



Cursus

**Concepteur en architecture informatique  
parcours Systèmes d'information**

Code : CPN8402A

UE UAAL0V

# **Expérience professionnelle et rapport d'activité**

Réalisé par **Grégory ROCHE**

N° auditeur CNAM : NAQ 410 558

Sous la direction de **David BRENET**

Enseignant au CNAM

**Résumé** Ce rapport d'activité introduit une vue d'ensemble de mon parcours, des missions et des réalisations effectuées en lien avec les systèmes d'informations. Elles sont ensuite développées et mettent en évidence les compétences techniques apportées par les unités d'enseignements de mon cursus de **Concepteur en architecture informatique parcours Systèmes d'information**.

6 octobre 2024

# TABLE DES MATIÈRES

Parcours académique et professionnel . . . . .	ii
Introduction . . . . .	iv
Ch. 1 WEBMASTER – PME ABC . . . . .	1
Ch. 2 DÉVELOPPEUR XML – ESN INDB . . . . .	2
Ch. 3 DÉVELOPPEUR WEB – ESN AJLSM . . . . .	3
Ch. 4 ESN AVANTEAM . . . . .	5
4.1 Présentation . . . . .	5
4.2 Activité et structure . . . . .	5
Ch. 5 PROJET . . . . .	6
5.1 Description du besoin . . . . .	6
5.2 Exigences et contraintes . . . . .	6
Ch. 6 CONCEPTION . . . . .	7
6.1 Conception technique . . . . .	7
6.1.1 Le processeur Apache FOP . . . . .	8
6.1.2 Le format XML . . . . .	8
6.1.3 La transformation XSLT . . . . .	8
6.1.4 Le langage XSL-FO . . . . .	8
6.1.5 Le format PDF . . . . .	9
6.2 Conception graphique . . . . .	9
6.2.1 Le zoning . . . . .	9
6.2.2 Le wireframe . . . . .	10
6.2.3 Le mockup . . . . .	10
Ch. 7 DÉVELOPPEMENT . . . . .	12
7.1 Environnement de développement . . . . .	12
7.1.1 Éclipse et ses outils . . . . .	12
7.1.2 Visionneuse PDF . . . . .	12
7.2 Recherche de qualité . . . . .	13
7.3 Paramétrisation . . . . .	14
7.4 Optimisation . . . . .	15
7.5 Internationalisation . . . . .	16
7.6 Documentation . . . . .	16
7.7 Création du livrable . . . . .	17
Ch. 8 TESTS . . . . .	18
8.1 Méthode . . . . .	18
8.2 Tests fonctionnels . . . . .	18
8.3 Tests de performance . . . . .	19
8.4 Résultats . . . . .	19
Ch. 9 MISE EN PRODUCTION . . . . .	20
9.1 Recette . . . . .	20
9.2 Production . . . . .	20
Conclusion . . . . .	21
Acronymes et glossaire . . . . .	22
Bibliographie . . . . .	24

# PARCOURS ACADÉMIQUE ET PROFESSIONNEL

## État civil

Nom	ROCHE
Prénom	Grégory
Date et lieu de naissance	30/07/1973 à Roanne (Loire, 42)
Nationalité	Français
Adresse postale	19 rue Solle 33 200 Bordeaux

## Coordonnées

Téléphone	07.82.21.97.73
Adresse e-mail	<a href="mailto:groche-cnam@polymorphisme.fr">groche-cnam@polymorphisme.fr</a> 

## Parcours académique

Diplôme obtenu le plus élevé :

Titre du diplôme	Licence <b>STS</b> , mention informatique
Date de l'obtention	2021
Diplôme	oui
Type d'établissement	public
Niveau de diplôme	Niveau 6

---

## Parcours professionnel

Durée totale de l'expérience professionnelle : 18 mois.

TAB. 1 La liste des emplois occupés.

Emploi	Entreprise	Durée (mois)	Fonctions
Technicien coef. 260	ABC	6	Webmaster
Technicien	ESN InDb	2	Développeur XML
Technicien analyste coef. 355	ESN AJLSM	4	Développeur Web
Technicien	ESN Avanteam	6	Développeur XML

La liste des entreprises citées dans le rapport et leur secteur d'activité :

TAB. 2 La liste des entreprises par secteur d'activité.

Entreprise	Secteur d'activité	Code NAF
ABC	Textile	
ESN InDb	Conseil en systèmes et logiciels informatiques	6220A
ESN AJLSM	Conseil en systèmes et logiciels informatiques	6220A
ESN Avanteam	Édition de logiciels applicatifs	5829C

## Remerciements

Je tiens à remercier en particulier :

- M. Hugues VAN LOO, ingénieur de la société [Avanteam](#), avec qui une collaboration fructueuse s'est construite au fil du temps;
- M. Damien GOUTTE-GATTAT, chercheur à la [Barts Cancer Institute](#) de Londres, qui rédige des documents avec le processeur FOP, suite à ma présentation de cet outil;
- M. Alain COUTHURES, ingénieur expert des langages XForms et XQuery au sein de l'entreprise [Agence XML](#) à Bordeaux;
- les acteurs enseignants et auditeurs du [CNAM Nouvelle-Aquitaine](#).

# INTRODUCTION

Je suis entré dans la vie active avec pour tout bagage un DEUG Mathématiques et Informatiques Appliquées aux Sciences obtenu en 1999 à l'université Lyon I. Ce diplôme n'augmente pas significativement l'employabilité.

Je me suis formé à la création artisanale de site Web. Ma démarche déboucha sur un CDD (Contrat à Durée Déterminée) pour un poste de webmaster dans l'entreprise ABC, dont le gérant recherchait en réalité un commercial ! Ce premier échec m'amena à reconsidérer ma position.

Après une errance de plusieurs années dans le milieu précaire des emplois de subsistance qui ne mènent à rien, j'ai décidé d'y mettre un terme en créant l'auto-entreprise Polymorphisme, consacrée au développement informatique.

Je me suis intéressé aux langages de programmation Java et XML (eXtensible Markup Language) mis en œuvre lors de projets, notamment avec les entreprises InDb, AJLSM et Avanteam, que je décris plus en détails au fil des chapitres. Plus que tout autre chose, ces expériences ont données un sens à mon engagement et un sentiment de revalorisation.

Toutefois, face au manque d'opportunités, j'ai remis en question mes compétences. En effet, les compétences techniques ne sont pas suffisantes pour entreprendre. Le marché montre une nette préférence pour les diplômés aux niveaux Bac + 4 ou Bac + 5.

Fort de ce constat et de ma situation, la Région Nouvelle-Aquitaine m'a permis de suivre des cours au CESI puis une POEC à Ynov. J'ai poursuivi ma démarche en optant pour un cursus en informatique au CNAM Nouvelle-Aquitaine [↗](#).

## Unités d'enseignements et compétences

Le contenu et les compétences mobilisées pour les deux unités ANG330 et SEC105 sont indispensables, transverses et multiples. Raison pour laquelle, il n'est pas utile de les mettre en avant dans les sections suivantes.

**ANG330 Anglais professionnel** [↗](#) : dans le domaine de l'informatique, l'anglais est fréquent lors des échanges directes, la documentation est généralement rédigée en anglais, la documentation en français fait figure d'exception.

**Sécurité SEC105 Architectures et bonnes pratiques de la sécurité des réseaux, des systèmes, des données et des applications** [↗](#) : la cybersécurité, plus qu'une nécessité, est devenue une exigence. Les outils que j'utilise le plus souvent sont un pare-feu, SSH, Gpg2, les sauvegardes de données et les bonnes pratiques dès que possible.

Note : des éléments de sémantique sont formatés dans le texte ainsi : nom de personne Mike KAY, terme du glossaire **archivistique**, nom de répertoire `dist/`, de fichier `README.txt`, tâche Ant «archifactory» référence à une figure ou une table `figure 3.1` ou `table 7.1`, référence bibliographique (MACLAUGHLIN, 2002), un lien hypertexte, p. ex. `TEX` [↗](#).

## Ch. 1

# WEBMASTER - PME ABC

Dans le bassin Roannais, le secteur textile survit encore. En particulier, la PME ABC offre un service de broderie industrielle sur divers supports textiles.

En tant que webmaster, je suis chargé de créer et de gérer le site vitrine de l'entreprise dans le but d'étendre sa visibilité au grand public. Le site Web doit être finalisé dans un délai de six mois.

Mon environnement de travail est une pièce lumineuse meublée d'un mobilier de rangement adossé aux murs et d'un poste de travail composé d'un bureau simple sur lequel sont disposés un micro-ordinateur et un téléphone fixe. L'entreprise a souscrit un service de connexion **RTC** (Réseau Téléphonique Commuté) vers l'Internet auprès du **FAI Free** [↗](#) (Fournisseur d'Accès Internet).

