>
11       </html>
12 </xsl:template>
13
14 </xsl:stylesheet>
```

Test

1. Quelle différence voyez-vous avec la feuille de style du Listing ?? (listeUMLHelloWorld.xsl) ?
2. Voyez-vous des espaces de noms déclarés dans ce fichier XML ? Si oui, lesquels ?
3. Sachant qu'au plus un espace de nom peut être associé à un élément, indiquez pour chaque élément si un namespace lui est associé et si oui lequel.

Transformation d'un fichier XML avec XSLT

Voici le résultat du fichier généré par Xalan en appliquant la feuille de style précédente au fichier du Listing 3 (listeUML.xml).

```
1 <?xml version="1.0" encoding="UTF-8"?><html xmlns="http://www.w3.org/1999/xhtml"><body
      ><h1>Hello world !</h1></body></html>
```

Listing 7 – Fichier XHTML (listeUML_xslcallHelloWorld3.xhtml) généré (par Xalan) par application de la feuille de style au fichier XML

Transformation d'un fichier XML avec XSLT (suite)

Pour forcer l'indentation du fichier de sortie (et éventuellement changer l'encodage de celui-ci) (ligne 6).

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0"
3                  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4                  xmlns="http://www.w3.org/1999/xhtml">
5
6  <xsl:output method="xml" indent="yes" encoding="ISO-8859-1"/>
7
8    <xsl:template match="/">
9        <html>
10         <body>
11           <h1>Hello world !</h1>
12         </body>
13       </html>
14     </xsl:template>
15
16   </xsl:stylesheet>
```

Transformation d'un fichier XML avec XSLT (suite)

```
1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <body>
4  <h1>Hello world !</h1>
5  </body>
6  </html>
```

Listing 8 – Fichier XHTML (listeUML_xslcallHelloWorld4.xhtml) généré (par Xalan) par application de la feuille de style au fichier XML

Quelques éléments de XSLT

Voici une feuille de style un peu plus complexe.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org
       /1999/xhtml">
3
4    <xsl:output method="xml" indent="yes" encoding="UTF-8"/>
5
6    <xsl:template match="/diagrammesUML">
7      <html>
8        <body>
9          <ul>
10           <xsl:for-each select="diagramme">
11             <li><b><xsl:value-of select="name"/>:</b><xsl:value-of select="description"/></li>
12           </xsl:for-each>
13         </ul>
14       </body>
15     </html>
16   </xsl:template>
17
18 </xsl:stylesheet>
```

Listing 9 – Fichier XSLT listeUMLAfficheListe.xsl

Quelques éléments de XSLT (suite)

Test

Sachant que

- `<xsl:for-each select = XPath_expr> </xsl:for-each>` contient un code instancié pour chaque nœud résultat de l'expression XPath `XPath_expr` évaluée dans le contexte courant.
- `<xsl:value-of select = XPath_expr />` évalue l'expression `XPath_expr` et convertit son résultat en chaîne de caractères.

1. Quel est le résultat de l'application de cette feuille de style au fichier du Listing 3 (listeUML.xml) ?
2. La ligne 5 devient `<xsl:template match="/">`. Modifier la suite de la feuille de style afin que celle-ci puisse produire le même fichier XHTML en sortie.

Quelques éléments de XSLT (suite)

Réponse à 1.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <body>
4  <ul>
5    <li>
6    <b>Diagramme de classes:</b>Modélise les objets du système étudié et leurs relations.</li>
7    <li>
8    <b>Diagramme de cas dutilisation:</b>Modélise l'ensemble des actions réalisées par le système
           étudié d'un utilisateur extérieur au système</li>
9    <li>
10   <b>Diagramme de séquence:</b>Modélise des interactions entre éléments du système</li>
11   <li>
12   <b>Diagramme d'activité:</b>Modélise une activité</li>
13   <li>
14   <b>Diagramme d'états-transitions:</b>Modélise le comportement d'un objet</li>
15   <li>
16   <b>Diagramme de communication:</b>Modélise les interactions (messages échangés) entre objets
           </li>
17   <li>
18   <b>Diagramme de composants:</b>Montre l'architecture physique d'une application en terme de
           modules : fichiers sources, exécutables, librairies,etc. </li>
19   <li>
20   <b>Diagramme de déploiement:</b>Montre la configuration physique des différents matériels
           nécessaires à lexécution du système, ainsi que des artefacts quils supportent.</li>
```

Listing 10 – Extrait du fichier XHTML (listeUML_xslcall_AfficheListe.xhtml) généré (par Xalan) par application de la feuille de style au fichier XML

Quelques éléments de XSLT (suite)

Réponse à 1.

