

Université Polytechnique de Bobo Dioulasso

.....  
Ecole Supérieure d'Informatique

.....  
Cycle des Ingénieurs de Conception en Informatique

.....  
Année académique 2007-2008



International Consulting and  
Management Group



10 BP 13405 Ouagadougou 10

Tel: (+226) 50 36 99 36

Fax: (+226) 50 36 99 36

Email: icomg@icomg.com

## Plateforme de Services Distants pour une Gestion de la Paie Multi Société

au Burkina Faso

**Mémoire de fin de cycle**

Pour l'obtention du

**Diplôme d'Ingénieur de Conception en Informatique**

Présenté et soutenu publiquement par

**Tong-Noma Firmin KABORE**



Superviseur

**Dr Mesmin DANDJINO**

Enseignant chercheur à l'ESI

Maitre de stage

**M. Halidou ROUAMBA**

Ingénieur de Développement Senior et Support à ICOMG

*A ma mère  
Qui donnera tout pour que je réussisse*



# Remerciements

- ✓ **Mr Moussa KONE**  
*Directeur Général de ICOMG*  
*Pour nous avoir acceptés dans son entreprise et pour sa disponibilité à répondre à toutes nos questions et préoccupations ;*
  
- ✓ **Mr Abdoulaye BILLA**  
*Directeur Technique Associé de ICOMG*  
*Pour ses précieux conseils, son aide et pour nous avoir mis dans de bonnes conditions de travail ;*
  
- ✓ **Mr Halidou ROUAMBA**  
*Ingénieur DBA, Développeur Senior et Support à ICOMG*  
*Pour son encadrement, son implication, ses conseils et toute l'expérience qu'il a bien voulu partagée avec nous ;*
  
- ✓ **Le personnel de ICOMG**  
*Pour avoir facilité notre intégration dans l'entreprise et pour l'ambiance joviale qui a prévalu tout au long de ces quatre (04) mois de stage ;*
  
- ✓ **L'Ecole Supérieure d'Informatique**  
*Pour la formation reçue durant ces cinq (05) ans ;*
  
- ✓ **Mes parents et amis**  
*Pour leur soutien sans faille et leurs conseils motivants.*

## Résumé

Il s'agit de concevoir et de réaliser un logiciel trois tiers, client léger, dans l'Environnement de Développement Intégré (EDI) Oracle JDeveloper 11G, avec le framework de développement d'applications Oracle ADF (totalement compatible Java EE 5). Le logiciel, baptisé PSD4GPMS (Plateforme de Services Distants pour une Gestion de la Paie Multi Société), doit couvrir toutes les fonctionnalités d'un logiciel de gestion de paie (gestion des informations salariales des employés, traitement des salaires des employés, etc.). Il doit être aussi accessible en ligne afin de permettre à des entreprises tierces de s'y abonner pour traiter leur paie.

Comme démarche dans la réalisation de notre mission, nous avons adopté le processus de développement unifié 2TUP (Two Tracks Unified Process). Ainsi, après l'étude préalable du futur système, nous nous sommes attelés sur la capture des besoins fonctionnels et techniques. L'étape suivante a consisté à développer les modèles statiques et dynamiques obtenus à l'étape précédente. Nous avons ensuite injecté les résultats obtenus dans une étape de conception préliminaire, puis dans une étape de conception détaillée. L'étape finale consistait à écrire le code de la plateforme et à concevoir un plan de déploiement. Le présent document récapitule donc nos travaux selon les grandes phases du processus 2TUP.

# sommaire

|   |           |
|---|-----------|
| <b>REMERCIEMENTS .....</b>                        | <b>2</b>  |
| <b>RESUME.....</b>                                | <b>3</b>  |
| <b>SOMMAIRE .....</b>                             | <b>4</b>  |
| <b>INTRODUCTION.....</b>                          | <b>7</b>  |
| <b>1. PRESENTATION DU CABINET ICOMG .....</b>     | <b>8</b>  |
| 1.1 DOMAINE DE COMPETENCE.....                    | 8         |
| 1.2 RESSOURCES HUMAINES.....                      | 9         |
| 1.2.1 Personnel administratif .....               | 9         |
| 1.2.2 Equipe technique .....                      | 9         |
| 1.3 EQUIPEMENTS .....                             | 9         |
| 1.3.1 Le laboratoire de développement .....       | 9         |
| 1.3.2 Le centre de formation .....                | 10        |
| <b>2. INTRODUCTION CONTEXTUELLE DU STAGE.....</b> | <b>11</b> |
| 2.1 CONTEXTE DU PROJET .....                      | 11        |
| 2.2 RESULTAT ET OBJECTIFS.....                    | 12        |
| 2.3 PRESENTATION DE LA DEMARCHE UTILISEE .....    | 12        |
| 2.3.1 Processus de développement.....             | 12        |
| 2.3.2 Aperçu du processus 2TUP .....              | 12        |
| 2.4 RESSOURCES A LA DISPOSITION DU PROJET.....    | 15        |
| 2.4.1 Ressources humaines .....                   | 15        |
| 2.4.2 Ressources matérielles .....                | 16        |

|  |                  |
|--|------------------|
| <b>2.5 PLANNING DE REALISATION</b> .....                                     | <b>16</b>        |
| <b><u>3. BESOINS FONCTIONNELS ET TECHNIQUES</u></b> .....                    | <b><u>20</u></b> |
| <b>3.1 ETUDE PREALABLE</b> .....   | <b>20</b>        |
| 3.1.1 Spécifications fonctionnelles .....                                    | 20               |
| 3.1.2 Spécifications techniques .....  | 22               |
| 3.1.3 Modélisation du contexte.....  | 23               |
| <b>3.2 CAPTURE DES BESOINS FONCTIONNELS</b> .....                            | <b>26</b>        |
| 3.2.1 Identification des cas d'utilisation .....                             | 27               |
| 3.2.2 Description des cas d'utilisation .....                                | 30               |
| 3.2.3 Organisation des cas d'utilisation.....                                | 35               |
| 3.2.4 Identification des classes candidates .....                            | 38               |
| <b>3.3 CAPTURE DES BESOINS TECHNIQUES</b> .....                              | <b>41</b>        |
| <b><u>4. DEVELOPPEMENT DES MODELES</u></b> .....                             | <b><u>42</u></b> |
| <b>4.1 DECOUPAGE EN CATEGORIES ET DEVELOPPEMENT DU MODELE STATIQUE</b> ..... | <b>42</b>        |
| <b>4.2 DEVELOPPEMENT DU MODELE DYNAMIQUE</b> .....                           | <b>52</b>        |
| 4.2.1 Diagramme de séquence : Préparer la paie .....                         | 52               |
| 4.2.2 Diagramme d'état : Employé .....                                       | 54               |
| <b><u>5. CONCEPTION ET REALISATION</u></b> .....                             | <b><u>55</u></b> |
| <b>5.1 PRESENTATION DE JAVA EE</b> .....                                     | <b>55</b>        |
| <b>5.2 PRESENTATION DE ORACLE ADF</b> .....                                  | <b>58</b>        |
| <b>5.3 CONCEPTION ET REALISATION DE LA PLATEFORME</b> .....                  | <b>59</b>        |
| 5.3.1 Conception de la couche des données.....                               | 59               |
| 5.3.2 Conception de la couche métier et de la couche présentation .....      | 61               |
| <b>5.4 PRESENTATION DES CAPTURES D'ECRANS</b> .....                          | <b>70</b>        |
| 5.4.1 Connexion à la plateforme .....  | 71               |
| 5.4.2 La page d'accueil.....   | 71               |
| 5.4.3 Configuration de l'entreprise .....                                    | 72               |
| 5.4.4 Eléments de salaire .....  | 73               |

|   |           |
|---|-----------|
| 5.4.5 Employés.....   | 74        |
| 5.4.6 Insertion d'un nouvel employé.....                        | 75        |
| 5.4.7 Traitement des salaires.....                              | 76        |
| 5.4.8 Liste des traitements .....                               | 77        |
| 5.4.9 Nouveau traitement .....                                  | 78        |
| 5.4.10 Bulletins de paie.....                                   | 79        |
| <b>6. DEPLOIEMENT ET SECURITE .....</b>                         | <b>80</b> |
| <b>6.1 ARCHITECTURE PRELIMINAIRE DE DEPLOIEMENT .....</b>       | <b>80</b> |
| 6.1.1 Oracle Weblogic Server 10.3 .....                         | 80        |
| 6.1.2 Oracle Database 10G .....                                 | 81        |
| 6.1.3 Diagramme préliminaire de déploiement .....               | 81        |
| <b>6.2 PROCEDURES DE SECURITE .....</b>                         | <b>83</b> |
| 6.2.1 Conception d'une solution haute disponibilité .....       | 83        |
| 6.2.2 Intégrité et confidentialité des données .....            | 90        |
| 6.2.3 Conclusion.....   | 94        |
| <b>6.4 ARCHITECTURE FINALE DE DEPLOIEMENT DE PSD4GPMS .....</b> | <b>94</b> |
| <b>CONCLUSION .....</b>   | <b>95</b> |
| <b>BIBLIOGRAPHIE.....</b>                                       | <b>97</b> |



# Introduction

L'Ecole Supérieure d'Informatique (ESI) est une école de l'Université Polytechnique de Bobo-Dioulasso (UPB). Elle dispose principalement de deux cycles de formation : le Cycle des Ingénieurs de Travaux Informatiques (3 ans de formation) et le Cycle des Ingénieurs de Conception en Informatique (5 ans de formation). Tout au long de la formation des ingénieurs de l'école, il est prévu des stages pratiques en entreprise pour leur permettre de mettre en pratique les cours théoriques suivis en classe et aussi leur permettre une intégration rapide et efficace en milieu professionnel.

C'est dans ce cadre qu'à la fin de cette dernière année du Cycle des Ingénieurs de Conception en Informatique (CICI), le cabinet ICOMG a bien voulu nous accepter en son sein pour notre mémoire de fin d'études qui s'est étalé sur quatre (04) mois. Tout au long de ce stage, nous avons travaillé sous l'encadrement des ingénieurs du cabinet et sous celui de notre superviseur, pour aboutir à une plateforme opérationnelle de services distants accessible sur Internet par des entreprises tierces qui s'y abonnent pour la gestion de leur paie en ligne.

Le présent document a été rédigé tout au long de notre stage et présente notamment les différentes phases de notre travail de conception qui ont abouti à la plateforme. Il sera question dans ce mémoire de :

- ✓ Présenter le cabinet ICOMG, structure dans laquelle nous avons effectué notre stage ;
- ✓ Faire une introduction contextuelle du thème d'étude ;
- ✓ Faire une capture des besoins fonctionnels et techniques que doit couvrir la plateforme ;
- ✓ Concevoir et réaliser la plateforme ;
- ✓ Proposer un plan de déploiement et des mesures de sécurité pour assurer une haute disponibilité de la plateforme, la confidentialité et l'intégrité des données manipulées.



# 1

## Présentation du cabinet ICOMG

ICOMG (International Consulting and Management Group) est une société à responsabilité limitée juridiquement constituée en 2005 au Burkina Faso. C'est une société de prestations de services informatiques. Il capitalise dans son domaine de compétence une grande, importante et valorisante expérience professionnelle de ses créateurs, acquise huit (08) années durant (1996-2003) à la direction de SORECO Consult, premier Country Partner de Oracle CORPORATION au Burkina Faso.

### 1.1 DOMAINE DE COMPETENCE

ICOMG est primordialement actif dans le domaine de l'ingénierie informatique. Ses offres de prestations couvrent les domaines suivants :

- **Etudes**
  - ✓ Audits Informatiques ;
  - ✓ Schémas directeurs et plans informatiques ;
  - ✓ Cahier des charges pour la sélection et l'achat de matériels et logiciels.
  
- **Génie logiciel**
  - ✓ Etudes et développements spécifiques ;
  - ✓ Mises en œuvre de solutions en bases de données ORACLE dans divers environnements d'exploitation ;
  - ✓ Installations et paramétrages de logiciels ORACLE ;
  - ✓ Installations et paramétrages de progiciels de gestion (Paie, Gestion du personnel, Comptabilité...).

- **Formation**
  - ✓ Formation aux outils ORACLE (Fondamentaux SQL, Forms, Reports, DBA, Sécurité-DB, Application Server, BI et DWH) ;
  - ✓ Formation des utilisateurs et administrateurs UNIX (Solaris) et LINUX ;
  - ✓ Formation à la carte (Bureautique, Assistance à l'exploitation, ...).

## **1.2 RESSOURCES HUMAINES**

Il s'agit du personnel administratif et de l'équipe technique.

### **1.2.1 Personnel administratif**

Le personnel administratif est constitué du directeur général, du directeur administratif, de l'agent comptable, de l'assistante de direction, du chauffeur coursier, du gardien et des agents de nettoyage et d'entretien.

### **1.2.2 Equipe technique**

Sous la houlette d'un Directeur Technique Associé, ICOMG dispose d'une équipe technique de professionnels expérimentés et de jeunes diplômés rigoureusement formés aux méthodes de travail du cabinet, régulièrement perfectionnés dans les meilleurs workshops et stages nationaux et internationaux.

## **1.3 EQUIPEMENTS**

En son siège, outre les bureaux entièrement équipés et fonctionnels, ICOMG dispose d'un laboratoire de développement et d'un centre de formation professionnelle.

### **1.3.1 Le laboratoire de développement**

Equipé de serveurs de bases de données Oracle 9i/10g, de postes de travaux évolués installés sur le réseau local, il abrite l'Atelier de Génie Logiciel (Oracle Designer) ainsi que les outils de développement client/serveur et Web (Oracle 9i/10g Internet Developer Suite) exploitant une base de données Oracle sous LINUX, entièrement dédiée aux travaux de conception et de développement. Le laboratoire de développement de ICOMG dispose d'une bibliothèque dotée des meilleurs manuels Oracle, ainsi que d'une batterie complète des outils Oracle, toutes versions confondues et dans la quasi-totalité des systèmes d'exploitation hôtes.

Les développeurs ICOMG bénéficient également des librairies de programmes standards développés en PL/SQL et en JAVA2, testés et éprouvés chez les clients depuis des années.

### 1.3.2 Le centre de formation

Agréé « Approved Oracle University Partner », le centre de formation ICOMG est logé dans une salle spécialisée, adaptée à la formation professionnelle. Salle climatisée de capacité « tout confort » de quinze (15) places, elle est équipée de PC individuels, d'un vidéo projecteur haute résolution, de matériel de production de supports de formations (imprimantes et photocopieuses de grande productivité).