Je met en place un environnement de développement **LAMP** (Linux Apache MySQL PHP) composé du système d'exploitation **Linux Debian** [↗](#), du serveur Web **Apache httpd** [↗](#), de la base de données relationnelle **MySQL** [↗](#) et du langage de programmation **PHP** (PHP Hypertext Preprocessor) [↗](#).

Des entretiens avec l'employeur permettent d'éclaircir ses besoins. J'ai rédigé un cahier des charges sommaire à partir des spécifications techniques du site Web.

En collaboration avec l'employeur, je recherche, choisis puis achète un nom de domaine auprès de la société **Gandi** [↗](#) ainsi qu'un hébergement Web sur les serveurs de la société **OVH** (On Vous Héberge) [↗](#).

Je conçois une charte graphique à partir du logo de l'entreprise. L'arborescence et la navigation du site Web sont validés. Le contenu du site Web est créé ou repris à partir de l'existant : le texte est rédigé et les images sont retouchées ; il est aussi question de la création d'une vidéo.

Je présente une première version du site Web au gérant de l'entreprise et aux employés de bureau. Suite à divers constats, j'améliore certains points du site Web au cours des semaines suivantes.

Finalement, je met en place une stratégie de référencement naturel puis l'applique et termine par une campagne de référencement.

Lors de mon passage dans cette entreprise, j'ai surtout pris conscience de l'importance de l'ambiance et des relations interpersonnelles.

Au terme de ce **CDD** de six mois, l'employeur me propose un poste de commercial. Préférant avoir moins que pire, j'ai aimablement décliné sa proposition.

## Ch. 2

# DÉVELOPPEUR XML - ESN INDB

La SARL InDb est une **ESN** (Entreprise de Service Numérique) implantée à Romans-Sur-Isère depuis 2002. Elle s'inscrit dans le secteur d'activité du conseil en systèmes et logiciels informatiques et elle s'est spécialisée dans la gestion de bases de données.

Chargée du développement d'un ensemble de feuilles de styles **XSLT** (eXtensible Stylesheet Language Transformations), dont elle ne détient pas le savoir-faire, elle fait appel à mon auto-entreprise Polymorphisme en tant que prestataire de services.

Dans un premier temps, j'échange avec le client afin de cerner ses besoins. Il me fait parvenir une maquette et m'explique le résultat attendu. Suite à nos échanges, il signe le devis détaillé que je lui soumet.

Je prends connaissance du standard **HR-XML** (Human Resource XML) [☞](#) qui formalise les besoins en gestion en ressources humaines à l'aide de la méthode **UML** (Unified Modeling Language) [☞](#). Je dois en acquérir un niveau de compréhension suffisant pour être en mesure d'appliquer le standard au cours de la phase de développement des feuilles de styles **XSLT**.

J'implémente les feuilles de styles **XSLT** qui transforment des fichiers de données au format **HR-XML** en documents **PDF** (Portable Document Format), qui sont des contrats tripartites, des fiches horaires et des fiches de paye intérimaires.

Je teste visuellement les documents **PDF** générés par les feuilles de styles **XSLT** à partir d'un jeu de données fourni par la société InDb. Quelques jours après réception du livrable, la société InDb effectue ses propres tests. Puis, le livrable est passé en revue avec le client et j'apporte des correctifs jusqu'à obtenir sa satisfaction.

Finalement, les feuilles de styles **XSLT** sont mises en production par la société InDb sur la **plateforme Pixid** [☞](#). Depuis, les trois classes de documents sont partagés par l'agence d'intérim employeur, l'intérimaire employé et l'entreprise cliente de l'agence d'intérim.

J'ai réalisé une prestation de service technique répondant aux besoins et donnant satisfaction au client. Néanmoins, la mise en application du standard **HR-XML** est coûteuse en temps.

Enfin, la société InDb s'est montrée bienveillante à mon égard en suggérant des axes d'améliorations possibles. Je prendrai en considération ses conseils lors de prochaines prestations de services.

### Unités d'enseignements et compétences

**NFE103 Méthodologies avancées d'informatisation** [☞](#) : la prise de connaissance et l'application du standard **HR-XML** passe nécessairement par une bonne compréhension de la méthode **UML** car il est entièrement rédigé à l'aide des schémas de cette méthode.

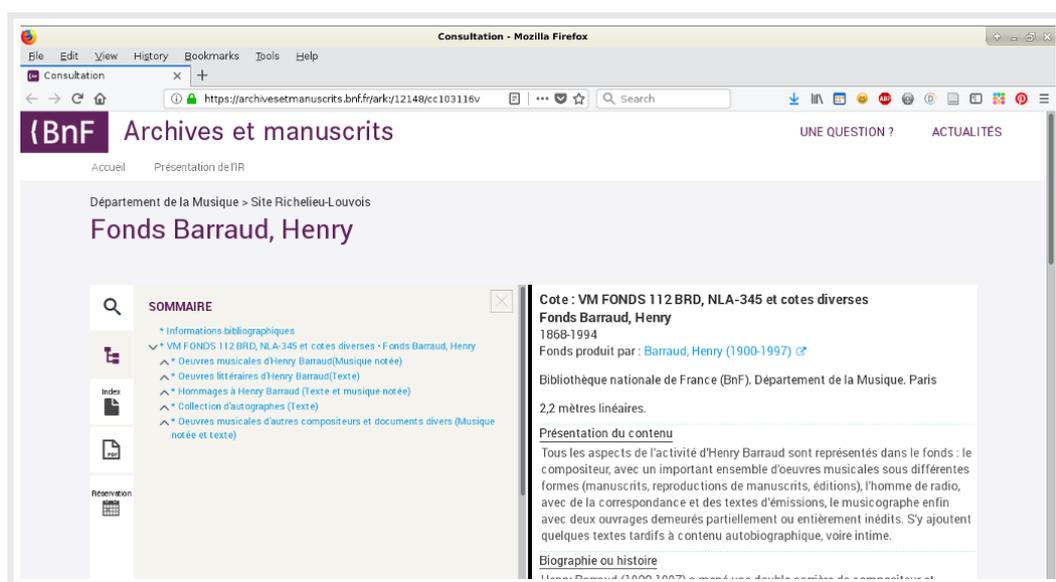
## Ch. 3

# DÉVELOPPEUR WEB - ESN AJLSM

Au cœur de la ville de Bordeaux, la société **AJLSM** (Arvers Jean-Luc Services et Maintenance) [☞](#) s'est spécialisée avec succès depuis 1999 dans le secteur d'activité du conseil en systèmes et logiciels informatiques. La société **AJLSM** conçoit et développe le progiciel **Pléade** [☞](#) qui s'inscrit dans le domaine de la **GED** (Gestion Electronique de Documentation). Pléade est un gestionnaire de corpus de documents d'archives, de documents scientifiques et d'icônographies. Il est destiné aux archives, bibliothèques et musées (figure 3.1).

Le progiciel Pléade est basé sur la servlet **Apache Cocoon** [☞](#) et le moteur de recherche **SDX** [☞](#) (Système Documentaire XML).

FIG. 3.1 Pléade est intégré au portail Web de la Bibliothèque Nationale de France.



Mes acquis d'expérience avec Apache Cocoon (BROGDEN et al., 2003; MOCZAR & ASTON, 2003) sont un marchepied pour l'analyse du progiciel Pléade. Lors de la période de montée en compétences, l'équipe m'explique la méthodologie de la recherche documentaire; la commissaire d'exposition Isabelle ARVERS assure un cours sur les relations entre les profils d'utilisateurs et le processus de construction de l'IHM (Interface Human Machine) de Pléade; par suite, j'ai pu analyser son architecture modulaire et rafraîchir des connaissances sur les langages de programmation mis en œuvre : Java, XML, XSLT, XHTML (eXtensible Hyper-Text Markup Language) et Javascript (MANGANO, 2003; NIEMEYER & KNUDSEN, 2002).

Enfin, je prends connaissance de formats de données EAD-XML (Encoded Archival Description XML) [☞](#), XML-MARC [☞](#), DC [☞](#) et XMP [☞](#) consacrés à l'archivistique (SOCIETY OF AMERICAN ARCHIVISTS, 1999).

Je met en œuvre les logiciels de gestion de synchronisation de fichiers Back In Time [☞](#) et Rsync (Remote SYNchronization) [☞](#), ainsi que le logiciel de gestion de versions de fichiers Apache Subversion [☞](#).

À l'aide du logiciel **Sélénium** [☞](#), je développe des suites de tests unitaires automatisés en Java couvrant les parties les plus utilisées de l'**IHM**. J'analyse ainsi l'**IHM** et améliore ma connaissance du fonctionnement de Pléade (LINK, 2003).

J'effectue la mise à jour de la librairie Javascript qui gère l'arborescence du menu de l'application. Je prends en compte la rétrocompatibilité, recherche de possibles impacts sur le code source de l'existant.

