

*Faculté des Sciences Économiques et Sociales de l'Université de  
Fribourg (Suisse)*

## **Génération automatique d'un catalogue des produits en pdf pour le webshop *eSarine***

Travail de Master

présenté par  
Nicolas Jaquier  
rte de Fribourg 12  
1723 Marly  
nicolas.jaquier@unifr.ch

auprès de  
Dr Henrik Stormer  
*Département d'informatique  
Université de Fribourg*

Fribourg, juillet 2005

# Table des matières

<b>Glossaire</b>	<b>IV</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Processus de transformation</b>	<b>3</b>
1.1 Présentation générale . . . . .	3
1.2 Différentes approches . . . . .	3
1.2.1 ConTeXt . . . . .	3
1.2.2 PHP-FPDF . . . . .	4
1.2.3 iText . . . . .	4
1.3 Approche choisie . . . . .	4
1.3.1 Processus de transformation . . . . .	4
1.3.2 Langages et formats . . . . .	5
1.3.3 FOP . . . . .	6
1.3.4 Les concurrents de FOP . . . . .	6
<b>2 Introduction à XSL-FO</b>	<b>8</b>
2.1 Syntaxe . . . . .	8
2.2 Structure d'une page en XSL-FO . . . . .	10
2.2.1 Des pages différenciées . . . . .	12
2.3 Éléments de mise en page du contenu . . . . .	14
2.3.1 Introduction . . . . .	14
2.3.2 CSS . . . . .	15
2.3.3 Objet blocs . . . . .	17
2.3.4 Objets en-ligne . . . . .	17
2.3.5 Tables . . . . .	18
2.3.6 Objets hors-ligne . . . . .	21
2.3.7 Listes . . . . .	22
2.3.8 Graphiques . . . . .	23
2.3.9 Liens . . . . .	24
<b>3 Structure des données en entrée</b>	<b>26</b>
3.1 Données générales . . . . .	28
3.1.1 company . . . . .	28

## TABLE DES MATIÈRES

---

3.1.2	title . . . . .	28
3.1.3	editorial . . . . .	28
3.2	Les produits . . . . .	28
3.2.1	occurences . . . . .	29
<b>4</b>	<b>Feuille de style</b>	<b>31</b>
4.1	Conception . . . . .	31
4.1.1	Fonctionnement général . . . . .	31
4.1.2	Traitement des données . . . . .	32
4.2	Structure du catalogue pdf . . . . .	32
4.2.1	Titre et sommaire . . . . .	32
4.2.2	Pages contenant les produits . . . . .	33
4.3	Implémentation . . . . .	36
4.3.1	Paramètres globaux . . . . .	36
4.3.2	Variables globales . . . . .	36
4.3.3	Template Main . . . . .	36
4.3.4	Template TitlePage . . . . .	39
4.3.5	Template TOC . . . . .	40
4.3.6	Template SmallProductBlock . . . . .	41
4.3.7	Template BigProductBlock . . . . .	44
4.3.8	Templates appelés par les deux types de bloc produit . . . . .	44
4.4	Utilisation . . . . .	46
4.4.1	Généralités . . . . .	46
4.4.2	Valeurs des paramètres pour la mise en page . . . . .	46
4.4.3	Valeurs des paramètres pour les données . . . . .	47
4.5	Limites de FOP . . . . .	47
4.6	Performances . . . . .	49
	<b>Conclusion</b>	<b>50</b>
	<b>A Environnement de développement et outils</b>	<b>52</b>
A.1	Xalan-J version 2.5.1 . . . . .	52
A.2	FOP version 0.20.5 . . . . .	52
A.3	Oxygen version 5.1 . . . . .	52
A.4	TestXSLT version 3.1 . . . . .	52
	<b>B Code de la feuille de style</b>	<b>53</b>
B.1	Variables du template BigProductBlock . . . . .	53
B.2	Template Main . . . . .	53
B.3	Template TitlePage . . . . .	55
B.4	Template TOC . . . . .	56
B.5	Template SmallProductBlock . . . . .	56
B.6	Template BigProductBlock . . . . .	59
B.7	Template EmptyLinesLoop . . . . .	62

## TABLE DES MATIÈRES

---

B.8	Template DisplayTableFooterAlarm . . . . .	63
B.9	Template DisplayImage . . . . .	64
B.10	Template DisplayProductInfo . . . . .	64
B.11	Templates SetColumns. . . . .	65
<b>C</b>	<b>Exemple <i>Parlement</i>, code et rendu</b>	<b>66</b>
	<b>Table des figures</b>	<b>74</b>
	<b>Liste des tableaux</b>	<b>75</b>
	<b>Références</b>	<b>77</b>

# Glossaire

CSS	Cascading Style Sheet
FOP	Formatting Objects Processor
FPDF	FreePDF
JSP	JavaServer Pages
MVC	Model-View-Controller
PDF	Portable Document Format
PHP	Hypertext Preprocessor
RGB	Red Green Blue
SVG	Scalable Vector Graphics
W3C	World Wide Web Consortium
XML	eXtensible Markup Language
XPath	XML Path Language
XSL	eXtensible Stylesheet Language
XSL-FO	eXtended Stylesheet Language - Formating Object
XSLT	eXtended Stylesheet Language Transformations

# Introduction générale

Ce travail de Master présente une solution de production de catalogue de produits vendus sur un site de commerce électronique, *eSarine*, en s'appuyant sur les technologies XML et XSL. Ce travail s'articule en quatre parties. La première présente le processus de transformation des données ainsi que différentes solutions techniques étudiées et présente celle retenue. La seconde partie est une introduction à XSL-FO, un dialecte XML permettant de décrire la mise en forme de documents. Le chapitre suivant décrit le format du document XML à traiter pour produire le catalogue. La dernière partie présente l'implémentation de la feuille de style XSLT permettant de produire le catalogue.

## *eSarine*

*eSarine* est un webshop développé par le groupe de recherche *Information Systems* de l'Université de Fribourg dont le but est de fournir une solution complète pour les petites et moyennes entreprises (PME) pour la vente de produits sur internet. L'avantage compétitif important pour les petites et moyennes entreprises offert par le commerce électronique est la motivation principale pour le développement de ce projet [22]. De plus les coûts importants de la mise en place d'une solution de commerce électronique pour une PME sont souvent rédhibitoires. *eSarine* apporte une solution simple et peu coûteuse.

Le projet open source est développé sous la forme de modules permettant une adaptation aux besoins spécifiques de chaque utilisateur. *eSarine* permet l'utilisation de plusieurs bases de données sous-jacentes différentes. L'adaptation à des systèmes de bases de données complémentaires est facilitée par le support de la technologie JDBC et le caractère open source du webshop. *eSarine* offre également la publication d'un webshop en plusieurs langues de manière très aisée.

*eSarine* repose sur un cadre d'applications<sup>1</sup> nommé *Struts* [9] qui permet de créer des applications Java pour le web. Basé sur le modèle Model-View-Controller (MVC), *eSarine* s'appuie sur des JSPs pour ce qui concerne la

---

<sup>1</sup>[anglais] Application Framework

vue<sup>2</sup>. La version open source disponible en ligne<sup>3</sup> utilise un serveur *Tomcat*[10] pour la production des JSPs pour les interfaces client et administrateur.

## Module de production du catalogue des produits en pdf

Le but fixé de ce travail était la création d'un catalogue des produits proposés dans le webshop qui soit distribuable électroniquement et physiquement. Le travail devait pouvoir s'intégrer sans difficulté et sans requérir d'ajout de technologie à la plateforme de développement sous-jacente du projet *eSarine*.

Actuellement le format d'échange de fichiers le plus largement utilisé en ligne est le format PDF. C'est donc ce format qui a été choisi comme cadre pour la publication du catalogue des produits. Le catalogue produit peut être imprimé ou consulté électroniquement.

Le catalogue des produits est généré dynamiquement lors du clic du client sur un lien qui appelle le module développé. Le catalogue est donc toujours à jour sans travail supplémentaire que la mise à jour de la base de données par le vendeur en ligne.

Le catalogue obtenu peut être aussi bien consulté directement par le client, imprimé pour être mis à disposition des clients dans un magasin par un revendeur, qu'envoyé par e-mail ou encore mis simplement à disposition sur un site internet pour une consultation en ligne.

---

<sup>2</sup>[anglais] View

<sup>3</sup>voir le site du projet [3]

# Chapitre 1

## Processus de transformation

### 1.1 Présentation générale

Le but du travail est de transformer des données provenant d'un webshop (*eSarine*) en un catalogue imprimable sous la forme d'un document PDF. La transformation devait comporter les étapes suivantes:

1. Extraction des données
2. Traitement des données
3. Construction du document PDF

Plusieurs approches ont été étudiées avant de choisir l'environnement de développement offrant la meilleure solution compte tenu des différentes contraintes imposées par le contexte de compatibilité avec *eSarine*. Dans un premier temps ce chapitre évoque différentes approches en donnant succinctement les raisons pour lesquelles elles n'ont pas été retenues. Ensuite la solution finalement appliquée est présentée.

### 1.2 Différentes approches

Il existe plusieurs approches pour produire un document PDF. Nous allons passer en revue quelques-uns des moyens à disposition puis étudier plus avant la solution retenue dans ce travail.

#### 1.2.1 ConTeXt

Developpé par la société *PRAGMA ADE* [7] ConTeXt est un environnement basé en grande partie sur TeX qui forme un système intégré d'outils typographiques. Cette solution offre la possibilité de transformer des documents XML lors de compilation du document ConTeXt. ConTeXt produit des documents PDF de haute qualité et permet de traiter nativement les documents XML. L'utilisation de ConTeXt est libre dans l'esprit de la licence GNU.

ConTeXt ne s'intégrant pas de manière aisée au serveur Tomcat utilisé par *eSarine*, cette solution n'a pas été retenue.

### 1.2.2 PHP-FPDF

Une autre approche possible est l'utilisation de la classe PHP appelée FPDF [5] qui permet de créer des documents PDF dynamiquement sur un serveur *Apache* ayant le module PHP installé. L'un des avantages de cette solution est d'être très flexible au niveau des données en entrée. En effet, il est possible de traiter directement les données en provenance d'une base de données, d'un document XML ou d'un document dans un autre format.

Bien qu'étant l'une des solutions les plus performantes, FPDF présente, dans le cadre du webshop *eSarine*, le désavantage d'ajouter une technologie supplémentaire ce qui ne correspond pas à l'esprit d'un module se greffant à l'environnement déjà en place.

### 1.2.3 iText

iText [6] est une bibliothèque Java qui permet de produire des PDF. C'est une solution gratuite. Son avantage est de pouvoir s'intégrer facilement dans un environnement Java. Elle n'a pas été retenue car ses possibilités sont limitées au niveau de éléments représentables et son support du XML laisse à désirer pour l'instant.

## 1.3 Approche choisie

Finalement c'est une autre solution qui a été choisie pour produire le catalogue des produits du webshop *eSarine*. Elle repose sur un traitement des fichiers XML en transformant avec une feuille de style XSLT le fichier des données. Cette solution est apparue la meilleure après avoir examiné les solutions proposées aux points précédents. La solution choisie présente l'avantage d'utiliser des éléments techniques facilement intégrables dans *eSarine*, éléments décrits dans l'introduction, pour plus de détails, se référer à l'introduction.

Le choix d'un processus de transformation basé sur XML, XSL et FOP dépasse les solutions ConTeXt et PHP-FPDF au niveau de la technologie utilisée et iText par un meilleur support du XML et un meilleur support du format PDF.

### 1.3.1 Processus de transformation

La solution choisie pour la réalisation du catalogue PDF du webshop *eSarine* prend en entrée un fichier XML [voir 1.3.2] au format décrit au point 4.2 qui est ensuite traité par une feuille de style écrite en langage

de transformation XSLT [voir 1.3.2]. Le résultat de cette transformation, effectuée à l'aide de Xalan, est un nouveau fichier XML dans le format XSL-FO. Le fichier est ensuite traité par FOP [voir 1.3.3] qui permet de produire un document au format PDF [voir 1.3.2].

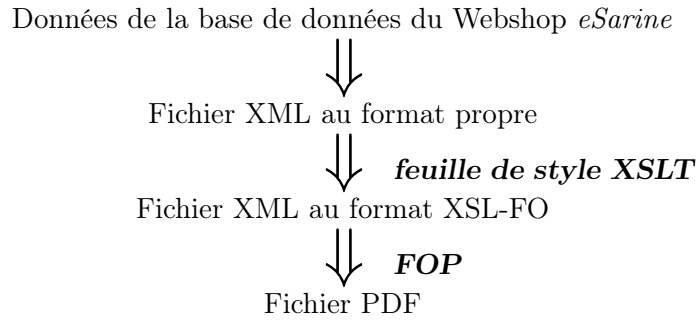


FIG. 1.1 – Processus de transformation

### 1.3.2 Langages et formats

#### XML

Ce standard établi par le W3C est un méta-langage utilisé pour décrire des données et du texte, il s'apparente à un format de données. De par sa généralisation il permet une grande souplesse dans la définition de dialectes. Sa description est disponible sous la forme d'une recommandation du consortium [17]. Tous les fichiers produits dans ce travail sont des fichiers XML à l'exception du résultat final qui est un fichier au format PDF.

#### Famille XSL

La famille des langages XSL<sup>1</sup> regroupe 3 langages permettant de définir une feuille de style pour le langage XML servant à préciser la présentation d'un document XML. Ces 3 langages sont brièvement décrits ci-après. XSL-FO fait l'objet d'une introduction plus complète au chapitre 2.

**XPath** Ce langage d'interrogation est utilisé pour la navigation dans un arbre XML à l'aide, par exemple, de prédicats. Ce langage utilise une syntaxe proche de la navigation dans un système de navigation UNIX. Il fait l'objet d'une recommandation du W3C [20].

<sup>1</sup>De manière stricte la dénomination XSL s'applique au langage dénommé ci-après XSL-FO, ce terme est utilisé par usage pour décrire la recommandation XSL.

**XSLT** XSLT est un dialecte de XML permettant d'appliquer des transformations à un document XML. Une transformation XSLT est réalisée à l'aide d'un processeur dont les deux plus connus sont *Saxon* et *Xalan* [11] qui est le processeur utilisé dans ce travail.

**XSL-FO** Le vocabulaire XSL-FO permet de décrire la mise en page d'un document XML pour différents médias. Il s'agit d'un outil typographique qui décrit des aires d'affichage et leur contenu. Un document XSL-FO est ensuite traité par un processeur comme par exemple FOP qui est décrit au point 1.3.3.

## PDF

Le format de document PDF a été créé par Adobe Systems. C'est un format ouvert dont les spécifications sont disponibles [15]. Ce format est indépendant de la plate-forme et c'est actuellement le format le plus utilisé pour l'échange de documents électroniques. Il est pour ainsi dire devenu le standard grâce à la large diffusion du lecteur de document gratuit fourni par Adobe et à l'essor des échanges électroniques.

### 1.3.3 FOP

FOP fait partie du Apache XML Graphics Project<sup>2</sup>. Il s'agit d'un processeur de documents XSL-FO qui produit un document dans de nombreux formats différents<sup>3</sup> dont principalement le format PDF. L'objectif de FOP est d'offrir un processeur de transformation de XSL-FO en PDF en assurant au moins le niveau de conformité de base de la recommandation XSL-FO version 1.0 [18].

La version de FOP utilisée lors de la réalisation de ce travail était la version 0.20.5 dont la conformité à la recommandation XSL-FO est disponible sur le site du projet [2].

FOP est écrit en Java et peut être utilisé comme une application indépendante ou intégrée à un système plus complexe sous forme, par exemple, de servlet avec Tomcat.

### 1.3.4 Les concurrents de FOP

Il existe plusieurs autres processeurs de document XSL-FO qui permettent de produire un document PDF. Nous allons passer en revue succinctement trois des plus importants. Il est relativement difficile de trouver un comparatif valable des différents outils à disposition. Un comparatif datant de mai 2004 est disponible sur le site de Antenna House [14]. Il reste

<sup>2</sup><http://xmlgraphics.apache.org/>

<sup>3</sup>PDF, PCL, PS, SVG, Print, AWT, MIF et TXT

cependant utile en montrant que XEP et XSL Formatter sont les processeurs les plus aboutis au niveau de la conformité avec la recommandation. FOP et XMLRoff sont largement en arrière à ce niveau mais cet aspect ne doit pas être le seul à prendre en compte lors de l'évaluation de ces différents produits. En effet le choix de FOP a été fait pour plusieurs raisons:

- c'est un programme opensource tout comme *eSarine* ce qui permet d'intégrer le module dans le webshop sans problème de licence,
- le prix des solutions commerciales est relativement élevé et ne convenait pas à la philosophie du projet,
- la technologie Java utilisée par FOP est compatible sans effort à l'implémentation d'*eSarine*.

### **XEP de RenderX**

Edité par la société RenderX [8], XEP est un processeur XSL-FO commercial qui offre un support étendu de la recommandation. XEP est écrit en Java et supporte plusieurs plateformes.

### **XSL Formatter de Antenna House**

Développé par Antenna House [1] ce processeur est également multi-plateformes et permet d'être appelé par plusieurs langage de programmation. Des extensions pour un formatage WYSIWYG sont également disponibles.

### **XMLRoff**

Développé initialement par SUN Microsystems, XMLRoff est écrit en C permettant de produire des documents PDF et PostScript à partir d'un document XSL-FO. Ce programme utilise les bibliothèques graphiques de GNOME pour la production de documents. Actuellement le projet est hébergé par Souceforge [12] après que son code source a été rendu public par SUN. Des différentes solutions présentées, c'est la moins aboutie.

## Chapitre 2

# Introduction à XSL-FO

L'introduction à XSL-FO contenue dans ce chapitre fait référence à la version 1.0 de la recommandation du W3C [18].

XSL-FO définit la structure typographique d'un document en se basant sur des régions. Un document est donc composé de plusieurs régions qui contiennent différentes sous-régions qui peuvent à nouveau se subdiviser. La recommandation précise la syntaxe, les éléments, leurs attributs et leurs valeurs et, pour finir, le résultat à produire au niveau conceptuel. Liberté d'implémentation est laissée quant aux méthodes pour assurer le rendu par les processeurs.