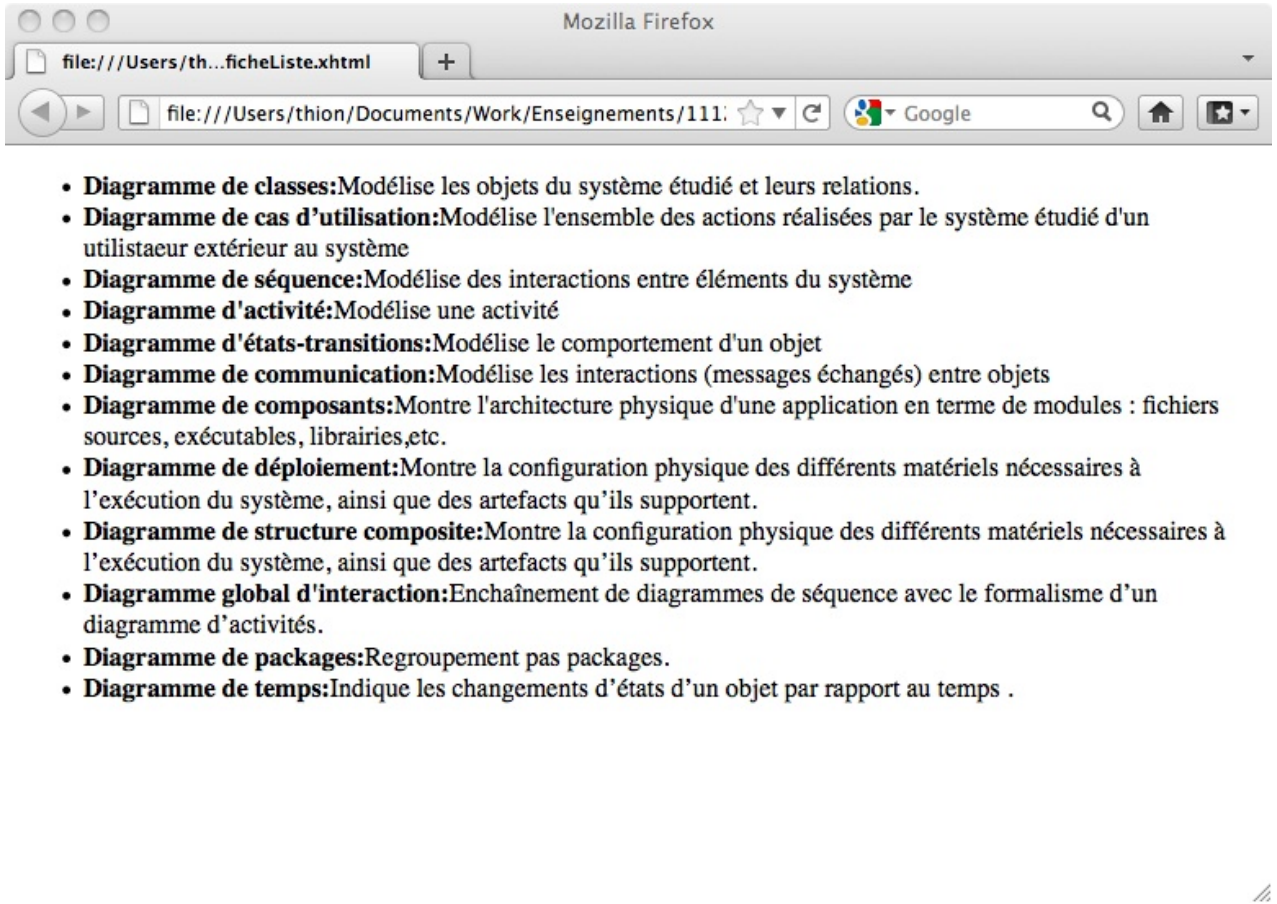


Figure 11: Ouverture du fichier avec Firefox

Quelques éléments de XSLT (suite)