Dans le cadre d'une activité de maintenance corrective, j'établis une liste d'anomalies de la visionneuse d'images. En menant ma recherche suivant une méthode naïve, je constate un nombre conséquent de dysfonctionnements. Afin d'en établir une liste compréhensible, je la structure en partitionnant l'**IHM** en composants et fixe des règles de rédaction. Dans un dernier temps, je restructure cette liste en tenant compte de la notion de scénario utilisateur. Finalement, un programmeur Javascript pourra améliorer le code source en partant de cette liste d'anomalies.

Je conçois et développe un module dédié aux principaux réseaux sociaux : **Facebook** [☞](#), **Twitter** [☞](#), **Pinterest** [☞](#) et **Google +**. Je prends ainsi connaissance des **API** (Application Programming Interface) disponibles. Ensuite, je développe le module à partir des langages Java, **XSLT**, Javascript et **EAD-XML**. Le chef de projet l'intègre à la version commerciale de Pléade (figure 3.2). En pratique, un administrateur de Pléade peut gérer ce module, c'est-à-dire l'installer, le configurer et le supprimer. Le **projet Salamandre** [☞](#) du **Collège de France** [☞](#) l'utilise au quotidien.

FIG. 3.2 Présentation du module dédié aux réseaux sociaux sur le site Web de Pléade.



D'un point de vue relationnel, je suis intégré à un groupe studieux, au sein duquel la communication ne va pas toujours de soi du fait de la diversité des langues maternelles.

Je présente mes réalisations à l'ensemble de l'équipe; une expérience à l'issue positive qui m'incite à la réitérer.

Au final, j'ai découvert le milieu de l'archivistique dans un environnement bienveillant et propice au développement tant personnel que professionnel, un milieu qui tire vers le haut.

### Unités d'enseignements et compétences

**NFP121 Programmation avancée** [☞](#) : les patrons de conceptions, la programmation orientée objet et des méthodes de test m'ont été particulièrement utiles pour mener à bien les divers travaux que l'ingénieur chef de projet m'avait confiés.

## Ch. 4

# ESN AVANTEAM

## 4.1 Présentation

Fondée en 2019, la [société Avanteam](#) est une SAS active depuis 16 ans.

La société Avanteam a établi son siège social au 13 avenue Morane Saulnier, immeuble Nungesser, 78 140 Vélizy-Villacoublay. Elle est secondée par son annexe.

## 4.2 Activité et structure

La société Avanteam œuvre dans le secteur des logiciels informatiques, en particulier dans l'édition de logiciels applicatifs (code NAF 5829C).

Elle apporte des solutions métiers dans plusieurs spécialités (figure 4.1) : la gestion de la qualité, la gestion des risques, la GED, l'ECM et la BPM.

FIG. 4.1 La plateforme Process Suite permet la gestion des factures fournisseurs ainsi que la dématérialisation et l'optimisation d'autres processus clés : demandes d'achats, gestion des contrats, évaluation des fournisseurs, qualité, risque, service client, congés, notes de frais, recrutement, entrée-sortie du personnel, juridique, informatique, etc.



Plus de 520 clients en Europe et 25 % des entreprises CAC40 (Cotation Assistée en Continu) utilisent la suite Avanteam pour rationaliser leurs processus, réduire les coûts et gagner en agilité et en conformité.

## Ch. 5

# PROJET

### 5.1 Description du besoin

Des clients d'Avanteam disposent de données comptables stockées dans une base de données. Certaines ont exprimées le besoin de les rendre disponibles sous forme de fichiers stockables, consultables et imprimables.

Elles choisissent ensemble le format de fichiers **PDF**. Les fichiers **PDF** de données comptables devront être stockés sur une plateforme informatique ouverte aux entreprises demandeuses. Enfin, un programme devra transformer les données comptables en fichiers **PDF**.

Au sein de la société Avanteam, l'ingénieur Hugues VAN LOO, chef du projet, sollicite à mon auto-entreprise de prestation de services Polymorphisme pour concevoir et développer ce programme.

### 5.2 Exigences et contraintes

Le programme commandité par la société Avanteam a pour fonction de transformer des données **XML** en documents **PDF**.

La transformation doit être implémentée par des feuilles de styles **XSLT**. Un moteur de rendu **XSL-FO** (XSL Formatting Objects) doit générer les documents **PDF**, lesquels doivent être stockables, visualisables et imprimables puisqu'ils sont destinés à : 1. l'archivage numérique à long terme sur support électronique ; 2. la consultation dans une visionneuse **PDF** ; 3. l'impression sur papier au format A4.

En remplissant ces exigences, le client final devrait être satisfait.

Le chef de projet propose les contraintes suivantes : 1. une **forêt XML** et une maquette de document **PDF** ; 2. seule la version 1.0 du langage **XSLT** peut être utilisée ; 3. un délai raisonnable pour rendre le livrable.

#### Unités d'enseignements et compétences

**MSE102 Management et organisation des entreprises** ☑ : la connaissance de l'organisation de l'entreprise m'a été utile auprès d'Avanteam, car chaque personne avait un seul rôle.

**NFP121 Programmation avancée** ☑ : de nouveau, les patrons de conceptions, la programmation orientée objet et des méthodes de test m'ont été particulièrement utiles.

## Ch. 6

# CONCEPTION

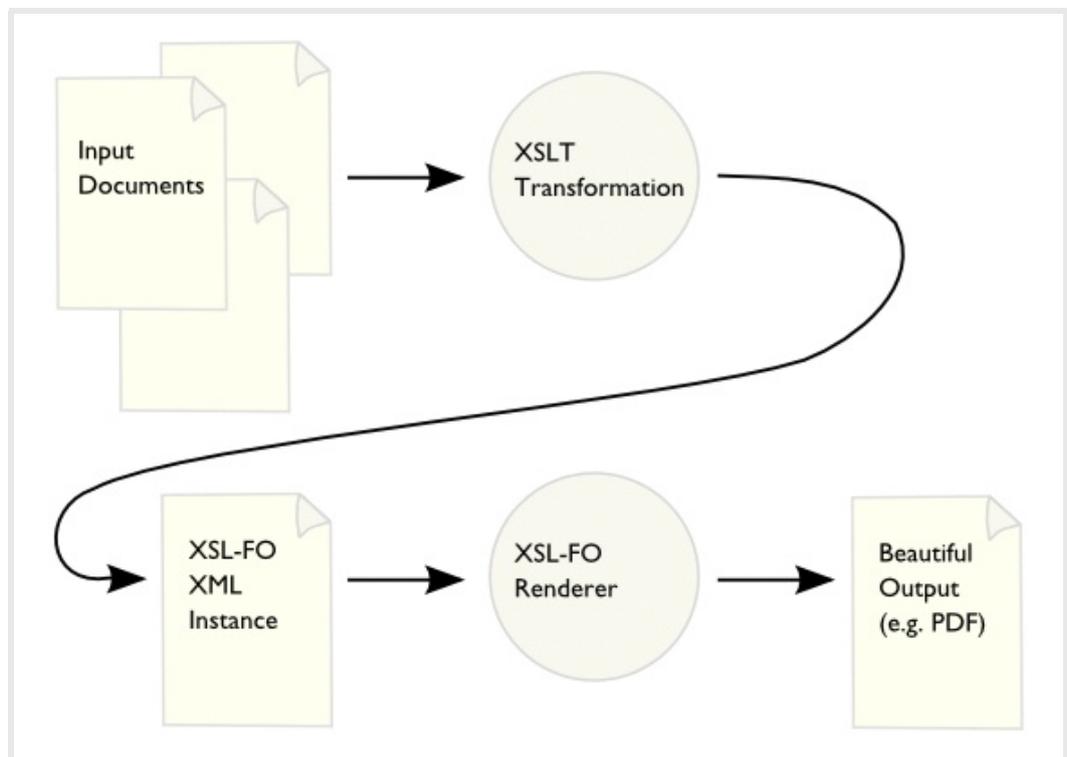
## 6.1 Conception technique

Les technologies **XML** apportent une réponse aux exigences du projet.

J'ai acquis un niveau de maîtrise de certaines technologies **XML** qui me permettent de produire des documents satisfaisant certaines exigences. Le processeur **FOP** (Formatting Objects Processor) implémente les étapes décrites par la figure 6.1 :

1. des documents en entrée représentent des données comptables sous la forme de fichiers **XML** ;
2. la transformation **XSLT** représente la transformation des fichiers **XML** en fichiers formatés en **XSL-FO** ;
3. un fichier **XSL-FO** définit la mise en page et le formatage des fichiers **XML** ;
4. à partir du fichier **XSL-FO**, le moteur de rendu **XSL-FO** génère un fichier en sortie ;
5. un document en sortie peut être un fichier **TXT** (TeXTe), **XML** ou encore **PDF**.

FIG. 6.1  
Les étapes du processus de transformation d'un document de données XML en document PDF.



### 6.1.1 Le processeur Apache FOP

Le **processeur Apache FOP** [↗](#) (Formatting Objects Processor) est un programme générant un fichier **PDF** à partir de données **XML** et d'une feuille de styles **XSLT**.