### 2.1 Syntaxe

Les éléments du vocabulaire XSL-FO font partie de l'espace de nommage<sup>1</sup> `fo:`. Chaque élément du langage sera donc préfixé<sup>2</sup> comme le montre le listing 2.1. Dans cet exemple, un nombre réduit d'éléments du vocabulaire XSL-FO sont utilisés. La recommandation en propose 56. Les attributs pouvant se rapporter à ces éléments sont un peu moins de 250 et sont dérivés pour partie des définitions de la CSS [16] comme discuté au point 2.3.2.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3   <fo:layout-master-set>
4     <fo:simple-page-master
5       master-name="page_first"
6       page-height="297mm">
```

---

<sup>1</sup>*[anglais]* namespace

Un namespace permet de différencier les différents dialectes XML utilisés dans un document.

<sup>2</sup>Il est cependant toujours possible de donner un espace de nommage par défaut ce qui rend la présence du préfixe inutile. Dans ce travail, les espaces de noms seront toujours spécifiés afin de faciliter la lecture et la compréhension

```
7     page-width="210mm"
8     margin="0mm">
9     <fo:region-body
10        region-name="xsl-region-body"
11        margin="20mm"
12        background-color="#E1E1E1" />
13 </fo:simple-page-master>
14 </fo:layout-master-set>
15 <fo:page-sequence master-reference="page_first">
16   <fo:flow flow-name="xsl-region-body">
17     <fo:block text-align="center">
18       Hello World!
19     </fo:block>
20   </fo:flow>
21 </fo:page-sequence>
22 </fo:root>
```

Listing 2.1 – Un document de base utilisant le vocabulaire XSL-FO

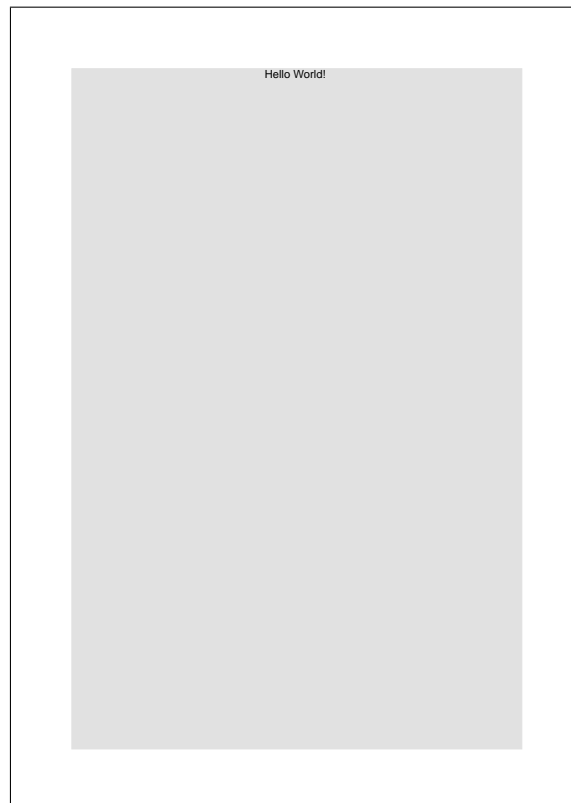


FIG. 2.1 – Résultat du listing 2.1

Un document XSL-FO est composé d'un élément racine<sup>3</sup> à l'intérieur duquel se trouvent trois types d'éléments comme le montre la figure 2.2 :

**fo:layout-master** qui contient le ou les modèles de page

**fo:page-sequence** qui précise le contenu, cet élément peut être répété selon les différents modèles de page définis.

**fo:declarations** qui permet de préciser des profils de couleurs. Cet élément est cité par souci d'exhaustivité mais ne sera pas commenté plus avant.

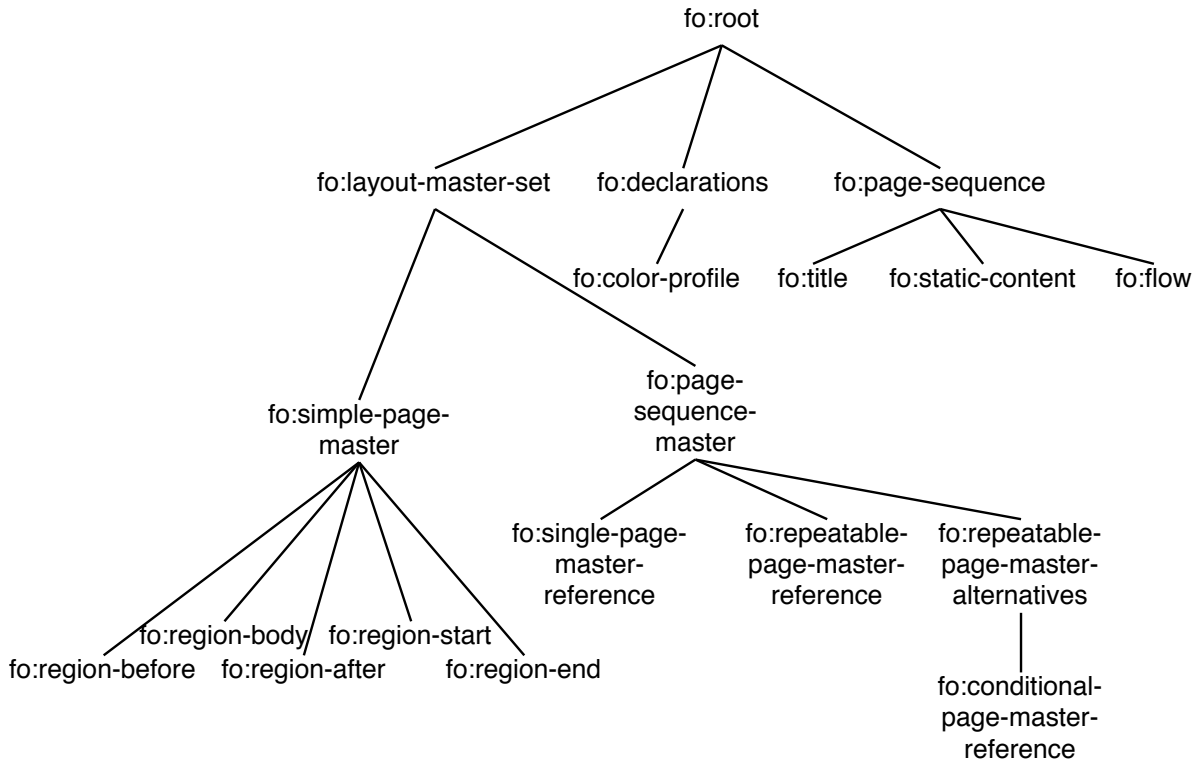


FIG. 2.2 – Structure d'une page XSL-FO sous forme d'arbre

La section 2.2 décrit la composition d'un modèle de page, la description du contenu de l'élément `fo:page-sequence` est fait dans la section 2.3.

## 2.2 Structure d'une page en XSL-FO

XSL-FO repose sur une structure de régions pour décrire une page comme le montre la figure 2.3. Une page se décompose en deux aires principales: les marges extérieures de la page qui sont représentées en blanc dans la figure

<sup>3</sup>fo:root

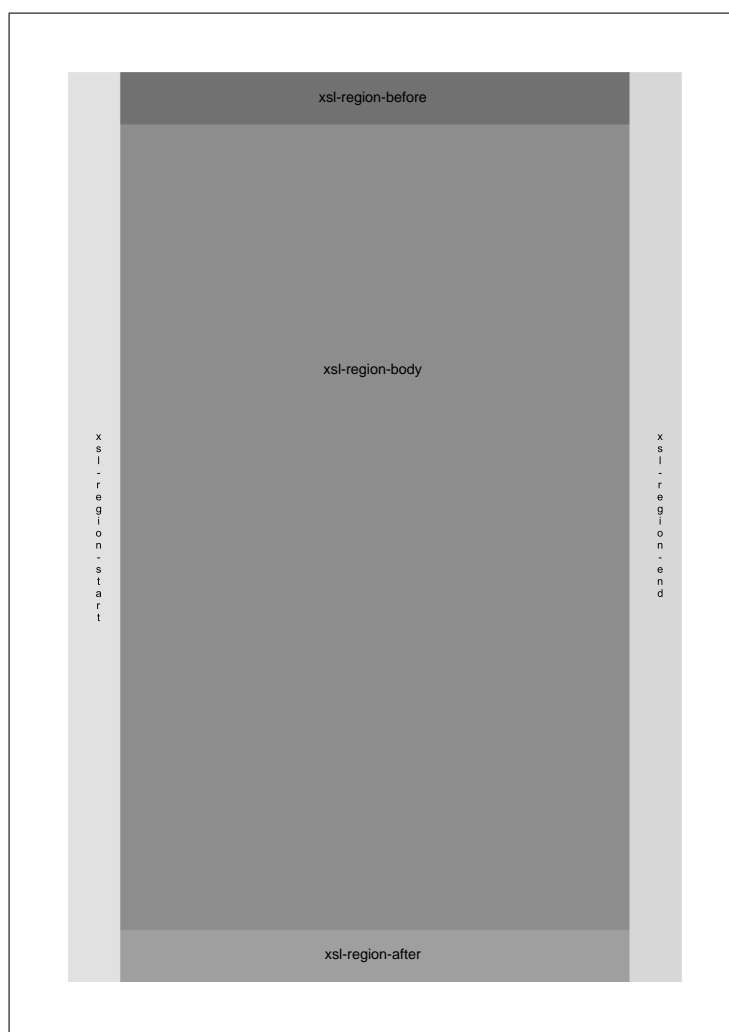


FIG. 2.3 – Structure d'une page

2.3 et le corps de la page<sup>4</sup> ( les zones grises de la figure 2.3). Cette dernière partie se subdivise ainsi:

- la région principale, nommée `xsl-region-body`
- l'en-tête, `xsl-region-before`
- le pied de page, `xsl-region-after`
- la région à gauche<sup>5</sup>, `xsl-region-start`
- la région à droite, `xsl-region-end`

<sup>4</sup>[anglais] page body

<sup>5</sup>Cette appellation est discutable car elle ne convient que dans le cas où l'écriture se fait de gauche à droite et de haut en bas. La recommandation prévoit la possibilité de définir d'autres sens d'écriture, d'où les termes *start* et *end* qui permettent une plus grande flexibilité conceptuelle.

```

3 <fo:layout-master-set>
4   <fo:simple-page-master master-name="page"
5     page-height="297mm"
6     page-width="210mm"
7     margin-left="15mm" margin-right="15mm"
8     margin-top="15mm" margin-bottom="15mm"
9     background-color="black">
10  <fo:region-body margin="15mm" background-color="#8D8D8D" />
11  <fo:region-start extent="15mm" background-color="#E1E1E1" />
12  <fo:region-end extent="15mm" background-color="#D7D7D7" />
13  <fo:region-before extent="15mm" background-color="#717171" />
14  <fo:region-after extent="15mm" background-color="#9F9F9F" />
15 </fo:simple-page-master>
16 </fo:layout-master-set>

```

Listing 2.2 – Exemple d'un élément `layout-master-set`

Un exemple de définition des régions d'une page est donné dans le listing 2.2.

Deux points importants sont à relever ici :

- Les tailles des régions `xsl-region-before`, `xsl-region-after`, `xsl-region-start`, `xsl-region-end` sont données avec l'attribut `extent` qui détermine la grandeur depuis la marge de la page.
- Les marges de la région `xsl-region-body` doivent être supérieures ou égales à la taille des éléments des quatre régions du point précédent, ceci pour éviter que ces régions ne soient recouvertes par le corps de la page.

Chaque région est ensuite composée des différents éléments de mise en page décrits dans la section 2.3.

### 2.2.1 Des pages différenciées

XSL-FO permet de créer des pages différentes selon les conditions (par exemple pour différencier la page de titre, les pages paires et les pages impaires). Ces différences sont décrites dans le jeu de mise en page `layout-master-set`. Tout d'abord les schémas de page nécessaires sont décrits comme vu dans le listing 2.2 à l'aide de modèles de page `fo:simple-page-master`. Ensuite l'élément `fo:page-sequence-master` permet de préciser le modèle de page simple à utiliser dans différentes situations :

- une page et une seule, l'élément `fo:single-page-master-reference` est utilisé
- pour obtenir plusieurs pages du même modèle c'est `fo:repeatable-page-master-reference` qui est utilisé

- pour le conditionnement du choix du modèle de page au placement de la page dans le document, un élément `fo:repeatable-page-master-alternatives` avec un fils de type `fo:conditional-page-master-reference` pour chaque situation est nécessaire. Les conditions peuvent se définir à l'aide de trois attributs: `page-position`, `odd-or-even` et `blank-or-not-blank` qui définissent les différentes pages possibles comme le montre le tableau 2.1.

Attribut	Valeurs	Description
<code>page-position</code>	<code>first</code> , <code>last</code> , <code>rest</code> ou <code>any</code>	Précise la position de la page dans le document.
<code>odd-or-even</code>	<code>odd</code> , <code>even</code> ou <code>any</code>	Différencie les pages paires et impaires.
<code>blank-or-not-blank</code>	<code>blank</code> , <code>not-blank</code> ou <code>any</code>	Sépare les pages selon qu'elles contiennent des données ou non.

TAB. 2.1 – Attributs de l'élément `fo:conditional-page-master-reference`

### Exemple

Le listing 2.3 donne un exemple des possibilités offertes par XSL-FO pour la gestion des pages. Dans cet exemple trois modèles de pages sont définis: `pageTitre`, `pagePaire` et `pageImpaire` qui ont des marges et des régions différentes selon la situation. Ces différents modèles sont appelés dans les alternatives proposées par chaque élément conditionnel aux lignes 27 à 34.

```

3 <fo:layout-master-set>
4   <fo:simple-page-master master-name="pageTitre"
5     page-height="297mm" page-width="210mm"
6     margin-left="15mm" margin-right="15mm"
7     margin-top="15mm" margin-bottom="15mm">
8     <fo:region-body margin="0mm" />
9   </fo:simple-page-master>
10  <fo:simple-page-master master-name="pagePaire"
11    page-height="297mm" page-width="210mm"
12    margin-left="25mm" margin-right="15mm"
13    margin-top="15mm" margin-bottom="15mm" >
14    <fo:region-body margin="15mm" />
15    <fo:region-start extent="15mm" /> <fo:region-end extent="15mm" />
16    <fo:region-before extent="15mm" /> <fo:region-after extent="15mm"
    />

```

```

17 </fo:simple-page-master>
18 <fo:simple-page-master master-name="pageImpaire"
19   page-height="297mm" page-width="210mm"
20   margin-left="15mm" margin-right="25mm"
21   margin-top="15mm" margin-bottom="15mm" >
22   <fo:region-body margin="15mm" />
23   <fo:region-start extent="15mm" /> <fo:region-end extent="15mm" />
24   <fo:region-before extent="15mm" /> <fo:region-after extent="15mm"
   />
25 </fo:simple-page-master>
26 <fo:page-sequence-master master-name="RectoVerso">
27   <fo:repeatable-page-master-alternatives>
28     <fo:conditional-page-master-reference master-name="Titre"
29       master-reference="pageTitre" page-position="first" />
30     <fo:conditional-page-master-reference master-name="Impaire"
31       master-reference="pageImpaire" odd-or-even="odd" />
32     <fo:conditional-page-master-reference master-name="Paire"
33       master-reference="pagePaire" odd-or-even="even" />
34   </fo:repeatable-page-master-alternatives>
35 </fo:page-sequence-master>
36 </fo:layout-master-set>

```

Listing 2.3 – Exemple d'un layout-master-set avec différenciation de la page de titre et des pages paires et impaires

## 2.3 Éléments de mise en page du contenu

### 2.3.1 Introduction

Il existe plusieurs types d'objets de mise en forme dans XSL-FO, voici ceux que nous allons étudier plus à fond:

- les objets de type bloc,
- les objets de type en-ligne,
- les objets de mise en forme des tables,
- les objets de mise en forme hors-ligne,
- les objets de type liste.

Il existe également des objets pour permettre l'intégration de graphiques et la création de liens.

Cette partie ne se veut pas exhaustive mais donne une aperçu des différentes possibilités offertes par XSL-FO.

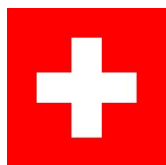
Dans cette section nous allons nous référer à un exemple complet mettant en oeuvre les différentes possibilités de mise en forme offertes par XSL-FO. Le code complet de cet exemple est donné en annexe C. Le rendu obtenu avec FOP fait l'objet d'une illustration dans la figure 2.4.

Cet exemple est un descriptif du parlement de la confédération helvétique ; par la suite, il y sera fait référence sous le terme *Parlement*. Le descriptif du parlement donné en exemple est une page de format A4 avec un en-tête avec des images, des titres, des paragraphes, un tableau, des notes de bas de page et des liens hypertextes.

### 2.3.2 CSS

Tous les objets définis dans la suite de cette section ont la possibilité, pour la plupart de se voir adjoindre de nombreux attributs. Les attributs utilisés dans la recommandation XSL-FO sont compatibles avec la recommandation CSS [16] s'ils ont un équivalent dans cette norme. CSS permet de définir les couleurs, les polices, le rendu et d'autres caractéristiques d'un document en utilisant la notion de style.

Le listing 2.4 donne un exemple sur les possibilités offertes par l'utilisation des attributs de mise en forme des polices avec les attributs `font-family`, `color`, `font-size`, `font-style` ou encore `text-decoration`. Il existe bien sûr de nombreux autres attributs CSS compatibles.



## Le Parlement de la Confédération helvétique

L'*Assemblée fédérale* suisse constitue ce qu'il est convenu d'appeler un parlement de milice dans la mesure où la majorité de ses membres exercent une activité professionnelle parallèlement à leur mandat parlementaire. Les députés doivent néanmoins faire face à un volume de travail considérable : ils sont en effet tenus de siéger quatre fois par an pendant les trois semaines que dure chaque session ordinaire de l'*Assemblée fédérale* (auxquelles s'ajoutent, le cas échéant, les sessions extraordinaires et les sessions spéciales), de participer dans l'intervalle à nombre de séances de commissions, de groupes parlementaires ou de groupes de travail, et enfin de se consacrer à la nécessaire préparation des dossiers.

### L'Assemblée fédérale - système bicaméral

L'Assemblée fédérale se compose de deux Chambres:

- le Conseil National, composé de 200 conseillers nationaux
- le Conseil des États, composé de 46 conseillers aux États (2 par canton)

Le tableau suivant présente la répartition des sièges dans les 2 chambres et la répartition des groupes:

Groupe parlementaire	Répartition	CN <sup>1</sup>	CE <sup>2</sup>	Total
Groupe de l'Union démocratique du centre	63 UDC, 1 Lega	56	8	64
Groupe socialiste	61 PS	52	9	61
Groupe radical-libéral	50 PRD, 4 PLS	40	14	54
Groupe démocrate-chrétien	43 PDC	28	15	43
Groupe des Verts	12 Verts, 1 PCS, 1 SGA	14	0	14
Groupe PEV-UDF	3 PEV, 2 UDF	5	0	5
Non inscrits	2 PdT, 1 AdG-Sol, 1 DS, 1 GLiZ	5	0	5

- [Le site internet du parlement](#)

---

1. Conseil National  
2. Conseil des États

### 2.3.3 Objet blocs

L'élément le plus couramment utilisé pour décrire un paragraphe ou un titre est `fo:block`.