Réponse à 2.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://
       www.w3.org/1999/xhtml">
3
4    <xsl:output method="xhtml" indent="yes" encoding="UTF-8"/>
5
6    <xsl:template match="/">
7      <html>
8        <body>
9          <ul>
10           <xsl:for-each select="diagrammesUML/diagramme">
11             <li><b><xsl:value-of select="name"/>:</b><xsl:value-of select="description"/></li
                   >
12           </xsl:for-each>
13         </ul>
14       </body>
15     </html>
16   </xsl:template>
17
18 </xsl:stylesheet>
```

Listing 11 – Fichier XSLT (listeUMLAfficheListeBis.xsl)

Quelques éléments de XSLT (suite)

Une autre feuille de style pour un résultat équivalent

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org
       /1999/xhtml">
3
4    <xsl:output method="xml" indent="yes" encoding="UTF-8"/>
5
6    <xsl:template match="diagrammesUML">
7      <html>
8        <body>
9          <ul>
10           <xsl:for-each select="diagramme">
11             <xsl:apply-templates select="."/>
12           </xsl:for-each>
13         </ul>
14       </body>
15     </html>
16   </xsl:template>
17
18   <xsl:template match="diagramme">
19     <li>
20       <b><xsl:value-of select="name"/>:</b><xsl:value-of select="description"/>
21     </li>
22   </xsl:template>
23
24 </xsl:stylesheet>
```

Listing 12 – Fichier XSLT listeUMLAfficheListe2.xsl

Quelques éléments de XSLT (suite)

Test

1. Écrire une feuille de style équivalente à la précédente en ajoutant une règle pour le pattern `"/"`.
2. Réduiser le fichier listeUML.xml à ses trois premiers diagrammes (voir listing 3). Pour chaque règle de votre nouvelle feuille de style, indiquer dans quel(s) contexte(s) la règle est instanciée si la feuille de style est appliquée au fichier XML contenant les trois diagrammes.

Rappel de l'intuition du mécanisme de base de l'exécution

Initialement, un processeur XSLT cherche à appliquer, à la racine du fichier XML à transformer (nœud *contexte* initial), un template ayant le pattern /.

Pour chaque règle, étant donné un nœud *contexte* dans le document à transformer, le processeur XSLT sélectionne un ensemble de nœuds $\mathcal{N}$ résultant de l'évaluation de l'expression XPath de son pattern. Pour chaque nœud $n \in \mathcal{N}$, le processeur XSLT instancie le corps de la règle, cela l'amenant éventuellement à choisir d'autres règles à appliquer avec le nœud contexte n.

Quelques éléments de XSLT (suite)

Encore une autre feuille de style pour un résultat équivalent

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org
       /1999/xhtml">
3
4    <xsl:output method="xml" indent="yes" encoding="UTF-8"/>
5
6    <xsl:template match="/diagrammesUML">
7      <html>
8        <body>
9          <ul>
10           <xsl:for-each select="diagramme">
11             <xsl:apply-templates select="."/>
12           </xsl:for-each>
13         </ul>
14       </body>
15     </html>
16   </xsl:template>
17
18   <xsl:template match="*">
19     <li>
20       <b><xsl:value-of select="name"/>:</b><xsl:value-of select="description"/>
21     </li>
22   </xsl:template>
23
24 </xsl:stylesheet>
```

Listing 13 – Fichier XSLT listeUMLAfficheListe3.xsl

Quelques éléments de XSLT (suite)

Test

Quel est le résultat de l'application de cette feuille de style au fichier du Listing 3 (listeUML.xml) ?

Associer différentes règles de transformation à un même ensemble d'éléments

Utilisation de l'attribut `mode` dans l'élément XSLT `template`.