Ce processeur délègue à chacun de ses composants toutes les tâches dont il ne s'acquie pas lui-même. Son cœur est un ensemble de moteurs de rendu **XSL-FO**.

Les sections suivantes décrivent plus en détails le fonctionnement de ces composants que sont Apache Xerces, Apache Xalan, Apache Batik et les moteurs de rendu **XSL-FO**.

### 6.1.2 Le format XML

Le **standard XML** [↗](#) définit le **langage XML** (eXtensible Markup Language) comme un format de texte créé pour représenter une information semi-structurée. Il est composé d'instances dont : XML Base, xml:id, les fragments **XML**, XInclude, XPointer, XLink et **XSLT**.

**FOP** délègue la lecture d'une instance de document **XML** à l'analyseur syntaxique **Apache Xerces** [↗](#).

Lors de cette lecture, Xerces peut analyser, valider, sérialiser et transformer des éléments **XML**. Il opère sur les inclusions, espaces de noms, résolutions d'entités, vérifications d'identifiant et de références, et plus encore.

Le résultat de la sérialisation est un arbre **XML** chargé en mémoire vive.

### 6.1.3 La transformation XSLT

Le **langage XSLT** [↗](#) (eXtensible Stylesheet Language Transformations) est une instance du langage **XML**, définit comme la première partie du **langage XSL** [↗](#).

Un arbre XML peut être transformé suivant les règles d'une feuille de styles **XSLT** par le processeur **Apache Xalan** [↗](#).

Le langage **XSLT** s'utilise conjointement avec le **langage XPath**.

Le **langage XPath** [↗](#) (XML Path) est une instance du langage **XML**, définit comme la deuxième partie du **langage XSL** [↗](#).

Une expression XPath définit un sous-arbre **XML**, souvent nommé « **fragment XML** ». Ainsi grâce au processeur XPath, une transformation **XSLT** peut s'appliquer uniquement au fragment **XML** pointé.

### 6.1.4 Le langage XSL-FO

Le **langage XSL-FO** [↗](#) (XSL Formatting Objects) est une instance du langage **XML**, définit comme la troisième partie du **langage XSL** [↗](#).

Le **langage XSL-FO** est un langage de description de mise en page et de formatage d'un document **XML**. Par extension, il définit un format de fichier **XML**.

Le processeur **FOP** implémente le **langage XSL-FO**. Il passe un arbre **XSL-FO** à l'entrée d'un de ses moteurs de rendu **XSL-FO** qui le sérialise en un fichier formaté. Le fichier peut en particulier être au format **PDF**. Parallèlement à cette opération, Apache Batik prend en charge le traitement des images.

### 6.1.5 Le format PDF

Depuis septembre 2005, la norme **ISO 19005-1:2005** [☐](#) définit le format de document PDF/A-1, c.-à-d. PDF Archive version 1, ainsi que les caractéristiques d'une visonneuse pour ce format. Il existe deux variations de PDF/A-1, basés sur le format PDF v1.4 :

- PDF/A-1a est définie par la norme complète;
- PDF/A-1b est conforme à la version allégée de la norme, cette variation préserve la lisibilité du document et sa bonne présentation à l'affichage et à l'impression.

## 6.2 Conception graphique

Le processus de conception graphique du document (figure 4.1 p. 5) se base sur les maquettes transmises par le client. Il se compose de trois étapes de création dont les livrables sont respectivement le zoning, le wirefire et le mockup. Il se termine par la validation du mockup par le client.

J'ai développé une tâche Ant « gabarit » pour créer le zoning, le wireframe et le mockup, en fonction de valeurs paramétrées. Au préalable, une tâche « addId » ajoute un identifiant à chaque nœud du modèle de données **XML**.

```
1 <target name="addId" depends="init"  
2   description="Ajout d'identifiant aux noeuds du modèle XML.">  
3 <target name="gabarit" depends="addId"  
4   description="Création d'un gabarit.">
```

### 6.2.1 Le zoning

Le zoning constitue le socle de la composition du document. Il reflète la structure générale des données **XML**.

Tout d'abord, je hiérarchise par niveau d'importance les fragments de données **XML** définis par leur nom et leur fonction. Je trace une grille multicolonne appropriée au type de document et en dessine les blocs de composition et les gouttières. Je met en correspondance chaque fragment XML avec un bloc de composition. Puis, j'équilibre, j'harmonise la composition obtenue.

Enfin, j'applique une palette de niveaux de gris à ma composition afin d'en distinguer chaque bloc et d'obtenir le zoning (figure 6.2a p. 11). Le zoning est le premier livrable du processus de conception, socle du wireframe.

### 6.2.2 Le wireframe

Le wireframe est d'une composition plus évoluée que le zoning. Au cours de cette étape créative, je choisis avec le client, l'emplacement et le formatage des éléments de données XML.

Pour le placement des éléments XML, je crée une correspondance de l'ensemble des éléments vers les parties de blocs de composition. Dans ce but, j'attribue un identifiant de nœud ID à chaque élément, généré automatiquement pour une donnée dynamique, telle le contenu d'une cellule de donnée de table, ou codé en dur pour une donnée statique, telle le contenu d'une cellule de l'en-tête de table. Puis, j'assigne un élément identifié à une partie de bloc.

Pour décider du formatage de donnée d'un élément XML, j'évalue le rapport entre le sens d'une donnée XML et le sens du rendu formaté de cette donnée. Dans ce contexte, les notions de signifiant et signifié définies par la sémiotique apportent des critères d'évaluation.

Finalement, le wireframe présente une disposition fine des données XML dans une composition de document agréable (figure 6.2b p. 11). Le wireframe est une validation du zoning. Il concrétise une représentation graphique des données qui peut être magnifiée lors de sa transformation en mockup.

### 6.2.3 Le mockup

Pour révéler le mockup, j'applique des opérations graphiques au wireframe.

Je retouche le logo d'entreprise au pixel si nécessaire. Le choix d'une palette de couleurs est simple pour ce type de documents, mais n'est pas à négliger.

Les attributs graphiques, tels que les cadres et les bordures, renforcent la hiérarchie entre les blocs de composition.

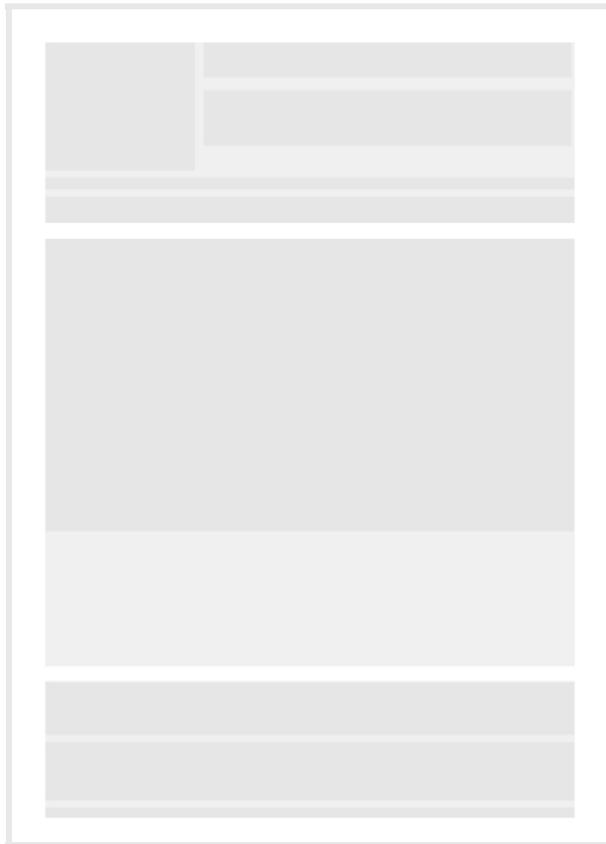
Un soin particulier est apporté à la typographie du document. Les fontes, la justification et l'interlignage participent à la lisibilité du texte. J'ai ainsi recherché parmi les caractéristiques des fontes, le support du standard Unicode ☑, la taille intrinsèque des glyphes, les niveaux de graisses supportés, etc. afin de sélectionner les meilleures fontes (ANDRIES, 2008 ; HARALAMBOUS, 2004).

Enfin, la qualité du ressenti du lecteur est évaluée à partir de l'harmonie de l'ensemble, du gris de la page, du rythme des éléments, de la finesse des espaces ou encore des saccades de l'œil (figure 6.2c p. 11).