```

25 <fo:block font-size="16pt" font-family="sans-serif" color="grey"
26   space-before="5mm"
27   space-after="5mm"> Le Parlement de la Confédération helvétique
28   </fo:block>
29 <fo:block text-indent="5mm" font-family="serif" font-size="12pt"
30   space-before="5mm"
31   space-after="5mm"> L'<fo:inline font-style="italic">Assemblée
32   fédérale</fo:inline>
33   suisse constitue ce qu'il est convenu d'appeler un parlement de milice
34   dans la mesure où ...
35   ... extraordinaires et les sessions spéciales), de participer dans
36   l'intervalle à nombre de
37   séances de commissions, de <fo:basic-link
38   internal-destination="tableau"
39   text-decoration="bold">groupes parlementaires</fo:basic-link> ou de
40   groupes de
41   travail, et enfin de se consacrer à la nécessaire préparation des
42   dossiers. </fo:block>
43 <fo:block font-size="14pt" font-family="sans-serif" color="grey"
44   space-before="5mm"
45   space-after="5mm">L'Assemblée fédérale – système bicaméral</fo:block>

```

Listing 2.4 – Exemple d'utilisation d'un élément `fo:block`

Dans cet exemple on remarque 3 blocs de textes distincts:

- un titre aux lignes 25-26
- un paragraphe avec indentation<sup>6</sup>, lignes 27 à 38
- un sous-titre aux lignes 39-40.

Chaque nouveau bloc provoque la création d'une nouvelle aire en dessous de la précédente. Les attributs `space-before` et `space-after` permettent de régler les espacements avant et après le bloc de texte.

Selon la recommandation `fo:block-container` permet de regrouper plusieurs éléments `fo:block`.

### 2.3.4 Objets en-ligne

Les objets de type en-ligne<sup>7</sup> permettent, en autres, de formater un partie dans un bloc de texte différemment du reste du paragraphe. Dans l'exemple 2.4, on a mis en évidence les mots *Assemblée fédérale* à chaque apparition

<sup>6</sup>attribut `text-indent`

<sup>7</sup>[anglais] `inline`

dans le paragraphe en les enfermant dans des éléments `fo:inline` et en leur attribuant un style *italique* [ligne 28, listing 2.4] comme le montre le résultat obtenu à la figure 2.5.

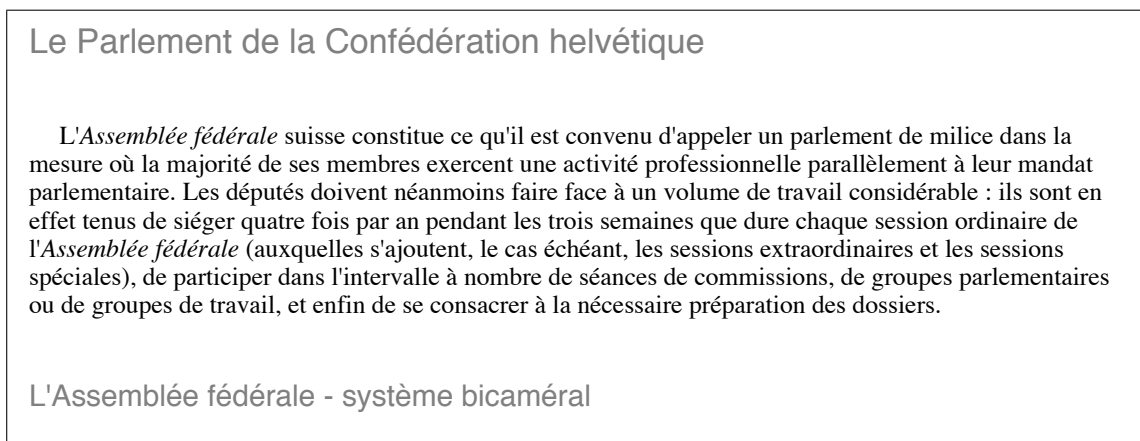


FIG. 2.5 – Résultat obtenu avec FOP du listing 2.4

Plusieurs autres objets de type en-ligne sont utiles pour la mise en page d'un document. On trouvera dans le tableau 2.2 les plus utiles.

Élément	Description
<code>fo:character</code>	Formatage d'un glyphe particulier.
<code>fo:external-graphic</code>	Voir la section 2.3.8
<code>fo:instream-foreign-object</code>	idem
<code>fo:leader</code>	Utilisé pour la création de règles de séparation horizontale et comme équivalent d'un tabulateur
<code>fo:page-number</code>	Retourne le numéro de page
<code>fo:page-number-citation</code>	Retourne le numéro de la page où apparaît un objet référencé. Utilisé pour les tables de matière par exemple.

TAB. 2.2 – Autres éléments de type en-ligne

### 2.3.5 Tables

XSL-FO permet de définir des tables dont la structure des éléments est explicité dans la figure 2.6. Une table compatible avec FOP comprend:

1. des colonnes, la largeur est précisée

2. un en-tête de table, composé d'un rang de cellule
3. un pied de table, composé d'un rang de cellule
4. un corps de table, qui contient plusieurs rangs de cellule

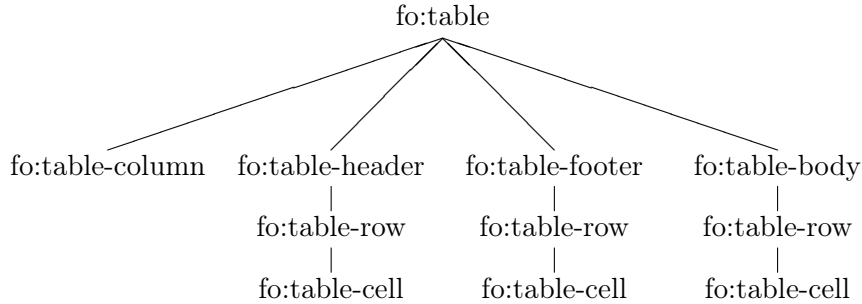


FIG. 2.6 – Structure des tables XSL-FO

Groupe parlementaire	Répartition	CN <sup>1</sup>	CE <sup>2</sup>	Total
Groupe de l'Union démocratique du centre	63 UDC, 1 Lega	56	8	64
Groupe socialiste	61 PS	52	9	61
Groupe radical-libéral	50 PRD, 4 PLS	40	14	54
Groupe démocrate-chrétien	43 PDC	28	15	43
Groupe des Verts	12 Verts, 1 PCS, 1 SGA	14	0	14
Groupe PEV-UDF	3 PEV, 2 UDF	5	0	5
Non inscrits	2 PdT, 1 AdG-Sol, 1 DS, 1 GLiZ	5	0	5

FIG. 2.7 – Table produite par le listing 2.5

Dans notre document exemple, une table a été générée pour donner les statistiques de composition des chambres fédérales comme le montre la figure 2.7. Dans l'extrait de code correspondant à cette table (listing 2.5), on peut remarquer les points suivants:

- les dimensions de la table sont fixes,
- les dimensions des colonnes également,
- l'en-tête de la table a les bordures visibles grâce à l'attribut `border="solid"`,

- le texte de certaines cellules est centré avec `text-align="center"`,
- la table possède un attribut `id="tableau"` qui permettra de pointer sur l'objet comme expliqué au point 2.3.9.

Remarquons finalement que cet extrait de code a été épuré afin d'en faciliter la lecture.

```
<fo:table id="tableau" border="solid" space-before="4mm" width="170mm">
  <fo:table-column column-width="50mm"/>
  ...
  <fo:table-header>
    <fo:table-row>
      <fo:table-cell border="solid">
        <fo:block font-weight="bold">Groupe parlementaire</fo:block>
      </fo:table-cell>
      ...
    </fo:table-row>
  </fo:table-header>
  <fo:table-body>
    <fo:table-row>
      <fo:table-cell>
        <fo:block>Groupe de l'Union démocratique du centre</fo:block>
      </fo:table-cell>
      <fo:table-cell>
        <fo:block>63 UDC, 1 Lega</fo:block>
      </fo:table-cell>
      <fo:table-cell>
        <fo:block text-align="center">56</fo:block>
      </fo:table-cell>
      <fo:table-cell>
        <fo:block text-align="center">8</fo:block>
      </fo:table-cell>
      <fo:table-cell>
        <fo:block text-align="center">64</fo:block>
      </fo:table-cell>
      ...
    </fo:table-row>
  </fo:table-body>
</fo:table>
```

Listing 2.5 – Exemple d'utilisation d'un élément `fo:table`

La recommandation XSL-FO permet également d'emballer une table avec sa légende dans un élément `fo:table-and-caption`.

### 2.3.6 Objets hors-ligne

Les objets de type hors-ligne<sup>8</sup> permettent d'ajouter des éléments flottants grâce à l'élément `fo:float` mais cet élément n'étant pas supporté par FOP nous n'allons pas nous concentrer sur cette fonctionnalité.

La possibilité d'inclure des notes de bas de page est également incluse dans ce type d'objet avec les éléments `fo:footelement` et `fo:footnote-body`. Dans notre exemple *Parlement*, deux notes de bas de page ont été insérées comme le montre la figure 2.8.

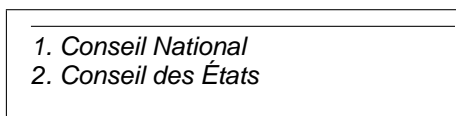


FIG. 2.8 – Les notes de pages produites

Pour insérer une ligne de séparation entre le corps du document et les notes de bas de page, un élément statique a été inséré comme indiqué dans le listing 2.6. Il produit une ligne en bas de page entre le texte et les notes de bas de page. La région entre ces deux zones est identifiée par la valeur `"xsl-footnote-separator"` pour le nom du flux.

```

18 <fo:static-content flow-name="xsl-footnote-separator">
19   <fo:block>
20     <fo:leader leader-pattern="rule" leader-length="100%"
21       rule-thickness="0.5pt"
22       rule-style="solid" color="black" />
23   </fo:block>
  </fo:static-content>

```

Listing 2.6 – Exemple de séparation du texte et des notes de bas de page

Il n'est pas possible d'assigner automatiquement la numérotation des notes de bas de page. C'est la raison pour laquelle il faut définir dans le code la numérotation de chaque note de bas de page, en précisant manuellement la mise en page du renvoi (ici la note est en exposant) et celle de la note de bas de page elle-même. Une note de bas de page se définit à la suite du texte à annoter dans le texte comme le montre le listing 2.7.

Les derniers éléments de type hors-ligne qui offrent des possibilités complémentaires sont `fo:marker` et `fo:retrieve-marker` qui permettent, par exemple, d'avoir un en-tête particulier pour chaque page, selon le chapitre.

<sup>8</sup> *[anglais]* out-of-line

```

83 <fo:block text-align="center" font-weight="bold">CN <fo:footnote>
84   <fo:inline vertical-align="super" font-size="8pt">1</fo:inline>
85   <fo:footnote-body>
86     <fo:block font-style="italic" font-weight="normal"
87       text-align="left">1. Conseil National</fo:block>
88   </fo:footnote-body>
89 </fo:footnote>
90 </fo:block>

```

Listing 2.7 – Exemple de note de bas de page

### 2.3.7 Listes

Les listes sont également supportées par XSL-FO. Pour créer une liste, il faut définir un bloc de liste qui sera composé des éléments de la liste. Un élément de la liste étant lui-même décomposé en un élément de marquage et un élément de contenu comme le montre la figure 2.9.

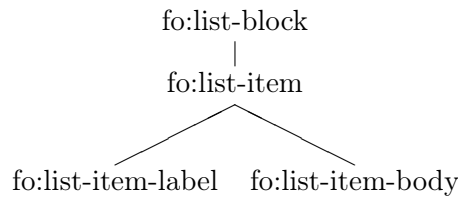


FIG. 2.9 – Structure des listes XSL-FO

Une liste des chambres fédérales est proposée dans notre exemple, une illustration du résultat est disponible (figure 2.10):

```

42 <fo:list-block>
43   <fo:list-item>
44     <fo:list-item-label end-indent="label-end()">
45       <fo:block></fo:block>
46     </fo:list-item-label>
47     <fo:list-item-body start-indent="body-start()">
48       <fo:block>le Conseil National, composé de 200 conseillers
49         nationaux</fo:block>
49     </fo:list-item-body>
50   </fo:list-item>
60 </fo:list-block>

```

Listing 2.8 – Exemple de liste

- L'Assemblée fédérale se compose de deux Chambres:

  - le Conseil National, composé de 200 conseillers nationaux
  - le Conseil des États, composé de 46 conseillers aux États (2 par canton)

FIG. 2.10 – Liste des chambres fédérales

### 2.3.8 Graphiques

Il est possible d'inclure des images dans différents formats (variables selon le processeur) avec XSL-FO. Nous allons voir ici deux solutions différentes illustrées dans notre exemple *Parlement*.

#### SVG

La première solution consiste à inclure du code SVG<sup>9</sup> directement dans le fichier XSL-FO, ceci est possible par l'intermédiaire de l'élément `fo:instream-foreign-object` qui autorise l'insertion de code d'un autre dialecte XML à l'intérieur d'un document XSL-FO comme le montre le code ci-après (listing 2.9) extrait de notre exemple.

```

267     <fo:instream-foreign-object>
268         <svg:svg width="30pt" height="12pt"
269             xmlns:svg="http://www.w3.org/2000/svg">
270             <svg:circle cx="15pt" cy="6pt" r="5pt"
271                 style="fill:blue;"/>
                </svg:svg>
            </fo:instream-foreign-object>

```

Listing 2.9 – Exemple d'insertion de code SVG avec un élément `fo:instream-foreign-object`

Cet exemple produit un simple rond bleu [voir figure 2.12].

#### Fichiers externes

L'en-tête de notre exemple *Parlement* présente deux images comme le montre la figure 2.11.

Ces images ont été insérées par l'intermédiaire de l'élément `fo:external-graphic` comme le montre l'extrait de code 2.10.

<sup>9</sup>Scalable Vector Graphics est une recommandation du W3C [19] pour décrire des objets graphiques vectoriels en XML dont l'espace de nom est `svg`. Le propos dans ce travail n'étant pas la syntaxe et les possibilités de SVG, le lecteur trouvera plus d'informations sur le site <http://www.svg.org>.

FIG. 2.11 – En-tête du document *Parlement*, résultat du listing 2.10

```

11 <fo:static-content flow-name="xsl-region-before">
12   <fo:block text-align-last="justify">
13     <fo:external-graphic height="3cm" src="suisse.jpg"
14       scaling="uniform"/>
15     <fo:leader/>
16     <fo:external-graphic height="3cm" src="palais.jpg"
17       scaling="uniform"/>
18   </fo:block>
19 </fo:static-content>

```

Listing 2.10 – Exemple d'insertion d'images avec un élément `fo:external-graphic`

L'attribut `scaling` avec la valeur "uniform" indique au processeur que si l'image doit être réduite pour respecter la hauteur ou la largeur limitée, la réduction des dimensions doit se faire de manière proportionnelle.

L'attribut `src` peut également contenir une url ou un chemin absolu d'accès au fichier.

### 2.3.9 Liens

Finalement, XSL-FO permet de définir des liens internes ou externes. Dans le cas de la production de document PDF avec FOP, ces liens sont totalement fonctionnels.

#### Liens internes

Un lien interne est un renvoi à l'intérieur d'un document. Dans notre exemple, le texte **groupes parlementaires**, en gras dans la figure 2.5, renvoie vers le tableau présenté dans la figure 2.7. Pour ce faire un attribut `id` a été donné à la table sus-mentionnée. Ensuite, le texte qui devait être lié a été enveloppé dans un élément `fo:basic-link` accompagné de l'attribut

`internal-destination` qui a la valeur de l'id de l'objet vers lequel pointe le lien. Le code du lien interne est donné dans le listing ci-après.

```
<fo:basic-link internal-destination="tableau" text-decoration="bold">
  groupes parlementaires
</fo:basic-link>
```

Listing 2.11 – Exemple d'un lien interne

### Liens externes

Un lien externe a un mécanisme très proche de celui du lien interne. Les deux différences avec le lien interne sont:

- l'utilisation d'un attribut `internal-destination`
- dans notre exemple, la destination est une url.

Pour le reste, les options de mise en forme sont les mêmes que pour le lien interne.

```
<fo:basic-link
  external-destination= "url('http://www.parlement.ch/')"
  text-decoration="underline"
  color="blue">
  Le site internet du parlement
</fo:basic-link>
```

Listing 2.12 – Exemple d'un lien externe

FIG. 2.12 – Un lien externe dans le document *Parlement*

## Chapitre 3

# Structure des données en entrée

Le catalogue des produits issu du webshop *eSarine* se présente sous la forme d'un fichier XML dont nous allons étudier la structure dans ce chapitre. La figure 3.1 donne un aperçu de la structure des noeuds du document XML décrivant les produits. Dans la suite de ce chapitre nous allons détailler les différents éléments de ce document.

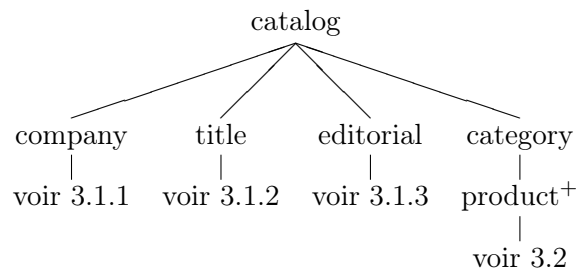


FIG. 3.1 – Structure de l'arbre XML du catalogue des produits

Un document XML exemple de catalogue de produits issu de *eSarine* est donné dans le listing 3.1 à la page suivante. Il n'existe pas de schéma XML ou de DTD pour ce type de fichier.