```
1  <xsl:template match="diagramme" mode="short">
2    <li> <xsl:value-of select="name"/></li>
3  </xsl:template>
4
5  <xsl:template match="diagramme" mode="long">
6    <li><b><xsl:value-of select="name"/>:</b><xsl:value-of select="description"/></li
       >
7  </xsl:template>
```

Listing 14 – Extrait d'un fichier XSLT utilisant l'attribut mode dans l'élément template

XSLT

Associer différentes règles de transformation à un même ensemble d'éléments (suite)

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org
       /1999/xhtml">
3    <xsl:output method="xml" indent="yes" encoding="UTF-8"/>
4
5    <xsl:template match="diagrammesUML">
6      <html>
7        <body>
8          Liste courte
9          <ul>
10           <xsl:for-each select="diagramme">
11             <xsl:apply-templates select="." mode="short"/>
12           </xsl:for-each>
13         </ul>
14         Liste avec description
15         <ul>
16           <xsl:for-each select="diagramme">
17             <xsl:apply-templates select="." mode="long"/>
18           </xsl:for-each>
19         </ul>
20       </body>
21     </html>
22   </xsl:template>
23
24   <xsl:template match="diagramme" mode="short">
25     <li> <xsl:value-of select="name"/></li>
26   </xsl:template>
27
28   <xsl:template match="diagramme" mode="long">
29     <li><b><xsl:value-of select="name"/>:</b><xsl:value-of select="description"/></li>
30   </xsl:template>
31
32 </xsl:stylesheet>
```

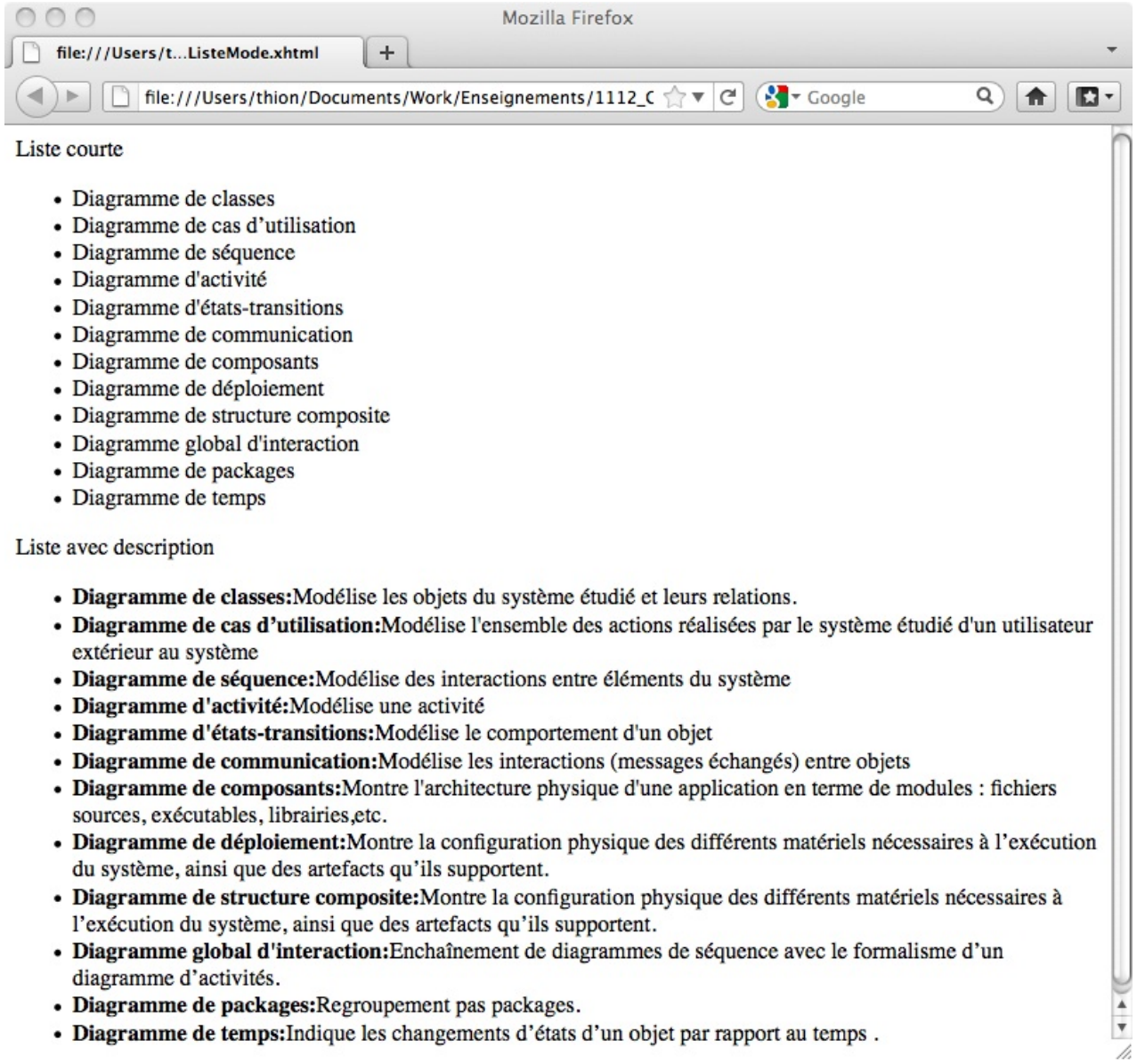
Listing 15 – Fichier XSLT listeUML.Affiche.Liste.Mode.xsl