Animé par des formateurs locaux et internationaux (partenariat) de très haut niveau, le centre ICOMG assure des sessions de formation régulières et à la carte.



# 2

## Introduction contextuelle du stage

Cette section présente la problématique du thème d'étude, la méthode de développement utilisée et le planning de réalisation du projet.

### 2.1 CONTEXTE DU PROJET

La motivation des employés est l'un des principaux soucis pour toute entreprise désirent atteindre les objectifs qu'elle s'est fixée. Cette motivation passe sans doute par un traitement efficace et dynamique des salaires des employés. La majorité des entreprises dans notre pays (Burkina Faso) ont recours aux progiciels, logiciels génériques souvent non adaptés à leurs besoins propres. En effet, les règles de calcul des éléments de salaire diffèrent d'un pays à un autre ; ces règles étant issues des conventions collectives et sectorielles de chaque Etat. De ce fait, la nécessité d'avoir des logiciels conformes à nos besoins propres (conçus et développés localement), sur le plan du traitement des salaires et dans bien d'autres domaines, s'est faite ressentir et constitue aujourd'hui l'un des principaux soucis du premier responsable de la maison ICOMG.

Aussi, la gestion de ces fonctions administratives (comptabilité, paie, etc.) dévie la majorité des entreprises de leurs missions propres. L'on trouve alors aujourd'hui des entreprises, qui afin de pouvoir se consacrer totalement aux aspects stratégiques de leurs métiers, confient la gestion de certaines fonctions comme la paie à des cabinets extérieurs qui s'en chargent. Notre mission est de faciliter le travail des ces cabinets.

Nous avons donc à notre charge, dans le cadre de notre mémoire de fin d'études, la mise en place d'une **Plateforme de Services Distants pour une Gestion de la Paie Multi Société au Burkina Faso (PSD4GPMS)**. Cette plateforme doit être accessible en ligne et permettre aux entreprises tierces une externalisation complète de leur paie.

## 2.2 RESULTAT ET OBJECTIFS

Le projet doit aboutir à un logiciel trois tiers, client léger, recouvrant toutes les fonctionnalités d'un logiciel de paie. Ce logiciel doit être déployé dans un cabinet et accessible en ligne, avec les deux objectifs (scénarii d'utilisation) suivants :

- ✓ **Gestion complète de la paie en ligne** permettant à des entreprises tierces de s'abonner à la plateforme et traiter leur paie en ligne ;
- ✓ **Externalisation complète de la paie** permettant la gestion complète de la paie des entreprises abonnées par les acteurs du cabinet.

Dans le souci de produire un logiciel de qualité qui répondra aux attentes qui nous ont été sollicitées et cela dans la marge de temps qui nous est imparti, il nous est indispensable de nous organiser autour d'une démarche.

## 2.3 PRESENTATION DE LA DEMARCHE UTILISEE

La complexité croissante des systèmes informatiques a conduit les concepteurs à s'intéresser aux processus de développement.

### 2.3.1 Processus de développement

Un processus de développement est un ensemble d'étapes, en partie ordonnées qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. L'objet d'un processus de développement est de produire des logiciels de qualité, qui répondent aux besoins de leurs utilisateurs dans des coûts et des délais prévisibles.

Le processus de développement adopté pour la réalisation de la plateforme est le processus 2TUP (2 Two Tracks Unified Process) détaillé dans [1].

### 2.3.2 Aperçu du processus 2TUP

Le processus 2TUP est un processus unifié et répond donc aux caractéristiques ci après.

- ✓ **Il est construit sur UML.** UML (Unified Modeling Language) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures

logicielles, concevoir des solutions et communiquer des points de vue. UML permet la communication avec le client sous les différents points de vue de modélisation (fonctionnel, statique, dynamique) à travers ses multiples diagrammes. Il unifie à la fois les notations et les concepts orientés objet ;

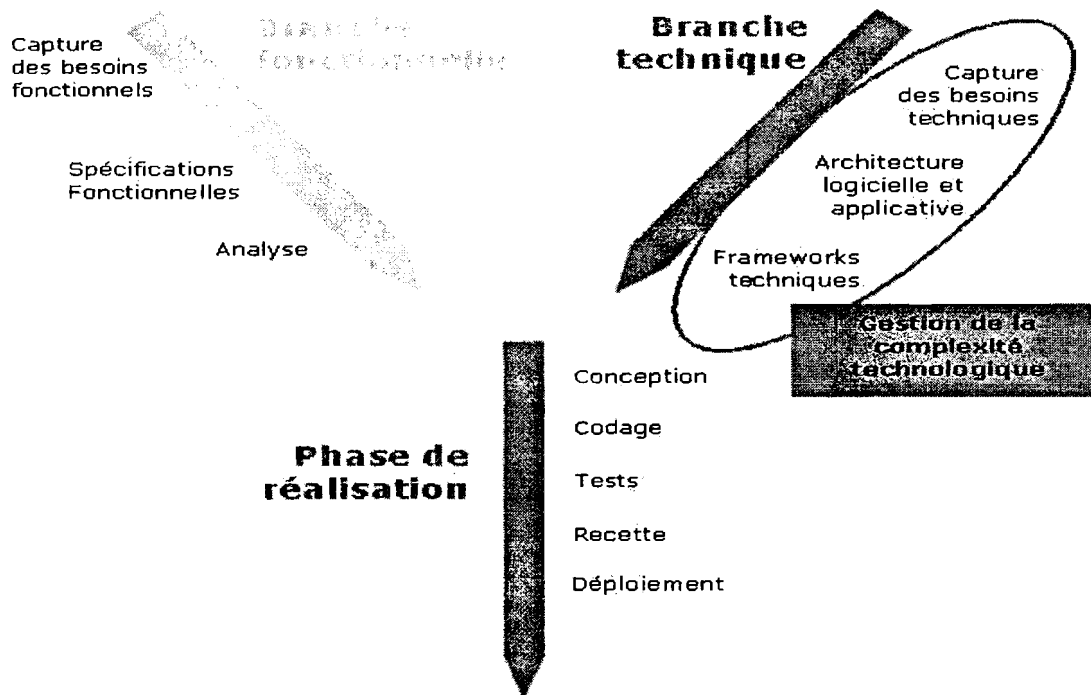
- ✓ **Il est itératif et incrémental.** Une itération est une séquence distincte d'activités avec un plan de base et des critères d'évaluation qui produit une release. Le contenu d'une itération est porteur d'améliorations ou d'évolutions du système et il peut donc être évalué par les utilisateurs ;
- ✓ **Il est piloté par les risques.** Dans ce cadre, les causes majeures d'échec d'un projet doivent être écartées en priorité ;
- ✓ **IL est orienté utilisateur** puisque que la spécification et la conception du logiciel sont construites à partir des modes d'utilisation attendus par les utilisateurs ;
- ✓ **Il est centré sur l'architecture.** L'architecture est l'ensemble des décisions d'organisation du système logiciel qui défend les intérêts de son propriétaire final. Les intérêts s'expriment en termes d'exigences fonctionnelles, techniques et économiques. Un processus centré sur l'architecture impose le respect des décisions d'architecture à chaque étape de construction du modèle.

« 2 Tracks » (soit «2 chemins ») pour dire que le processus suit deux chemins. Il s'agit des chemins « fonctionnel » et « d'architecture technique » qui correspondent aux deux axes de changement imposés au système informatique. L'axiome fondateur du processus 2TUP est de constater que toute évolution imposée au système d'information peut se décomposer et se traiter parallèlement. A l'issue de l'étude du modèle fonctionnel et de celle de l'architecture technique, la réalisation du système consiste à fusionner les résultats des deux branches. Cette fusion conduit à l'obtention d'un processus en forme de Y<sup>1</sup> comme l'indique la figure 2.1.

Le processus 2TUP (encore appelé processus en Y) est construit autour des trois (03) branches suivantes : la branche gauche, la branche droite et la branche du milieu.

---

<sup>1</sup> C'est pour cela que le processus 2TUP est aussi appelé processus en Y



**Figure 2.1 :** processus 2TUP

La branche gauche (fonctionnelle) comporte la capture et l'analyse des besoins fonctionnels du système à mettre en place. Elle définit les fonctionnalités du système d'une manière exhaustive et précise, en se focalisant sur le métier des utilisateurs (cas d'utilisation) et qualifie le risque de produire un système inadapté aux besoins des utilisateurs.

La branche droite (architecture technique) comporte la capture des besoins techniques qui recensent toutes les contraintes et choix techniques dimensionnant le système. Elle comporte la conception générique qui définit les composants nécessaires à la construction de l'architecture technique.

La branche du milieu comporte la conception préliminaire qui intègre le modèle d'analyse dans l'architecture technique et la conception détaillée qui est le développement du modèle d'analyse. Elle comporte aussi l'étape de codage qui produit les composants et teste au fur et à mesure les unités de code réalisées et l'étape de recette qui valide les fonctions du système développé.

## 2.4 RESSOURCES A LA DISPOSITION DU PROJET

La réussite du projet nécessite des ressources humaines et matérielles.

### 2.4.1 Ressources humaines

Suivant les rôles des différents intervenants dans le projet, nous pouvons catégoriser les intervenants en quatre (04) groupes principaux : le comité de pilotage, le comité technique, le groupe des utilisateurs et le groupe de projet.

#### 2.4.1.1 Le Comité de pilotage

Cette équipe va servir de relais de pouvoir, elle va permettre de faire approuver au plus haut niveau les décisions majeures et les passages d'étapes. Les membres de ce comité, dont les noms sont consignés dans le tableau 2.1, ont un pouvoir direct ou indirect sur les moyens affectés au projet.

**Tableau 2.1** : membres du comité de pilotage

| NOM                | FONCTION                    | SOCIETE |
|--------------------|-----------------------------|---------|
| M. Moussa KONE     | Directeur Général           | ICOMG   |
| M. Abdoulaye BILLA | Directeur Technique Associé | ICOMG   |

#### 2.4.1.2 Le Comité technique

Le comité technique assure la coordination technique du projet et contrôle la qualité des résultats. Les noms des membres de ce groupe sont consignés dans le tableau 2.2.

**Tableau 2.2** : membres du comité technique

| NOM                | FONCTION                              | SOCIETE |
|--------------------|---------------------------------------|---------|
| M. Halidou ROUAMBA | Ingénieur de développement et Support | ICOMG   |
| M. Abdoulaye BILLA | Directeur Technique Associé           | ICOMG   |

#### 2.4.1.3 Le Groupe des utilisateurs

Ce groupe est l'entité porteuse du besoin ; il définit clairement les objectifs et les résultats attendus du projet. Aussi, il aide le groupe de projet à l'apprentissage du métier et s'assure à



chaque étape que les besoins exprimés sont pris en compte. Le groupe des utilisateurs est l'ensemble des intervenants cités dans les tableaux précédents.

#### **2.4.1.4 Le Groupe de projet**

Ce groupe se résume au stagiaire que nous sommes. Il est chargé de l'exécution des différentes tâches du projet.

#### **2.4.2 Ressources matérielles**

Pour le projet, nous disposons d'un ordinateur connecté à Internet. Tous les documents électroniques et « physiques » dont dispose le cabinet, nous ont été présentés et nous y avons grandement accès. Nous avons aussi accès aux ressources partagées du réseau telles que l'imprimante.

### **2.5 PLANNING DE REALISATION**

Le processus de développement 2TUP s'articule autour des différentes étapes suivantes :

- ✓ Etude préliminaire ;
- ✓ Capture des besoins fonctionnels ;
- ✓ Capture des besoins techniques ;
- ✓ Découpage des classes candidates en catégories ;
- ✓ Développement du modèle statique ;
- ✓ Développement du modèle dynamique ;
- ✓ Conception générique ;
- ✓ Conception préliminaire ;
- ✓ Conception détaillée ;
- ✓ Codage et test des unités de code ;
- ✓ Recette.

Connaissant le processus de développement à utiliser et les différentes étapes de celui-ci, nous pouvons donc construire un planning pour l'exécution du projet. Le planning ci-dessous (tableaux 2.4 à 2.12) prend en compte les différentes étapes du processus 2TUP (citées ci-dessus) et fournit une décomposition plus atomique des tâches.

Tableau 2.4 : planning de l'étude préliminaire (7 jours)

| <b>ETAPE 0 : ETUDE PRELIMINAIRE</b>   |               |             |               |   |
|---|---------------|-------------|---------------|---|
| <b>Objectif : Etude succincte des besoins fonctionnels et opérationnels</b> |               |             |               |   |
| Tâches  | Date de début | Date de fin | Durée (jours) | Livrables   |
| Rédaction du cahier des charges préliminaires                               | 13/08/08      | 17/08/08    | 5             | • Cahier des charges  |
| Modélisation du contexte  | 18/08/08      | 19/08/08    | 2             | • Diagramme de contexte dynamique<br>• Diagramme de contexte statique |

Tableau 2.5 : planning pour la capture des besoins fonctionnels (16 jours)

| <b>ETAPE I : CAPTURE DES BESOINS FONCTIONNELS</b>   |               |             |               |   |
|---|---------------|-------------|---------------|---|
| <b>objectif : Formaliser et détailler la partie « fonctionnelle » ébauchée au cours de l'étude préliminaire</b> |               |             |               |   |
| Tâches  | Date de début | Date de fin | Durée (jours) | Livrables   |
| Identification des cas d'utilisations (CU)  | 20/08/08      | 21/08/08    | 2             | • Diagramme de CU   |
| Description des cas d'utilisation   | 22/08/08      | 26/08/08    | 5             | • Fiche de description des CU<br>• Diagrammes dynamiques                            |
| Structuration des cas d'utilisation   | 27/08/08      | 28/09/08    | 2             | • Diagramme de CU raffiné et consolidé<br>• Packages de spécification fonctionnelle |
| Identification des classes candidates   | 28/08/08      | 02/09/08    | 6             | • Diagramme de classes participantes  |
| Validation et consolidation   | 03/09/08      | 03/09/08    | 1             | • Matrice de traçabilité  |

