

Introduction à XSL/FO

Code: xml-xslfo

Originaux

[url: http://tecfa.unige.ch/guides/tie/html/xml-xslfo/xml-xslfo.html](http://tecfa.unige.ch/guides/tie/html/xml-xslfo/xml-xslfo.html)

[url: http://tecfa.unige.ch/guides/tie/pdf/files/xml-xslfo.pdf](http://tecfa.unige.ch/guides/tie/pdf/files/xml-xslfo.pdf)

Auteurs et version

- Daniel K. Schneider - Vivian Synteta
- Version: 0.8 (modifiée le 8/12/05 par DKS)

Prérequis

- XML de base

Module technique précédent: xml-dom

Module technique précédent: xml-tech (matière obligatoire!)

Module technique précédent: xml-xslt (transformations XSL)

Autres modules

Module technique suppl.: xml-docu (Xml et documentation)

Abstract

XSL/FO est un langage qui permet de formater l'affichage et/ou l'impression d'un document XML. Il s'agit de l'équivalent (en plus puissant) des style-sheet CSS.

Objectifs

- Formattage de base avec XSLT et XSL/FO
- Utilisation de Apache/FOP

1. Table des matières détaillée

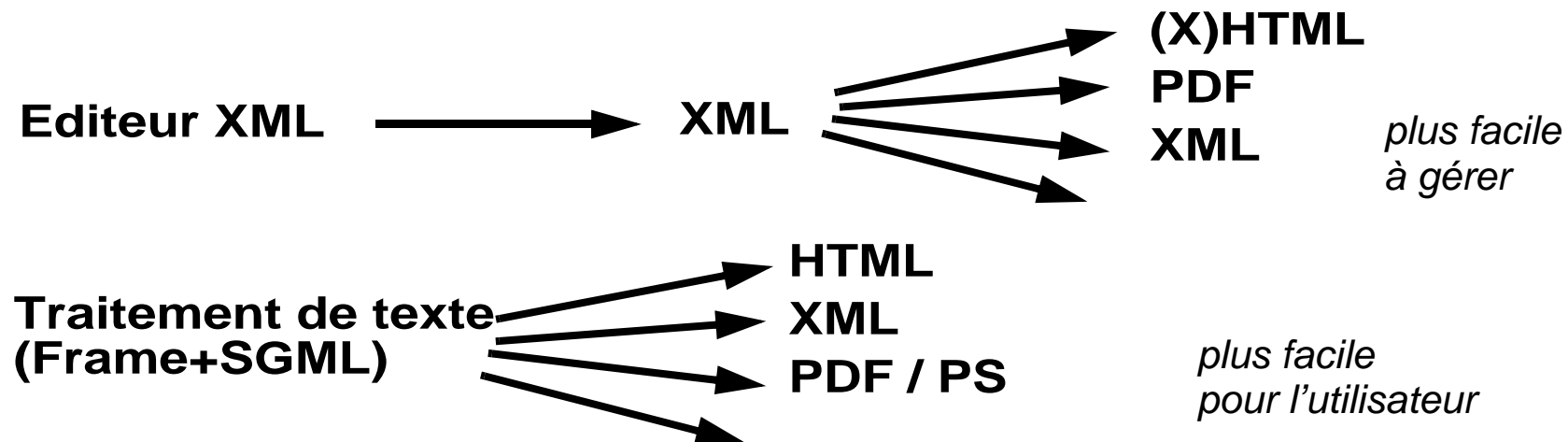
1. Table des matières détaillée.....	3
2. Le contexte: publier avec XML	4
2.1 Edition "manuelle" de XML	4
2.2 Langages de base pour systèmes de "textes" on-line	5
2.3 Exemple d'une chaîne de production "Web publishing"	6
3. Principes de base XSL/FO	7
3.1 Organisation d'une "page" FO	8
Exemple 3-1: Un simple stylesheet XSLT + FO	8
4. Pagination	14
Exemple 4-1: Pagination	14
5. Formattage des blocs et éléments spéciaux	18
Exemple 5-1: Formattage CSS vs. XSL	18
5.1 Listes	19
Exemple 5-2: Une simple "bullet-list"	20
5.2 Tables	21
Exemple 5-3: Table à deux colonnes fixes	22
6. FOP en batch	23
6.1 Installation	23
Exemple 6-1: Exemple d'un fichier fop.bat à placer dans c:\bin de Windos	23
6.2 Arguments de l'application FOP	24