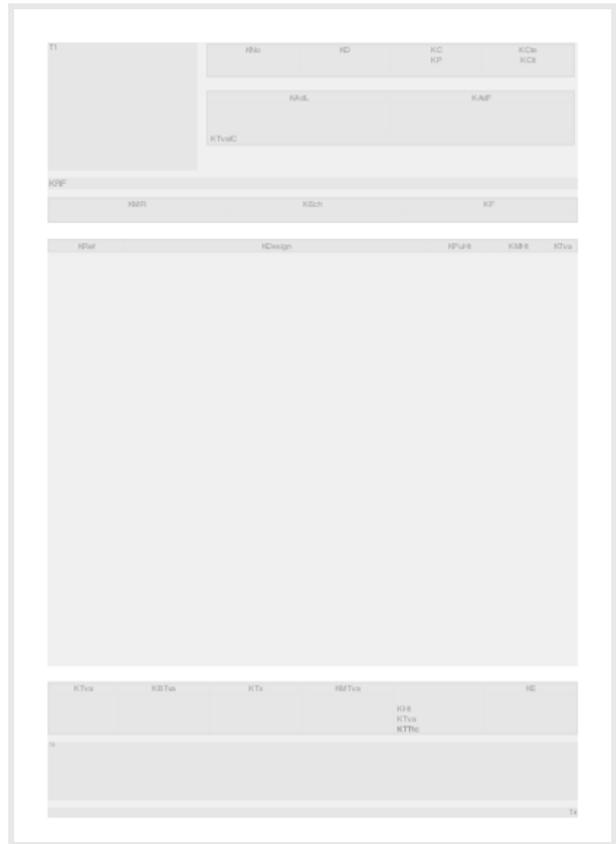
Le mockup est le livrable dont le rendu graphique doit impérativement séduire le client.

La figure 6.2d de la page suivante présente le rendu final d'un type de document PDF obtenu.

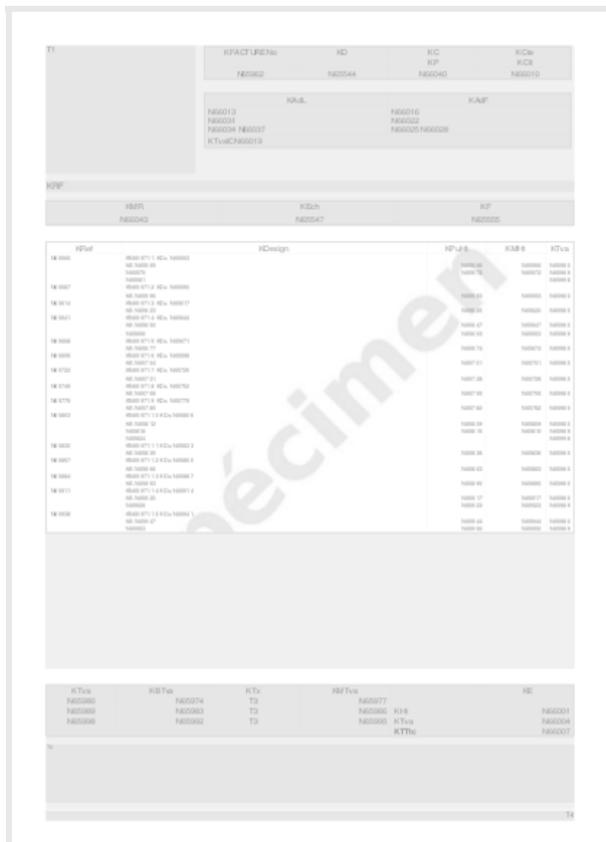
FIG. 6.2 Les étapes du processus de conception graphique.



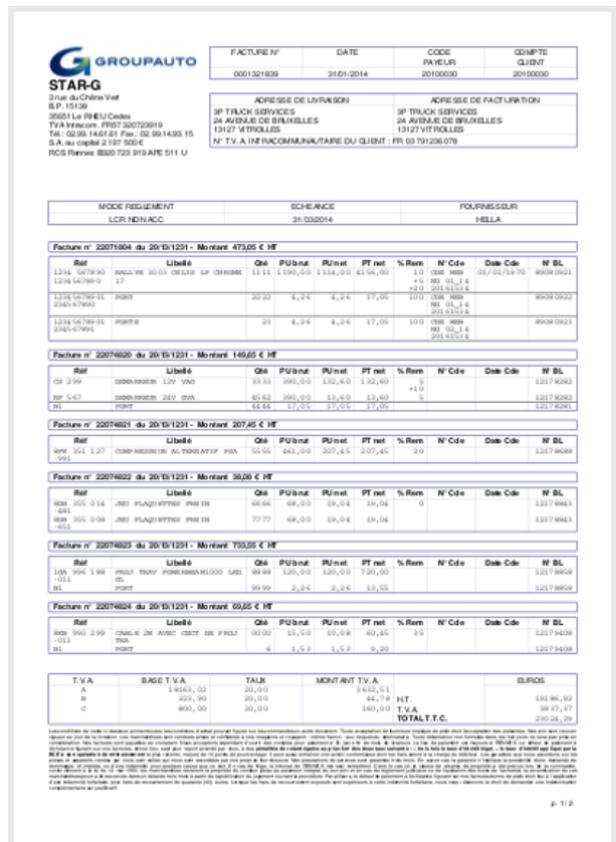
(A) Le zoning.



(B) Le wireframe.



(C) Le mockup.



(D) Un exemple de facture.

## Ch. 7

# DÉVELOPPEMENT

## 7.1 Environnement de développement

Je me suis familiarisé avec un environnement basé sur le système d'exploitation **Linux Debian** [↗](#). Il se compose d'outils classiques et performants.

### 7.1.1 Éclipse et ses outils

J'ai investi du temps pour maîtriser l'IDE **Éclipse** [↗](#) (Integrated Development Environment).

Je gère mes tâches à l'aide de l'ALM **Éclipse Mylyn** [↗](#) (Application Lifecycle Management).

J'ai programmé avec le logiciel d'automatisation des opérations **Apache Ant** [↗](#) (Another Neat Tool) pour créer des fichiers de configuration, des suites de tâches et des macros, dans le but de rationaliser une partie de mon organisation.

J'ai redécouvert la gestion de versions de code source avec l'outil de versionning **Git** [↗](#), et en particulier le plug-in **EGit** [↗](#) qui l'intègre à l'IDE Éclipse.

Enfin, le projet **XSL Tools** [↗](#) d'Éclipse procure des outils fiables pour réaliser un ensemble d'opérations telles que la création, la lecture, la validation de document XML, la recherche de chemin XPath et l'application de transformation XSLT.

### 7.1.2 Visionneuse PDF

La visionneuse PDF expose à l'approbation du client final la partie émergente du résultat du processus de création. Le choix de la visionneuse PDF est par conséquent crucial.

La visionneuse PDF doit être compatible avec les **OS** (Operating System) courants et d'une prise en main agréable.

La **table 7.1** de la page suivante résume les caractéristiques des visionneuses susceptibles de répondre aux besoins du client final.

La visionneuse **Pdf4Eclipse** [↗](#) est intéressante pour le développeur sous l'IDE Éclipse (figure 7.1 p. 13). En revanche, ce plug-in n'offre pas d'outil de recherche.

La visionneuse **Évince** [↗](#) dispose des outils nécessaires pour répondre à de nombreux besoins du développeur et peut être un complément du plug-in Pdf4Eclipse.

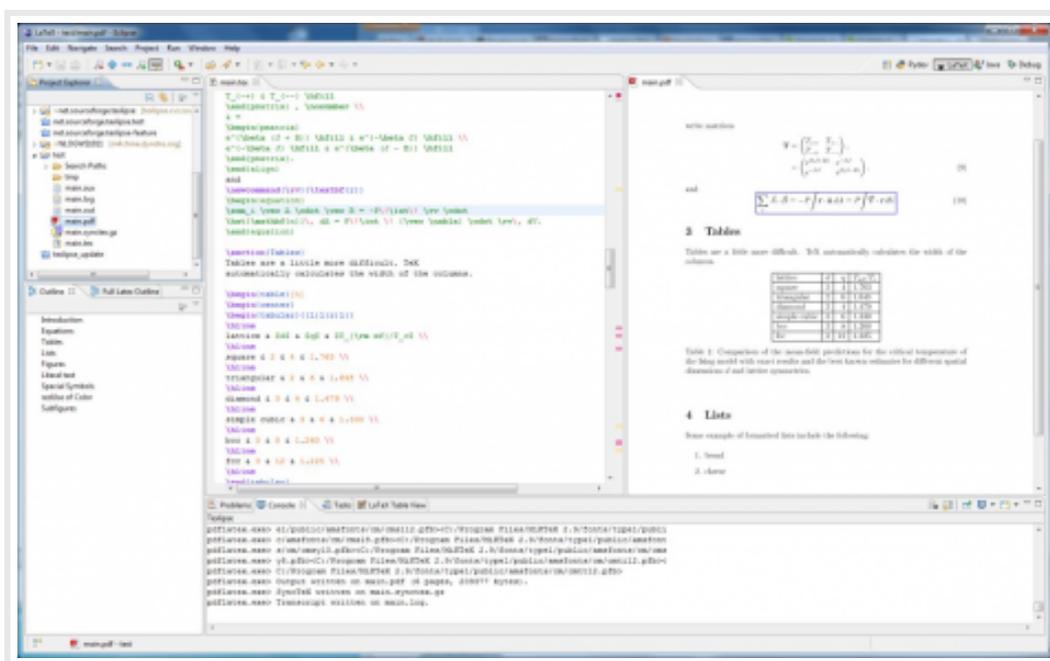
TAB. 7.1 Le comparatif des visionneuses PDF.