Tableau 2.6 : planning pour la capture des besoins techniques (6 jours)

| <b>ETAPE II : CAPTURE DES BESOINS TECHNIQUES</b>  |               |             |               |   |
|---|---------------|-------------|---------------|---|
| <b>objectif : Formaliser et détailler la partie « non fonctionnelle » ébauchée au cours de l'étude préliminaire</b> |               |             |               |   |
| Tâches  | Date de début | Date de fin | Durée (jours) | Livrables   |
| Capture des spécifications techniques   | 04/09/08      | 05/09/08    | 2             | • Diagramme de configuration matérielle<br>• Diagramme de déploiement |
| Capture des spécifications logicielles  | 06/09/08      | 09/09/09    | 4             | • Modèle de spécification logicielle                                  |

Tableau 2.7 : planning pour le découpage en catégorie (6 jours)

| <b>ETAPE III : DECOUPAGE EN CATEGORIES</b>  |               |             |               |  |
|---|---------------|-------------|---------------|--|
| <b>Objectif : passage d'une structuration fonctionnelle (cas d'utilisation, package) à une structuration objet (classes et catégories).</b> |               |             |               |  |
| Tâches  | Date de début | Date de fin | Durée (jours) | Livrables  |
| Organisation des classes en catégories.   | 10/09/08      | 15/09/08    | 6             | <ul style="list-style-type: none"> <li>Diagramme des packages des classes</li> </ul> |

Tableau 2.8 : planning pour la capture des besoins fonctionnels (8 jours)

| <b>ETAPE IV : DEVELOPPEMENT DU MODELE STATIQUE</b>  |               |             |               |  |
|---|---------------|-------------|---------------|--|
| <b>Objectif : Détailler, compléter, optimiser les diagrammes de classe de l'étape précédente.</b>   |               |             |               |  |
| Tâches  | Date de début | Date de fin | Durée (jours) | Livrables  |
| <ul style="list-style-type: none"> <li>Affinage des classes</li> <li>Affinage des associations</li> <li>Ajout des attributs</li> <li>Ajout des opérations</li> <li>Optimisation avec la généralisation</li> </ul> | 16/09/08      | 23/09/08    | 8             | <ul style="list-style-type: none"> <li>Diagrammes de classes complets</li> </ul> |

Tableau 2.9 : planning pour le développement du modèle dynamique (5 jours)

| <b>ETAPE VI : DEVELOPPEMENT DU MODELE DYNAMIQUE</b>   |               |             |               |   |
|---|---------------|-------------|---------------|---|
| <b>Objectif : Description des interactions entre objets, description du cycle de vie des objets</b> |               |             |               |   |
| Tâches  | Date de début | Date de fin | Durée (jours) | Livrables   |
| Formalisation des scénarii  | 24/09/08      | 28/09/08    | 5             | <ul style="list-style-type: none"> <li>Diagrammes de communication</li> <li>Diagrammes de séquence</li> <li>Diagrammes d'états</li> </ul> |

Tableau 2.10 : planning pour la conception générique du projet (3 jours)

| <b>ETAPE VII : CONCEPTION GENERIQUE</b>                       |               |             |               |           |
|---|---------------|-------------|---------------|-----------|
| <b>Objectif : Définir les frameworks et outils à utiliser</b> |               |             |               |           |
| Tâches  | Date de début | Date de fin | Durée (jours) | Livrables |
| Choix des frameworks  | 29/09/08      | 01/10/08    | 3             |           |

Tableau 2.11 : planning pour la conception préliminaire du projet (7 jours)

| <b>ETAPE VIII : CONCEPTION PRELIMINAIRE</b>   |               |             |              |   |
|---|---------------|-------------|--------------|---|
| <b>Objectif : Fusionner l'analyse technique et l'analyse fonctionnelle pour aboutir à l'architecture générale du système.</b> |               |             |              |   |
| Tâches  | Date de début | Date de fin | Durée (jour) | Livrables                                       |
| Esquisse de l'architecture générale   | 02/10/08      | 08/10/08    | 7            | Diagramme des composants et des unités logiques |

Tableau 2.12 : planning pour la conception détaillée du projet (28 jours)

| <b>ETAPE IX CONCEPTION DETAILLEE</b>  |               |             |              |  |
|---|---------------|-------------|--------------|--|
| <b>Objectif : Développer la conception préliminaire pour donner une image prête à coder de l'application.</b>   |               |             |              |  |
| Tâches  | Date de début | Date de fin | Durée (jour) | Livrables  |
| <b>Micro processus de conception logique</b><br>1. Conception des classes<br>2. Conception des associations<br>3. Conception des attributs<br>4. Conception des opérations<br>5. Validation du modèle | 09/10/08      | 15/10/08    | 7            | Diagrammes des classes complets (ajouts des classes de conception, classes techniques, des designs patterns) |
| Conception de la couche présentation  | 16/10/08      | 22/10/08    | 7            | Maquettes d'interfaces utilisateurs  |
| Conception de la couche Métier  | 23/10/08      | 29/10/08    | 7            |  |
| Conception du stockage des données  | 30/10/08      | 05/11/08    | 7            |  |

Tableau 2.13 : planning pour le codage et les tests (37 jours)

| <b>ETAPE X : CODAGE ET TESTS</b>                            |               |             |              |                     |
|---|---------------|-------------|--------------|---------------------|
| <b>Objectif : Coder l'application et procéder aux tests</b> |               |             |              |                     |
| Tâches  | Date de début | Date de fin | Durée (jour) | Livrables           |
| Codage et tests   | 06/11/08      | 12/12/08    | 37           | Le logiciel 3 tiers |

# 3

## Besoins fonctionnels et techniques

Cette section présente l'analyse conceptuelle de la plateforme à réaliser. Elle se focalise notamment sur la capture des besoins fonctionnels et techniques que doit satisfaire la plateforme.

### 3.1 ETUDE PREALABLE

La toute première étape du processus 2TUP est l'étude préliminaire du futur système. Cette étape a pour objet d'effectuer une étude succincte des besoins fonctionnels et opérationnels que doit satisfaire la plateforme. Elle doit aboutir à l'obtention du cahier des charges préliminaires définissant les spécifications de base du système à réaliser.

#### 3.1.1 Spécifications fonctionnelles

Les spécifications fonctionnelles décrivent les besoins, en termes de fonctionnalités, que doit couvrir le nouveau système. Ces besoins sont généralement obtenus auprès des experts métiers qui sont les spécialistes du domaine d'étude. Après les interviews réalisées, il a été résumé les exigences et besoins suivants :

- ✓ Le cœur de la plateforme est le traitement **périodique** des salaires des employés. Cette période, appelée période de base doit être un paramètre variable et sa durée est relative à chaque entreprise ;
- ✓ Chaque employé, en fonction de son classement et de son poste dans l'entreprise, dispose d'avantages permanents, qui doivent être pris en compte dans tous les traitements des salaires qui le concernent. A chaque nouveau traitement des salaires, on peut accorder à l'employé d'autres avantages appelés avantages variables (prime,

heures supplémentaires, etc.). On peut aussi lui faire des retenues qui sont liées aux prêts en cours qui lui sont accordés par l'entreprise ;

- ✓ Des prêts (ou avances sur salaire) peuvent être accordés à un employé par son entreprise ou par toute autre entreprise (banques, caisses, ...). Ces prêts ont des répercussions sur le traitement des salaires comme mentionné au point précédent ;
- ✓ Le traitement des salaires consiste à déterminer, à partir des informations salariales de chaque employé (avantages permanents et éléments variables) et des paramètres étatiques<sup>1</sup> en vigueur, tous les éléments calculables (Impôt Unique sur le Traitement des Salaires, Taxe Patronale d'Apprentissage, salaire net, salaire net à payer, etc.). En principe tous les éléments calculables sont les mêmes pour toutes les entreprises, mais souvent quelques spécificités peuvent être identifiées ;
- ✓ Généralement, chaque entreprise a son propre modèle de bulletins de paie. Pour ce faire, la plateforme se doit d'être fortement paramétrable pour prendre en compte les spécificités de chaque entreprise. Elle doit notamment permettre la personnalisation des bulletins de paie ;
- ✓ Chaque employé peut effectuer des heures supplémentaires. Ces heures supplémentaires effectuées ont des répercussions positives sur le traitement de son salaire. De même, toutes les absences non justifiées auront des répercussions négatives sur les salaires des employés ;
- ✓ Un employé peut avoir des relations avec les institutions de la place (banques, pharmacies, stations, assurances, ...). Les relations de l'employé peuvent induire des entrées dans le traitement de son salaire. Un employé peut par exemple décider que les frais de gestion de son contrat d'assurance soient directement prélevés de son salaire ;
- ✓ Les cessations de travail (départ à la retraite, congé, licenciement) nécessitent des traitements de salaire particuliers ;

---

<sup>1</sup> La paie est réglementée par un ensemble de lois élaborées par les conventions collectives et sectorielles de l'Etat.

- ✓ Pour faciliter la gestion des avantages des employés, certaines entreprises disposent d'une grille salariale. La grille salariale permet de classer les employés de l'entreprise et d'harmoniser les avantages permanents ;
- ✓ Le traitement des salaires consiste aussi à l'édition d'un ensemble d'états, le but de la plateforme est de permettre l'élaboration et l'impression de ces états (ordre de virement dans les banques des employés, états des retenues IUTS<sup>1</sup> et TPA<sup>2</sup>, états des cotisations CNSS<sup>3</sup>, statistiques diverses, etc.) ;
- ✓ Le traitement des salaires nécessite la gestion du parcours salarial de l'employé dans l'entreprise (affectation, avancement, nomination, etc.) ;
- ✓ Les procédures de calcul des éléments de salaire sont réglementées par les conventions collectives et sectorielles de l'Etat ;
- ✓ Chaque entreprise peut s'abonner à la plateforme pour une durée déterminée et accéder à distance aux services de celle ci pour l'administration de sa paie.

### 3.1.2 Spécifications techniques

Les spécifications techniques s'expriment, en termes de sécurité, de performance, d'extensibilité, de disponibilité, de frameworks et d'outils à utiliser. Elles décrivent les besoins opérationnels que doit couvrir la plateforme. Les spécifications techniques pour la plateforme PSD4GPMS peuvent être résumées comme suit :

- ✓ La plateforme doit être développée en 3 tiers (client léger) suivant la technologie Java EE (ou technologie implémentant Java EE) ;
- ✓ Elle doit être implémentée sur toutes ses composantes en environnement Oracle ;
- ✓ Chaque entreprise abonnée accède à la plateforme à travers un module de sécurité et voit uniquement les données qui la concernent ;

---

<sup>1</sup> IUTS : Impôt unique sur le traitement des salaires

<sup>2</sup> TPA : Taxe Patronale d'Apprentissage

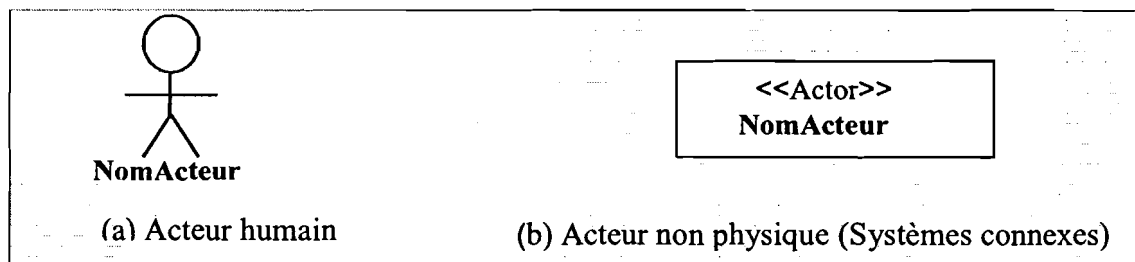
<sup>3</sup> CNSS : Caisse Nationale de Sécurité Sociale

- ✓ Plusieurs entreprises doivent pouvoir se connecter simultanément au système, chaque entreprise ayant l'impression d'être la seule connectée (isolation) ;
- ✓ La plateforme doit être performante et hautement disponible ;
- ✓ Elle doit être fortement sécurisée, extensible et facilement évolutive ;

### 3.1.3 Modélisation du contexte

En plus du cahier des charges préliminaires, cette étape présente aussi les interactions entre les différents acteurs de la plateforme et la plateforme elle-même. Les diagrammes de contexte statique et de contexte dynamique permettent de modéliser le contexte de la plateforme à réaliser. Les concepts clefs employés ici sont les notions d'acteur et de message.

A cette étape du processus, un acteur se définira comme toute entité externe du système mais qui interagit avec le système. La communication se fait par l'envoi de simples messages. UML n'a pas standardisé la notation des acteurs, mais dans la modélisation de la plateforme, il a été adopté les notations suivantes (figure 3.1):



**Figure 3.1** : représentation d'un acteur

Un message représente la spécification d'une communication unidirectionnelle entre objets (entre la plateforme et un acteur par exemple) qui transporte de l'information avec l'intention de déclencher une réaction chez le récepteur.

#### 3.1.3.1 Identifications des acteurs

Pour la plateforme PSD4GPMS, il convient de définir deux catégories ou types d'acteurs :

- ✓ **Les acteurs niveau « entreprise »** qui sont liés à une seule et unique entreprise. Ils ignorent la présence d'autres entreprises et se connectent à la plateforme à travers la souscription de leur entreprise ;



- ✓ Les acteurs niveau « central » qui peuvent « voir » plusieurs entreprises. Ils interagissent avec le système au nom de plusieurs entreprises. C'est notamment le cas de l'administrateur central de la plateforme.

Les acteurs<sup>1</sup> interagissant avec la plateforme sont consignés dans le tableau 3.1.

**Tableau 3.1** : acteurs interagissant avec la plateforme

| Acteur                                     | Type                  | Rôle  |
|--|-----------------------|---|
| Administrateur central (super utilisateur) | Niveau central        | <ul style="list-style-type: none"> <li>✓ Administre le système ;</li> <li>✓ Maintient à jour les paramètres étatiques ;</li> <li>✓ Gère les comptes des autres utilisateurs ;</li> <li>✓ Gère les abonnements des entreprises.</li> </ul> |
| Acteur cabinet                             | Niveau central        | Prend en charge l'intégralité du processus de paie de plusieurs entreprises   |
| RH   | Niveau « entreprise » | <ul style="list-style-type: none"> <li>✓ Gère les paramètres de l'entreprise ;</li> <li>✓ Gère les informations liées aux employés.</li> </ul>  |
| Solde                                      | Niveau « entreprise » | Prend en charge le traitement des salaires des employés de son entreprise   |
| Employé                                    | Niveau « entreprise » | Consulte les informations le concernant uniquement (salaire, avantages, prêts, heures supplémentaires, etc.)  |

La figure 3.2 présente les différentes relations existant entre les différents acteurs de la plateforme. Dans ce diagramme, il est question essentiellement d'une relation d'héritage. Comme le montre la figure 3.2, les acteurs niveau « entreprise » peuvent être des employés de l'entreprise.