Associer différentes règles de transformation à un même ensemble d'éléments (suite)

```
1  <?xml version="1.0" encoding="UTF-8"?><html xmlns="http://www.w3.org/1999/xhtml">
2  <body>
3  Liste courte
4  <ul>
5    <li>Diagramme de classes</li>
6    <li>Diagramme de cas dutilisation</li>
7    <li>Diagramme de séquence</li>
8    <li>Diagramme d'activité</li>
9    <li>Diagramme d'états-transitions</li>
10   <li>Diagramme de communication</li>
11   <li>Diagramme de composants</li>
12   <li>Diagramme de déploiement</li>
13   <li>Diagramme de structure composite</li>
14   <li>Diagramme global d'interaction</li>
15   <li>Diagramme de packages</li>
16   <li>Diagramme de temps</li>
17 </ul>
18 Liste avec description
19 <ul>
20   <li>
```

Listing 16 – Extrait du fichier XHTML (listeUML_xslcall_AfficheListeMode.xhtml) généré (par Xalan) par application de la feuille de style au fichier XML

Associer différentes règles de transformation à un même ensemble d'éléments (suite)



Liste courte

- Diagramme de classes
- Diagramme de cas d'utilisation
- Diagramme de séquence
- Diagramme d'activité
- Diagramme d'états-transitions
- Diagramme de communication
- Diagramme de composants
- Diagramme de déploiement
- Diagramme de structure composite
- Diagramme global d'interaction
- Diagramme de packages
- Diagramme de temps

Liste avec description

- **Diagramme de classes:**Modélise les objets du système étudié et leurs relations.
- **Diagramme de cas d'utilisation:**Modélise l'ensemble des actions réalisées par le système étudié d'un utilisateur extérieur au système
- **Diagramme de séquence:**Modélise des interactions entre éléments du système
- **Diagramme d'activité:**Modélise une activité
- **Diagramme d'états-transitions:**Modélise le comportement d'un objet
- **Diagramme de communication:**Modélise les interactions (messages échangés) entre objets
- **Diagramme de composants:**Montre l'architecture physique d'une application en terme de modules : fichiers sources, exécutables, librairies,etc.
- **Diagramme de déploiement:**Montre la configuration physique des différents matériels nécessaires à l'exécution du système, ainsi que des artefacts qu'ils supportent.
- **Diagramme de structure composite:**Montre la configuration physique des différents matériels nécessaires à l'exécution du système, ainsi que des artefacts qu'ils supportent.
- **Diagramme global d'interaction:**Enchaînement de diagrammes de séquence avec le formalisme d'un diagramme d'activités.
- **Diagramme de packages:**Regroupement pas packages.
- **Diagramme de temps:**Indique les changements d'états d'un objet par rapport au temps .

Conflit entre règles

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org
       /1999/xhtml">
3    <xsl:output method="xml" indent="yes" encoding="UTF-8"/>
4
5    <xsl:template match="/">
6      <html>
7        <body>
8          <ul>
9            <xsl:for-each select="diagrammesUML/diagramme">
10             <xsl:apply-templates select="."/>
11           </xsl:for-each>
12         </ul>
13       </body>
14     </html>
15   </xsl:template>
16
17   <xsl:template match="*">
18     <li>
19       <xsl:copy-of select="."/>
20     </li>
21   </xsl:template>
22
23   <xsl:template match="diagramme">
24     <li>
25       <b><xsl:value-of select="name"/>:</b><xsl:value-of select="description"/>
26     </li>
27   </xsl:template>
28
29 </xsl:stylesheet>
```

Listing 17 – Fichier XSLT listeUMLAfficheListeConflit.xsl

Conflit entre règles (suite)

En cas de conflit de règles

La règle la plus spécifique est appliquée.

Conflit entre règles (suite)