Visionneuse	Licence	Recherche	Zoom	Chargement	Liens
Adobe Reader	Acrobat	oui	oui	oui	oui
Évince	GPL-3.0-or-later <sup>1</sup>	oui	oui	oui	oui
Pdf4Eclipse	EPL-2.0 <sup>1</sup>	non	oui	oui	oui

<sup>1</sup> Les identifiants SPDX sont décrits dans la page [Web Licences de réutilisation](#).

La visionneuse [Adobe reader](#) est un outil bien connu des professionnels et du grand public. Elle satisfait l'ensemble des besoins attendus. Les visionneuses PDF suggérées au client sont Évince et Adobe reader, cette dernière bénéficiant d'un effet de halo.

FIG. 7.1 La visionneuse PDF du plug-in Pdf4Eclipse.



## 7.2 Recherche de qualité

Cette recherche est née d'une part de la demande de portabilité et d'optimisation du programme et d'autre part de mon questionnement au sujet de l'évolutivité de FOP.

La portabilité est résolue en appliquant les standards correctement puis en prenant en compte les spécificités de l'environnement ciblé.

Par contre, la notion d'optimisation m'est floue et soulève des problèmes d'influence. Par exemple, la substitution d'une fonction XSLT par une règle XSLT peut améliorer la vitesse de traitement mais diminuer la maintenabilité. Comment solutionner ?

Je trouve un début de réponse avec les facteurs de qualité de MACCALL. Ils m'ont apporté un éclairage sur certaines notions et sur l'influence qu'ils exercent entre eux.

Enfin, de nouvelles versions de langage étant disponibles, la question de l'évolutivité s'est posée puisque **FOP** ne les prend pas encore en charge.

Mais sachant que **FOP** implémente des **API** normalisées pour intégrer ces composants, je leur recherche des alternatives. Je crée un tableau comparatif des caractéristiques de composants récents pour :

- le processeur **XML** Apache Xerces ;
- le processeur **XSLT** Apache Xalan ;
- le processeur **XPath** (XML Path).

À partir des résultats d'une recherche documentaire, j'établis une liste de processeurs alternatifs. Grâce à leur documentation, je retiens certaines caractéristiques :

- l'environnement :
  - le coût financier d'acquisition et de maintenance ;
  - la dynamique de l'équipe de développement : date de création, nouveautés, projection dans l'avenir ;
  - la pertinence de la documentation ;
  - une représentation des applications et des clients ;
- le langage informatique mis en œuvre ;
- la compatibilité avec les standards et les versions : XML 1.0 ou 1.1, XPath 1.0 ou 2.0, XSLT 1.0 ou 2.0 ;
- d'autres caractéristiques dont la vitesse de traitement des données XML.

L'ensemble de ses caractéristiques, associées aux facteurs de qualité de MACCALL, me permettent de retenir par exemple le processeur XSLT [Saxon](#) et le processeur XPath [Psycho-Path](#).

### 7.3 Paramétrisation

Nouvelle demande du chef de projet : des clients souhaitent obtenir des documents PDF personnalisés. J'aborde alors la paramétrisation de feuilles de styles **XSLT**.

J'ai dû distinguer les deux termes **paramétriser** et **paramétrer**.

Pour chaque propriété d'un objet à paramétrer, quatre tâches sont à effectuer dans la feuille de styles **XSLT** :

1. créer une variable constante pour la propriété à paramétrer ;
2. trouver l'ensemble de définition de la valeur de cette variable constante ;
3. bien choisir la valeur par défaut de cette variable constante, qui appartient nécessairement à son ensemble de définition ;
4. remplacer la variable constante par un paramètre.

Ensuite, j'ai créé un fichier de propriétés `personal.properties` afin de protéger la feuille de styles **XSLT** d'une modification malencontreuse et de simplifier la personnalisation effectuée par le client.

Finalement, le client peut :

- ne définir aucune valeur de paramètre : un paramètre est défini avec une valeur par défaut dans la feuille de styles **XSLT** ;
- rechercher sa valeur préférée d'un paramètre en utilisant la commande `java` ;
- fixer la valeur d'une paramètre dans le fichier `personal.properties`.

Ainsi, par exemple, le client peut réutiliser des tonalités de sa palette graphique pour définir la couleur de bordure des tables d'un document PDF.

Au première abord simple, la mise en œuvre de la paramétrisation peut se révéler délicate. En effet, certaines propriétés d'objet peuvent impacter d'autres propriétés de ce même objet ou d'autres objets du document. Il en est ainsi des propriétés de dimension par exemple.

## 7.4 Optimisation

Dans le cadre de la responsabilité sociétale des entreprises, la société Avanteam m'a demandé d'effectuer plusieurs optimisations portant sur la génération des documents PDF et sur les documents **XML** et PDF.

L'optimisation du **XML** consiste principalement à supprimer les espaces de noms et les entités **XML**, puis à minimifier le **XML**.

Le cœur de **FOP** dispose de possibilités d'optimisation. L'optimisation de **FOP** dépend principalement de l'optimisation de ces composants, décrite ci-dessous. En particulier, **FOP** commence le rendu après la fin de la lecture d'une séquence de pages.

L'analyseur syntaxique Apache Xerces **XML** permet de désactiver des caractéristiques et des propriétés. Une désactivation allège le traitement des données.

Pour optimiser les feuilles de styles **XSLT**, j'ai appliqué les principales règles de performances préconisées par Mike KAY.

Concernant l'optimisation XPath. Le parcour d'arbre **XML** nécessite souvent des ressources non négligeables en espace et en temps. Dans une classe d'expressions XPath équivalentes pour un chemin donné, il existe un sous-ensemble d'expressions plus efficaces. Des fonctions XPath sont substituables par une template **XSLT**.

Pour optimiser un document **PDF**, on peut diminuer le nombre de pages ou encore minimiser son poids.

Concernant le nombre de pages, j'ai reconsidéré les éléments du document PDF suivant une contrainte de lisibilité de l'information, tant sur support écran que sur support papier.

Pour diminuer le poids d'un document PDF, j'ai supprimé toute l'information non nécessaire, telle les méta-données XMP, les annotations, le menu, la signature, etc. Je me suis de nouveau intéressé aux fontes intégrées dans un document PDF et en particulier à l'inclusion des seuls glyphes utilisés.

D'autres optimisations sont possibles. Apache Xalan traite des instances de document **XML** compilées. Le moteur de rendu PDF de **FOP** prend directement en entrée un document **XSL-FO**, laissant ainsi le choix des outils de pré-formatage. Enfin, un document PDF peut être compresser.

Les résultats sont significatifs. Le temps de traitement d'une page de document à diminuer d'environ 25 %. Le poids d'une page de document PDF est relativement faible. La quantité de papier a chuté d'environ 10 %.

La but de la demande n'est pas aisé à atteindre mais sa finalité reste très satisfaisante et valorisante.

## 7.5 Internationalisation

Suite à la sollicitation du chef de projet en vue d'internationaliser le programme afin de fournir des entreprises anglophones, en particulier au Royaume-Uni, je me suis penché sur cette problématique.

L'internationalisation d'un contenu, souvent abrégé « i18n », consiste à l'adapter aux publics d'une langue, d'une culture ou encore d'une région donnée. Je me suis approprié les concepts et définitions a) de l'internationalisation et la localisation, b) du contenu international ou multilingue, et c) du contenu monolingue ou multilingue.

Ces derniers n'ont ammenés à m'intéresser aux systèmes d'écritures et au **standard Unicode** [☞](#) (ANDRIES, 2008), aux caractères et encodages ainsi qu'aux fontes et glyphes (HARALAMBOUS, 2004); et finalement à l'ensemble de caractères d'un document **XML**.

J'ai fourni une version internationnalisée du document PDF et répondu à la demande du client.

## 7.6 Documentation

La gestion de la documentation est nécessaire : une ressource documentaire couplée à un système de recherche documentaire augmente le potentiel informationnel et le gain de temps.