<sup>1</sup> Les acteurs de la plateforme sont tous des acteurs humains

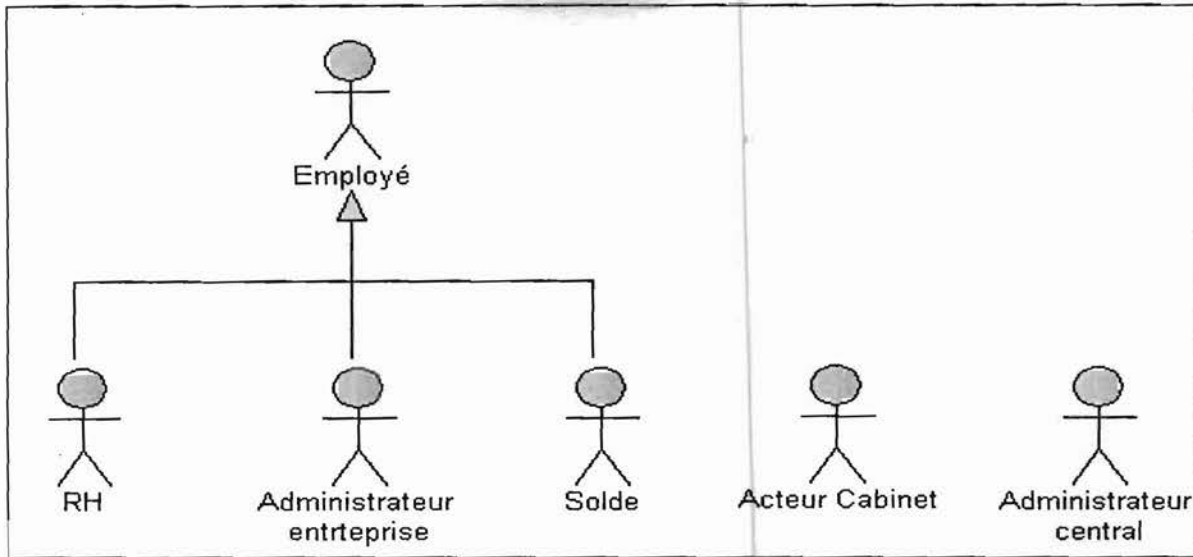


Figure 3.2 : relation entre les acteurs de la plateforme

### 3.1.3.2 Diagramme de contexte statique

Le diagramme de contexte statique est un diagramme dans lequel chaque acteur est relié au

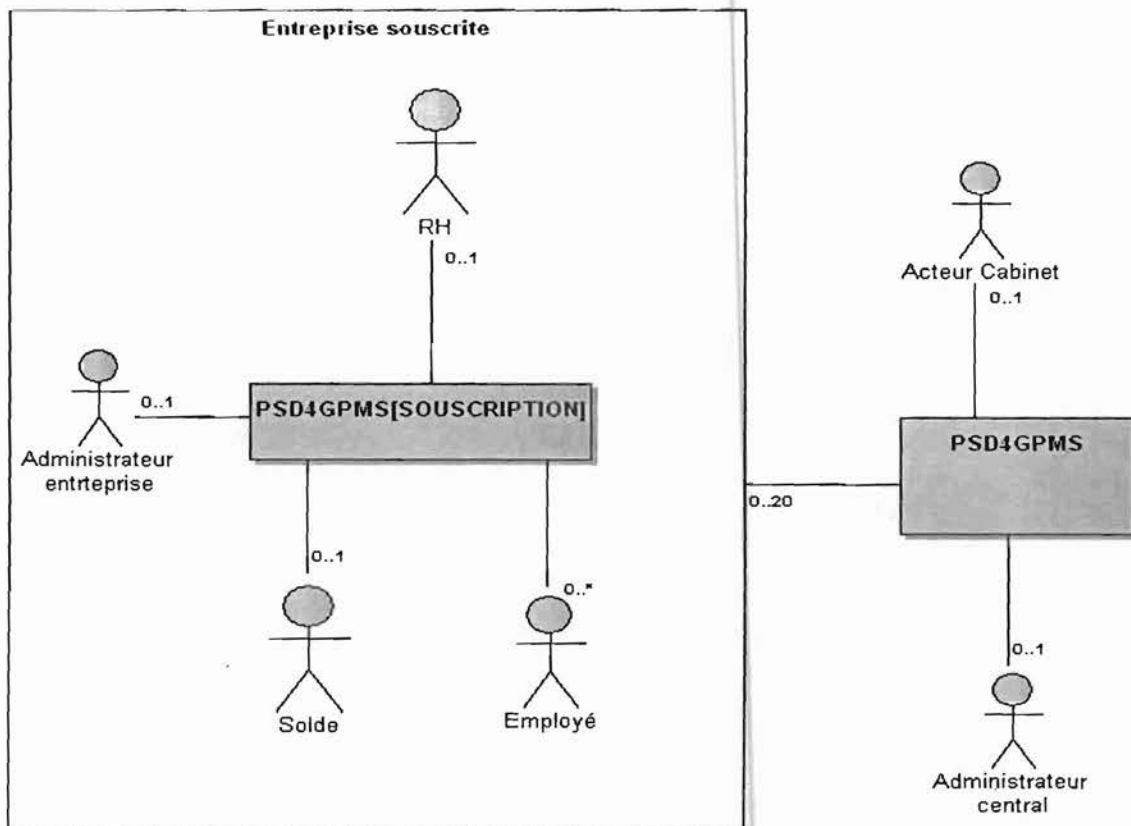


Figure 3.3 : diagramme de contexte statique

système<sup>1</sup> par une association à laquelle sont rattachées des cardinalités. Ce diagramme précise ainsi le nombre d'instances d'acteurs pouvant se connectés simultanément à notre système. Comme le montre la figure 3.3, il est prévu pour cette version de la plateforme, un nombre limite de vingt (20) abonnements.

### 3.2 CAPTURE DES BESOINS FONCTIONNELS

Cette étape vient compléter l'étude des besoins fonctionnels ébauchée dans l'étude préalable. La technique des cas d'utilisation est la pierre angulaire de celle-ci. Ces cas d'utilisation présentent les différentes façons qu'auront les acteurs d'utiliser la plateforme. Les concepts clefs utilisés dans cette partie sont les notions de cas d'utilisation et d'acteur.

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions réalisé par le système et produisant un résultat observable intéressant pour un acteur particulier. L'ensemble des cas d'utilisation doit décrire exhaustivement les exigences fonctionnelles du système. Chaque cas d'utilisation correspond donc à une fonction métier du système, selon le point de vue d'un acteur particulier. Un cas d'utilisation est représenté par une ellipse à l'intérieur de laquelle figure le nom du cas d'utilisation (figure 3.4). Il comporte obligatoirement un acteur principal et éventuellement des acteurs secondaires ;

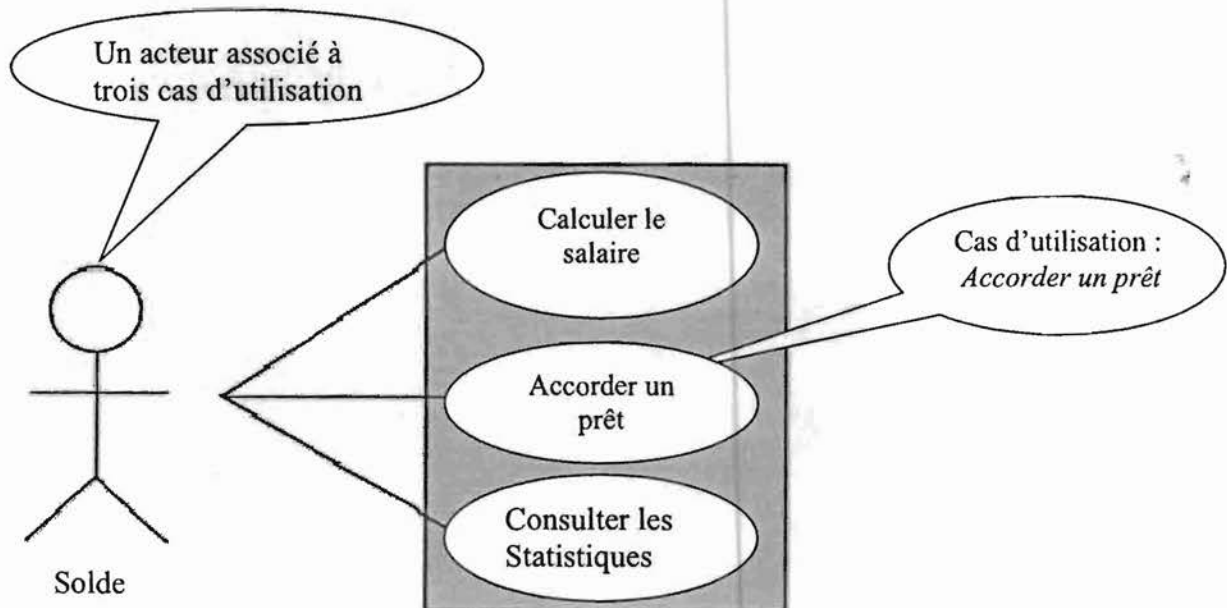


Figure 3.4 : un acteur associé à trois (03) cas d'utilisation

<sup>1</sup> Le système est vu ici comme une boîte noire

On appelle acteur principal celui pour qui le cas d'utilisation produit la plus-value métier. Par opposition, nous qualifions d'acteurs secondaires les autres participants du cas d'utilisation. Les acteurs secondaires sont typiquement sollicités à leur tour par le système pour obtenir des informations complémentaires.

Les acteurs principaux sont généralement placés à gauche et les acteurs secondaires à droite des cas d'utilisation. La figure 3.4 montre un acteur associé à trois (03) cas d'utilisation.

### 3.2.1 Identification des cas d'utilisation

A partir du cahier des charges préliminaires et/ou des diagrammes de contexte, on peut dégager les cas d'utilisation du système. Les tableaux 3.1 à 3.5 résument les cas d'utilisation préliminaires de la plateforme. Chaque tableau décrit les cas d'utilisation identifiés par acteur principal.

**Tableau 3.1 : cas d'utilisation de l'administrateur central**

| Acteur                        | Cas d'utilisation                                     | Résumé   |
|-------------------------------|---|--|
| <b>Administrateur central</b> | Maintenir les paramètres étatiques                    | Ce cas d'utilisation permet à l'administrateur central de maintenir dans le système les paramètres de l'Etat issus des conventions collectives et sectorielles et qui déterminent les modes de calcul des éléments de salaire.                   |
|                               | Souscrire une entreprise                              | Ce cas d'utilisation permet de: <ul style="list-style-type: none"> <li>✓ Souscrire une entreprise ;</li> <li>✓ Créer un compte administrateur associé à l'entreprise souscrite;</li> <li>✓ Gérer les contrats liés aux souscriptions.</li> </ul> |
|                               | Gérer les utilisateurs (niveau central ou entreprise) | Gérer les profils des administrateurs niveau « entreprise ». Créer, activer, suspendre ou désactiver des comptes niveau entreprise.  |

Tableau 3.2 : cas d'utilisation de l'acteur RH

| Acteur | Cas d'utilisation                                      | Résumé  |
|--------|--|---|
| RH     | Gérer les paramètres de l'entreprise                   | Ce cas d'utilisation permet de définir : <ul style="list-style-type: none"> <li>✓ Les éléments de salaire de l'entreprise ;</li> <li>✓ Les fonctions et les services qui forment l'entreprise ;</li> <li>✓ La grille salariale de l'entreprise ;</li> <li>✓ Les modes de paiement autorisés de l'entreprise ;</li> </ul>  |
|        | Enregistrer un employé                                 | Ce cas d'utilisation permet à l'acteur Solde de : <ul style="list-style-type: none"> <li>✓ Saisir les informations de l'employé qui vont permettre son identification (matricule, nom, date d'embauche, ...);</li> <li>✓ Déclarer les membres de famille de l'employé et indiquer ainsi le nombre de charge de l'employé ;</li> <li>✓ Attribuer une fonction/poste à l'employé ;</li> <li>✓ Classer l'employé sur la grille salariale.</li> </ul> |
|        | Gérer les prêts et avances sur salaires                | Ce cas d'utilisation permet le suivi des avances et prêts accordés aux employés par l'entreprise ou des sociétés tierces  |
|        | Gérer l'absentéisme                                    | Ce cas d'utilisation permet le suivi des absences des employés.   |
|        | Gérer les fins de contrat                              | Ce cas d'utilisation permet le suivi des cessations de travail (départ en retraite, licenciement, démission)  |
|        | Gérer les heures supplémentaires                       | Ce cas d'utilisation permet le suivi des heures supplémentaires des employés.   |
|        | Affecter un employé                                    | Ce cas d'utilisation permet le suivi des affectations des employés.   |
|        | Gérer les relations des employés avec les institutions | Ce cas d'utilisation permet la gestion des relations des employés avec les institutions.  |

**Tableau 3.3 : cas d'utilisation de l'acteur Solde**

| Acteur | Cas d'utilisation                             | Résumé  |
|--------|---|---|
| Solde  | Préparer la paie                              | <p>Permet à l'acteur Solde de préparer la paie périodique des employés par:</p> <ul style="list-style-type: none"> <li>✓ La reconduction automatique des éléments permanents ;</li> <li>✓ La saisie des éléments variables ;</li> <li>✓ Le paramétrage de la paie.</li> </ul> |
|        | Calculer les salaires                         | Permet à l'acteur Solde de déclencher le traitement batch du calcul des salaires suite à la préparation de la paie.   |
|        | Editer les bulletins de paie                  | Permet à l'acteur Solde de visionner à l'écran ou d'imprimer autant qu'il veut les bulletins de paie pour un traitement effectué.   |
|        | Editer les ordres de virement                 | Permet à l'acteur Solde de visionner à l'écran ou d'imprimer les ordres de virement pour les employés affiliés aux banques.   |
|        | Editer les états de retenues IUTS/TPA         | Permet à l'acteur Solde de visionner ou d'imprimer les états de retenue IUTS et TPA.  |
|        | Editer les états de versement des cotisations | Permet de visionner ou d'imprimer les états de retenue de cotisation (CNSS, etc.)   |
|        | Editer les états statistiques                 | Permet de visionner ou d'imprimer les différents états statistiques élaborés par le système.  |