```

1 <catalog>
2   <company>
3     <name>Entreprise Exemple</name>
4   </company>
5   <title>
6     <main>Catalogue des produits 2005</main>
7     <sub>Une longueur d'avance</sub>
8     <date>27 avril 2005 14:37:21</date>
9     <author>Paul Personne</author>
10    <copyright>Exemple SA, Fribourg, Suisse</copyright>
11    <location>Suisse</location>
12    <annotation>L'excellence au rendez-vous!</annotation>
13  </title>
14  <editorial>
15    <title>Chers clients,</title>
16    <content>Voici le nouveau catalogue !</content>
17  </editorial>
18  <category>
19    <name>T-Shirts</name>
20    <product>
21      <name>T-Shirt Geek binaire</name>
22      <description>T-Shirt en coton "Geek"</description>
23      <img-path>/product/t-shirt/geek.jpg</img-path>
24      <occurrences>
25        <occurrences-header>
26          <option-value productTypeId="1">
27            Taille
28          </option-value>
29        </occurrences-header>
30        <occurrence>
31          <value productTypeId="1">XL</value>
32          <item-number>200456</item-number>
33          <price>20.0</price>
34          <quantity>1</quantity>
35          <discount>0</discount>
36          <quantity-in-stock>10</quantity-in-stock>
37          <status>1</status>
38          <weight-points>0</weight-points>
39          <reorder-time>0</reorder-time>
40          <trigger-level>0</trigger-level>
41        </occurrence>
42      </occurrences>
43      <product-type-field name="Couleur">blanc</product-type-field>
44    </product>
45  </category>
46 </catalog>

```

Listing 3.1 – Un exemple de fichier XML produit par *eSarine*

## 3.1 Données générales

### 3.1.1 company

Le noeud `company` possède un fils `name` qui permet de donner le nom de l'entreprise qui publie le catalogue.

### 3.1.2 title

Le noeud titre possède les fils décrits dans le tableau 3.1.

Élément	Description
<code>main</code>	Le titre du catalogue
<code>sub</code>	Sous-titre
<code>date</code>	La date d'édition
<code>author</code>	Auteur
<code>copyright</code>	Les droits de propriété intellectuelle
<code>location</code>	La localisation de l'entreprise
<code>annotation</code>	Une note complémentaire

TAB. 3.1 – Fils de l'élément `title`

### 3.1.3 editorial

Les éléments fils `title` et `content` permettent de définir le titre et le contenu de l'éditorial du catalogue.

## 3.2 Les produits

Les produits sont classés par catégorie. Une catégorie porte un nom donné par un fils `name` et regroupe plusieurs éléments `product` dont le contenu est détaillé dans la figure 3.2.

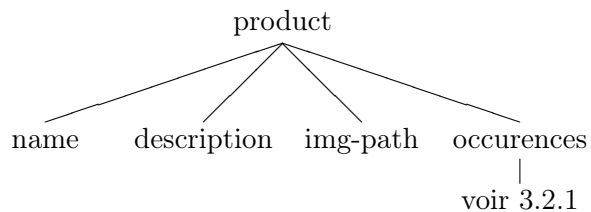


FIG. 3.2 – Structure de l'arbre XML d'un produit du catalogue

Dans le tableau 3.2 se trouve la description des trois premières informations composant un produit:

Élément	Description
<code>name</code>	Le nom du produit
<code>description</code>	La description du produit
<code>img-path</code>	Le chemin d'accès à l'image illustrant le produit

TAB. 3.2 – Éléments non répétés d'un produit

### 3.2.1 occurrences

Cet élément contient toutes les informations concernant les différentes déclinaisons d'un produit et les caractéristiques relatives à ce produit et à ses différentes déclinaisons.

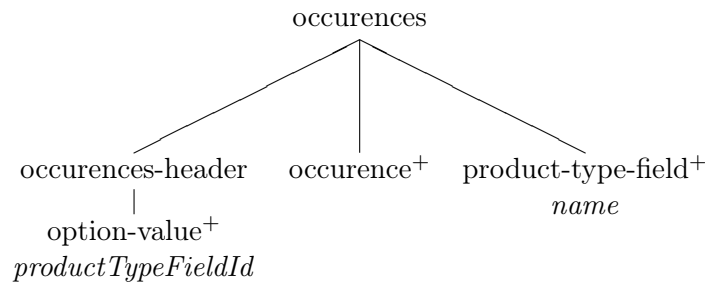


FIG. 3.3 – Structure de l'arbre XML des déclinaisons d'un produit

#### occurrences-header

Cet élément contient un ou plusieurs fils `option-value` qui permettent de définir les différents types d'informations disponibles sur le produit comme par exemple la taille, la couleur ou une option qui varie selon chaque déclinaison d'un produit. L'attribut `productTypeFieldId` est un numéro d'identification de cette information qui sera utilisée par le fils `value` d'une déclinaison<sup>1</sup> du produit.

#### occurrence

Le tableau 3.3 présente l'ensemble des descendants d'un élément `occurrence` avec leur description.

<sup>1</sup>[anglais] occurrence

Élément	Description
value <sup>+</sup>	Cet élément peut être présent une ou plusieurs fois, il possède un attribut <i>productTypeFieldId</i> qui lie la valeur de l'élément à un type d'information contenu dans <i>occurences-header</i>
item-number	le numéro d'identification du produit
price	le prix du produit
quantity	la quantité de produit dans cette déclinaison
discount	le statut d'un éventuel rabais sur ce produit
quantity-in-stock	la quantité de produit en stock
status	le statut du produit
weight-point	le poids accordé au produit pour les calculs de rabais
reorder-time	la date à laquelle à eu lieu la dernière commande
trigger-level	le niveau d'alerte pour provoquer une nouvelle commande

TAB. 3.3 – Fils de l'élément occurrence

### product-type-field

Le dernier fils de l'élément **product** permet de définir des caractéristiques communes à toutes les déclinaisons d'un produit comme par exemple la couleur. Cet élément possède un attribut **name** qui donne la description de la caractéristique, par exemple **name="couleur"**. Le contenu de l'élément, lui, donne la valeur de la caractéristique, par exemple **rouge**.

## Chapitre 4

# Feuille de style

La production du catalogue des produits est réalisée par FOP qui reçoit en entrée le résultat d'une transformation XSLT comme le montre la figure 4.1. Ce chapitre décrit en détail le fonctionnement de la feuille de style qui applique des transformations à un document XML au format décrit au chapitre 3 pour obtenir un document au format XSL-FO. La conception générale du système sera tout d'abord exposée, puis le résultat obtenu pour terminer avec l'implémentation. De larges extraits du code de la feuille de style sont disponibles dans l'annexe B et présentent le code des templates de la feuille de style XSLT. Dans ce chapitre, le résultat XSL-FO sera présenté et illustré selon les besoins.

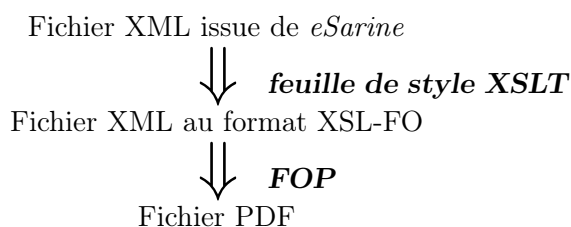


FIG. 4.1 – Processus de transformation

## 4.1 Conception

### 4.1.1 Fonctionnement général

L'objectif de la création du catalogue des produits était de fournir une solution efficace produisant un catalogue de produits utilisables par des clients aussi bien sous forme électronique que sous forme papier. La feuille de style traite le document XML issu du webshop *eSarine* en parcourant les différents noeuds. Elle est conçue pour permettre une certaine flexibilité pour l'utilisa-

teur qui peut personnaliser un certain nombre de paramètres. La figure 4.2 présente le catalogue au niveau conceptuel.

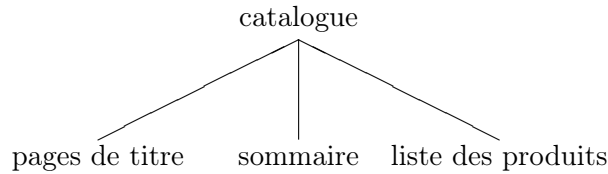


FIG. 4.2 – Catalogue au niveau conceptuel

### 4.1.2 Traitement des données

Les données du fichier XML *eSarine* ne subissent aucune modification particulière durant le processus de transformation. Cependant toutes les données fournies ne sont pas utilisées pour la production du catalogue des produits. Comme nous le verrons ultérieurement, la possibilité de limiter la longueur des chaînes de caractères a été étudiée puis écartée. Cette solution présentait le désavantage de couper arbitrairement des informations, ce qui pouvait altérer la cohérence de certaines informations dans le catalogue finalement produit.

La conception de la mise en page est décrite dans la section suivante présentant la structure du document PDF obtenu à la fin du processus de transformation.

## 4.2 Structure du catalogue pdf

Dans cette section, le résultat produit par l'implémentation décrite au point 4.3 est présentée. Il s'agit d'un document PDF comportant deux parties principales:

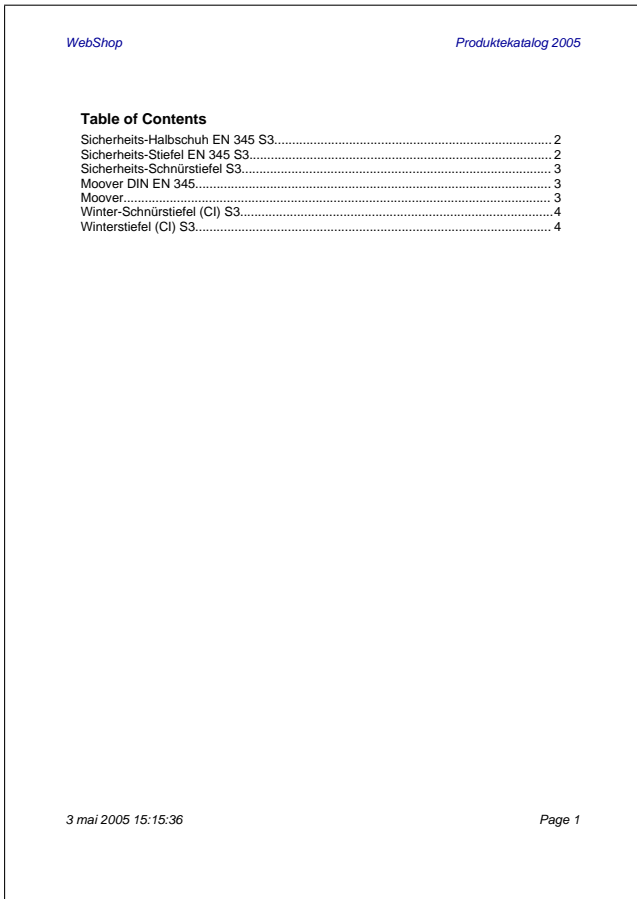
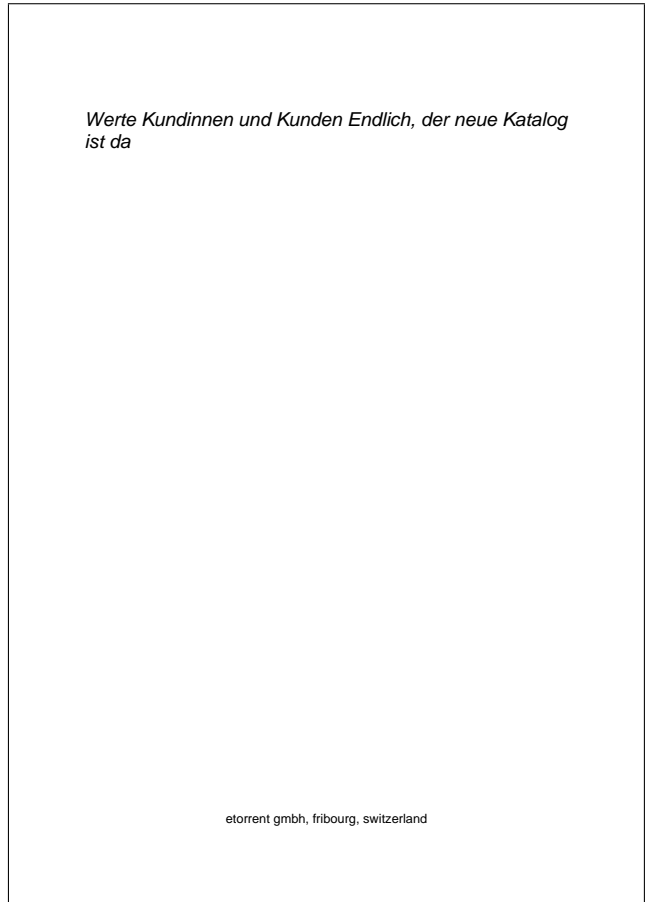
- le titre, la page de garde et le sommaire,
- les pages de description des produits.

### 4.2.1 Titre et sommaire

À la page 34 se trouve un exemple des trois premières pages d'un catalogue. Dans la partie supérieure, la page de titre et celle de garde sont présentées. Dans la partie inférieure, une page de sommaire est proposée. Un sommaire peut s'étendre sur plusieurs pages si nécessaire.

### 4.2.2 Pages contenant les produits

Les pages contenant les produits peuvent se présenter sous différentes formes comme le montrent les illustrations de la page 35. Les différentes mises en forme des pages de présentation des produits sont liées au nombre de déclinaisons que possède un produit comme expliqué au point 4.3.



WebShop Produktkatalog 2005

**Sicherheits-Halbschuh EN 345 S3**



hochwertiges geschmeidiges Rindleder, Sicherheits - Reflexstreifen, Funktionsfutter, modische Textil-Schnürschlaufen, Nähte funkenfest, naturgegerbte Lederbrandsohle

Schutzklasse:S3  
Größe:39 - 44,  
Farbe:schwarz, hellgraues Futter

Item #	Options	Price
2601	Größe: 39	0.00 EUR
2601	Größe: 40	0.00 EUR
2601	Größe: 41	0.00 EUR
2601	Größe: 42	0.00 EUR
2601	Größe: 43	0.00 EUR
2601	Größe: 44	0.00 EUR

**Sicherheits-Stiefel EN 345 S3**



hochwertiges geschmeidiges Rindleder, Sicherheits - Reflexstreifen, Funktionsfutter, modische Textil-Schnürelemente, Nähte funkenfest, naturgegerbte Lederbrandsohle

Farbe:schwarz, hellgraues Futter  
Größe:39 - 47  
Schutzklasse:S3


Item #	Options	Price
2607	Größe: 39	0.00 EUR
2607	Größe: 40	0.00 EUR
2607	Größe: 41	0.00 EUR
2607	Größe: 42	0.00 EUR
2607	Größe: 43	0.00 EUR
2607	Größe: 44	0.00 EUR

There are 9 items for this product in our catalog. There are only 6 items displayed.

3 mai 2005 15:15:36 Page 2

WebShop Produktkatalog 2005

**Sicherheits-Schnürstiefel S3**




Vollrindleder, weiche Knöchelpolsterung, durchgehend Lederfutter, naturgegerbte Lederbrandsohle

Schutzklasse:S3  
Größe:39 - 41,  
Farbe:schwarz/blau

Item #	Options	Price
3609	Größe: 39	0.00 EUR
3609	Größe: 40	0.00 EUR
3609	Größe: 41	0.00 EUR

**Moover DIN EN 345**




PUR-Duo-Sohle

Farbe:schwarz  
Größe:39 - 40  
Art.-Nr.:8072u

Item #	Options	Price
8072u	Größe: 39	0.00 EUR
8072u	Größe: 40	0.00 EUR

**Moover**



Bau-Halbschuh, PUR-DUO-Sohle


Farbe:schwarz  
Schutzklasse:S3  
Größe:39 - 41

Item #	Options	Price
8073	Größe: 39	0.00 EUR
8073	Größe: 40	0.00 EUR
8073	Größe: 41	0.00 EUR

3 mai 2005 15:15:36 Page 3

WebShop Produktkatalog 2005

**Winter-Schnürstiefel (CI) S3**




Rindleder, Warmfutter aus Schurwolle, mit Überkappe, Stahlzwischensohle, DUO-Compound Sohle, Weite 10 plus

Schutzklasse:S3  
Farbe:schwarz/grau  
Art.-Nr.:3657

Item #	Options	Price
3657	Größe: 39	0.00 EUR
3657	Größe: 40	0.00 EUR
3657	Größe: 41	0.00 EUR
3657	Größe: 42	0.00 EUR
3657	Größe: 43	0.00 EUR
3657	Größe: 44	0.00 EUR

There are 9 items for this product in our catalog. There are only 6 items displayed.

**Winterstiefel (CI) S3**



Rindleder, Reißverschluss innen, Warmfutter aus Schurwolle, Stahlzwischensohle und Überkappe, DUO-Compound Sohle, Weite 10 plus

Schutzklasse:S3  
Farbe:schwarz  
Größe:39

Item #	Options	Price
3655	Größe: 39	0.00 EUR

3 mai 2005 15:15:36 Page 4

## 4.3 Implémentation

L'implémentation de la feuille de style est décortiquée dans cette section. La transformation du document XML *eSarine* se fait par l'intermédiaire de plusieurs templates qui sont appelés en fonction des conditions rencontrées. La figure 4.3 présente la hiérarchie d'appel des différents templates.

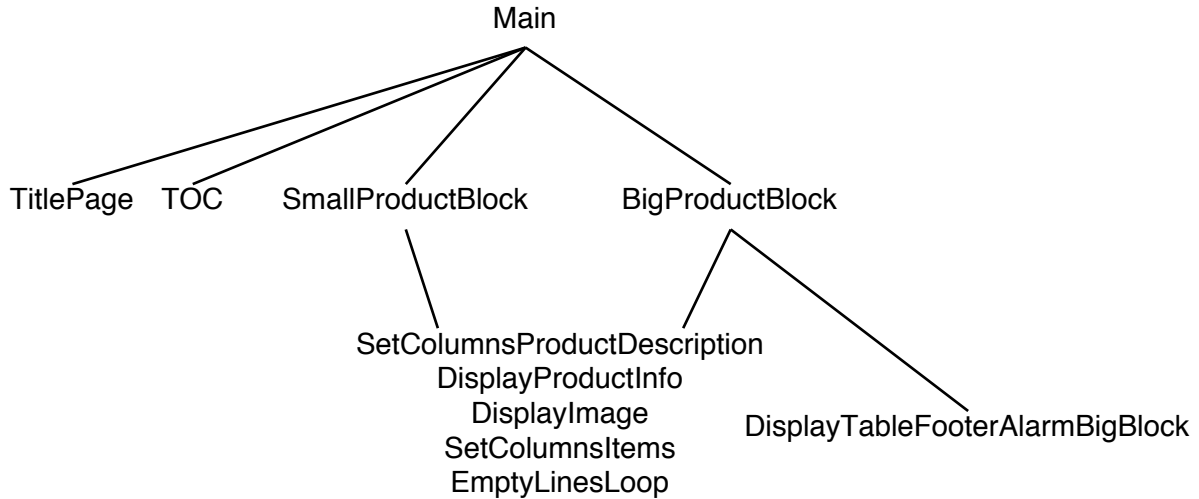


FIG. 4.3 – Structure des templates

### 4.3.1 Paramètres globaux

Les paramètres globaux sont modifiables par l'utilisateur, ils sont décrits dans le tableau 4.1. Les valeurs des paramètres sont données en exemple. Vu la façon dont la feuille de style a été implémentée, les distances doivent toujours être exprimées en *mm*. Pour plus d'information sur les valeurs des paramètres, se référer à la section 4.4 qui précise l'utilisation des paramètres.

### 4.3.2 Variables globales

Les deux variables globales utilisées servent à calculer les dimensions des blocs de produit qui sont en fait des tableaux. Les dimensions des tableaux doivent en effet être explicitement données afin de permettre certaines opérations avec FOP. Ces variables sont présentées dans le tableau 4.2.

### 4.3.3 Template Main

C'est la porte d'entrée de la feuille de style XSLT. Il définit tout d'abord le `fo:layout-master-set` du résultat XSL-FO. Pour se faire, les paramètres des dimensions de la page, des marges de la page et de marges du corps du

Paramètre	Valeur exemple	Description
PageBackgroundColor	"rgb(255, 255, 255)"	couleur du fond de la page
BorderColor	"rgb(100,149,237)"	couleur des bordures
BorderWidth	"2pt"	taille des bordures
BackgroundHeaderColor	"rgb(208, 227, 250)"	couleur du fond de l'en-tête des tables
BackgroundFooterColor	"rgb(208, 227, 250)"	couleur du fond du pied des tables
BackgroundBodyColor	"rgb(255, 255, 255)"	couleur du fond du corps de la page
BackgroundCellColor	"rgb(239, 246, 255)"	couleur des cellules des tables
HeaderTextColor	"rgb(0,0,150)"	couleur du texte de l'en-tête
PageWidth	"210mm"	largeur de la page
PageHeight	"297mm"	hauteur de la page
PageMarginTop	"10mm"	marge supérieure de la page
PageMarginBottom	"10mm"	marge inférieure de la page
PageMarginLeft	"10mm"	marge gauche de la page
PageMarginRight	"10mm"	marge droite de la page
BodyMarginTop	"25mm"	marge supérieure du corps de la page
BodyMarginBottom	"25mm"	marge inférieure du corps de la page
BodyMarginLeft	"15mm "	marge gauche du corps de la page
BodyMarginRight	"15mm"	marge droite du corps de la page
FontFamily	'Helvetica'	Police de base du document
GlobalFontSize	'12pt'	Taille de base de la police dans le document
SmallProductBlockThreshold	3	seuil du nombre de déclinaisons pour un petit produit
BigProductBlockThreshold	6	seuil du nombre de déclinaisons maximum

TAB. 4.1 – Paramètres globaux

texte sont utilisés ainsi que les variables locales du template décrites dans le tableau 4.3.

**Variables** Les quatre premières variables du template `Main` sont des variables calculées établissant la taille des régions du pourtour du corps du texte en fonctions des paramètres globaux définissant les marges du corps de la page.

La dernière variable compte le nombre de déclinaisons d'un produit, en comptant les petit-fils occurrence d'un produit.

Ensuite, le template `Main` appelle le template `TitlePage` décrit au point 4.3.4 qui construit la page de titre et la page de garde du catalogue. Une fois ces pages construites, le template crée les pages du catalogue contenant les

Variable et description	Valeur
BodyHorizontalWriteArea <i>taille de la zone horizontale libre</i>	substring-before(\$PageWidth,'mm') - substring-before(\$PageMarginLeft,'mm') - substring-before(\$PageMarginRight,'mm') - substring-before(\$BodyMarginLeft,'mm') - substring-before(\$BodyMarginRight,'mm') - 10
BodyHorizontalWriteAreaString <i>chaîne de caractères de la variable précédente</i>	concat(\$BodyHorizontalWriteArea,'mm')
BodyVerticalWriteArea <i>taille de la zone verticale libre</i>	substring-before(\$PageHeight,'mm') - substring-before(\$PageMarginTop,'mm') - substring-before(\$PageMarginBottom,'mm') - substring-before(\$BodyMarginTop,'mm') - substring-before(\$BodyMarginBottom,'mm') - 10
BodyVerticalWriteAreaString <i>chaîne de caractères de la variable précédente</i>	concat(\$BodyVerticalWriteArea,'mm')

TAB. 4.2 – Variables globales

Variable	Valeur
RegionBeforeExtent	concat(substring-before(\$BodyMarginTop,'mm') - 5, 'mm')
RegionAfterExtent	concat(substring-before(\$BodyMarginBottom,'mm') - 5, 'mm')
RegionStartExtent	concat(substring-before(\$BodyMarginLeft,'mm') - 5, 'mm')
RegionEndExtent	concat(substring-before(\$BodyMarginRight,'mm') - 5, 'mm')
OccurencesCounter	count(occurences/occurrence)

TAB. 4.3 – Variables du template Main

descriptions des produits.

Pour créer les pages le template Main effectue les opérations suivantes:

1. Création de l'en-tête
2. Création du pied de page
3. Construction de la table des matières à l'aide du template TOC décrit au point 4.3.5.
4. Construction de la liste des produits et de leur description.

L'en-tête contient le nom de l'entreprise et le nom du catalogue. Le pied de page la date de production du fichier XML *eSarine* et le numéro de la page. Le code XSL-FO obtenu après application de la feuille de style illustrant ces éléments est affiché dans le listing 4.1 . Le résultat PDF correspond aux en-têtes et pieds de pages des illustrations de la page 35.