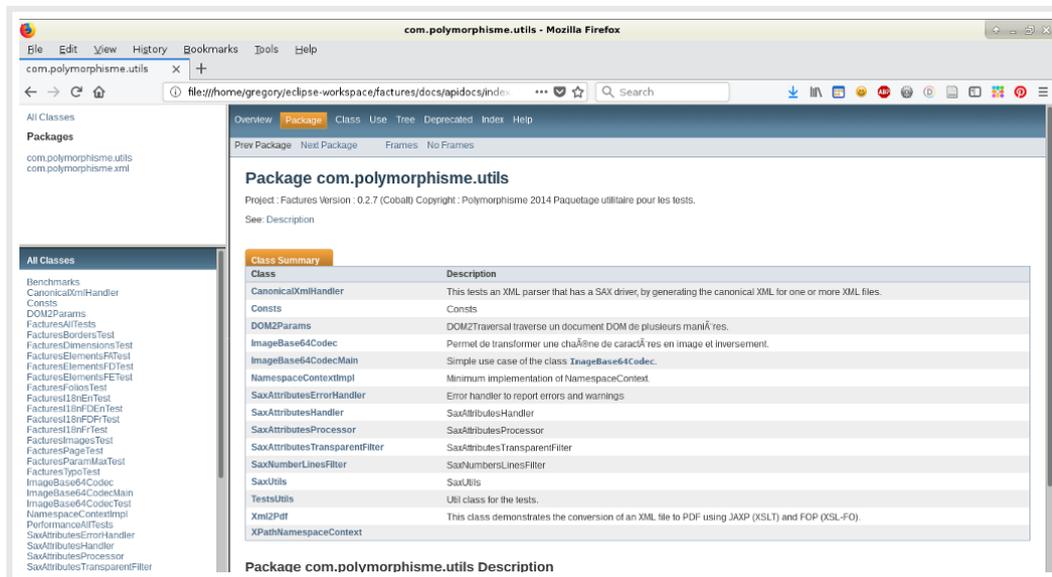
La documentation des outils et des langages de programmation de mon environnement de travail est un prérequis à la conception et au développement.

J'ai organisé ma documentation dans un unique répertoire suivant un plan de classement et des règles de nommage à l'aide des outils informatiques que je connaissais sous **Linux Debian** [☞](#). Le logiciel de gestion de références **Zotero** [☞](#) s'est révélé être un complément très efficace.

Au cours du développement, je génère de la documentation (figure 7.2) à l'aide de l'outil [Javadoc](#) : la documentation du code source Java, le rapport des tests unitaires et le rapport des tests de performance.

Enfin pour chaque livrable, je rédige manuellement le fichier `README.txt`.

FIG. 7.2 La documentation du code source des tests.



## 7.7 Création du livrable

La création du livrable est la dernière étape avant sa vérification et sa validation par le chef de projet.

La tâche Ant « dist » copie l'archive compressée dans le répertoire `dist/`, préalablement créée par la tâche « arch-facture » :

```

1 <target name="arch-facture"
2   description="Création d'une archive zip.">
3   ...
4 <target name="dist" depends="arch-facture,arch-javadoc,arch-junitdoc"
5   description="Création d'une archive.">

```

Le livrable consiste en une archive compressée disponible dans le répertoire `dist/` du système de fichiers local.

Dans une prochaine version, l'archive pourrait être chiffrée, certifiée et signée à l'aide de [Gpg2](#).

## Ch. 8

# TESTS

Le développement d'un programme gérant des données comptables ne semble pas envisageable sans une phase de tests. J'ai opté pour l'approche dite de test boîte noire. J'ai créé des suites de tests fonctionnels et de tests de performance.

## 8.1 Méthode

Dans un premier temps, par manque de connaissances, j'ai codé puis effectué des tests visuels sur le résultat obtenu. Dans un second temps, je me suis intéressé à la méthode **TDD** (Test Driven Development), qui entre dans le cadre méthodologique Agile. Enfin, j'ai automatisé les tests (LINK, 2003).

L'IDE Eclipse intègre le plug-in **JUnit-Tools** [↗](#). De plus, les bibliothèques de tests de la famille xUnit complète cet outil : **JUnit** [↗](#), **XmlUnit** [↗](#), **JUnitPerf** [↗](#) et **JUnit-Benchmarks** [↗](#). Aussi, Ant dispose de tâches interfaçant Eclipse et ces bibliothèques de tests. En complément, le processeur FOP apporte des bibliothèques de tests.

Bien que l'entreprise Avanteam m'ait fourni un jeu de données, la création de données complémentaires s'est imposée afin d'augmenter sa surface. Afin de valider ce jeu de données, j'écris un schéma **XSD** (XML Schema Definition) (van der VLIST, 2002).

Afin de limiter les ressources matérielles lors de l'exécution des tests, j'alloue à la **JVM** (Java Virtual Machine) : 1 Go d'espace disque, 1 Go de mémoire vive et 1 GHz de puissance processeur.

## 8.2 Tests fonctionnels

Les tests fonctionnels permettent de valider les entrées et les sorties. Aussi, un test fonctionnel peut décrire une erreur ou une anomalie et démontrer qu'une correction a été apportée. Enfin, une suite de tests fonctionnels assure la non régression, indispensable lors de l'évolution du code source.

J'ai écrit des tâches Ant de lancement des tests unitaires Java et de création de leur documentation, dont le résultat est présenté par la figure 8.1 p. 19.

```
1 <target name="junit" depends="junit-compile"
2   description="Tests unitaires.">
3 <target name="docs-junit4" depends="junit4"
4   description="Création de documentation : le rapport JUnit4.">
```

FIG. 8.1 La documentation des résultats des tests unitaires.

Unit Test Results.

Class com.polymorphisme.xml.FacturesElementsFDTest

Name	Tests	Errors	Failures	Skipped	Time(s)	Time Stamp	Host
FacturesElementsFDTest	13	0	0	0	0.936	2014-05-03T15:03:22	localhost

Tests

Name	Status	Type	Time(s)
testNumberFDElements[0]	Success		0.099
testNumberFDElements[1]	Success		0.073
testNumberFDElements[2]	Success		0.066
testNumberFDElements[3]	Success		0.058
testNumberFDElements[4]	Success		0.064
testNumberFDElements[5]	Success		0.064
testNumberFDElements[6]	Success		0.063
testNumberFDElements[7]	Success		0.070
testNumberFDElements[8]	Success		0.061
testNumberFDElements[9]	Success		0.059
testNumberFDElements[10]	Success		0.080
testNumberFDElements[11]	Success		0.081
testNumberFDElements[12]	Success		0.078

## 8.3 Tests de performance

Les tests de performance permettent de vérifier si la performance attendue du logiciel est atteinte dans certaines conditions de montée en charge.

De nouveau, j'ai écrit des tâches Ant de lancement des tests de performance et de génération de la documentation de leur résultats :

```

1 <target name="junit-perf" depends="junit-compile"
2   description="Tests de performance.">
3 <target name="benchmarks" depends="junit-compile"
4   description="Benchmarks.">
5 <target name="docs-junit-perf" depends="junit-perf"
6   description="Création de documentation : le rapport JUnit performances.">

```

## 8.4 Résultats

Il faut prendre le temps de construire un jeu de donnée représentatif. La validation des données permet de vérifier le format des données. En disposant de plus de temps, je construirai un jeu de données aléatoires.

Les suites de tests unitaires contribuent à l'amélioration de la fiabilité du code source. Elles constituent un argument qualitatif à faire valoir auprès du client.

La documentation des résultats des tests de performances fournit des indicateurs forts. La mesure du temps d'exécution permet de cibler ou de recadrer les améliorations de la vitesse de production horaire du nombre de page. Ces indicateurs servent de base de calcul pour les ressources matérielles nécessaires en réponse à des contraintes de mémoire ou de temps.

## Ch. 9

# MISE EN PRODUCTION

Lorsque le livrable est terminé, je prépare sa livraison. En effet, je dois être en mesure de le présenter et d'effectuer les tâches administratives qui vont de pair.

## 9.1 Recette

Après réception d'un nouveau livrable, le chef de projet effectue la recette dans son environnement de recette.

Le livrable contient le nécessaire indispensable. Le fichier `README.txt` décrit le contenu du livrable, la procédure d'installation et la procédure d'exécution du programme. Le script Bash `factures.sh` pour Linux ou le script Batch `factures.bat` pour Windows permet de lancer la génération de documents **PDF**.

Au fil des itérations, une ébauche puis un prototype de programme a vu le jour, lequel a été mis en production chez un unique client qui faisait remonter les éventuelles anomalies.

Ensuite, j'ai ajouté une à une les nouvelles fonctionnalités au prototype. Pour chaque nouvelle version du programme, un nouveau client l'utilisait. Une nouvelle fonctionnalité n'est envisagé que lorsque la version courante semble fiable.

## 9.2 Production

Au fil du temps, plusieurs clients ont adoptés ce programme et il a généré des centaines de milliers de factures fournisseurs. Puis, une fois internationalisé, il a été vendu en Angleterre.

La maintenance du programme est effectuée par un membre de l'équipe de la société Avanteam. Seules des modifications mineures devraient voir le jour en fonction des textes de loi.

Finalement en 2020, le programme est au cœur du service de factures fournisseurs de la plateforme Process Suite de l'entreprise Avanteam depuis maintenant plusieurs années.

# CONCLUSION

Au fil des expériences et des unités d'enseignements, j'ai adopté une posture plus fiable, en considérant les projets d'un point de vue plus distant et sous diverses facettes.

L'acquisition de nouvelles connaissances, sommes de processus, de méthodes, illustrées de leurs vocabulaires, est propice à l'émergence et l'évolution de compétences qui s'éprouvent, se confirment et se renforcent par leur mise en œuvre.