**Tableau 3.4 : cas d'utilisation de l'acteur « Administrateur entreprise »**

| Acteur                    | Cas d'utilisation                            | Résumé   |
|---------------------------|--|--|
| Administrateur entreprise | Gérer les utilisateurs niveau « entreprise » | Gérer les profils des utilisateurs niveau « entreprise » |

**Tableau 3.5 : cas d'utilisation de l'acteur Employé**

| Acteur  | Cas d'utilisation | Résumé   |
|---------|-------------------|--|
| Employé | Consulter infos   | Permettre à l'employé de consulter les informations qui le concernent (éléments permanents, éléments variables, heures supplémentaires, prêts, etc.) |

### 3.2.2 Description des cas d'utilisation

L'ensemble des cas d'utilisation doit être décrit textuellement en vue de compléter les diagrammes de cas d'utilisation et faciliter leur compréhension au client. Il n'y a pas de syntaxe normalisée pour la description des cas d'utilisation. Pour notre part, nous retenons la syntaxe suivante :

- ✓ **Le sommaire d'identification** qui est obligatoire et qui inclut le titre, le but et le résumé du cas d'utilisation, la date de la description de celui ci, la version, le responsable et les acteurs de l'équipe de réalisation ;
- ✓ **La description des enchaînements (obligatoire)** qui décrit le scénario nominal, les scénarii alternatifs, les scénarii d'exception mais aussi les pré conditions et les post conditions. Un scénario représente une succession particulière d'enchaînements, qui s'exécute du début à la fin du cas d'utilisation ;
- ✓ **Les contraintes non fonctionnelles** qui ajoutent éventuellement les informations opérationnelles (fréquence d'utilisation de la fonctionnalité décrite par le cas d'utilisation, volumétrie des données engendrée par celui-ci, disponibilité de la fonctionnalité du système décrite par le cas d'utilisation, fiabilité, intégrité, confidentialité, performances, concurrence, etc.) ;

Etant donné la longue liste des cas d'utilisation de la plateforme et pour des raisons de clarté, nous avons décidé de ne présenter que la description de quatre (04) cas d'utilisation (tableaux 3.6 à 3.9)

**Tableau 3.6** : description du cas d'utilisation « Souscrire une entreprise »

| <b>Cas d'utilisation : Souscrire une entreprise</b>  |
|--|
| <p><b>But :</b> Souscrire une entreprise à la plateforme afin qu'elle bénéficie des services offerts pour le traitement de la paie de ses employés.</p> <p><b>Résumé :</b> Saisir partiellement les informations de l'entreprise dans le système, créer un compte administrateur niveau « entreprise » pour l'entreprise partiellement enregistrée.</p> <p><b>Acteurs :</b> Administrateur central (principal)</p> <p><b>Date de création :</b> 05/09/08 <b>Date de mise à jour :</b> 05/09/08 <b>Version :</b> 1.0</p> <p><b>Responsable :</b> Firmin T. KABORE</p> <p><b>Préconditions</b></p> <ol style="list-style-type: none"> <li>1. L'administrateur central est authentifié</li> </ol> <p><b>Scénario nominal</b><br/>Ce scénario commence lorsque l'administrateur central désire souscrire une entreprise à la plateforme.</p> <p><b>Enchaînement a : Enregistrer l'entreprise</b><br/>L'administrateur central entre les informations de l'entreprise dans le système (Sigle, immatriculation, type entreprise, etc.).</p> <p><b>Enchaînement b : Créer un compte « administrateur entreprise »</b><br/>L'administrateur central crée un compte « administrateur entreprise » et l'associe à l'entreprise souscrite</p> <p><b>Enchaînement c : Indiquer la période de validité de la souscription</b><br/>L'administrateur central indique également la période pendant laquelle la souscription est valide. Cependant une souscription peut être renouvelée.</p> <p><b>Enchaînement d: Valider la souscription</b><br/>L'administrateur central valide la souscription de l'entreprise et le système enregistre la souscription.</p> <p><b>Postconditions :</b></p> <ol style="list-style-type: none"> <li>1. La souscription de l'entreprise est enregistrée dans le système.</li> <li>2. L'entreprise est en attente de paramétrage.</li> </ol> <p><b>Contraintes non fonctionnelles</b></p> <p><b>Temps de réponse :</b> L'interface de l'administrateur doit réagir dans l'espace de deux secondes</p> <p><b>Disponibilité :</b> Cette fonctionnalité doit être permanemment disponible.</p> |



Tableau 3.7 : description du cas d'utilisation « Paramétrer l'entreprise »

|   |
|---|
| <p><b>Cas d'utilisation : Paramétrer l'entreprise</b></p> <p><b>But :</b> Maintenir les paramètres de l'entreprise dans le système</p> <p><b>Résumé :</b> Saisir les informations de l'entreprise dans le système, définir les services, les fonctions, les éléments de salaire, etc.</p> <p><b>Acteurs :</b> RH (principal)</p> <p><b>Date de création :</b> 05/09/08 <b>Date de mise à jour :</b> 05/09/08 <b>Version :</b> 1.0</p> <p><b>Responsable :</b> Firmin T. KABORE</p> <p><b>Préconditions</b></p> <ol style="list-style-type: none"> <li>1. L'entreprise est déjà souscrite par l'administrateur central ;</li> <li>2. Le RH est authentifié.</li> </ol> <p><b>Scénario nominal</b></p> <p>Ce cas d'utilisation commence lorsque l'acteur RH désire utiliser le système pour paramétrer son entreprise.</p> <p><b>Enchaînement a : Confirmer la souscription de l'entreprise</b></p> <p>L'acteur RH termine la création de l'entreprise ébauchée par l'administrateur central (lors de la souscription de l'entreprise) en apportant des informations supplémentaires et confirme ainsi la souscription de son entreprise à la plateforme.</p> <p><b>Enchaînement b : Définir les services et les postes</b></p> <p>L'acteur RH définit les différents services et les postes de travail qui composent son entreprise. Il définit également la hiérarchie entre les différents services.</p> <p><b>Enchaînement c : Définir les éléments de salaire</b></p> <p>Le RH définit les éléments de salaire de son entreprise. Pour chaque élément de salaire, il fournit son code, son libellé et le mode de calcul qui lui est associé.</p> <p><b>Enchaînement d : Définir les modes de virement</b></p> <p>L'acteur RH définit les différents modes de virement acceptés par l'entreprise. Si l'entreprise accepte le mode de virement bancaire, alors l'acteur RH doit indiquer les comptes bancaires de l'entreprise.</p> <p><b>Enchaînement e : Indiquer la base de salaire acceptée de l'entreprise.</b></p> <p>L'acteur RH devrait pouvoir indiquer la base de salaire acceptée par l'entreprise. Mais dans cette première version de notre plateforme, la base de salaire est fixée au mois.</p> <p><b>Enchaînement f : Configurer la grille salariale de l'entreprise</b></p> <p>Si l'entreprise dispose d'une grille salariale, l'acteur RH peut également la configurer en définissant les avantages permanents pour chaque partition de la grille.</p> <p><b>Postconditions</b></p> <ol style="list-style-type: none"> <li>1. L'entreprise est paramétrée.</li> <li>2. Les autres utilisateurs pourront maintenant utiliser la plateforme.</li> </ol> <p><b>Contraintes non fonctionnelles</b></p> <p><b>Temps de réponse :</b> L'interface du RH doit réagir dans l'espace de deux secondes</p> <p><b>Disponibilité :</b> Cette fonctionnalité doit être permanemment disponible.</p> |
|---|

**Tableau 3.8** : description du cas d'utilisation « Enregistrer un employé »

|  |
|--|
| <p><b>Cas d'utilisation : Enregistrer un employé</b></p> <p><b>But :</b> Maintenir dans le système les informations sur les employés travaillant dans l'entreprise.</p> <p><b>Résumé :</b> Enregistrer les informations des nouveaux employés dans le système, mettre à jour les informations des employés déjà enregistrées.</p> <p><b>Acteurs :</b> RH (principal)</p> <p><b>Date de création :</b> 05/09/08 <b>Date de mise à jour :</b> 05/09/08 <b>Version :</b> 1.0</p> <p><b>Responsable :</b> Firmin T. KABORE</p> <p><b>Préconditions</b></p> <ol style="list-style-type: none"> <li>1. Le RH est authentifié</li> <li>2. L'entreprise est paramétrée</li> </ol> <p><b>Scénario nominal</b></p> <p>Ce cas d'utilisation commence lorsque l'acteur RH désire enregistrer ou mettre à jour les informations d'un employé dans le système.</p> <p><b>Enchaînement a : Ajouter un nouvel employé</b></p> <p>L'acteur RH fournit les informations nécessaires (numéro matricule, nationalité, type employé, date d'embauche, etc.) pour l'enregistrement de l'employé dans le système.</p> <p><b>Enchaînement b : Classer sur la grille salariale</b></p> <p>Le RH peut au cas où l'entreprise dispose d'une grille salariale, classer l'employé sur celle-ci ou lui accorder des avantages permanents (salaire de base, indemnités, etc.)</p> <p><b>Enchaînement c : Déclarer les membres de famille de l'employé</b></p> <p>Le RH peut éventuellement déclarer les membres de famille de l'employé et indiquer ainsi le nombre de charges de l'employé.</p> <p><b>Enchaînement d : Attribuer une fonction/poste à l'employé</b></p> <p>Le RH doit aussi indiquer le service de l'employé et lui affecter un poste. Si le poste affecté possède des avantages permanents, ces avantages sont automatiquement acquis par l'employé.</p> <p><b>Enchaînement e : valider l'ajout de l'employé</b></p> <p>Le RH valide l'ajout de l'employé et le système enregistre toutes les informations de l'employé fournies</p> <p><b>Scénario alternatif</b></p> <p><b>Enchaînement f : Mettre à jour les infos d'un employé</b></p> <p>Le RH peut modifier toutes les informations validées auparavant quand il le désire.</p> <p><b>Postconditions :</b></p> <ol style="list-style-type: none"> <li>1. Les informations de l'employé sont disponibles dans le système.</li> </ol> <p><b>Contraintes non fonctionnelles</b></p> <p><b>Temps de réponse :</b> L'interface du RH doit réagir dans l'espace de deux secondes</p> <p><b>Concurrence :</b> La validation de l'ajout de l'employé doit être notifiée chez les utilisateurs RH et Solde connectés.</p> <p><b>Disponibilité :</b> Cette fonctionnalité doit être permanentement disponible</p> |
|--|

Tableau 3.9 : description du cas d'utilisation « Maintenir les paramètres étatiques »

| Cas d'utilisation : Maintenir les paramètres étatiques  |
|---|
| <p><b>But :</b> Maintenir à jour les paramètres étatiques nécessaires au traitement de la paie.</p> <p><b>Résumé :</b> Enregistrer et mettre à jour les différents paramètres étatiques qui interviennent dans le processus de traitement de la paie.</p> <p><b>Acteurs :</b> Administrateur central (principal) <b>Responsable :</b> Firmin T. KABORE</p> <p><b>Date de création :</b> 05/09/08 <b>Date de mise à jour :</b> 05/09/08 <b>Version :</b> 1.0</p> <p><b>Précondition :</b> L'administrateur central est authentifié.</p> <p><b>Scénario nominal</b><br/>Ce cas d'utilisation commence lorsque l'administrateur central désire enregistrer ou mettre à jour un paramètre étatique dans le système.</p> <p><b>Enchaînement a : Editer les paramètres CNSS</b><br/>L'administrateur central renseigne le plafond de la base CNSS, le taux CNSS de la cotisation « employé » et le taux CNSS de la cotisation « Employeur ».</p> <p><b>Enchaînement b : Editer les paramètres IUTS</b><br/>L'administrateur central édite les différents taux IUTS et les montants correspondants.</p> <p><b>Enchaînement c : Editer les paramètres TPA</b><br/>L'administrateur central édite le taux TPA pour les nationaux et le taux TPA pour les étrangers.</p> <p><b>Enchaînement d : Editer les paramètres « travail »</b><br/>L'administrateur central édite les différents paramètres de travail (nombre de jours ouvrables, nombre de jours de repos, taux horaire, etc.).</p> <p><b>Enchaînement e : Editer les paramètres sur les abattements de charges</b><br/>L'administrateur central édite la borne maximale des abattements charges, le nombre de charges et les taux correspondants.</p> <p><b>Enchaînement f : Editer les paramètres sur les abattements d'IUTS</b><br/>L'administrateur central édite les différents plafonds sur les indemnités (plafond fonction, plafond logement, plafond transport) et les taux d'abattements sur ces indemnités. Il renseigne aussi le taux d'abattement sur le salaire de base.</p> <p><b>Enchaînement g : Editer les paramètres d'ancienneté</b><br/>L'administrateur central édite les paramètres sur l'ancienneté.</p> <p><b>Enchaînement h : Editer les paramètres sur les heures supplémentaires</b><br/>L'administrateur central définit les différents types d'heure supplémentaire et les taux de majoration correspondants.</p> <p><b>Enchaînement i : Enregistrer les paramètres étatiques</b><br/>L'administrateur central valide l'enregistrement des différents paramètres et le système enregistre les paramètres étatiques.</p> <p><b>Postcondition :</b> Les paramètres étatiques sont à jour dans le système</p> <p><b>Contraintes non fonctionnelles</b><br/><b>Temps de réponse :</b> L'interface de l'acteur Solde doit réagir dans l'espace de deux secondes<br/><b>Disponibilité :</b> Cette fonctionnalité doit être disponible à l'initialisation de la plateforme.</p> |



La description textuelle peut être complétée par un ou plusieurs diagrammes dynamiques tels que les diagrammes d'activité ou de séquence système. Cependant, pour des raisons de volume du document, nous allons nous en tenir à la description textuelle.

### 3.2.3 Organisation des cas d'utilisation

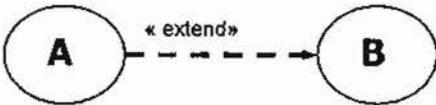
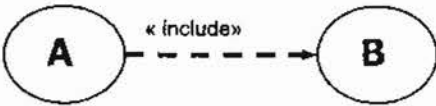

On peut maintenant procéder à l'organisation des cas d'utilisation. Cela peut se faire de deux façons différentes et complémentaires :

- ✓ Le rajout des relations d'inclusion, d'extension et de généralisation ;
- ✓ Le regroupement des cas d'utilisation en paquetages pour former des blocs fonctionnels homogènes de haut niveau.