2. Le contexte: publier avec XML

2.1 Edition "manuelle" de XML

- Outils permettant d'éditer un "arbre" (quelques programmes Java gratuits)
- Outils d'édition de texte structuré (éditeurs de programmation comme Emacs)
- Outils semi-professionnels (comme XMetal ou EpcEdit): assez chers.
- Outils professionnels SGML/XML comme FrameMaker+SGML: chers.)
- Plug-ins pour traitement de texte (médiocres encore)
- Filtres vers XML (HTML, RTF, Latex, etc.): médiocres par nature

2 grandes options:

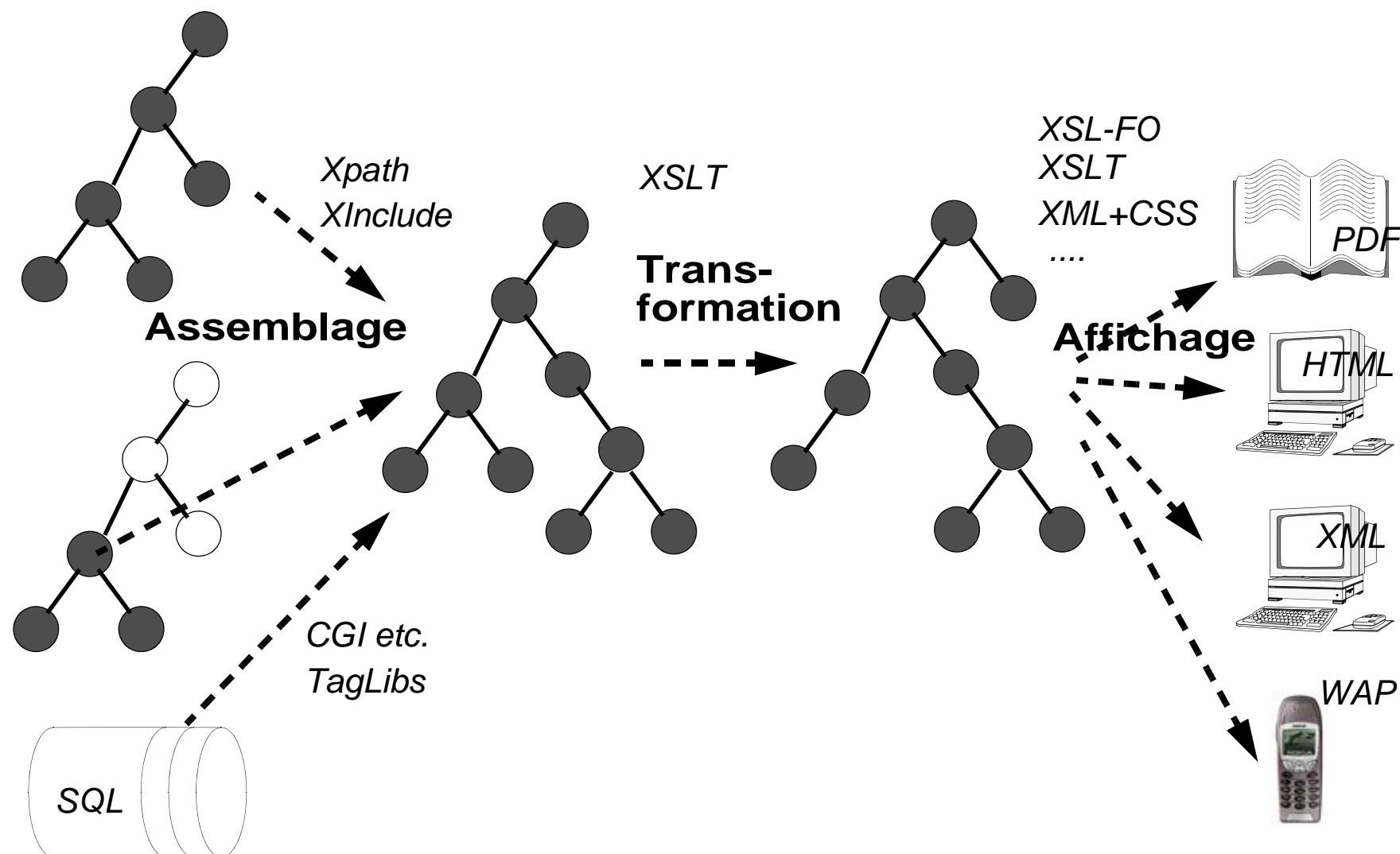


2.2 Langages de base pour systèmes de "textes" on-line

- Markup: Langage pour caractériser des éléments d'information
- Style: Langage pour définir la mise en page d'une classe d'objets
- Linking: Langage pour représenter des liens entre éléments et objets
- Assemblage: Assembler du texte à partir de fragments d'autres textes
- Scripting: Interface et langages pour créer des applications client-side

	monde HTML	monde XML	monde SGML
Linking	(<A> Tag dans HTML)	Xlink (+ Xpointer & Xpath)	HyTime & TEI
Assemblage	"calculs server-side"	XInclude (+ Xpointer & Xpath) ou entités ou "calculs server-side"	Entités SGML
Style	CSS2	XSL (CSS)	DSSL
	CSS1		
Markup	HTML	applications XML (XHTML, Docbook)	applications SGML (Docbook, TEI,)
Multimédia	formats "exotiques" (Flash, Gif, Jpeg)	formalismes XML (SVG, SMIL, MathML)	
Interface entre Markup et Scripting	Document Object Model (DOM)		
Scripting	Javascript, JScript, ECMAScript,		

2.3 Exemple d'une chaîne de production "Web publishing"



3. Principes de base XSL/FO