```

<fo:static-content font-style="italic"
    flow-name="xsl-region-before"
    color="rgb(0,0,150)">
  <fo:block text-align-last="justify">
    WebShop<fo:leader />Produktekatalog 2005
  </fo:block>
</fo:static-content>
<fo:static-content font-style="italic" flow-name="xsl-region-after">
  <fo:block text-align-last="justify">3 mai 2005 15:15:36
    <fo:leader /> Page <fo:page-number />
  </fo:block>
</fo:static-content>

```

Listing 4.1 – Code FO pour l'en-tête et le pied de page

Nous allons maintenant passer en revue le processus du dernier point. Le template sélectionne tout noeud `product` rencontré dans le document XML *eSarine*, ceci sans tenir compte de la catégorie dans laquelle il se trouve. Ensuite la variable `OccurencesCounter` décrite précédemment compte le nombre de déclinaisons du produit. Si le produit compte un nombre de déclinaisons inférieur ou égal au paramètre global `SmallProductBlockThresold`<sup>1</sup>, il s'agit d'un produit qui aura un bloc de petite taille. Dans ce cas le template `SmallProductBlock` (voir 4.3.6) est appelé. Dans le cas contraire, c'est le template `BigProductBlock`, décrit au point 4.3.7, qui est utilisé.

Une fois tous les produits traités, le document est terminé.

#### 4.3.4 Template TitlePage

Le template `TitlePage` crée une page de titre comme démontré au point 4.2.1. Il y a trois paramètres ajustables par l'utilisateur, paramètres présentés dans le tableau 4.4. Un pied de page spécifique contenant le copyright est créé pour la page de titre. Le listing 4.2 présente le code FO produisant les deux pages de titre et de garde dont une illustration est donné dans partie supérieure de la page 34.

Paramètre	Valeur exemple	Description
<code>TitleSize</code>	'30pt'	taille du titre
<code>TitleCompanyNameSize</code>	'24pt'	taille du nom de l'entreprise sur la page de titre
<code>TitleColor</code>	'blue'	couleur du titre et sous-titre

TAB. 4.4 – Paramètres du template `TitlePage`

<sup>1</sup>Ce paramètre fixe le seuil de déclinaison pour qu'un produit passe du groupe des *petits* produits à celui des *grands*

```

<fo:page-sequence master-reference="Page">
  <fo:static-content flow-name="xsl-region-after">
    <fo:block text-align="center">
      etorrent gmbh, fribourg, switzerland
    </fo:block>
  </fo:static-content>
  <fo:flow flow-name="xsl-region-body">
    <fo:block text-align="center" space-after="3cm"
      font-style="italic" font-size="30pt" color="blue">
      Produktkatalog 2005
    </fo:block>
    <fo:block text-align="center" space-after="3cm"
      font-size="24pt" color="blue" break-after="page">
      WebShop
    </fo:block>
    <fo:block font-style="italic" font-size="18pt">
      Werte Kundinnen und Kunden
      Endlich, der neue Katalog ist da
    </fo:block>
  </fo:flow>
</fo:page-sequence>

```

Listing 4.2 – Code FO pour la page de titre et celle de garde

### 4.3.5 Template TOC

Le template TOC crée la table des matières ou le sommaire du catalogue. Le processus de création de cette table des matières se déroule en passant chaque produit dans une boucle `xsl:for-each`. Pour chaque produit, une ligne contenant le nom de ce produit, un série de points de suspension et le numéro de la page où se trouve la description dudit produit est écrite. Les liens internes proposés par XSL-FO comme expliqué au point 2.3.9 sont utilisés en adéquation avec la fonction `generate-id()` qui permet d'obtenir un identifiant unique sous forme de chaîne de caractères. Cet identifiant unique permet de réaliser un lien interne vers chaque produit et vers la page où se trouve le produit<sup>2</sup>. Un extrait du code FO obtenu pour produire la table des matières présentée à la page 34 est présenté dans le listing 4.3.

```

<fo:block space-after="5pt" font-weight="bold" font-size="14pt">
  Table of Contents
</fo:block>
<fo:block break-after="page">
  <fo:block text-align-last="justify">

```

<sup>2</sup>Pour ce point on utilise un élément `fo:page-number-citation`

```

<fo:basic-link internal-destination="N10030">
  Sicherheits-Halbschuh EN 345 S3
  <fo:leader leader-pattern="dots" />
  <fo:page-number-citation ref-id="N10030" />
</fo:basic-link>
</fo:block>
...
</fo:block>

```

Listing 4.3 – Code FO pour la table des matières

### 4.3.6 Template SmallProductBlock

Ce template permet de créer un bloc pour un produit de petite taille. Le principe adopté est le suivant: pour des produits dont le nombre de déclinaisons est inférieur ou égal au seuil défini par l'utilisateur ( `SmallProductBlock-Threshold` ), on affiche trois blocs de produit par page. Pour calculer la taille nécessaire aux différentes parties de la présentation d'un produit on utilise les quatre premières variables décrites dans le tableau 4.5.

Variable	Valeur
<code>SpaceAfterSmallProductBlock</code>	<code>concat(\$BodyVerticalWriteArea * 0.2 div 3, 'mm')</code>
<code>HeightItemsSmallProductBlock</code>	<code>concat((\$SmallProductBlockThreshold + 1) * 5, 'mm')</code>
<code>HeightSmallProductBlock</code>	<code>concat(\$BodyVerticalWriteArea * 0.8 div 3, 'mm')</code>
<code>HeightDescriptionSmallProductBlock</code>	<code>concat(   substring-before(\$HeightSmallProductBlock, 'mm' )   - substring-before(\$HeightItemsSmallProductBlock, 'mm' )   , 'mm')</code>
<code>OccurencesCounter</code>	<code>count(occurences/occurence)</code>
<code>currentItem</code>	<code>item-number</code>
<code>currentproductTypeFieldId</code>	<code>@productTypeFieldId</code>

TAB. 4.5 – Variables du template `SmallProductBlock`

La taille de l'espace vertical disponible sur une page est divisée en trois. Les 4/5 de l'espace alloué à un produit sont occupés par l'image, la description et les déclinaisons du produit. Le 1/5 restant est un espace à la suite du bloc du produit.

Les 4/5 réservés à la présentation du produit sont divisés en différents sous-espaces. Un espace de 5mm est réservé par déclinaison de produit affichée, on ajoute en plus 5mm pour aérer la présentation. L'espace vertical consacré à l'image et à la description est la différence entre l'espace vertical total du bloc produit diminué de la zone réservée à la liste des déclinaisons.

Un bloc produit de petite taille est présenté dans la figure 4.4 et le code FO correspondant dans le listing 4.4.


Moover		
Bau-Halbschuh, PUR-DUO-Sohle		
<i>Farbe:schwarz</i> <i>Schutzklasse:S3</i> <i>Größe:39 - 41</i>		
		
Item #	Options	Price
8073	Größe: 39	0.00 EUR
8073	Größe: 40	0.00 EUR
8073	Größe: 41	0.00 EUR

FIG. 4.4 – Bloc produit issu du listing 4.4

Il est important de souligner que les valeurs calculées ne seront pas respectées si la description du produit nécessite un espace plus important et le bloc du produit s'étendra en hauteur. Ce comportement est lié aux limitations imposées par FOP qui sont décrites dans la section 4.5.

Une fois les tailles des différentes parties définies, ces différents aspects du produit sont présentés à l'aide de tableaux. Afin d'éviter qu'un produit ne soit réparti sur deux pages et pour qu'il soit centré, chaque produit est encapsulé dans un tableau qui contient lui-même deux tableaux:

- un tableau contenant l'image et la description du produit,
- un tableau contenant les déclinaisons du produit.

Les colonnes du premier tableau sont définies par l'intermédiaire d'un template `SetColumnsProductDescription`. Le tableau contenant la description et l'image fait appel à deux templates auxiliaires `DisplayImage` et `DisplayProductInfo` – décrits au point 4.3.8 – qui sont appelés dans un ordre alterné. Un produit affichera l'image à gauche, le suivant à droite et ainsi de suite... Ce mode de présentation utilise la position du produit dans le fichier XML `eSarine` pour déterminer la position à adopter.

Pour chaque produit, on intègre l'image du produit et sa description dans le premier tableau qui correspond à la partie supérieure de la figure 4.4 qui a une couleur de fond blanche.

Le tableau suivant, accolé au premier et présentant les différentes déclinaisons du produit fait également appel à un template particulier pour définir les colonnes du tableau : `SetColumnsItems`. Dans ce tableau on affiche le code de la déclinaison du produit, les caractéristiques particulières et le prix de chacune des déclinaisons du produit qui correspond à la partie ayant des teintes de bleu en arrière-plan sur la figure 4.4.