#### 3.2.3.1 Rajout des relations d'inclusion, d'extension et de généralisation

Il existe essentiellement trois (03) types de relations standardisées entre cas d'utilisation comme le montre le tableau 3.10.

Tableau 3.10 : type de relation en cas d'utilisation

| Relation       | Notation  | Explication  |
|----------------|---|--|
| Extension      |  | <b>A étend B (extend) :</b> la réalisation de B peut s'étendre sous certaines conditions spécifiques à la réalisation de A |
| Inclusion      |  | <b>A inclut B (include) :</b> la réalisation de A contient (nécessite) la réalisation de B                                 |
| Généralisation |  | <b>Une généralisation de A vers B :</b> A est une spécialisation de B  |

La figure 3.5 montre les cas d'utilisation de la plateforme organisés suivant les relations d'extension, d'inclusion et de généralisation.

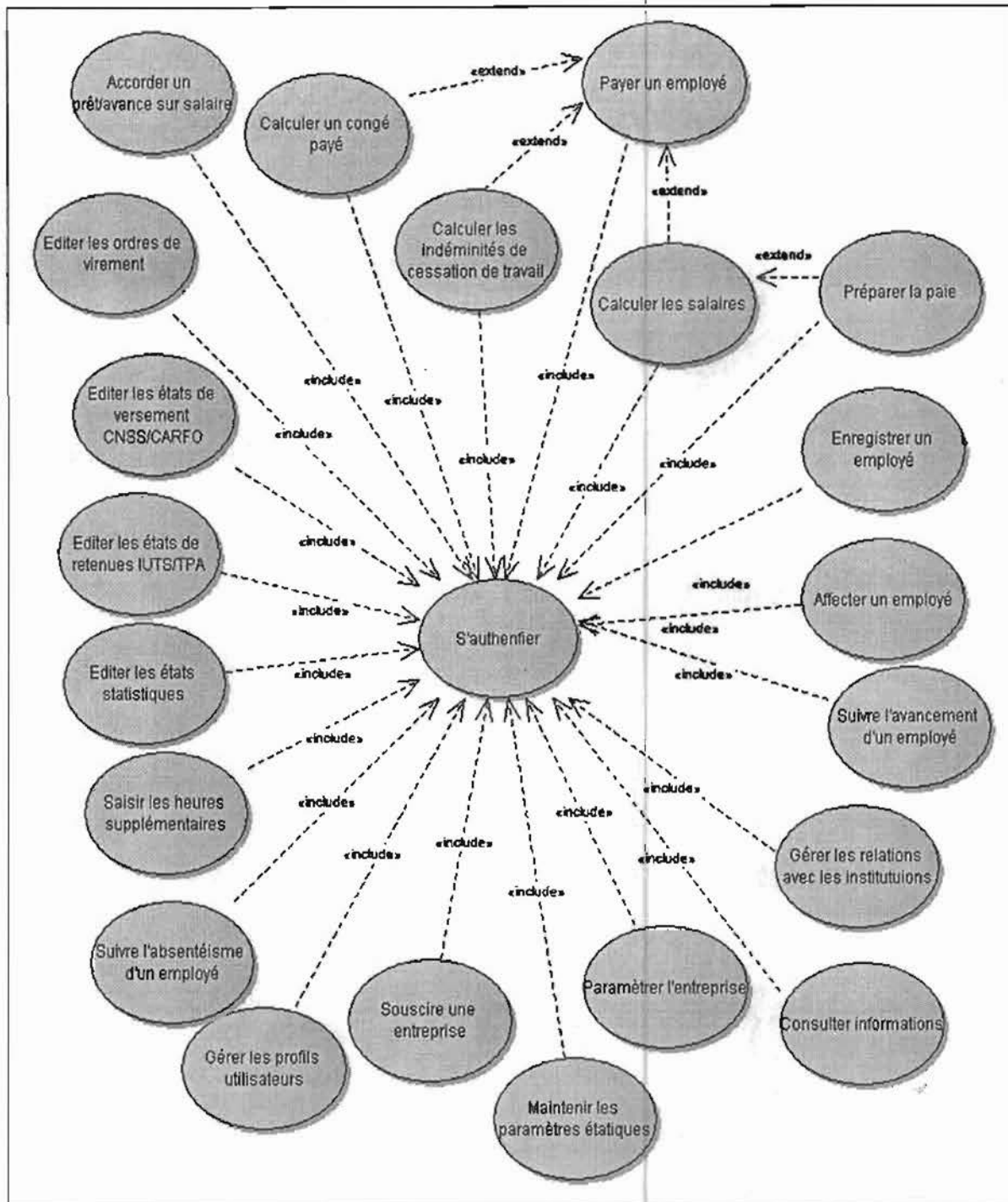


Figure 3.5 : cas d'utilisation organisés suivant les relations d'inclusion, d'extension et de généralisation

### 3.2.3.2 Organisation en blocs fonctionnels

Pour définir la stratégie de regroupement des cas d'utilisation pour un projet, il convient de recourir à la liste suivante de critères :

- ✓ **par domaine d'expertise métier** qui est plus intuitif et souvent plus efficace. Il facilite la spécialisation des analystes et permet d'organiser la disponibilité des différents experts ;
- ✓ **par acteur** qui est simple à mettre en œuvre uniquement si chaque cas d'utilisation est relié à un et un seul acteur, sinon il s'apparente souvent au critère précédent ;
- ✓ **par lot de livraison** qui dans le cadre d'un développement itératif et incrémental, permet de regrouper dans un même paquetage les cas d'utilisation qui seront livrés ensemble au client.

Le mécanisme générique de regroupement d'éléments en UML est le paquetage. Un paquetage UML représente un espace de nommage qui peut contenir :

- ✓ des éléments d'un modèle ;
- ✓ des diagrammes représentant les éléments du modèle ;
- ✓ d'autres paquetages.

Le critère retenu pour le regroupement de nos cas d'utilisation en blocs fonctionnels homogènes est le premier critère c'est-à-dire par domaine d'expertise métier. Le tableau 3.7 donne les paquetages en blocs fonctionnels homogènes de la plateforme PSD4GPMS.

**Tableau 3.11** : organisation des cas d'utilisation en paquetages

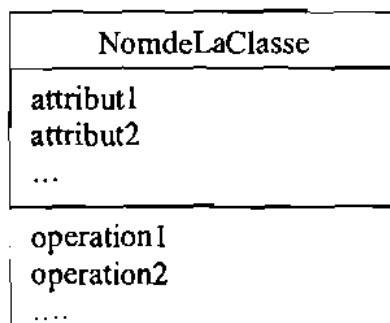
| <b>Paquetage</b>                      | <b>Cas d'utilisation</b>                               |
|---------------------------------------|--|
| <b>Traitement de la paie</b>          | Préparer la paie                                       |
|                                       | Calculer les salaires                                  |
|                                       | Editer les bulletins de paie                           |
|                                       | Editer les ordres de virement                          |
|                                       | Editer les états de retenue IUTS/TPA                   |
|                                       | Editer les états de cotisation                         |
|                                       | Editer les états statistiques                          |
| <b>Situation Employé</b>              | Enregistrer un employé                                 |
|                                       | Gérer les prêts et avances sur salaire                 |
|                                       | Gérer l'absentéisme                                    |
|                                       | Gérer les fins de contrat                              |
|                                       | Gérer les heures supplémentaires                       |
|                                       | Affecter un employé                                    |
|                                       | Consulter infos  |
| <b>Gestion des profils</b>            | Gérer les utilisateurs                                 |
| <b>Relation avec les institutions</b> | Gérer les relations des employés avec les institutions |
| <b>Gestion des paramètres</b>         | Gérer les paramètres Entreprise                        |
|                                       | Maintenir les paramètres étatiques                     |
| <b>Souscription</b>                   | Souscrire une entreprise                               |

### 3.2.4 Identification des classes candidates

La capture des besoins fonctionnels se termine par l'identification des classes candidates. Cette partie entame l'analyse proprement dite. Pour chacun des cas d'utilisation listés dans le

tableau 3.7 (deuxième colonne), on identifie les classes candidates (objets métiers réifierables en objets). Une classe candidate est une classe qui peut être retenue ou rejetée après une analyse approfondie dans les phases ultérieures.

En UML, une classe est une entité métier encapsulant un état et des méthodes. Elle est représentée par un rectangle avec trois compartiments (figure 3.6):

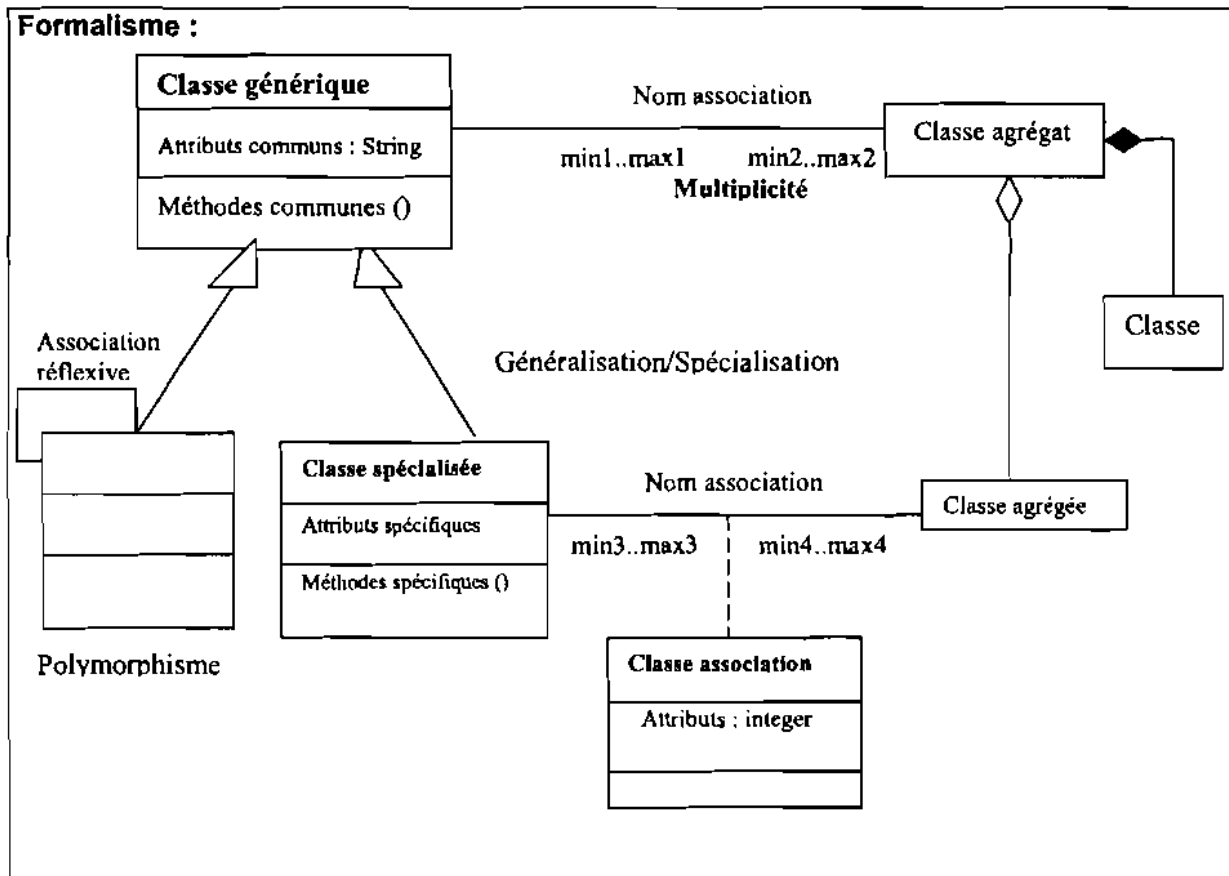


**Figure 3.6 :** notation simplifiée d'une classe en UML

Le diagramme de classes met en œuvre des classes reliées entre elles par des associations ou des généralisations. C'est le point central dans tout développement orienté objet. En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs. Le tableau 3.12 qui suit, détaille tout le formalisme et les concepts utilisés dans les diagrammes de classes UML.



Tableau 3.12 : formalisme du diagramme de classes UML



**Principaux concepts**

| Concept                        | Définition  |
|--------------------------------|---|
| Nom classe                     | Identifiant de la classe.   |
| Attribut                       | Information élémentaire composant une classe. Un attribut peut permettre d'identifier la classe.  |
| Opération                      | Fonctionnalité assurée par la classe.   |
| Association                    | Relation entre classes.   |
| Association réflexive          | Association mettant en relation une classe avec elle-même.  |
| Classe association             | Association porteuse d'attribut.  |
| Multiplicité                   | Nombre d'instances impliquées dans l'association.   |
| Agrégation                     | Type d'association mettant en évidence une classe agrégée et une classe agrégat. Chaque objet de la classe agrégat est associé à un ou plusieurs objets de la classe agrégée. |
| Généralisation/ Spécialisation | Permet d'identifier parmi les objets d'une classe (générique) des sous-ensembles d'objets (des classes spécialisées) ayant des caractéristiques spécifiques.                  |
| Polymorphisme                  | Possibilité pour un même message de déclencher des traitements différents, suivant les objets spécialisés auxquels il est adressé.  |

A cette étape, les classes sont représentées uniquement par le compartiment comportant le nom de la classe. La figure 3.7 présente le diagramme de classes participantes du cas d'utilisation « Préparer la paie ». Il s'agit là de simples diagrammes de classes qui seront développés dans les étapes ultérieures. On identifie les concepts métier qui peuvent être réifiés comme classe sans rentrer en profondeur avec les associations et les cardinalités.