Spécification de XSL

[url: http://www.w3.org/TR/xsl/](http://www.w3.org/TR/xsl/)

But

- Qualité d'affichage de haut niveau (équivalent à celle d'un bon traitement de texte)
- Adaptation aux média (browser, imprimante, ...)
- Multi-culturel

Usages

1. XML + XSLT => XML + XSL/FO -> format imprimable / affichable
 - ici on ne présente que cette solution
2. XSL/FO -> format pour affichage / impression

Statut

- XSL/FO est une recommandation du W3C depuis 2001
- XSL-FO ne marche pas encore dans les navigateurs habituels (mais il existe de viewers)
- Il existe plusieurs outils "server-side" (Cocoon, Axkit) qui incluent un processeur FO.
- Des éditeurs XML comme Oxygen peuvent lancer des processeurs FO comme FOP
[url: http://xmlgraphics.apache.org/fop/](http://xmlgraphics.apache.org/fop/) (**Processur XSL/FO de Apache**)
- Note: XSL/FO = XSL-FO = XSL = FO (en tout cas pour moi)

3.1 Organisation d'une "page" FO

Exemple 3-1: Un simple stylesheet XSLT + FO

[url: http://tecfa.unige.ch/guides/xml/examples/xslfo/hello-page.xml](http://tecfa.unige.ch/guides/xml/examples/xslfo/hello-page.xml)

[url: http://tecfa.unige.ch/guides/xml/examples/xslfo/hello-page.xml.text](http://tecfa.unige.ch/guides/xml/examples/xslfo/hello-page.xml.text)

[url: http://tecfa.unige.ch/guides/xml/examples/xslfo/hello-page-xslfo.xsl](http://tecfa.unige.ch/guides/xml/examples/xslfo/hello-page-xslfo.xsl)

- Note: L'exemple ci-dessus est composé d'un fichier XML et d'un fichier XSLT/XSL-FO et il a été traité "manuellement" avec un processeur XSL/FO