```
1  <?xml version="1.0" encoding="UTF-8"?><html xmlns="http://www.w3.org/1999/xhtml">
2  <body>
3  <ul>
4  <li>
5  <b>Diagramme de classes:</b>Modélise les objets du système étudié et leurs relations.</li>
6  <li>
7  <b>Diagramme de cas dutilisation:</b>Modélise l'ensemble des actions réalisées par le système
       étudié d'un utilisateur extérieur au système</li>
8  <li>
9  <b>Diagramme de séquence:</b>Modélise des interactions entre éléments du système</li>
10 <li>
11 <b>Diagramme d'activité:</b>Modélise une activité</li>
12 <li>
13 <b>Diagramme d'états-transitions:</b>Modélise le comportement d'un objet</li>
14 <li>
15 <b>Diagramme de communication:</b>Modélise les interactions (messages échangés) entre objets
       </li>
16 <li>
17 <b>Diagramme de composants:</b>Montre l'architecture physique d'une application en terme de
       modules : fichiers sources, exécutables, librairies,etc. </li>
18 <li>
19 <b>Diagramme de déploiement:</b>Montre la configuration physique des différents matériels
       nécessaires à lexécution du système, ainsi que des artefacts quils supportent.</li>
20 <li>
```

Listing 18 – Extrait du fichier XHTML (listeUML_xslcall_AfficheListeConflit.xhtml) généré (par Xalan) par application de la feuille de style au fichier XML

XSLT

On s'arrête là mais XSLT peut aller beaucoup plus loin (variables, etc).

Pour aller plus loin :

- [?] Wed Data Management and Distribution de Serge Abiteboul, Ioana Manolescu, Philippe Rigaux, Marie-Christine Rousset, et Pierre Senellart.
- [?] Comprendre XSLT (2002) de Philippe Rigaux et Bernd Amann.
- [?] XSLT : Mastering XML Transformations (2008) de Doug Tidwell.

Exercice à effectuer en cours

Voici un fichier XML (*produitsBB.xml*).

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="feuilleproduits.xsl" ?>
<Produits>
  <Magasin>
    <name>Theo et Louise</name>
    <Produit ID="7-9997-B">
      <lib>Mc Traveler - Poussette</lib>
      <Quantity>57</Quantity>
      <Marque>Jojo et Marjolaine</Marque>
    </Produit>
    <Produit ID="3-5670-C">
      <lib>Dady Mam - biberon</lib>
      <Quantity>91</Quantity>
      <Marque>Les grignottes</Marque>
    </Produit>
    <Produit ID="2-3478-A">
      <lib>BB trop confort - Table a langer</lib>
      <Quantity>27</Quantity>
      <Marque>Jojo et Marjolaine</Marque>
    </Produit>
  </Magasin>
  <Magasin>
    <name>Chez BB</name>
    <Produit ID="5-6767-A">
      <lib>BB Naturel - Set bain</lib>
      <Quantity>28</Quantity>
      <Marque>Mariette la Chouette</Marque>
    </Produit>
    <Produit ID="5-6867-A">
      <lib>BB Naturel - Set 3 bodies</lib>
      <Quantity>23</Quantity>
      <Marque>Mariette la Chouette</Marque>
    </Produit>
  </Magasin>
  <Magasin>
    <name>Autour du bebe</name>
    <Produit ID="5-6755-A">
      <lib>BB Naturel - Ensemble chaussettes antidérapantes
      <Quantity>28</Quantity>
      <Marque>Mariette la Chouette</Marque>
    </Produit>
    <Produit ID="5-6868-A">
      <lib>BB Naturel - set 3 bodies</lib>
      <Quantity>23</Quantity>
      <Marque>Jojo et Marjolaine</Marque>
    </Produit>
  </Magasin>
</Produits>
```

Exercice à effectuer en cours (suite)

Test

1. Écrire un fichier XSLT permettant d'écrire sous forme d'un fichier XHTML la liste des noms des magasins contenus dans *produitsBB.xml*.
2. Compléter ce fichier en lui permettant d'afficher, en dessous de la liste des magasins, pour chaque magasin la liste des produits vendus dans le magasin (libellé du produit et sa marque entre parenthèses).

Vous pouvez trouver une solution pour cet exercice dans le fichier *feuilleproduits.xsl* (téléchargeable sur Plei@d)