La plus importante difficulté rencontrée est certainement le manque de relations. D'une part, sans diplôme ou presque, il n'est pas évident de susciter l'intérêt, d'autre part, j'ai eu peu d'occasions d'échanger de façon constructive à propos de sujets que j'avais à traiter.

La seconde difficulté concerne la négociation dont je n'avais aucune notion. Savoir négocier un forfait, un délai ou encore un partage de responsabilités s'apprend ; cette compétence est primordiale.

Aussi, je me suis aussi rendu compte de manques dans certaines compétences telles que la valorisation de mon savoir-faire et des résultats obtenus, la prospection de nouveaux clients et la construction d'un réseau de connaissances. C'est en train de changer.

Fort de ce constat, je consacre d'avantage de temps sur la connaissance de l'environnement et du contexte métier : le marché, ses lois, ses normes, ses procédures, ses aspects financiers, etc.

De plus, j'élargi mes bases en mettant l'accent sur l'apprentissage de l'anglais et la maîtrise de nouveaux vocables. Je suis beaucoup plus attentif aux croyances et aux attentes.

J'ai pris conscience de l'importance de l'image personnelle. J'ai ainsi tendance à moins mettre en avant le savoir-faire technique au bénéfice de la valorisation du savoir-être et du relationnel. Dans ce sens, j'ai significativement amélioré mes compétences en communication.

À ce jour, je compte poursuivre mon cursus en informatique aussi longtemps que je le pourrais dans le but d'être nécessairement utile au sein d'une entreprise.

# ACRONYMES ET GLOSSAIRE

## Acronymes

### A

- AJLSM** Arvers Jean-Luc Services et Maintenance (p. 3)
- ALM** Application Lifecycle Management (p. 12)
- ANT** Another Neat Tool (p. 12)
- API** Application Programming Interface (pp. 4, 14)

### C

- CAC** Cotation Assistée en Continu (p. 5)
- CDD** Contrat à Durée Déterminée (pp. iv, 1)

### E

- EAD-XML** Encoded Archival Description XML (pp. 3, 4)
- ESN** Entreprise de Service Numérique (p. 2)

### F

- FAI** Fournisseur d'Accès Internet (p. 1)
- FOP** Formatting Objects Processor (pp. 7, 8, 9, 13, 14, 15, 16)

### G

- GED** Gestion Electronique de Documentation (p. 3)

### H

- HR-XML** Human Resource XML (p. 2)

### I

- IDE** Integrated Development Environment (p. 12)
- IHM** Interface Human Machine (pp. 3, 4)

### J

- JVM** Java Virtual Machin (p. 18)

### L

- LAMP** Linux Apache MySQL PHP (p. 1)

### M

- MIAS** Mathématiques et Informatiques Appliquées aux Sciences (p. iv)

### N

- NAF** Nomenclature d'Activité Française (p. 5)

### O

- OS** Operating System (p. 12)
- OVH** On Vous Héberge (p. 1)

### P

- PDF** Portable Document Format (pp. 2, 6, 7, 8, 9, 15, 20)
- PHP** PHP Hypertext Preprocessor (p. 1)

### R

- Rsync** Remote SYNchronization (p. 3)
- RTC** Réseau Téléphonique Commuté (p. 1)

### S

- SDX** Système Documentaire XML (p. 3)
- STS** Sciences, Technologies, Santé (p. ii)

### T

- TDD** Test Driven Development (p. 18)
- TXT** TeXTe (p. 7)

### U

- UML** Unified Modeling Language (p. 2)

### X

- XHTML** eXtensible HyperText Markup Language (p. 3)
- XML** eXtensible Markup Language (pp. iv, 3, 6, 7, 8, 9, 10, 14, 15, 16)
- XPath** XML Path (pp. 8, 14)
- XSD** XML Schema Definition (p. 18)
- XSL-FO** XSL Formatting Objects (pp. 6, 7, 8, 9, 16)
- XSLT** eXtensible Stylesheet Language Transformations (pp. 2, 3, 4, 6, 7, 8, 14, 15)

## Glossaire

### A

**archivistique** science qui étudie les principes et les méthodes appliquées à la collecte, au traitement, à la conservation, à la communication et à la mise en valeur des documents d'archives. (pp. iv, 3)

### B

**BPM** permet d'avoir une vue d'ensemble des processus métiers de l'organisation et de leurs interactions pour les optimiser et les automatiser autant que possible. (p. 5)

### E

**ECM** vise à gérer l'ensemble des contenus d'une organisation. Il s'agit de prendre en compte sous forme électronique les informations qui ne sont pas structurées. (p. 5)

### F

**forêt XML** collection de documents XML, donc d'arbres éléments. (p. 6)

**fragment XML** portion de documents XML pouvant être composée d'un ou plusieurs éléments et pouvant posséder plusieurs racines. (p. 8)

### G

**GED** ensemble des techniques permettant la gestion des documents dans l'entreprise, incluant notamment la gestion de bases de données documentaires. (p. 5)

### L

**langage XSL-FO** langage inclus dans XSL permettant de transformer un document XML en une séquence de blocs formatés pour la sérialisation. (pp. 8, 9)

**langage XPath** langage permettant d'accéder des éléments, attributs, etc. par navigation à l'intérieur d'un document XML, standardisé par le W3C. (p. 8)

**langage XSL** langage de feuille de styles extensible basé sur des règles de productions. (p. 8)

**langage XSLT** langage inclus dans XSL permettant de transformer un document XML en un nouveau document XML ayant une structure éventuellement différente. (p. 8)

**langage XML** métalangage développé par le W3C permettant de définir des langages de marquage de documents ou de messages, au centre d'un ensemble de standards dédiés à la communication dans les SI. (p. 8)

### P

**paramétrer** fixer la valeur d'un paramètre. (p. 14)

**paramétriser** faire dépendre d'un paramètre. (p. 14)

**processeur** module analysant et décodant les balises d'un document XML afin de permettre à l'application utilisant cet analyseur de traiter les données des éléments. (p. 8)

# BIBLIOGRAPHIE

- AMANN, B., & RIGAUX, PHILIPPE. (2002). *Comprendre XSLT*. O'Reilly Éditions
- ANDRIES, P. (2008). *Unicode 5.0 en pratique : Codage des caractères et internationalisation des logiciels et des documents*. Dunod. (pp. 10, 16)
- BOUKHORS, A., KASZYCKI, ALEXANDRE, LAPLACE, JÉRÔME, MUNEROT, SANDRINE & POUBLAN, LAURENT. (2002). *XML, la synthèse : Intégrez XML dans vos architectures*. Dunod
- BROGDEN, B., D'CRUZ, C., & GAITHER, M. (2003). *Cocoon 2 Programming : Web Publishing with XML and Java* (L. STEPHENSON, Éd.). Sybex. (p. 3)
- BURKE, E. M. (2002). *Java et XSLT* (D. BURAUD & CUNEO, SAMUEL, Éd. ; K. CARVER & LAURENT, FRÉDÉRIC, Trad.). O'Reilly Éditions
- DARWIN, I. F. (2002). *Java en action* (D. BURAUD, Éd. ; H. SOULARD, Trad.). O'Reilly Éditions
- EISENBERG, J. D. (2003). *SVG : Production orientée XML de graphiques vectoriels* (S. CUNEO, Éd. ; F. DELHOUME, Trad.). O'Reilly Éditions
- GARDARIN, G. (2002). *XML : Des bases de données aux services Web*. Dunod
- HARALAMBOUS, Y. (2004). *Fontes & codages* (X. CAZIN, Éd.). O'Reilly Éditions. (pp. 10, 16)
- LINK, J. (2003). *Tests unitaires en Java* (RAIMOND, CLAUDE, Trad.). Dunod. (pp. 4, 18)
- MACLAUGHLIN, B. (2002). *Java et XML* (X. CAZIN, Éd. ; A. GACHET, Trad. ; 2<sup>e</sup> éd.). O'Reilly Éditions. (p. iv)
- MANGANO, S. (2003). *XSLT en action* (F. LEROY, Éd. ; É. JACOBONI & A. KETTERLIN, Trad.). O'Reilly Éditions. (p. 3)
- MOCZAR, L., & ASTON, J. (2003). *Cocoon Developer's Handbook*. Developer's library. (p. 3)
- NIEMEYER, P., & KNUDSEN, J. B. (2002). *Introduction à Java* (O'REILLY AND ASSOCIATES INC & BURAUD, DOMINIQUE, Éd. ; V. EYT & F. PETILLOT, Trad. ; 2<sup>e</sup> éd.). O'Reilly Éditions. (p. 3)
- SOCIETY OF AMERICAN ARCHIVISTS (Éd.). (1999). *Encoded Archival Description application guidelines version 1.0*. Society of American Archivists. (p. 3)
- van der VLIST, E. (2002). *XML Schéma* (X. CAZIN, Éd. ; THOMASSON, JEAN-JACQUES, Trad.). O'Reilly Éditions. (p. 18)