A. Source XML

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<page>
  <title>Hello XSL-FO Fiends</title>
  <content>
    Here is some totally cool content. It demonstrates that content can be
    rendered in PDF.
  </content>
  <content>
    Here is some more content.
    It is slightly uninteresting. It is totally uninteresting. It is totally
    uninteresting. It is totally uninteresting. It is totally uninteresting. It
    is totally uninteresting. Don't smile .... :)
  </content>
  <comment>Written by DKS/Tecfa 6/2001 - Revised 12/2005</comment>
</page>
```


B. <Feuille de style XSL/FO

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format"
  version="1.0" >

<!-- rule for the whole document: root element is page -->

<xsl:template match="page">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <fo:layout-master-set>

      <!-- Definition of a single master page. It is simple (no headers etc.) -->
      <fo:simple-page-master
        master-name="first"
        margin-left="2cm" margin-right="2cm"
        margin-bottom="0.5cm" margin-top="0.75cm"
        page-width="21cm" page-height="29.7cm"
        >

        <!-- required element body -->
        <fo:region-body />
      </fo:simple-page-master>
    </fo:layout-master-set>
  </fo:root>
</xsl:template>

```

```
<!-- Definition of a page sequence -->
<fo:page-sequence master-reference="first">
  <fo:flow flow-name="xsl-region-body" font-size="14pt" line-height="14pt">
    <xsl:apply-templates/>
  </fo:flow>
</fo:page-sequence>
</fo:root>
</xsl:template>

<!-- A series of XSLT rules that produce fo:blocks to be inserted above -->

<xsl:template match="page/title">
  <fo:block font-size="36pt" text-align="center" line-height="40pt" space-
before="0.5cm" space-after="1.0cm">
    <xsl:apply-templates/></fo:block>
  </xsl:template>

<xsl:template match="content">
  <fo:block text-align="justify" space-before="0.5cm">
    <xsl:apply-templates/></fo:block>
  </xsl:template>

<xsl:template match="comment">
  <fo:block font-size="12pt" text-align="start" space-before="0.7cm" font-
style="italic">
    <xsl:apply-templates/></fo:block>
  </xsl:template>

</xsl:stylesheet>
```

C. Voici la squelette XSLT vers XSL-FO de base

```
<xsl:template match="page">
  <fo:root>
    <fo:layout-master-set>

      <!-- Definition of a single master page. It is simple (no headers etc.) -->
      <fo:simple-page-master
        master-name="first"  >
        <!-- required element body -->
        <fo:region-body/>
      </fo:simple-page-master>
    </fo:layout-master-set>

    <!-- Definition of a page sequence -->
    <fo:page-sequence master-reference="first">
      <fo:flow flow-name="xsl-region-body" font-size="14pt" line-height="14pt">
        <xsl:apply-templates/>
      </fo:flow>
    </fo:page-sequence>
  </fo:root>
</xsl:template>
```

Une règle XSLT

- qui définit la racine FO pour la racine XML
- qui définit au moins un "master-page", c.à.d. un possible layout
- qui définit les séquences des pages et l'usage des master pages.
- et qui lance les autres règles