```

<fo:table height="57.866mm" width="100%" space-after="14.466mm"
  table-layout="fixed">
  <fo:table-column column-width="proportional-column-width(1)"/>
  <fo:table-column column-width="150mm"/>
  <fo:table-column column-width="proportional-column-width(1)"/>
  <fo:table-body>
    <fo:table-row keep-with-next="always">
      <fo:table-cell><fo:block> </fo:block></fo:table-cell><fo:table-cell>
        <fo:table border-right-color="rgb(100,149,237)" height="37.866mm"...>
          <fo:table-column column-width="proportional-column-width(1.5)"/>...
          <fo:table-body background-color="rgb(255, 255, 255)">
            <fo:table-row>
              <fo:table-cell padding-left="5mm">
                <fo:block text-align="center" id="N103AC" font-size="18pt"> Moover
                  </fo:block>
                <fo:block space-before="5mm" space-after="5mm" wrap-option="wrap"
                  font-size="11pt"> Bau-Halbschuh, PUR-DUO-Sohle </fo:block>
                <fo:block font-style="italic" wrap-option="wrap" font-size="11pt">
                  Farbe:schwarz
                  </fo:block>
                ...</fo:table-cell>
              <fo:table-cell display-align="center">
                <fo:block text-align="center">
                  <fo:external-graphic src="178-2.jpg" scaling="uniform"
                    height="37.866mm"/>
                  </fo:block>
                </fo:table-cell></fo:table-row>
            </fo:table-body></fo:table></fo:table-cell>
          <fo:table-cell><fo:block> </fo:block></fo:table-cell>
        </fo:table-row></fo:table-row></fo:table-cell></fo:block>
      </fo:table-cell>
      <fo:table font-size="12pt"
        border-right-color="rgb(100,149,237)" height="20mm"...>
        <fo:table-column column-width="proportional-column-width(1)"/>...
        <fo:table-header background-color="rgb(208, 227, 250)">
          <fo:table-row>
            <fo:table-cell><fo:block>Item #</fo:block></fo:table-cell>
            <fo:table-cell><fo:block>Options</fo:block></fo:table-cell>
            <fo:table-cell><fo:block>Price</fo:block></fo:table-cell>
          </fo:table-row></fo:table-header>
        <fo:table-body>
          <fo:table-row background-color="rgb(239, 246, 255)"
            keep-with-previous="always">
            <fo:table-cell><fo:block>8073</fo:block></fo:table-cell>
            <fo:table-cell><fo:block>Grösse: 39</fo:block></fo:table-cell>
            <fo:table-cell><fo:block>0.00 EUR</fo:block></fo:table-cell>
          </fo:table-row>
          ...</fo:table-body>
        </fo:table></fo:table-cell></fo:table-cell></fo:block>
      </fo:table-cell>
    </fo:table-row>
  </fo:table-body>
</fo:table>

```

Listing 4.4 – Code FO pour un bloc produit de petite taille

Pour terminer le tableau, si le nombre de déclinaisons présentes est inférieur au nombre total de déclinaisons prévues à l’affichage, on affiche des lignes vides en utilisant un template particulier : `EmptyLinesLoop`. Ceci permet d’assurer une unité de présentation des blocs de produit.

#### 4.3.7 Template `BigProductBlock`

Le template `BigProductBlock` affiche un produit dont le nombre de déclinaisons est supérieur à la limite fixée. Il fonctionne de manière très similaire au bloc de petite taille. Seules vont être soulignées les différences avec le bloc de petite taille. La première différence se situe au niveau de la division de la page : la page est divisée en deux, ce qui limite l’affichage à deux produits. Pour le reste la structure du bloc et la répartition des différents espaces est identique. Un ajout important se situe lors de la création de la liste des déclinaisons du produit. En effet, au cas où certaines déclinaisons du produit ne peuvent être affichées car elles dépassent la limite supérieure ( `BigProductBlockThreshold` ), une ligne d’alerte va être affichée en fin de tableau en appelant le template `DisplayTableFooterAlarmBigBlock` décrit au point suivant.

La dernière différence concerne les valeurs des variables du template qui sont présentées dans l’annexe B.1 vu leurs similarités avec celles du template `SmallProductBlock`.

#### Template `DisplayTableFooterAlarmBigBlock`

Ce template est appelé avec un paramètre `total` qui contient le nombre de déclinaisons différentes du produit en cours de traitement. Si le paramètre a une valeur supérieure au paramètre global `BigProductBlockThreshold`, la dernière ligne du tableau des déclinaisons du produit contient un texte signalant le nombre de déclinaisons affichées et le nombre de déclinaisons réellement disponibles comme le montre le produit affiché en bas de la première illustration de la page 35.

#### 4.3.8 Templates appelés par les deux types de bloc produit

Comme décrit auparavant, certains templates sont appelés par les deux types de bloc produit. Ces templates ont des fonctions particulières qui sont explicitées dans la suite de cette partie. Le listing 4.4 et le résultat produit illustré par la figure 4.4 contiennent des exemples du résultat produit avec ces différents templates.

#### Template `SetColumnsProductDescription`

Le template définit la largeur relative des colonnes du tableau contenant l’image et la description du produit en cours de traitement. La largeur de

la colonne contenant la description du produit est une fois et demie plus importante que celle de l'image.

### Template `DisplayProductInfo`

Avec l'application de ce template, les informations générales sur un produit sont affichées: le nom, la description et les caractéristiques générales du produit. Seules les trois premières caractéristiques globales du produit sont affichées.

De plus, un attribut `id` pour permettre de pointer sur le produit lors de la création du sommaire est ajouté au bloc de texte contenant le nom du produit. Deux paramètres sont utilisés pour définir la taille de la police utilisée pour le nom du produit et pour la description. Ces paramètres se trouvent dans le tableau 4.6.

Paramètre	Valeur exemple	Description
<code>NameSize</code>	<code>'18pt'</code>	taille du nom du produit
<code>DescriptionSize</code>	<code>'11pt'</code>	taille de la description

TAB. 4.6 – Paramètres du template `DisplayProductInfo`

### Template `DisplayImage`

L'insertion de l'image pose plusieurs problèmes de réduction de la taille de l'image. Afin de limiter la hauteur de l'image, le template est utilisé avec un paramètre `MaxHeight` qui donne la hauteur maximale autorisée pour l'image, hauteur précisée avec l'attribut `height` de l'élément `fo:external-graphic`. Un autre attribut, `scaling` utilisé avec la valeur `"uniform"` permet de garder les proportions de l'image malgré la limitation de sa hauteur. Une variable est utilisée par ce template pour récupérer le chemin d'accès à l'image.

### Template `SetColumnsItems`

Les largeurs relatives des colonnes de la liste des déclinaisons du produit sont définies par ce template. Les rapports sont les suivants : la colonne contenant les options de chaque déclinaison est quatre fois plus large que celle du numéro et que celle du prix.

### Template `EmptyLinesLoop`

Effectuer des boucles de type "pour `i` allant de `x` à `y`" n'est pas une possibilité offerte directement par XSLT. Pour pallier à cette absence, un appel récursif à un template est effectué. Le template `EmptyLinesLoop` est appelé avec deux paramètres : `i` et `count`. Le template produit un rang de

tableau vide à chaque boucle. Chaque cellule de ce rang contient un espace insécable dans le but de forcer l’affichage du rang. Un exemple des lignes vides produites est donné à la page 35 sur le deuxième produit de l’illustration en bas à gauche.

## 4.4 Utilisation

### 4.4.1 Généralités

La feuille de style contient toutes les informations nécessaires au traitement par Xalan [11]. Une fois la transformation effectuée, le résultat doit être traité par FOP. Pour l’utilisateur final, la production du catalogue en PDF peut être facilement rendue transparente, grâce aux possibilités offertes par le serveur Tomcat utilisé par *eSarine*.

### 4.4.2 Valeurs des paramètres pour la mise en page

Les paramètres globaux et ceux de certains templates permettent à l’utilisateur de personnaliser son catalogue PDF au niveau de la mise en forme. Les remarques faites ci-après sont à prendre en compte lors du choix des valeurs.

- La taille des pages est configurable et doit être exprimée en *mm* comme le montrent les valeurs exemples du tableau 4.1.
- Les marges de la page sont également modifiables selon le résultat désiré. Les imprimantes ayant souvent des marges d’impression minimales, il est préférable de ne pas réduire les marges de la page à zéro.
- Les marges du corps de la page ne doivent pas être inférieures à 10mm pour assurer un fonctionnement correct des en-têtes et des pieds de page. En tout les cas, elles ne peuvent pas être inférieures à 5mm.
- Les tailles des polices et celles des bordures sont données en points (pt) dans les exemples mais d’autres unités sont utilisables. Pour plus d’informations, se référer à la documentation de FOP [4].
- Les couleurs sont, dans les exemples, décrites au format RGB<sup>3</sup> ou par leur nom. D’autres possibilités sont également offertes, se référer à la documentation de FOP.

Voici les éléments principaux que nous avons estimé utiles de rendre paramétrables. Plusieurs autres éléments de la feuille de style sont modifiables pour une personnalisation plus poussée au sein même du code de cette feuille de style<sup>4</sup> mais leur modification demande une attention certaine à d’éventuels effets de bord.

Vu la grande diversité des formats et des produits pouvant être proposés, un compromis a été fait entre un environnement totalement paramétrable

<sup>3</sup>Format de codage de couleur, basé sur les trois couleurs primaires

<sup>4</sup>par exemple le nombre de blocs produit par page

et une structure plus rigide. L'offre de personnalisation du catalogue PDF permet de produire une mise en page attrayante avec la modification de quelques paramètres.

#### 4.4.3 Valeurs des paramètres pour les données

En ce qui concerne les données, les deux paramètres modifiables sont les seuils limitant le nombre de déclinaisons du produit à afficher dans le cas d'un *petit* produit et d'un *grand* produit. La seule limitation pour les valeurs de ces paramètres est que `BigProductBlockThreshold` doit être supérieur à la limite pour un *petit* produit, `SmallProductBlockThreshold`.

En conclusion, le paramétrage des différents éléments de mise en page et des données doit se faire de manière coordonnée afin d'assurer le meilleur rendu possible. Il est illusoire de vouloir afficher les vingt déclinaisons d'un produit sur une page d'un petit format (comme A5). Loin d'être une limitation propre à ce système de mise en page, il s'agit plutôt d'une contrainte générale à toute entreprise de mise en page d'informations. Dans tous les cas, des tests doivent être effectués avant de pouvoir mettre en production un catalogue généré dynamiquement.

Les valeurs par défaut sont les valeurs données en exemple dans les différents tableaux présentant les paramètres tout au long de ce chapitre.

## 4.5 Limites de FOP

La version utilisée de FOP propose un support partiel de la recommandation XSL-FO [18]. Ce support limité de la norme, impose des limites dans l'utilisation de FOP. La suite de cette section sera consacrée à la présentation de certaines limitations rencontrées ainsi qu'aux solutions développées pour y pallier.

Les attributs `keep-together`, `keep-with-next` et `keep-with-previous` n'étant supportés que dans l'élément `fo:table-row`, les tableaux de description d'un produit et celui de la liste des déclinaisons sont enveloppés dans un tableau global. Ceci permet de garantir, dans la mesure du possible, que ces tableaux soient affichés ensemble sur la même page.

Pour limiter la taille des images afin de les mettre à l'échelle pour l'affichage, la définition d'une hauteur pour l'image est nécessaire sans pouvoir utiliser l'attribut `max-height` qui n'est pas supporté. Cette solution a le défaut d'agrandir éventuellement une image très petite.

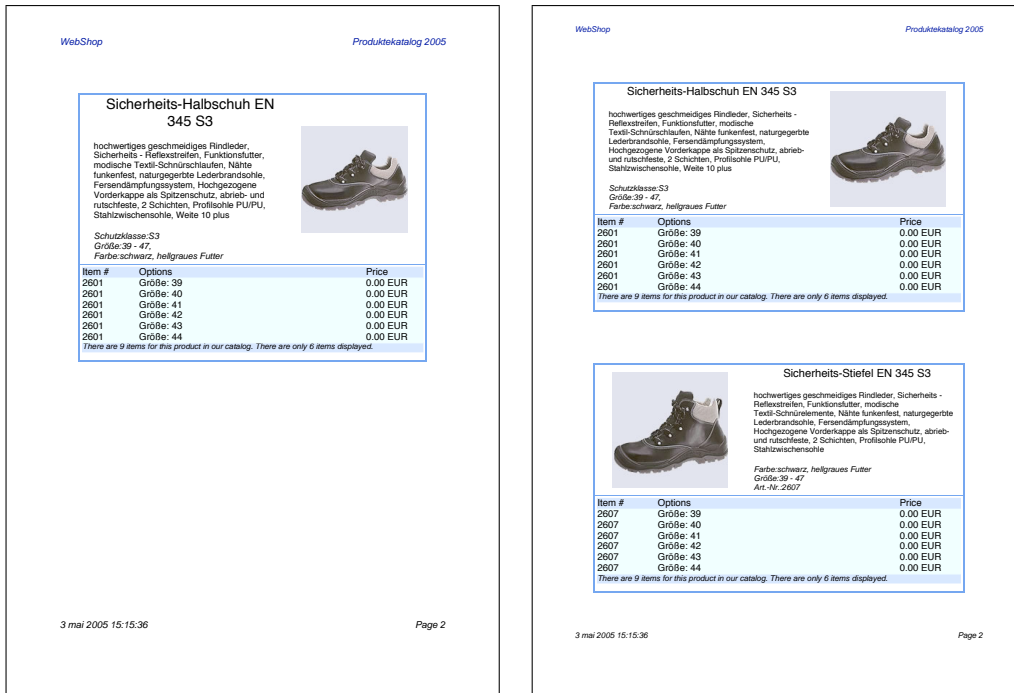
La principale limite imposée par FOP pour ce travail se rattache à la gestion du texte. En effet, il n'est pas possible de connaître à priori la taille occupée par du texte et donc de prévoir un espace suffisant pour ce texte.

Une solution possible est de limiter le nombre de lignes pour un texte mais il n'est pas possible de cacher le texte dépassant la zone réservée avec les attributs `overflow` et `clip` car ils ne sont pas implémentés. Ces attributs

permettent de gérer les débordements dus à un texte trop long par exemple. Après l'étude de plusieurs solutions, l'option choisie est de ne pas couper arbitrairement du texte afin de ne pas rendre le contenu incompréhensible. Cela a pour effet d'augmenter la taille des blocs de produit si le texte nécessite un espace supérieur à la taille réservée pour ces blocs. Il est donc possible dans certains cas qu'un bloc produit soit affiché sur deux pages. Il peut également arriver qu'un nombre plus réduit de produits soit affiché sur la page que celui prévu comme le montre l'illustration gauche de la figure 4.5. Il existe plusieurs possibilités pour obtenir une mise en page plus satisfaisante – illustrée par la partie droite de la figure 4.5 – si le résultat obtenu ne convient pas:

- diminuer les tailles de police pour le nom du produit et la description,
- diminuer les marges,
- réduire la description du produit ou le nom du produit dans le fichier XML *eSarine*.

Dans l'exemple illustré, ce sont les deux premières solutions qui ont été utilisées.



Avant mise au point des paramètres

Après mise au point des paramètres

FIG. 4.5 – Démonstration de mise au point de la mise en page

La gestion des tableaux offerte par FOP a rendu indispensable la définition de leur taille pour permettre un affichage correct en écartant tout risque

de débordement des tableaux en dehors des zones d'écriture.

De manière générale, l'implémentation partielle de XSL-FO par FOP nécessite de trouver des solutions détournées pour offrir certaines fonctionnalités.

Décrire de manière exhaustive les limites de FOP n'est pas possible ici car cela reviendrait à écrire une feuille de style idéale selon la recommandation puis expliquer chaque modification effectuée pour la rendre compatible avec FOP. Ce n'est pas le principe adopté lors du développement de la feuille de style car elle a été, dès le départ, pensée comme devant fonctionner avec FOP.

## 4.6 Performances

Afin de réduire le temps de traitement, la possibilité de trier les noeuds par l'intermédiaire de la feuille de style n'a pas été utilisée. Ce tri demande un temps de traitement bien trop important et la solution du fichier déjà trié auparavant est bien moins coûteuse en ressources.

Pour la production d'un catalogue de 8 pages pour 7 produits le temps de traitement est d'environ 45 secondes sur avec la machine de développement, la transformation demande des ressources importantes. Les performances peuvent être améliorées avec un serveur dédié et optimisé pour l'utilisation de FOP.

Dans la situation actuelle, l'outil convient mieux à une production semi-dynamique du catalogue. Au regard du temps de traitement, la production d'un catalogue personnalisé pour chaque utilisateur à la volée n'est pas envisageable sans une machine offrant des performances accrues. Par contre, un catalogue mis à jour une ou même plusieurs fois par jour convient bien vu les performances. De plus le catalogue des produits n'est, en principe, pas susceptible de changer plus d'une à deux fois par jour. Avant une mise en production des tests de charge en conditions réelles sont encore nécessaires.

# Conclusion

La feuille de style développée au cours de ce travail permet de produire de manière dynamique un catalogue des produits présents dans le webshop *eSarine*. Le catalogue s'adapte à tout produit décrit et à tout format de catalogue. La présentation des produits se présente sous un format comportant une description et une image accompagnées d'une liste des déclinaisons disponibles du produit.

## Améliorations et développements

Les nombreuses possibilités de personnalisation de la feuille de style oblige l'utilisateur à modifier directement les paramètres dans la feuille de style. Une interface plus conviviale pour l'administration est une des améliorations à apporter au module afin de l'intégrer totalement à *eSarine*.

Le module développé pour *eSarine* n'est pas encore totalement intégré à la version distribuée mais sera intégré avec relativement peu d'efforts supplémentaires.

Les limites imposées par la version actuelle de FOP ont conduit à certains choix d'implémentation et de présentation. Tout en gardant la structure et les grands principes de ce travail, l'utilisation de la nouvelle version de FOP (voir l'annexe A) permettra d'améliorer la gestion des chaînes de caractères et les performances entre autres.

Une des difficultés rencontrées lors de ce travail est l'absence de format d'exportation en XML défini par un schéma XML dans l'environnement *eSarine*. Le format présenté au chapitre 3 est donc susceptible de changer. En cas de modification du format, un certain travail d'adaptation devrait se faire au niveau de la feuille de style.

Dans la même optique de création d'un catalogue de produits, le développement d'une solution produisant des bulletins de commande PDF permettant une impression pour archivage ou pour envoi est un module qui pourrait compléter l'offre d'*eSarine*

## Conclusions personnelles

Le développement de la feuille de style a nécessité une étude approfondie des technologies XML et surtout de XSL. Ces technologies sont prometteuses mais force est de constater que les outils actuellement disponibles ne sont pas toujours arrivés à maturité pour une utilisation dans un environnement de production.

La solution proposée repose sur des bases conceptuelles solides. Les problèmes rencontrés ont été le plus souvent au niveau de la mise en page et découlaient de l'implémentation partielle de la recommandation XSL-FO. En gardant la structure de la feuille de style il sera aisé de modifier la mise en page des produits présentés dans le catalogue.

## Annexe A

# Environnement de développement et outils

Le développement de la feuille de style a été réalisé dans le système d'exploitation Mac OS X 10.3.

### A.1 Xalan-J version 2.5.1

C'est la version qui a réalisé toutes les transformations XSL. Elle est disponible sur le site donné en référence [11].

### A.2 FOP version 0.20.5

La version 0.20.5 utilisée est un version de maintenance dont la dernière révision date du mois de juillet 2003. Le projet est en train de travailler à la version 1.0 mais la version disponible n'est qu'une version DR (development release). La nouvelle version devrait être disponible dans le courant du deuxième semestre 2005.

### A.3 Oxygen version 5.1

*Oxygen* est un éditeur XML et un débogueur XSL/XSLT qui permet l'automatisation des processus de transformation. Il est édité par la société *SyncRO Soft* et un site internet lui est consacré: <http://www.oxygenxml.com/>.

### A.4 TestXSLT version 3.1

*TestXSLT* est une application pour Mac OS X développée par Marc Liyanage et disponible à l'adresse suivante: <http://www.entropy.ch/software/macosx/>. Cette application propose une interface simple pour effectuer des transformations XSL.

## Annexe B

# Code de la feuille de style

Le code des templates de la feuille de style est donné ci-après. Ce code est documenté par l'intermédiaire d'une feuille de style spécialement dédiée qui produit du code HTML. La documentation HTML ainsi que le code complet de la feuille de style se trouve sur le CD accompagnant ce travail.

### B.1 Variables du template **BigProductBlock**

Variable	Valeur
SpaceAfterBigProductBlock	<code>concat(\$BodyVerticalWriteArea * 0.2 div 2, 'mm')</code>
HeightItemsBigProductBlock	<code>concat((\$BigProductBlockThreshold + 2) * 5, 'mm')</code>
HeightBigProductBlock	<code>concat(\$BodyVerticalWriteArea * 0.8 div 2, 'mm')</code>
HeightDescriptionBigProductBlock	<code>concat( substring-before(\$HeightBigProductBlock, 'mm' ) - substring-before(\$HeightItemsBigProductBlock, 'mm' ) , 'mm')</code>
OccurencesCounter	<code>count(occurences/occurrence)</code>
currentItem	<code>item-number</code>
currentproductTypeFieldId	<code>@productTypeFieldId</code>

TAB. B.1 – Variables du template **BigProductBlock**

### B.2 Template **Main**

```
<xsl:template focus:comment="Main template" match="/" name="Main">  
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format"  
    font-family="{ $FontFamily }" font-size="{ $GlobalFontSize }">  
    <fo:layout-master-set>
```

ANNEXE B. CODE DE LA FEUILLE DE STYLE

---

```

89 <fo:simple-page-master margin-bottom="{ $PageMarginBottom }"
    margin-left="{ $PageMarginLeft }" margin-right="{ $PageMarginRight }"
    margin-top="{ $PageMarginTop }" master-name="Paper"
    page-height="{ $PageHeight }"
    page-width="{ $PageWidth }">
    <xsl:variable focus:comment="Automatic calculation"
        name="RegionBeforeExtent"
        select="concat(substring-before($BodyMarginTop, 'mm') - 5,
            'mm')"/>
    <xsl:variable focus:comment="Automatic calculation"
        name="RegionAfterExtent"
        select="concat(substring-before($BodyMarginBottom, 'mm') - 5,
            'mm')"/>
    <xsl:variable focus:comment="Automatic calculation"
        name="RegionStartExtent"
94     select="concat(substring-before($BodyMarginLeft, 'mm') - 5,
            'mm')"/>
    <xsl:variable focus:comment="Automatic calculation"
        name="RegionEndExtent"
        select="concat(substring-before($BodyMarginRight, 'mm') - 5,
            'mm')"/>
    <fo:region-body background-color="{ $PageBackgroundColor }"
        margin-bottom="{ $BodyMarginBottom }"
        margin-left="{ $BodyMarginLeft }"
99     margin-right="{ $BodyMarginRight }"
        margin-top="{ $BodyMarginTop }"/>
    <fo:region-before background-color="{ $PageBackgroundColor }"
        extent="{ $RegionBeforeExtent }"/>
    <fo:region-after background-color="{ $PageBackgroundColor }"
        extent="{ $RegionAfterExtent }"/>
104 <fo:region-start background-color="transparent"
        extent="{ $RegionStartExtent }"/>
    <fo:region-end background-color="transparent"
        extent="{ $RegionEndExtent }"/>
</fo:simple-page-master>
<fo:simple-page-master margin-bottom="{ $PageMarginBottom }"
    margin-left="{ $PageMarginLeft }" margin-right="{ $PageMarginRight }"
109     margin-top="{ $PageMarginTop }" master-name="PaperBlank"
    page-height="{ $PageHeight }"
    page-width="{ $PageWidth }"/>
<fo:page-sequence-master master-name="Page">
    <fo:repeatable-page-master-alternatives>
        <fo:conditional-page-master-reference master-reference="Paper"
114         page-position="first"/>
        <fo:conditional-page-master-reference master-reference="Paper"
            page-position="rest"/>
        <fo:conditional-page-master-reference blank-or-not-blank="blank"
            master-reference="PaperBlank"/>
119     </fo:repeatable-page-master-alternatives>
    </fo:page-sequence-master>
</fo:layout-master-set>
<xsl:call-template name="TitlePage"/>
<fo:page-sequence initial-page-number="1" master-reference="Page">

```

```

124     <focas:comment>The initial page number has an effect on the layout: an
        odd page will
        always be on the right in a book, similar behavior
        here</focas:comment>
    <fo:static-content color="{ $HeaderTextColor }"
        flow-name="xsl-region-before"
        font-style="italic">
    <fo:block text-align-last="justify">
129         <xsl:value-of select="/catalog/company/name"/>
        <fo:leader/>
        <xsl:value-of select="/catalog/title/main"/>
    </fo:block>
    </fo:static-content>
134 <fo:static-content flow-name="xsl-region-after" font-style="italic">
    <fo:block text-align-last="justify">
        <xsl:value-of select="/catalog/title/date"/>
        <fo:leader/> Page <fo:page-number/>
    </fo:block>
139 </fo:static-content>
    <fo:flow flow-name="xsl-region-body">
        <xsl:call-template name="TOC"/>
        <xsl:for-each select="//product">
            <xsl:variable name="OccurrencesCounter"
                select="count(occurrences/occurrence)"/>
144         <xsl:choose>
            <xsl:when test="$OccurrencesCounter <=
                $SmallProductBlockThreshold">
                <xsl:call-template name="SmallProductBlock"/>
            </xsl:when>
            <xsl:otherwise>
149         <xsl:call-template name="BigProductBlock"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:for-each>
    </fo:flow>
154 </fo:page-sequence>
    </fo:root>
</xsl:template>

```

Listing B.1 – Code du template Main

### B.3 Template TitlePage

```

<xsl:template focas:comment="Creation of title page, including special footer"
    name="TitlePage">
    <xsl:param focas:comment="Size of title" name="TitleSize" select="'30pt'"/>
    <xsl:param focas:comment="Size of Company name on title Page"
        name="TitleCompanyNameSize"
160     select="'24pt'"/>
    <xsl:param focas:comment="Color of Title and subtitle" name="TitleColor"
        select="'blue'"/>

```

```
<fo:page-sequence master-reference="Page">
  <fo:static-content flow-name="xsl-region-after">
    <fo:block text-align="center">
165     <xsl:value-of select="/catalog/title/copyright"/>
    </fo:block>
  </fo:static-content>
  <fo:flow flow-name="xsl-region-body">
    <fo:block color="{TitleColor}" font-size="{TitleSize}"
170     font-style="italic"
    space-after="3cm" text-align="center">
    <xsl:value-of select="/catalog/title/main"/>
    </fo:block>
    <fo:block break-after="page" color="{TitleColor}"
    font-size="{TitleCompanyNameSize}"
175     space-after="3cm" text-align="center">
    <xsl:value-of select="/catalog/company/name"/>
    </fo:block>
    <fo:block font-size="18pt" font-style="italic">
    <xsl:value-of select="/catalog/editorial"/>
    </fo:block>
180  </fo:flow>
  </fo:page-sequence>
</xsl:template>
```

Listing B.2 – Code du template TitlePage

## B.4 Template TOC

```
<xsl:template focus:comment="Make table of contents" name="TOC">
  <fo:block font-size="14pt" font-weight="bold" space-after="5pt">Table of
  Contents</fo:block>
185  <fo:block break-after="page">
    <xsl:for-each select="//product">
      <fo:block text-align-last="justify">
        <fo:basic-link internal-destination="{generate-id(.)}">
          <xsl:value-of select="name"/>
190          <fo:leader leader-pattern="dots"/>
          <fo:page-number-citation ref-id="{generate-id(.)}"/>
        </fo:basic-link>
      </fo:block>
    </xsl:for-each>
195  </fo:block>
</xsl:template>
```

Listing B.3 – Code du template TOC

## B.5 Template SmallProductBlock

ANNEXE B. CODE DE LA FEUILLE DE STYLE

---

```

<xsl:template focas:comment="Draw a block for a product with a few items (3 per
pages)"
name="SmallProductBlock">
200 <xsl:variable name="SpaceAfterSmallProductBlock"
select="concat($BodyVerticalWriteArea * 0.2 div 3, 'mm')"/>
<xsl:variable name="HeightItemsSmallProductBlock"
select="concat(($SmallProductBlockThreshold + 1) * 5, 'mm')"/>
<xsl:variable name="HeightSmallProductBlock"
205 <xsl:variable name="HeightDescriptionSmallProductBlock"
select="concat(substring-before($HeightSmallProductBlock, 'mm') -
substring-before($HeightItemsSmallProductBlock, 'mm'), 'mm')"/>
<fo:table table-layout="fixed" space-after="{SpaceAfterSmallProductBlock}"
width="100%"
height="{HeightSmallProductBlock}">
210 <fo:table-column column-width="proportional-column-width(1)"/>
<fo:table-column column-width="{BodyHorizontalWriteAreaString}"/>
<fo:table-column column-width="proportional-column-width(1)"/>
<fo:table-body>
<fo:table-row keep-with-next="always">
215 <fo:table-cell>
<fo:block>&#160;</fo:block>
</fo:table-cell>
<fo:table-cell>
<fo:table height="{HeightDescriptionSmallProductBlock}"
border-after-color="{BorderColor}"
border-after-style="solid"
220 border-after-width="0pt" border-before-color="{BorderColor}"
border-before-style="solid"
border-before-width="{BorderWidth}"
border-left-color="{BorderColor}" border-left-style="solid"
border-left-width="{BorderWidth}"
border-right-color="{BorderColor}"
border-right-style="solid"
border-right-width="{BorderWidth}"
225 padding-after="3pt" padding-before="3pt" padding-left="3pt"
padding-right="3pt">
<xsl:call-template name="SetColumnsProductDescription"/>
<fo:table-body background-color="{BackgroundBodyColor}">
<fo:table-row>
<xsl:choose>
230 <xsl:when test="(position() mod 2)= 0">
<xsl:call-template name="DisplayImage">
<xsl:with-param name="MaxHeight"
select="$HeightDescriptionSmallProductBlock"/>
</xsl:call-template>
235 <xsl:call-template name="DisplayProductInfo"/>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="DisplayProductInfo"/>
<xsl:call-template name="DisplayImage">
240 <xsl:with-param name="MaxHeight"
select="$HeightDescriptionSmallProductBlock"/>
</xsl:call-template>

```