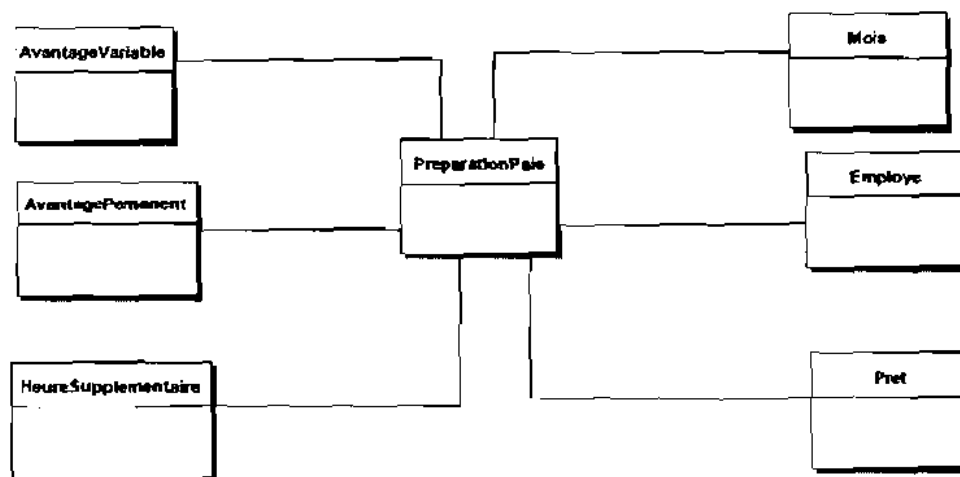


Figure 3.7 : diagramme de classes participantes du cas d'utilisation « Préparer la paie »

### 3.3 CAPTURE DES BESOINS TECHNIQUES

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative. Elle a lieu lorsque les architectes ont obtenu suffisamment d'informations sur les prérequis techniques. Ils doivent à priori connaître au moins le matériel à savoir les machines et réseaux, les progiciels à intégrer et les outils retenus pour le développement. Cependant, cette activité est généralement peu formalisée, soit par manque de notation et de processus approprié, soit parce que les architectes recourent rarement à une approche formelle. De ce fait, nous avons passé cette activité en revue tout au long des étapes suivantes.

# 4

## Développement des modèles

Cette section se focalise sur le développement des modèles statiques et des modèles dynamiques.

### 4.1 DECOUPAGE EN CATEGORIES ET DEVELOPPEMENT DU MODELE STATIQUE

La capture des besoins fonctionnels a conduit à l'identification des classes candidates. A présent, il faut regrouper ces classes candidates en catégories afin de les développer. Une catégorie<sup>1</sup> est un ensemble de classes fortement liées par des associations, des agrégations et des compositions. C'est donc un ensemble de classes à forte cohésion interne et à faible couplage externe. Les classes candidates identifiées peuvent être regroupées suivant les catégories commentées dans le tableau 4.1.

<sup>1</sup> La notion de catégorie correspond exactement à la notion de paquetage UML.

**Tableau 4.1** : regroupement des classes en catégories

| Catégorie                           | Commentaire  |
|-------------------------------------|--|
| Adresse                             | Cette catégorie s'occupe de l'adressage des éléments « <i>adressables</i> » de la plateforme (Employés, entreprises, services, etc.) |
| Eléments Employés                   | Modélise les différents éléments de salaire qui forment le salaire de l'employé.   |
| Collaboration avec les institutions | Modélise la collaboration des employés avec les entreprises de la place (station, assurance, pharmacie)                              |
| Direction                           | Cette catégorie n'est pas indispensable à la réalisation de la plateforme. Elle est juste là pour des raisons d'extension.           |
| Paie                                | Cette catégorie modélise le processus de traitement des salaires.  |
| Infos SocioProfessionnelles Employé | Modélise les informations socioprofessionnelles de l'employé.  |
| Paramètres Entreprise               | Modélise les paramètres de l'entreprise (éléments de salaire, mode de virement, fonctions, etc.)                                     |
| Paramètres Etatiques                | Modélise les paramètres de l'Etat à prendre en compte dans le traitement des salaires.   |
| Profils                             | Gère les profils des utilisateurs et les contrôles d'accès à la plateforme.  |
| Situation Employé                   | Modélise la situation de l'employé (ses prêts en cours, ses heures supplémentaires, ses absences)                                    |
| Souscription                        | Gère les abonnements des entreprises tierces à la plateforme.  |

Chaque catégorie identifiée nécessite la construction d'un diagramme de classes détaillées. Pour des raisons de clarté et de volume du mémoire, nous allons présenter seulement les diagrammes de classes de huit (08) catégories (figure 4.1 à 4.8).

La figure 4.1 montre le diagramme de classes détaillées de la catégorie « Souscription ». La catégorie « Souscription » gère les abonnements des entreprises tierces à la plateforme. Le comité de pilotage du projet prévoit pour la plateforme plusieurs types d'abonnement. Une entreprise souscrit un type d'abonnement pour une durée déterminée.

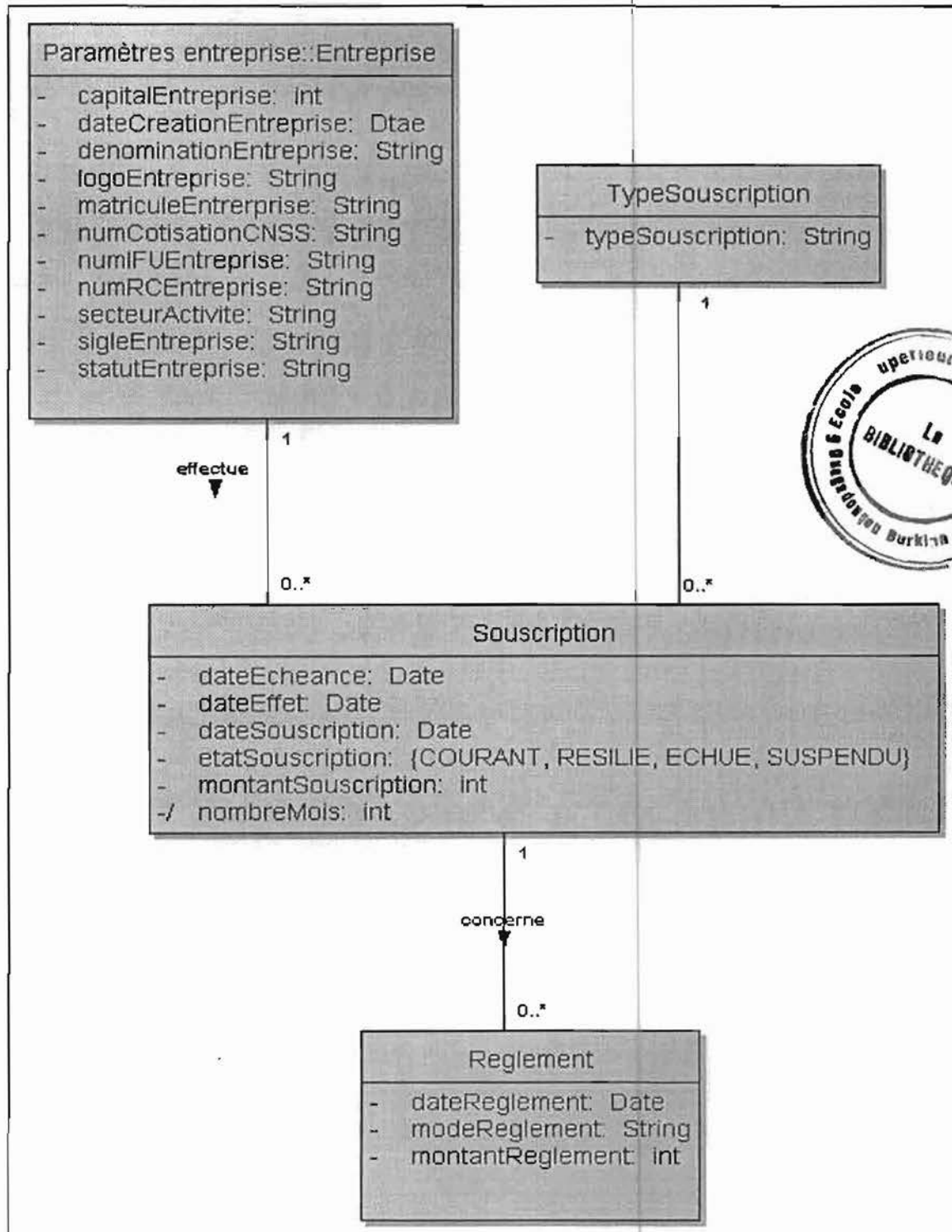


Figure 4.1 : diagramme de classe de la catégorie « Souscription »

La catégorie « Paramètres Etatiques » (figure 4.2) modélise les paramètres de l'Etat (Burkina Faso) issus des conventions collectives et sectorielles entrant dans le cadre du traitement des salaires. Pour cette version de la plateforme, l'étude est circonscrite au Burkina Faso, l'extension aux autres pays viendra dans les versions ultérieures.

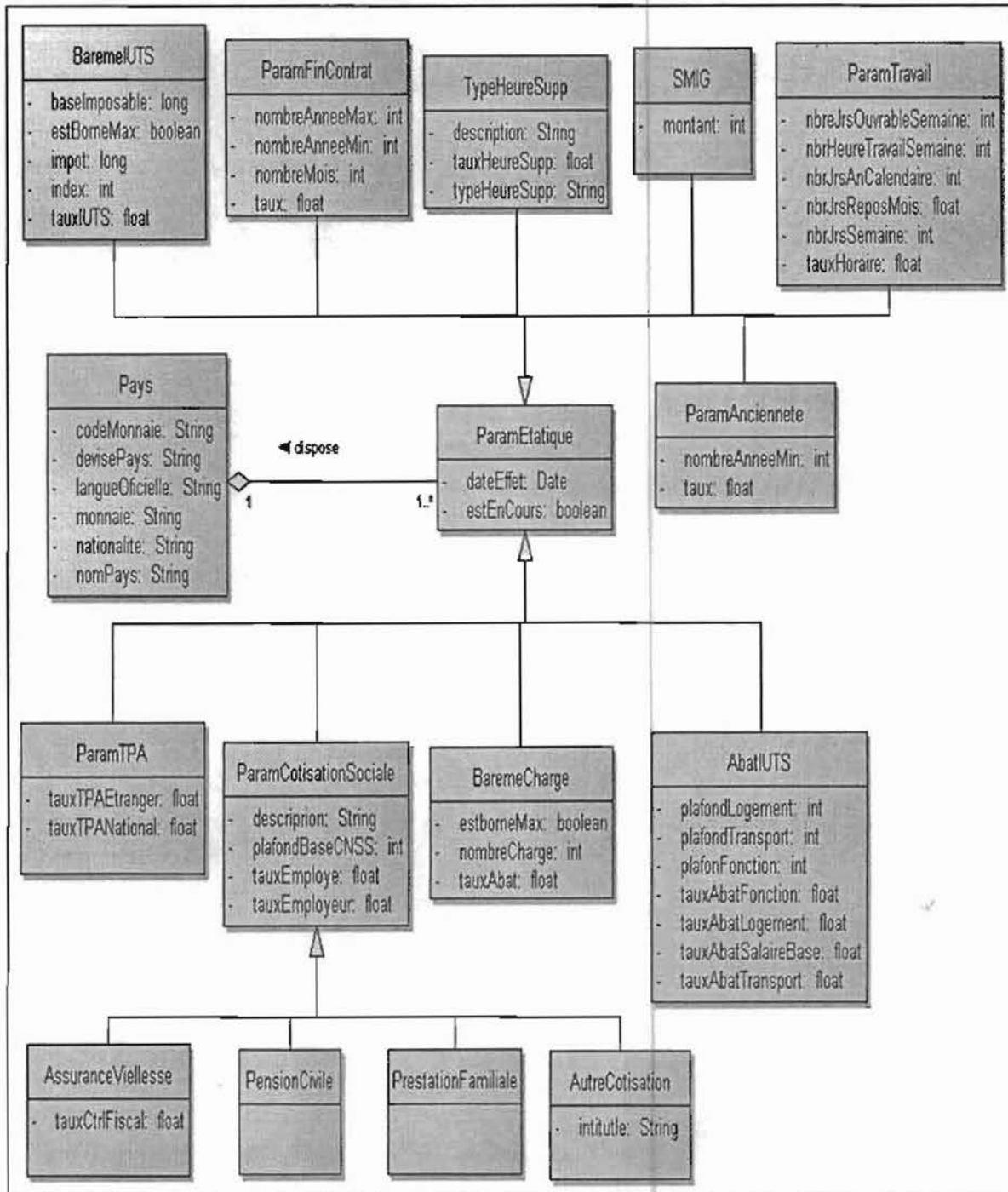


Figure 4.2 : diagramme de classes de la catégorie « Paramètres Etatiques »

La figure 4.3 est le diagramme de classes de la catégorie « Paramètres Entreprises ». Cette catégorie permet d'associer un contexte au sein de la plateforme à chaque entreprise abonnée, en définissant les paramètres qui sont propres à chaque entreprise tels que les éléments de salaire, la grille salariale, les modes de virements, etc.

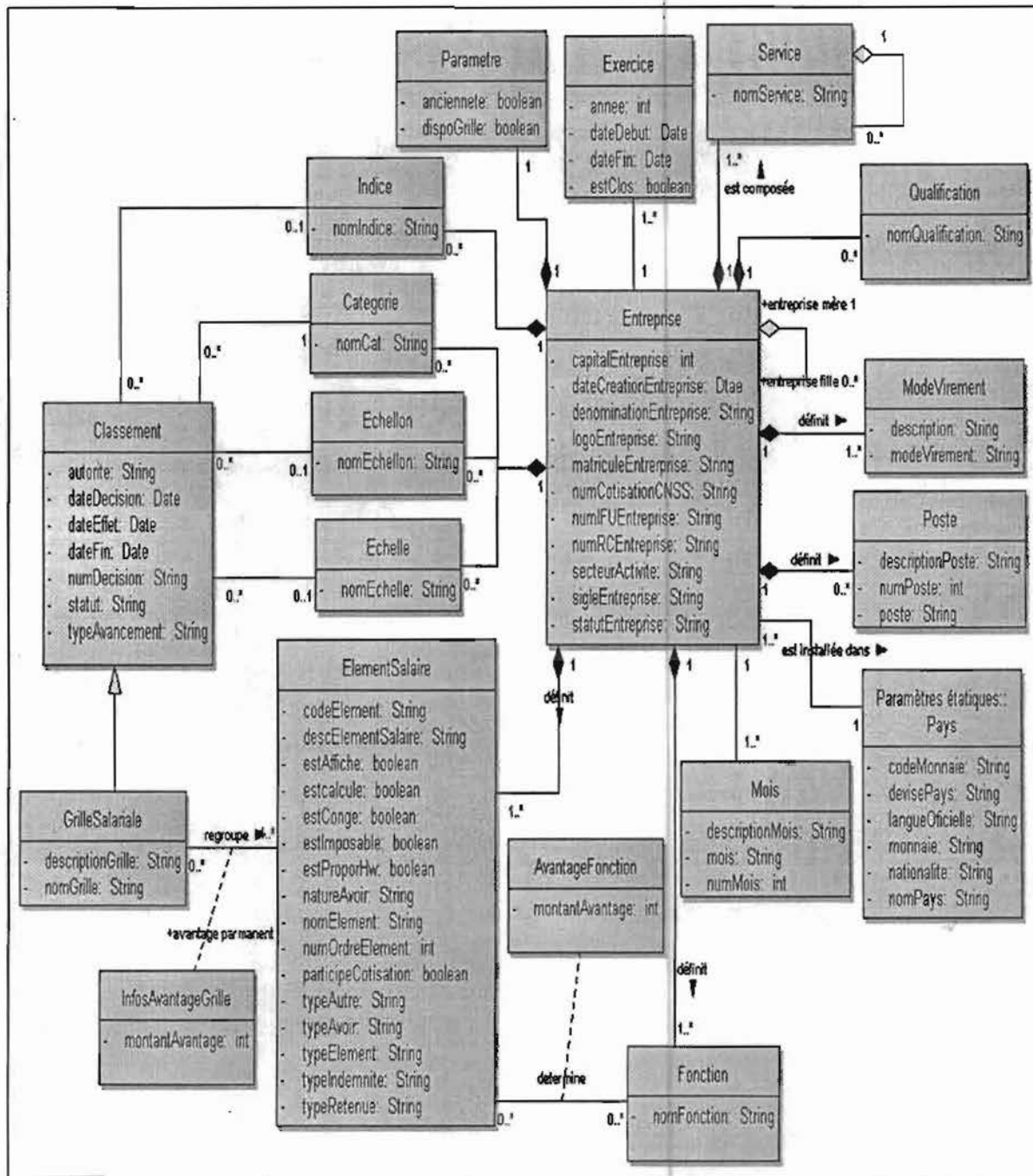


Figure 4.3 : diagramme de classes de la catégorie « Paramètres Entreprises »

La catégorie « InfosSocioProfessionnels » dont le diagramme est présenté dans la figure 4.3, modélise les informations socioprofessionnelles des employés intéressantes pour le traitement des salaires.