Voici le résultat XML+XSL/FO qu'on ne consulte pas d'habitude

- Un processeur FOP fait xml+xslt -> xml+xsl/fo -> PDF

```
<?xml version="1.0" encoding="UTF-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="first" margin-left="2cm" margin-right="2cm"
      margin-bottom="0.5cm" margin-top="0.75cm" page-width="21cm" page-height="29.7cm">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>

  <fo:page-sequence master-name="first">
    <fo:flow flow-name="xsl-region-body" font-size="14pt" line-height="14pt">
      <fo:block font-size="36pt" text-align="center" line-height="40pt" space-
before="0.5cm" space-after="1.0cm">Hello Apache/FOP and Apache/Cocoon friends</fo:block>
      <fo:block text-align="justify" space-before="0.5cm">
Here is some content. It should work with FOP 0.18 (and hopefully above).
It is totally uninteresting. It is totally uninteresting. It is totally uninteresting.
It is totally uninteresting. It is totally uninteresting.... :)
      </fo:block>
      <fo:block text-align="justify" space-before="0.5cm">
Here is some more content.
It is slightly uninteresting. It is totally uninteresting. It is totally uninteresting.
It is totally uninteresting. Don't smile .... :)
      </fo:block>
      <fo:block font-size="12pt" text-align="start" space-before="0.7cm" font-
style="italic">Written by DKS/Tecfa 6/01</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

Architecture de la racine FO:

Sous la racine fo:root il y a toujours d'abord les "réglages"

1. un `fo:layout-master-set` qui définit
2. un ou plusieurs page layouts définis avec `fo:simple-page-master`
3. des déclarations à option `fo:declarations`

Ensuite on trouve:

- une ou plusieurs `fo:page-sequences` qui contiennent du texte et des instructions de formatage.

4. Pagination

Exemple 4-1: Pagination

[url: http://tecfa.unige.ch/tecfa/teaching/formcont/webmaster2005/programme/](http://tecfa.unige.ch/tecfa/teaching/formcont/webmaster2005/programme/)

[url: http://tecfa.unige.ch/tecfa/teaching/formcont/webmaster2005/programme/programme.xml](http://tecfa.unige.ch/tecfa/teaching/formcont/webmaster2005/programme/programme.xml)

- Faire view source pour voir le XML

[url: http://tecfa.unige.ch/tecfa/teaching/formcont/webmaster2005/programme/programme-fo.xsl](http://tecfa.unige.ch/tecfa/teaching/formcont/webmaster2005/programme/programme-fo.xsl)

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format">

<xsl:template match="course">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <fo:layout-master-set>

      <fo:simple-page-master master-name="first"
        page-height="29.7cm"
        page-width="21cm"
        margin-top="1.5cm"
        margin-bottom="2cm"
        margin-left="2.5cm"
        margin-right="2.5cm">
        <fo:region-body margin-top="2cm"/>
        <fo:region-before extent="3cm"/>
        <fo:region-after extent="1.5cm"/>
      </fo:simple-page-master>
```

```
<fo:simple-page-master master-name="right"
  page-height="29.7cm"
  page-width="21cm"
  margin-top="1.5cm"
  margin-bottom="2cm"
  margin-left="2.5cm"
  margin-right="2.5cm">
  <fo:region-body margin-top="2cm"/>
  <fo:region-before extent="2.5cm"/>
  <fo:region-after extent="1.5cm"/>
</fo:simple-page-master>
```

```
<fo:simple-page-master master-name="left"
  page-height="29.7cm"
  page-width="21cm"
  margin-top="1.5cm"
  margin-bottom="2cm"
  margin-left="2.5cm"
  margin-right="2.5cm">
  <fo:region-body margin-top="2cm"/>
  <fo:region-before extent="2.5cm"/>
  <fo:region-after extent="1.5cm"/>
</fo:simple-page-master>
```

```
<fo:page-sequence-master master-name="run">
  <fo:repeatable-page-master-alternatives maximum-repeats="no-limit" >
    <fo:conditional-page-master-reference
      master-reference="left"
      odd-or-even="even" />
    <fo:conditional-page-master-reference
      master-reference="right"
      odd-or-even="odd" />
    <fo:conditional-page-master-reference
      master-reference="title"
      />
  </fo:repeatable-page-master-alternatives>
</fo:page-sequence-master>

</fo:layout-master-set>
<!-- end: defines page layout -->
```



```
<!-- actual layout -->

<fo:page-sequence master-reference="run" initial-page-number="1">

  <fo:static-content flow-name="xsl-region-before">
    <fo:block text-align="end" font-size="10pt" font-family="serif" line-
height="11pt" color="red" > Programme - <fo:page-number/> </fo:block>
  </fo:static-content>

  <fo:flow flow-name="xsl-region-body" font-size="10pt" line-height="11pt">

    <fo:block color="red" font-size="30pt" line-height="32pt" text-
align="center" space-before.optimum="22pt"> <xsl:value-of select="course-
title"/></fo:block>

    <fo:block font-size="10pt" line-height="11pt" text-align="center" space-
before.optimum="14pt">
      Attention: Il s'agit ici d'un fichier généré automatiquement depuis XML
    </fo:block>

  <!-- ici sont appelées les règles pour le reste .... -->
  <xsl:apply-templates/>

</fo:flow>
</fo:page-sequence>

</fo:root>
</xsl:template>
```

5. Formattage des blocs et éléments spéciaux

- Le principe ressemble assez fortement à CSS (sauf qu'on utilise une syntaxe XML)
- Note: certains attributs CSS deviennent des éléments XSL ! (listes par exemple)