```

                </xsl:otherwise>
            </xsl:choose>
245         </fo:table-row>
        </fo:table-body>
    </fo:table>
</fo:table-cell>
<fo:table-cell>
250     <fo:block>&#160;</fo:block>
</fo:table-cell>
</fo:table-row>
<fo:table-row>
    <fo:table-cell>
255     <fo:block>&#160;</fo:block>
</fo:table-cell>
<fo:table-cell>
    <fo:table height="{ $HeightItemsSmallProductBlock }"
        border-after-color="{ $BorderColor }"
        border-after-style=" solid "
260     border-after-width="{ $BorderWidth }"
        border-before-color="{ $BorderColor }"
        border-before-style=" solid " border-before-width="0 pt"
        border-left-color="{ $BorderColor }" border-left-style=" solid "
        border-left-width="{ $BorderWidth }"
        border-right-color="{ $BorderColor }"
        border-right-style=" solid "
        border-right-width="{ $BorderWidth }"
        padding-after="3pt" padding-before="3pt" padding-left="3pt"
        padding-right="3pt" font-size="12pt">
    <xsl:call-template name="SetColumnsItems"/>
    <fo:table-header background-color="{ $BackgroundHeaderColor }">
        <fo:table-row>
            <fo:table-cell>
270             <fo:block>Item #</fo:block>
            </fo:table-cell>
            <fo:table-cell>
                <fo:block>Options</fo:block>
            </fo:table-cell>
275             <fo:table-cell>
                <fo:block>Price</fo:block>
            </fo:table-cell>
        </fo:table-row>
    </fo:table-header>
280 </fo:table-body>
    <xsl:variable name="OccurencesCounter"
        select="count(occurences/occurrence)"/>
    <xsl:for-each select="occurences/occurrence">
        <fo:table-row keep-with-previous="always"
            background-color="{ $BackgroundCellColor }">
285         <fo:table-cell>
            <fo:block>
                <xsl:variable name="currentItem"
                    select="item-number"/>
                <xsl:value-of select="item-number"/>
            </fo:block>
        </fo:table-cell>
    </fo:table-row>
</fo:table>

```

```

290         </fo:table-cell>
        <fo:table-cell>
            <xsl:for-each select="value">
                <xsl:variable name="currentproductTypeFieldId"
295                 select="@productTypeFieldId"/>
                <fo:block>
                    <xsl:value-of
                        select=" ../.. / occurrences-header / option-value [ @product
                        />: <xsl:value-of select="text ()" />
                    </fo:block>
                </xsl:for-each>
            </fo:table-cell>
            <fo:table-cell>
                <fo:block>
                    <xsl:value-of select="price"/>
305                </fo:block>
            </fo:table-cell>
        </fo:table-row>
    </xsl:for-each>
    <xsl:call-template name="EmptyLinesLoop">
310        <xsl:with-param name="i">1</xsl:with-param>
        <xsl:with-param name="count"
            select="$SmallProductBlockThreshold -
                $OccurrencesCounter" />
    </xsl:call-template>
    </fo:table-body>
315 </fo:table>
    </fo:table-cell>
    <fo:table-cell>
        <fo:block>&#160;</fo:block>
    </fo:table-cell>
320 </fo:table-row>
    </fo:table-body>
    </fo:table>
</xsl:template>

```

Listing B.4 – Code du template SmallProductBlock

## B.6 Template BigProductBlock

```

<xsl:template focus:comment="Draw a block for a product with more items (2 per
page)"
325 name="BigProductBlock">
    <xsl:variable name="SpaceAfterBigProductBlock"
        select="concat($BodyVerticalWriteArea * 0.2 div 2, 'mm')"/>
    <xsl:variable name="HeightItemsBigProductBlock"
        select="concat(($BigProductBlockThreshold + 2) * 5, 'mm')"/>
330 <xsl:variable name="HeightBigProductBlock"
        select="concat($BodyVerticalWriteArea * 0.8 div 2, 'mm')"/>
    <xsl:variable name="HeightDescriptionBigProductBlock"

```

## ANNEXE B. CODE DE LA FEUILLE DE STYLE

---

```

select="concat(substring-before($HeightBigProductBlock, 'mm') -
  substring-before($HeightItemsBigProductBlock, 'mm'), 'mm')"/>
<fo:table table-layout="fixed" space-after="{ $SpaceAfterBigProductBlock}"
width="100%"
335 height="{ $HeightBigProductBlock}">
<fo:table-column column-width="proportional-column-width(1)"/>
<fo:table-column column-width="{ $BodyHorizontalWriteAreaString}"/>
<fo:table-column column-width="proportional-column-width(1)"/>
<fo:table-body>
340 <fo:table-row keep-with-next="always">
<fo:table-cell>
<fo:block>&#160;</fo:block>
</fo:table-cell>
<fo:table-cell>
345 <fo:table height="{ $HeightDescriptionBigProductBlock}"
border-after-color="{ $BorderColor}"
border-after-style="solid"
border-after-width="0pt" border-before-color="{ $BorderColor}"
border-before-style="solid"
border-before-width="{ $BorderWidth}"
border-left-color="{ $BorderColor}" border-left-style="solid"
350 border-left-width="{ $BorderWidth}"
border-right-color="{ $BorderColor}"
border-right-style="solid"
border-right-width="{ $BorderWidth}"
padding-after="3pt" padding-before="3pt" padding-left="3pt"
padding-right="3pt">
<xsl:call-template name="SetColumnsProductDescription"/>
<fo:table-body background-color="{ $BackgroundBodyColor}">
355 <fo:table-row>
<xsl:choose>
<xsl:when test="(position() mod 2)= 0">
<xsl:call-template name="DisplayImage">
<xsl:with-param name="MaxHeight"
360 select="$HeightDescriptionBigProductBlock"/>
</xsl:call-template>
<xsl:call-template name="DisplayProductInfo"/>
</xsl:when>
<xsl:otherwise>
365 <xsl:call-template name="DisplayProductInfo"/>
<xsl:call-template name="DisplayImage">
<xsl:with-param name="MaxHeight"
select="$HeightDescriptionBigProductBlock"/>
</xsl:call-template>
370 </xsl:otherwise>
</xsl:choose>
</fo:table-row>
</fo:table-body>
</fo:table>
375 </fo:table-cell>
<fo:table-cell>
<fo:block>&#160;</fo:block>
</fo:table-cell>
</fo:table-row>

```

```

380     <fo:table-row>
        <fo:table-cell>
            <fo:block>&#160;</fo:block>
        </fo:table-cell>
        <fo:table-cell>
385     <fo:table height="{ $HeightItemsBigProductBlock }"
            border-after-color="{ $BorderColor }"
                border-after-style=" solid "
                border-after-width="{ $BorderWidth }"
                border-before-color="{ $BorderColor }"
                border-before-style=" solid " border-before-width="0 pt"
                border-left-color="{ $BorderColor }" border-left-style=" solid "
390     border-left-width="{ $BorderWidth }"
                border-right-color="{ $BorderColor }"
                border-right-style=" solid "
                border-right-width="{ $BorderWidth }"
                padding-after="3pt" padding-before="3pt" padding-left="3pt"
                padding-right="3pt" font-size="12pt">
        <xsl:variable name="OccurencesCounter"
            select="count(occurences/occurrence)"/>
        <xsl:call-template name="SetColumnsItems"/>
395     <fo:table-header background-color="{ $BackgroundHeaderColor }">
        <fo:table-row>
            <fo:table-cell>
                <fo:block>Item #</fo:block>
            </fo:table-cell>
            <fo:table-cell>
                <fo:block>Options</fo:block>
            </fo:table-cell>
            <fo:table-cell>
                <fo:block>Price</fo:block>
            </fo:table-cell>
400     </fo:table-row>
        </fo:table-header>
        <!-- Warning: The specs of fop clearly state that you have to
            put table-footer before table-body-->
        <xsl:call-template name="DisplayTableFooterAlarmBigBlock">
            <xsl:with-param name="total" select="$OccurencesCounter"/>
410     </xsl:call-template>
        <fo:table-body>
            <xsl:for-each
                select="occurences/occurrence[position() &lt;=
                    $BigProductBlockThreshold]">
415     <fo:table-row keep-with-previous="always"
                background-color="{ $BackgroundCellColor }">
            <fo:table-cell>
                <fo:block>
                    <xsl:variable name="currentItem"
                        select="item-number"/>
420     <xsl:value-of select="item-number"/>
                </fo:block>
            </fo:table-cell>
            <fo:table-cell>
                <xsl:for-each select="value">

```

```

425         <xsl:variable name="currentproductTypeFieldId"
           select="@productTypeFieldId" />
         <fo:block>
           <xsl:value-of
430             select=" ../.. / occurrences-header / option-value [ @product
           />: <xsl:value-of select="text ()" />
           </fo:block>
         </xsl:for-each>
       </fo:table-cell>
       <fo:table-cell>
435         <fo:block>
           <xsl:value-of select="price" />
         </fo:block>
       </fo:table-cell>
     </fo:table-row>
440 </xsl:for-each>
     <xsl:call-template name="EmptyLinesLoop">
       <xsl:with-param name="i">1</xsl:with-param>
       <xsl:with-param name="count"
445         select="$BigProductBlockThreshold -
           $OccurrencesCounter" />
     </xsl:call-template>
   </fo:table-body>
 </fo:table>
 </fo:table-cell>
 <fo:table-cell>
450   <fo:block>&#160;</fo:block>
 </fo:table-cell>
 </fo:table-row>
 </fo:table-body>
 </fo:table>
455 </xsl:template>

```

Listing B.5 – Code du template BigProductBlock

## B.7 Template EmptyLinesLoop

```

<xsl:template focus:comment="Draw empty lines at the end of a table"
  name="EmptyLinesLoop">
  <xsl:param name="i" />
  <xsl:param name="count" />
460 <xsl:if test="$i &lt;= $count">
    <!-- body of the loop goes here -->
    <!-- &#160; is the unicode for a nbsp -->
    <fo:table-row keep-with-previous="always"
      background-color="{ $BackgroundCellColor }">
      <fo:table-cell>
        <fo:block>&#160;</fo:block>
465      </fo:table-cell>
      <fo:table-cell>
        <fo:block>&#160;</fo:block>

```

```
470     </fo:table-cell>
        <fo:table-cell>
            <fo:block>&#160;</fo:block>
        </fo:table-cell>
    </fo:table-row>
</xsl:if>
475 <xsl:if test="$i <= $count">
    <xsl:call-template name="EmptyLinesLoop">
        <xsl:with-param name="i">
            <!-- Increment index-->
            <xsl:value-of select="$i + 1" />
        </xsl:with-param>
480     <xsl:with-param name="count">
        <xsl:value-of select="$count" />
    </xsl:with-param>
    </xsl:call-template>
</xsl:if>
485 </xsl:template>
```

Listing B.6 – Code du template EmptyLinesLoop

## B.8 Template DisplayTableFooterAlarm

```
<xsl:template focus:comment="Draw a warning rank at the end of a table"
name="DisplayTableFooterAlarmBigBlock">
  <xsl:param name="total" />
  <fo:table-footer background-color="{ $BackgroundFooterColor }">
    <fo:table-row>
490     <fo:table-cell number-columns-spanned="3">
        <xsl:choose>
            <!-- if there more than BigProductBlockThreshold product types,
                 only BigProductBlockThreshold items will be displayed -->
            <xsl:when test="$total > $BigProductBlockThreshold">
                <fo:block font-size="10pt" font-style="italic"> There are
495                 <xsl:value-of
                    select="$total" /> items for this product in our
                    catalog. There are only
                    <xsl:value-of select="$BigProductBlockThreshold" />
                    items displayed.
                </fo:block>
            </xsl:when>
            <xsl:otherwise />
500     </xsl:choose>
        </fo:table-cell>
    </fo:table-row>
  </fo:table-footer>
</xsl:template>
```

Listing B.7 – Code du template DisplayTableFooterAlarm

## B.9 Template DisplayImage

```
505 <xsl:template focus:comment="Insert product image" name="DisplayImage">
    <xsl:param name="MaxHeight" />
    <fo:table-cell display-align="center">
        <fo:block text-align="center">
            <xsl:variable focus:comment="Path to the image" name="image-path"
            select="img-path" />
510     <fo:external-graphic height="{ $MaxHeight }" scaling="uniform"
            src="{ $image-path }" />
        </fo:block>
    </fo:table-cell>
</xsl:template>
```

Listing B.8 – Code du template DisplayImage

## B.10 Template DisplayProductInfo

```
<xsl:template focus:comment="Display information about product"
name="DisplayProductInfo">
515 <xsl:param focus:comment="Size of name" name="NameSize" select="'18pt'" />
<xsl:param focus:comment="Size of description" name="DescriptionSize"
select="'11pt'" />
<fo:table-cell padding-left="5mm">
    <fo:block font-size="{ $NameSize }" id="{ generate-id(.) }"
    text-align="center">
        <xsl:value-of select="name" />
520 </fo:block>
    <fo:block font-size="{ $DescriptionSize }" wrap-option="wrap"
    space-after="5mm" space-before="5mm">
        <xsl:value-of select="description" />
    </fo:block>
    <xsl:for-each select="product-type-field">
525 <xsl:choose>
        <xsl:when test="position() &lt;= 3">
            <fo:block font-size="{ $DescriptionSize }" wrap-option="wrap"
            font-style="italic">
                <xsl:value-of select="@name" /> : <xsl:value-of
                select="text()" />
            </fo:block>
530 </xsl:when>
            <xsl:otherwise/>
        </xsl:choose>
    </xsl:for-each>
</fo:table-cell>
535 </xsl:template>
```

Listing B.9 – Code du template DisplayProductInfo

## B.11 Templates SetColumns...

```
<xsl:template focus:comment="Set columns relative size for description"
  name="SetColumnsProductDescription">
  <xsl:choose>
    <xsl:when test="(position() mod 2)= 0">
      <fo:table-column column-width="proportional-column-width(1)"/>
540     <fo:table-column column-width="proportional-column-width(1.5)"/>
    </xsl:when>
    <xsl:otherwise>
      <fo:table-column column-width="proportional-column-width(1.5)"/>
545     <fo:table-column column-width="proportional-column-width(1)"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
<xsl:template focus:comment="Set columns relative size for items list"
  name="SetColumnsItems">
550 <fo:table-column column-width="proportional-column-width(1)"/>
  <fo:table-column column-width="proportional-column-width(4)"/>
  <fo:table-column column-width="proportional-column-width(1)"/>
</xsl:template>
```

Listing B.10 – Code des templates SetColumns...

## Annexe C

# Exemple *Parlement*, code et rendu

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3   <fo:layout-master-set>
      <fo:simple-page-master master-name="page" page-height="297mm"
        page-width="210mm"
        margin-left="15mm" margin-right="15mm" margin-top="15mm"
        margin-bottom="15mm">
        <fo:region-body margin-top="3.5cm" />
        <fo:region-before extent="3cm" />
8     </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="page">
    <fo:static-content flow-name="xsl-region-before">
      <fo:block text-align="justify">
13      <fo:external-graphic height="3cm" src="suisse.jpg" scaling="uniform" />
      <fo:leader />
      <fo:external-graphic height="3cm" src="palais.jpg" scaling="uniform" />
    </fo:block>
    </fo:static-content>
18    <fo:static-content flow-name="xsl-footnote-separator">
      <fo:block>
        <fo:leader leader-pattern="rule" leader-length="100%"
          rule-thickness="0.5pt"
          rule-style="solid" color="black" />
      </fo:block>
23    </fo:static-content>
    <fo:flow flow-name="xsl-region-body">
      <fo:block font-size="16pt" font-family="sans-serif" color="grey"
        space-before="5mm"
        space-after="5mm"> Le Parlement de la Confédération helvétique
      </fo:block>
      <fo:block text-indent="5mm" font-family="serif" font-size="12pt"
        space-before="5mm"
```

```

28      space-after="5mm"> L'<fo:inline font-style="italic">Assemblée
        fédérale</fo:inline>
        suisse constitue ce qu'il est convenu d'appeler un parlement de milice
        dans la mesure où
        la majorité de ses membres exercent une activité professionnelle
        parallèlement à leur
        mandat parlementaire. Les députés doivent néanmoins faire face à un
        volume de travail
        considérable : ils sont en effet tenus de siéger quatre fois par an
        pendant les trois
33      semaines que dure chaque session ordinaire de l'<fo:inline
        font-style="italic">Assemblée
        fédérale</fo:inline> (auxquelles s'ajoutent, le cas échéant, les
        sessions
        extraordinaires et les sessions spéciales), de participer dans
        l'intervalle à nombre de
        séances de commissions, de <fo:basic-link
        internal-destination="tableau"
        text-decoration="bold">groupes parlementaires</fo:basic-link> ou de
38      groupes de
        travail, et enfin de se consacrer à la nécessaire préparation des
        dossiers. </fo:block>
<fo:block font-size="14pt" font-family="sans-serif" color="grey"
        space-before="5mm"
        space-after="5mm">L'Assemblée fédérale – système bicaméral</fo:block>
<fo:block>L'Assemblée fédérale se compose de deux Chambres:</fo:block>
<fo:list-block>
43      <fo:list-item>
        <fo:list-item-label end-indent="label-end()">
          <fo:block></fo:block>
        </fo:list-item-label>
        <fo:list-item-body start-indent="body-start()">
48          <fo:block>le Conseil National, composé de 200 conseillers
            nationaux</fo:block>
        </fo:list-item-body>
        </fo:list-item>
        <fo:list-item>
        <fo:list-item-label end-indent="label-end()">
53          <fo:block></fo:block>
        </fo:list-item-label>
        <fo:list-item-body start-indent="body-start()">
          <fo:block>le Conseil des États, composé de 46 conseillers aux
            États (2 par
            canton)</fo:block>
58          </fo:list-item-body>
        </fo:list-item>
      </fo:list-block>
<fo:block space-before="5mm" space-after="3mm">
        <fo:inline>Le tableau suivant présente la répartition des sièges dans
        les 2 chambres et
63      la répartition des groupes:</fo:inline>
        <fo:table id="tableau" padding-after="3pt" padding-before="3pt"
        padding-start="3pt"