Exemple 5-1: Formattage CSS vs. XSL

CSS:

```
page > title {  
    display: block;  
    text-align: center; line-height: 40pt; ....  
}
```

XSLT + XSL/FO:

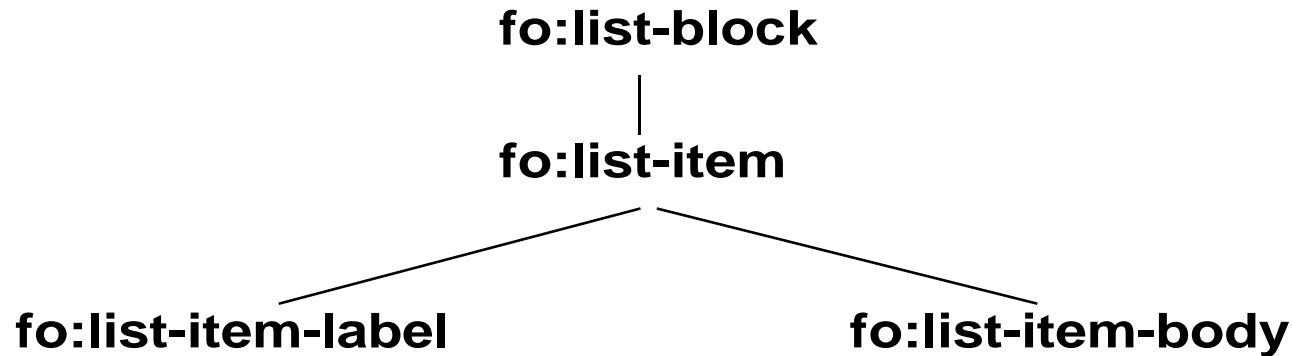
```
<xsl:template match="page/title">  
    <fo:block font-size="36pt" text-align="center"  
        line-height="40pt" space-before="0.5cm"  
        space-after="1.0cm">  
    <xsl:apply-templates/></fo:block>  
</xsl:template>
```

XSL/FO simple:

```
<fo:block font-size="36pt" text-align="center"  
    line-height="40pt" space-before="0.5cm"  
    space-after="1.0cm">  
Hello Apache/FOP and Apache/Cocoon friends  
</fo:block>
```

5.1 Listes

- Mécanisme puissant: listes ordinaires, notes en bas de page, simples tables, etc.



- **fo:list-block**: contient la liste et contient quelques définitions pour la géométrie
- **fo:list-item**: enfants de **fo:list-block**, c.a.d. des items qui contiennent un label et un body (voir ci-dessous)
- **fo: fo:list-item-label**: contient le contenu du label (typiquement un **fo:block** avec un nombre, un caractère dingbat, etc.)
- The **fo:list-item-body** contient le corps d'un item, un ou plusieurs **fo:block**

Exemple 5-2: Une simple "bullet-list"