```

```

padding-end="3pt" border="solid" space-before="4mm" width="170mm">
<fo:table-column column-width="50mm"/>
<fo:table-column column-width="40mm"/>
68 <fo:table-column column-width="20mm"/>
<fo:table-column column-width="20mm"/>
<fo:table-column column-width="20mm"/>
<fo:table-header>
  <fo:table-row>
73   <fo:table-cell padding-after="3pt" padding-before="3pt"
        padding-start="3pt"
        padding-end="3pt" border="solid">
        <fo:block font-weight="bold">Groupe
          parlementaire</fo:block>
        </fo:table-cell>
      <fo:table-cell padding-after="3pt" padding-before="3pt"
        padding-start="3pt"
78   padding-end="3pt" border="solid">
        <fo:block font-weight="bold">Répartition</fo:block>
      </fo:table-cell>
      <fo:table-cell padding-after="3pt" padding-before="3pt"
        padding-start="3pt"
        padding-end="3pt" border="solid">
83   <fo:block text-align="center" font-weight="bold">CN
        <fo:footnote>
          <fo:inline vertical-align="super"
            font-size="8pt">1</fo:inline>
          <fo:footnote-body>
            <fo:block font-style="italic"
              font-weight="normal"
              text-align="left">1. Conseil
                National</fo:block>
88   </fo:footnote-body>
          </fo:footnote>
        </fo:block>
      </fo:table-cell>
      <fo:table-cell padding-after="3pt" padding-before="3pt"
        padding-start="3pt"
93   padding-end="3pt" border="solid">
        <fo:block text-align="center"
          font-weight="bold">CE<fo:footnote>
          <fo:inline vertical-align="super"
            font-size="8pt">2</fo:inline>
          <fo:footnote-body>
            <fo:block font-style="italic"
              font-weight="normal"
98   text-align="left">2. Conseil des
              États</fo:block>
            </fo:footnote-body>
          </fo:footnote>
        </fo:block>
      </fo:table-cell>
103 <fo:table-cell padding-after="3pt" padding-before="3pt"
        padding-start="3pt"
        padding-end="3pt" border="solid">

```

```

        <fo:block text-align="center"
            font-weight="bold">Total</fo:block>
    </fo:table-cell>
</fo:table-row>
108 </fo:table-header>
    <fo:table-body>
        <fo:table-row>
            <fo:table-cell padding-after="3pt" padding-before="3pt"
                padding-start="3pt"
                padding-end="3pt">
113         <fo:block>Groupe de l'Union démocratique du
                centre</fo:block>
            </fo:table-cell>
            <fo:table-cell padding-after="3pt" padding-before="3pt"
                padding-start="3pt"
                padding-end="3pt">
118         <fo:block>63 UDC, 1 Lega</fo:block>
            </fo:table-cell>
            <fo:table-cell padding-after="3pt" padding-before="3pt"
                padding-start="3pt"
                padding-end="3pt">
123         <fo:block text-align="center">56</fo:block>
            </fo:table-cell>
            <fo:table-cell padding-after="3pt" padding-before="3pt"
                padding-start="3pt"
                padding-end="3pt">
128         <fo:block text-align="center">8</fo:block>
            </fo:table-cell>
            <fo:table-cell padding-after="3pt" padding-before="3pt"
                padding-start="3pt"
                padding-end="3pt">
133         <fo:block text-align="center">64</fo:block>
            </fo:table-cell>
        </fo:table-row>
        <fo:table-row>
            <fo:table-cell padding-after="3pt" padding-before="3pt"
                padding-start="3pt"
                padding-end="3pt">
138         <fo:block>Groupe socialiste</fo:block>
            </fo:table-cell>
            <fo:table-cell padding-after="3pt" padding-before="3pt"
                padding-start="3pt"
                padding-end="3pt">
143         <fo:block>61 PS</fo:block>
            </fo:table-cell>
            <fo:table-cell padding-after="3pt" padding-before="3pt"
                padding-start="3pt"
                padding-end="3pt">
                <fo:block text-align="center">52</fo:block>
            </fo:table-cell>
            <fo:table-cell padding-after="3pt" padding-before="3pt"
                padding-start="3pt"
                padding-end="3pt">
                <fo:block text-align="center">9</fo:block>
            </fo:table-cell>
        </fo:table-row>
    </fo:table-body>
</fo:table>

```

```

148         </fo:table-cell >
          <fo:table-cell padding-after="3pt" padding-before="3pt"
            padding-start="3pt"
            padding-end="3pt">
            <fo:block text-align="center">61</fo:block >
          </fo:table-cell >
153 </fo:table-row >
      <fo:table-row >
        <fo:table-cell padding-after="3pt" padding-before="3pt"
          padding-start="3pt"
          padding-end="3pt">
158         <fo:block >Groupe radical-libéral</fo:block >
        </fo:table-cell >
        <fo:table-cell >
          <fo:block >50 PRD, 4 PLS</fo:block >
        </fo:table-cell >
        <fo:table-cell padding-after="3pt" padding-before="3pt"
          padding-start="3pt"
163         padding-end="3pt">
          <fo:block text-align="center">40</fo:block >
        </fo:table-cell >
        <fo:table-cell padding-after="3pt" padding-before="3pt"
          padding-start="3pt"
          padding-end="3pt">
168         <fo:block text-align="center">14</fo:block >
        </fo:table-cell >
        <fo:table-cell padding-after="3pt" padding-before="3pt"
          padding-start="3pt"
          padding-end="3pt">
          <fo:block text-align="center">54</fo:block >
173 </fo:table-cell >
      </fo:table-row >
      <fo:table-row >
        <fo:table-cell padding-after="3pt" padding-before="3pt"
          padding-start="3pt"
          padding-end="3pt">
178         <fo:block >Groupe démocrate-chrétien</fo:block >
        </fo:table-cell >
        <fo:table-cell padding-after="3pt" padding-before="3pt"
          padding-start="3pt"
          padding-end="3pt">
          <fo:block >43 PDC</fo:block >
183 </fo:table-cell >
        <fo:table-cell padding-after="3pt" padding-before="3pt"
          padding-start="3pt"
          padding-end="3pt">
          <fo:block text-align="center">28</fo:block >
        </fo:table-cell >
188 <fo:table-cell padding-after="3pt" padding-before="3pt"
          padding-start="3pt"
          padding-end="3pt">
          <fo:block text-align="center">15</fo:block >
        </fo:table-cell >

```

```

193     <fo:table-cell padding-after="3pt" padding-before="3pt"
padding-start="3pt"
padding-end="3pt">
<fo:block text-align="center">43</fo:block >
</fo:table-cell >
</fo:table-row >
198 <fo:table-row >
<fo:table-cell padding-after="3pt" padding-before="3pt"
padding-start="3pt"
padding-end="3pt">
<fo:block>Groupe des Verts</fo:block >
</fo:table-cell >
<fo:table-cell padding-after="3pt" padding-before="3pt"
padding-start="3pt"
padding-end="3pt">
203 <fo:block>12 Verts , 1 PCS, 1 SGA</fo:block >
</fo:table-cell >
<fo:table-cell padding-after="3pt" padding-before="3pt"
padding-start="3pt"
padding-end="3pt">
208 <fo:block text-align="center">14</fo:block >
</fo:table-cell >
<fo:table-cell padding-after="3pt" padding-before="3pt"
padding-start="3pt"
padding-end="3pt">
<fo:block text-align="center">0</fo:block >
213 </fo:table-cell >
<fo:table-cell padding-after="3pt" padding-before="3pt"
padding-start="3pt"
padding-end="3pt">
<fo:block text-align="center">14</fo:block >
</fo:table-cell >
218 </fo:table-row >
<fo:table-row >
<fo:table-cell padding-after="3pt" padding-before="3pt"
padding-start="3pt"
padding-end="3pt">
<fo:block>Groupe PEV-UDF</fo:block >
223 </fo:table-cell >
<fo:table-cell padding-after="3pt" padding-before="3pt"
padding-start="3pt"
padding-end="3pt">
<fo:block>3 PEV, 2 UDF</fo:block >
</fo:table-cell >
228 <fo:table-cell padding-after="3pt" padding-before="3pt"
padding-start="3pt"
padding-end="3pt">
<fo:block text-align="center">5</fo:block >
</fo:table-cell >
<fo:table-cell padding-after="3pt" padding-before="3pt"
padding-start="3pt"
padding-end="3pt">
233 <fo:block text-align="center">0</fo:block >
</fo:table-cell >

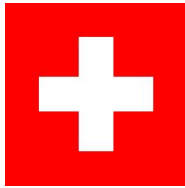
```

```

238     <fo:table-cell padding-after="3pt" padding-before="3pt"
        padding-start="3pt"
        padding-end="3pt">
        <fo:block text-align="center">5</fo:block>
    </fo:table-cell>
</fo:table-row>
<fo:table-row>
243     <fo:table-cell padding-after="3pt" padding-before="3pt"
        padding-start="3pt"
        padding-end="3pt">
        <fo:block>Non inscrits</fo:block>
    </fo:table-cell>
    <fo:table-cell padding-after="3pt" padding-before="3pt"
        padding-start="3pt"
        padding-end="3pt">
248     <fo:block>2 PdT, 1 AdG-Sol, 1 DS, 1 GLiZ</fo:block>
    </fo:table-cell>
    <fo:table-cell padding-after="3pt" padding-before="3pt"
        padding-start="3pt"
        padding-end="3pt">
253     <fo:block text-align="center">5</fo:block>
    </fo:table-cell>
    <fo:table-cell padding-after="3pt" padding-before="3pt"
        padding-start="3pt"
        padding-end="3pt">
258     <fo:block text-align="center">0</fo:block>
    </fo:table-cell>
    <fo:table-cell padding-after="3pt" padding-before="3pt"
        padding-start="3pt"
        padding-end="3pt">
        <fo:block text-align="center">5</fo:block>
    </fo:table-cell>
</fo:table-row>
263 </fo:table-body>
</fo:table>
</fo:block>
<fo:block>
    <fo:instream-foreign-object>
268     <svg:svg width="30pt" height="12pt"
        xmlns:svg="http://www.w3.org/2000/svg">
        <svg:circle cx="15pt" cy="6pt" r="5pt" style="fill:blue;"/>
    </svg:svg>
    </fo:instream-foreign-object>
    <fo:basic-link external-destination="url('http://www.parlement.ch/')"
273     text-decoration="underline" color="blue">Le site internet du
    parlement</fo:basic-link>
</fo:block>
</fo:flow>
</fo:page-sequence>
278 </fo:root>

```

Listing C.1 – Code de l'exemple *Parlement*



## Le Parlement de la Confédération helvétique

L'*Assemblée fédérale* suisse constitue ce qu'il est convenu d'appeler un parlement de milice dans la mesure où la majorité de ses membres exercent une activité professionnelle parallèlement à leur mandat parlementaire. Les députés doivent néanmoins faire face à un volume de travail considérable : ils sont en effet tenus de siéger quatre fois par an pendant les trois semaines que dure chaque session ordinaire de l'*Assemblée fédérale* (auxquelles s'ajoutent, le cas échéant, les sessions extraordinaires et les sessions spéciales), de participer dans l'intervalle à nombre de séances de commissions, de groupes parlementaires ou de groupes de travail, et enfin de se consacrer à la nécessaire préparation des dossiers.

### L'Assemblée fédérale - système bicaméral

L'Assemblée fédérale se compose de deux Chambres:

- le Conseil National, composé de 200 conseillers nationaux
- le Conseil des États, composé de 46 conseillers aux États (2 par canton)

Le tableau suivant présente la répartition des sièges dans les 2 chambres et la répartition des groupes:

Groupe parlementaire	Répartition	CN <sup>1</sup>	CE <sup>2</sup>	Total
Groupe de l'Union démocratique du centre	63 UDC, 1 Lega	56	8	64
Groupe socialiste	61 PS	52	9	61
Groupe radical-libéral	50 PRD, 4 PLS	40	14	54
Groupe démocrate-chrétien	43 PDC	28	15	43
Groupe des Verts	12 Verts, 1 PCS, 1 SGA	14	0	14
Groupe PEV-UDF	3 PEV, 2 UDF	5	0	5
Non inscrits	2 PdT, 1 AdG-Sol, 1 DS, 1 GLiZ	5	0	5

- [Le site internet du parlement](#)

---

1. Conseil National  
2. Conseil des États

# Table des figures

1.1	Processus de transformation . . . . .	5
2.1	Résultat du listing 2.1 . . . . .	9
2.2	Structure d'une page XSL-FO sous forme d'arbre . . . . .	10
2.3	Structure d'une page . . . . .	11
2.4	Résultat obtenu avec FOP de l'exemple <i>Parlement</i> . . . . .	16
2.5	Résultat obtenu avec FOP du listing 2.4 . . . . .	18
2.6	Structure des tables XSL-FO . . . . .	19
2.7	Table produite par le listing 2.5 . . . . .	19
2.8	Les notes de pages produites . . . . .	21
2.9	Structure des listes XSL-FO . . . . .	22
2.10	Liste des chambres fédérales . . . . .	23
2.11	En-tête du document <i>Parlement</i> , résultat du listing 2.10 . . . . .	24
2.12	Un lien externe dans le document <i>Parlement</i> . . . . .	25
3.1	Structure de l'arbre XML du catalogue des produits . . . . .	26
3.2	Structure de l'arbre XML d'un produit du catalogue . . . . .	28
3.3	Structure de l'arbre XML des déclinaisons d'un produit . . . . .	29
4.1	Processus de transformation . . . . .	31
4.2	Catalogue au niveau conceptuel . . . . .	32
	Exemples des premières pages d'un catalogue PDF . . . . .	34
	Exemples des pages contenant les produits d'un catalogue PDF . . . . .	35
4.3	Structure des templates . . . . .	36
4.4	Bloc produit issu du listing 4.4 . . . . .	42
4.5	Démonstration de mise au point de la mise en page . . . . .	48
C.1	Résultat obtenu avec FOP de l'exemple <i>Parlement</i> . . . . .	73

# Liste des tableaux

2.1	Attributs de l'élément fo:conditional-page-master-reference . . .	13
2.2	Autres éléments de type en-ligne . . . . .	18
3.1	Fils de l'élément title . . . . .	28
3.2	Éléments non répétés d'un produit . . . . .	29
3.3	Fils de l'élément occurrence . . . . .	30
4.1	Paramètres globaux . . . . .	37
4.2	Variables globales . . . . .	38
4.3	Variables du template Main . . . . .	38
4.4	Paramètres du template TitlePage . . . . .	39
4.5	Variables du template SmallProductBlock . . . . .	41
4.6	Paramètres du template DisplayProductInfo . . . . .	45
B.1	Variables du template BigProductBlock . . . . .	53

# Listings

2.1	Un document de base utilisant le vocabulaire XSL-FO . . . . .	8
2.2	Exemple d'un élément <code>layout-master-set</code> . . . . .	12
2.3	Exemple d'un <code>layout-master-set</code> avec différenciation de la page de titre et des pages paires et impaires . . . . .	13
2.4	Exemple d'utilisation d'un élément <code>fo:block</code> . . . . .	17
2.5	Exemple d'utilisation d'un élément <code>fo:table</code> . . . . .	20
2.6	Exemple de séparation du texte et des notes de bas de page .	21
2.7	Exemple de note de bas de page . . . . .	21
2.8	Exemple de liste . . . . .	22
2.9	Exemple d'insertion de code SVG avec un élément <code>fo:instream- foreign-object</code> . . . . .	23
2.10	Exemple d'insertion d'images avec un élément <code>fo:external-graphic</code>	24
2.11	Exemple d'un lien interne . . . . .	25
2.12	Exemple d'un lien externe . . . . .	25
3.1	Un exemple de fichier XML produit par <i>eSarine</i> . . . . .	27
4.1	Code FO pour l'en-tête et le pied de page . . . . .	39
4.2	Code FO pour la page de titre et celle de garde . . . . .	40
4.3	Code FO pour la table des matières . . . . .	40
4.4	Code FO pour un bloc produit de petite taille . . . . .	43
B.1	Code du template <code>Main</code> . . . . .	53
B.2	Code du template <code>TitlePage</code> . . . . .	55
B.3	Code du template <code>TOC</code> . . . . .	56
B.4	Code du template <code>SmallProductBlock</code> . . . . .	57
B.5	Code du template <code>BigProductBlock</code> . . . . .	59
B.6	Code du template <code>EmptyLinesLoop</code> . . . . .	62
B.7	Code du template <code>DisplayTableFooterAlarm</code> . . . . .	63
B.8	Code du template <code>DisplayImage</code> . . . . .	64
B.9	Code du template <code>DisplayProductInfo</code> . . . . .	64
B.10	Code des templates <code>SetColumns</code> . . . . .	65
C.1	Code de l'exemple <i>Parlement</i> . . . . .	66

# Références

- [1] Antenna house homepage [en ligne, disponible le 25 avril 2005]. Disponible sur internet: <http://www.antennahouse.com/>.
- [2] Apache fop compliance page [en ligne, disponible le 11 avril 2005]. Disponible sur internet: <http://xml.apache.org/fop/compliance.html>.
- [3] e-sarine homepage [en ligne, disponible le 25 avril 2005]. Disponible sur internet: <http://diuf.unifr.ch/is/research/esarine/introduction.php>.
- [4] Fop homepage [en ligne, disponible le 9 mai 2005]. Disponible sur internet: <http://xml.apache.org/fop/>.
- [5] Fpdf homepage [en ligne, disponible le 8 avril 2005]. Disponible sur internet: <http://www.fpdf.org/>.
- [6] itext homepage [en ligne, disponible le 7 mai 2005]. Disponible sur internet: <http://www.lowagie.com/iText/>.
- [7] Pragma homepage [en ligne, disponible le 20 avril 2005]. Disponible sur internet: <http://www.pragma-ade.com/>.
- [8] Renderx homepage [en ligne, disponible le 25 avril 2005]. Disponible sur internet: <http://www.renderx.com/>.
- [9] Struts project homepage [en ligne, disponible le 7 mai 2005]. Disponible sur internet: <http://struts.apache.org/>.
- [10] Tomcat homepage [en ligne, disponible le 7 mai 2005]. Disponible sur internet: <http://jakarta.apache.org/tomcat/>.
- [11] Xalan-java homepage [en ligne, disponible le 7 mai 2005]. Disponible sur internet: <http://xml.apache.org/xalan-j/>.
- [12] Xmlroff homepage [en ligne, disponible le 25 avril 2005]. Disponible sur internet: <http://xmlroff.sourceforge.net/>.
- [13] Eliotte Rusty Harold and W.Scott Means. *XML in a Nutshell*. O'Reilly, 2nd edition, 2002.
- [14] Tokushige Kobayashi. Implementation comparison of xsl-fo processors. Disponible sur internet: <http://www.antennahouse.com/xslfo/impl-comparison-fo.htm> [disponible le 25 Avril 2005].

## RÉFÉRENCES

---

- [15] Adobe Systems. Pdf reference. Disponible sur internet: [http://partners.adobe.com/public/developer/pdf/index\\_reference.html](http://partners.adobe.com/public/developer/pdf/index_reference.html) [disponible le 11 avril 2005].
- [16] W3C. Cascading style sheets, level 2 css2 specification, w3c recommendation. Disponible sur internet: <http://www.w3.org/TR/1998/REC-CSS2-19980512/> [disponible le 27 avril 2005].
- [17] W3C. Extensible markup language (xml) 1.0 recommendation (third edition). Disponible sur internet: <http://www.w3.org/TR/2004/REC-xml-20040204/> [disponible le 7 avril 2005].
- [18] W3C. Extensible stylesheet language (xsl) recommendation version 1.0. Disponible sur internet: <http://www.w3.org/TR/2001/REC-xsl-20011015/> [disponible le 7 avril 2005].
- [19] W3C. Scalable vector graphics (svg) 1.1 specification, w3c recommendation. Disponible sur internet: <http://www.w3.org/TR/2003/REC-SVG11-20030114/> [disponible le 27 avril 2005].
- [20] W3C. Xml path language (xpath) version 1.0. Disponible sur internet: <http://www.w3.org/TR/1999/REC-xpath-19991116> [disponible le 11 avril 2005].
- [21] W3C. Xsl transformations (xslt) recommendation version 1.0. Disponible sur internet: <http://www.w3.org/TR/1999/REC-xslt-19991116> [disponible le 7 avril 2005].
- [22] Nicolas Werro, Henrik Stormer, Daniel Frauchiger, and Andreas Meier. esarine - a struts-based webshop for small and medium-sized enterprises. In *the EMISA Conference - Information Systems in E-Business and E-Government*, October 2004.