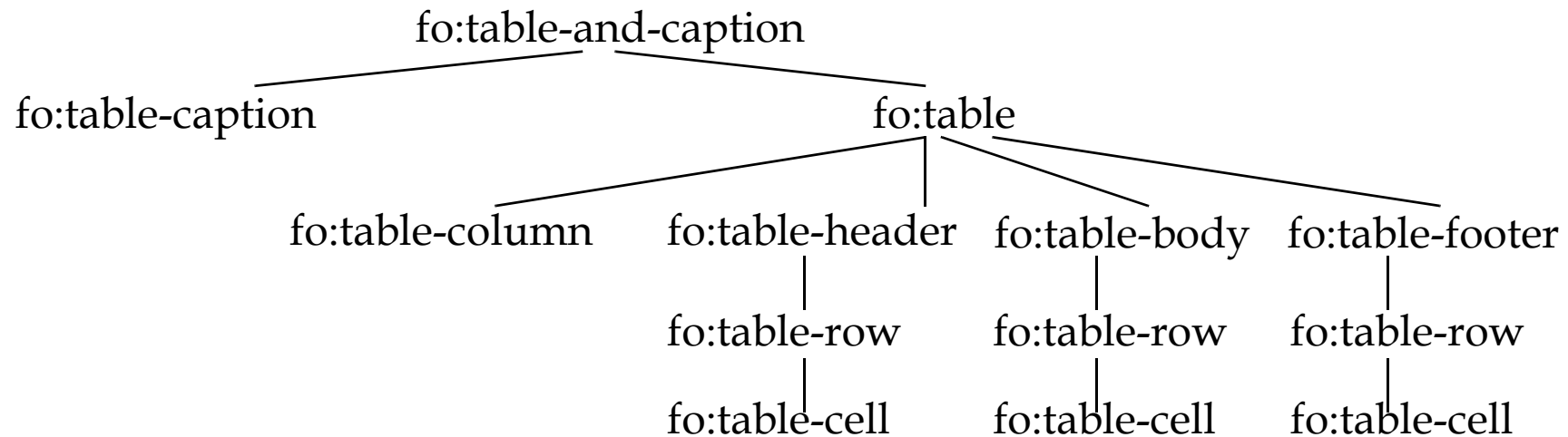
```
<xsl:template match="session-materials">

  Matériaux de cours:
  <fo:list-block space-before.optimum="6pt">
    <xsl:apply-templates select="session-material"/>
  </fo:list-block>
</xsl:template>

<xsl:template match="session-material">
  <fo:list-item space-before.optimum="8pt">
    <fo:list-item-label end-indent="label-end()">
      <fo:block>&#x2022;</fo:block>
    </fo:list-item-label>
    <fo:list-item-body start-indent="body-start()">
      <fo:block text-align="justify">
        <xsl:value-of select="@material-title"/>
      </fo:block>
      <fo:block text-align="justify" font-size="10pt">
        (URL:<xsl:value-of select="@material-url"/>)
        <xsl:value-of select="@material-comment"/>
      </fo:block>
    </fo:list-item-body>
  </fo:list-item>
</xsl:template>
```

5.2 Tables

- ressemblent un peu aux tables HTML



- `fo:table-and-caption`:
- `fo:table-caption`: La caption d'une table
- `fo:table`: la table proprement dite. Elle contient header et footer à option et un body.
- `fo:table-column`: permet de spécifier notamment la longueur d'une colonne
- `fo:table-header`: Ligne entête, contient des lignes ou cellules
- `fo:table-footer`: Ligne "footer", contient des lignes ou cellules
- `fo:table-body`: contient des lignes ou cellules
- `fo:table-row`: contient des cellules qui contiennent des `fo:blocks`

Exemple 5-3: Table à deux colonnes fixes

```

<xsl:template match="session">
  <fo:table space-before.optimum="6pt" text-align="center">
    <fo:table-column column-width="3cm"/> <fo:table-column column-width="13cm"/>
    <fo:table-body>
      <fo:table-row space-before.optimum="6pt">
        <fo:table-cell>
          <fo:block font-size="12pt" text-align="start" space-
before.optimum="5pt"><xsl:value-of select="session-date/@session-day"/><xsl:value-of
select="session-date/@session-month"/><xsl:value-of select="session-date/@session-
year"/></fo:block>
          <fo:block font-size="12pt" text-align="start" space-
before.optimum="5pt"><xsl:value-of select="session-component[position()=1]/starthour"/>
-
          <xsl:value-of select="session-component[position()=last()]/endhour"/>    </
fo:block>
        </fo:table-cell>
        <fo:table-cell>
          <fo:block color="olive" font-size="12pt" text-align="start" space-
before.optimum="5pt"><xsl:value-of select="session-title"/>
          </fo:block>
          <fo:block font-size="12pt" text-align="start" space-
before.optimum="5pt"><xsl:apply-templates select="session-component"/>
          </fo:block>
        </fo:table-cell>
      </fo:table-row>
    </fo:table-body> </fo:table>

```

6. FOP en batch

6.1 Installation

- Cf les transparents xml-xslt ! (on vous explique à quoi sert un fichier *.bat etc.)
- Il faut installer un Java 1.4 ou 1.5 (si ce n'est pas déjà fait).
- Décompresser l'archive Apache FOP binaire qq part dans votre système. Ensuite il faut chercher dans le répertoire où se trouvent les archives ".jar utilisées ci-dessous.
- Pour la version 0.20.8 il faut faire/installer un script pour lancer l'application FOP qui se trouve dans la classe Java `org.apache.fop.apps.Fop`

Exemple 6-1: Exemple d'un fichier fop.bat à placer dans c:\bin de Windos

```
rem @ECHO OFF
set LOCALCLASSPATH=c:\bin\fop.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:\bin\xml-apis.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:\bin\xercesImpl-2.2.1.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:\bin\xalan-2.4.1.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:\bin\batik.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:\bin\avalon-framework-cvs-20020806.jar
set LOCALCLASSPATH=%LOCALCLASSPATH%;c:\bin\jimi.jar
REM set LOCALCLASSPATH=%LOCALCLASSPATH%;c:\bin\jai_core.jar
REM set LOCALCLASSPATH=%LOCALCLASSPATH%;c:\bin\jai_codec.jar
java -cp "%LOCALCLASSPATH%" org.apache.fop.apps.Fop %1 %2 %3 %4 %5 %6 %7 %8
```

- Note: c:\bin doit se trouver dans le path ! Dans la distribution il y a un fichier.bat à adapter.

Utilisation typique (également sur Solaris à TECFA):

- `fop -xml fichier.xml -xsl fichier.xsl output.pdf`
- Si cela vous fait peur, achetez un éditeur XML comme Oxygen où on effectue cette opération avec un menu (mêmes outils en arrière-plan)

6.2 Arguments de l'application FOP

tapez le nom de votre fichier de commandes (par ex. fop.cmd) dans un terminal

USAGE

```
Fop [options] [-fo|-xml] infile [-xsl file] [-awt|-pdf|-mif|-pcl|-ps|-txt|-at|-print]
<outfile>
```

[OPTIONS]

```
-d          debug mode
-x          dump configuration settings
-q          quiet mode
-c cfg.xml  use additional configuration file cfg.xml
-l lang     the language to use for user information
-s          (-at output) omit tree below block areas
-txt.encoding (-txt output encoding use the encoding for the output file.
           The encoding must be a valid java encoding.
-o [password] pdf file will be encrypted with option owner password
-u [password] pdf file will be encrypted with option user password
-noprint    pdf file will be encrypted without printing permission
-nocopy     pdf file will be encrypted without copy content permission
-noedit     pdf file will be encrypted without edit content permission
-noannotations pdf file will be encrypted without edit annotation permission
```

[INPUT]

```
infile      xsl:fo input file (the same as the next)
-fo infile  xsl:fo input file
-xml infile xml input file, must be used together with -xsl
-xsl stylesheet xslt stylesheet
```


[OUTPUT]

```
outfile          input will be rendered as pdf file into outfile
-pdf outfile     input will be rendered as pdf file (outfile req'd)
-awt             input will be displayed on screen
-mif outfile     input will be rendered as mif file (outfile req'd)
-pcl outfile     input will be rendered as pcl file (outfile req'd)
-ps outfile      input will be rendered as PostScript file (outfile req'd)
-txt outfile     input will be rendered as text file (outfile req'd)
-svg outfile     input will be rendered as an svg slides file (outfile req'd)
-at outfile      representation of area tree as XML (outfile req'd)
-print          input file will be rendered and sent to the printer
               see print specific options with "-print help"
```

[Examples]

```
Fop foo.fo foo.pdf
Fop -fo foo.fo -pdf foo.pdf (does the same as the previous line)
Fop -xsl foo.xsl -xml foo.xml -pdf foo.pdf
Fop foo.fo -mif foo.mif
Fop foo.fo -print or Fop -print foo.fo
Fop foo.fo -awt
```

Exemple typique pour faire du xml + xsl -> (fo) -> pdf:

```
fop -xml fichier.xml -xsl fichier-style.xsl -pdf beau-resultat.pdf
```

Debogage !!!!!!! !!!!!

- Pour trouver des erreurs logiques dans votre feuille de style, on conseille de produire du xsl-fo avec un processeur XML et ensuite de valider...