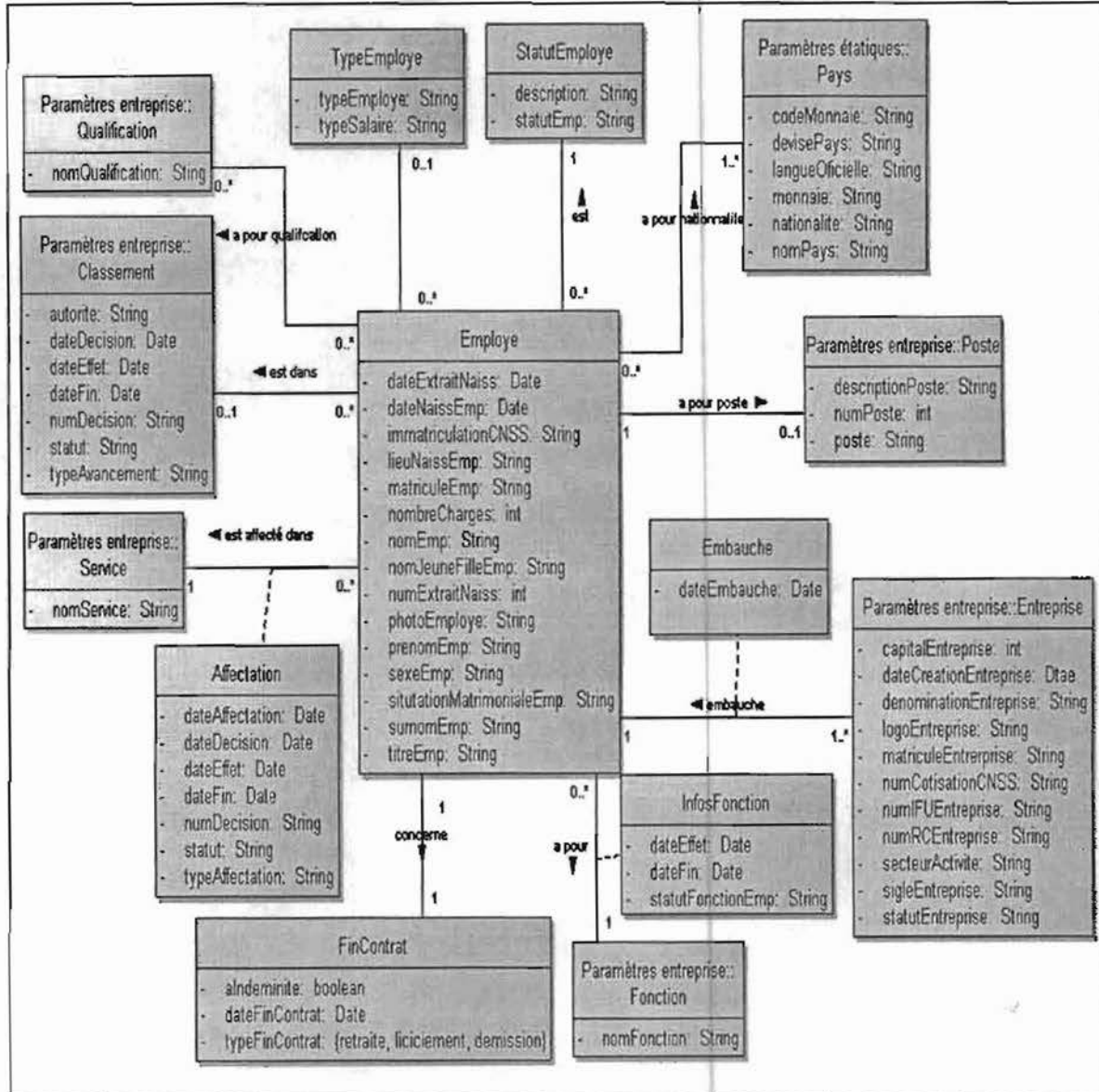


Figure 4.4 : diagramme de classes de la catégorie « InfosSocioProfessionnels Employé »

La catégorie « Eléments de salaire Employé » (figure 4.5) regroupe les éléments de salaire (avantages et retenues) qui forment le salaire de l'employé. Elle prend en compte les heures supplémentaires, les éléments variables comme les primes d'encouragement, les prêts et avances sur salaire, etc.



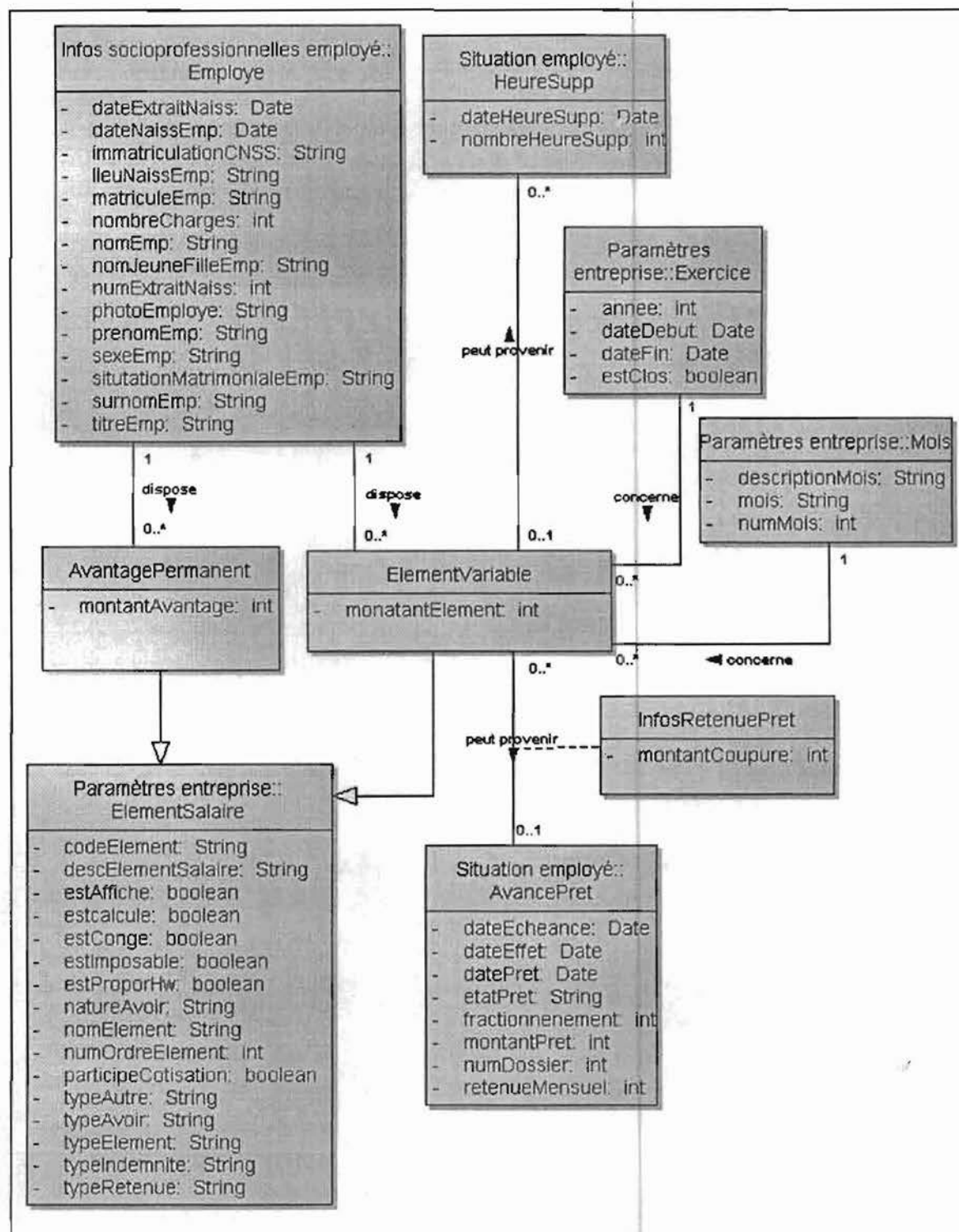


Figure 4.5 : diagramme de classes de la catégorie « Eléments de salaire Employé »

Le diagramme de classes de la catégorie « Situation Employé » (figure 4.6) donne le contexte salarial de l'employé dans l'entreprise. Cette catégorie permet de se situer sur les heures supplémentaires effectuées par l'employé, les prêts qui lui sont accordés et qui sont en cours, ses absences, etc.

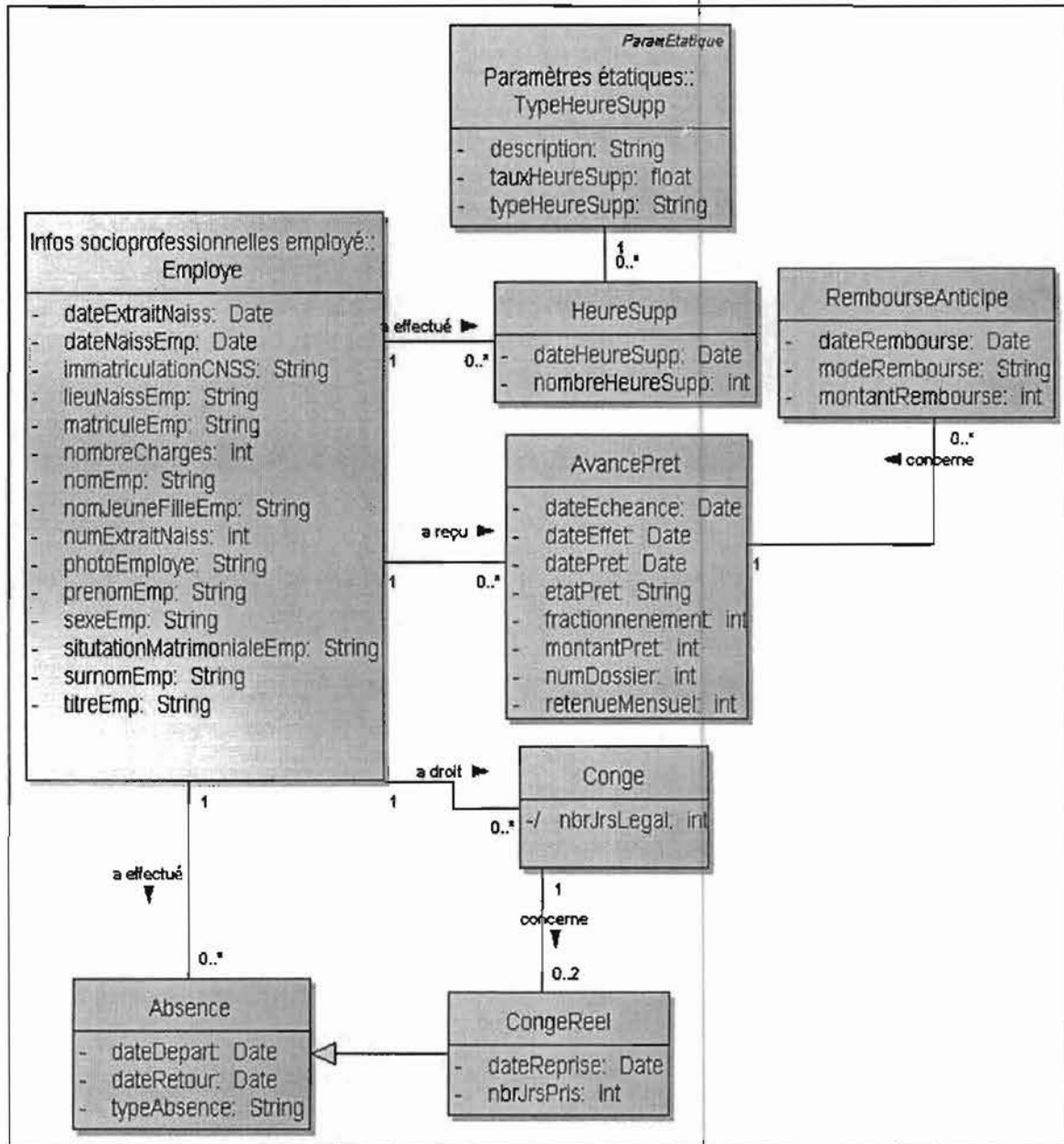


Figure 4.6 : diagramme de classes de la catégorie « Situation Employé »

Le diagramme de classes de la catégorie « Paie » (figure 4.7) est sans doute le plus important. La catégorie « Paie » permet le calcul des salaires des employés et la formation des bulletins de paie.

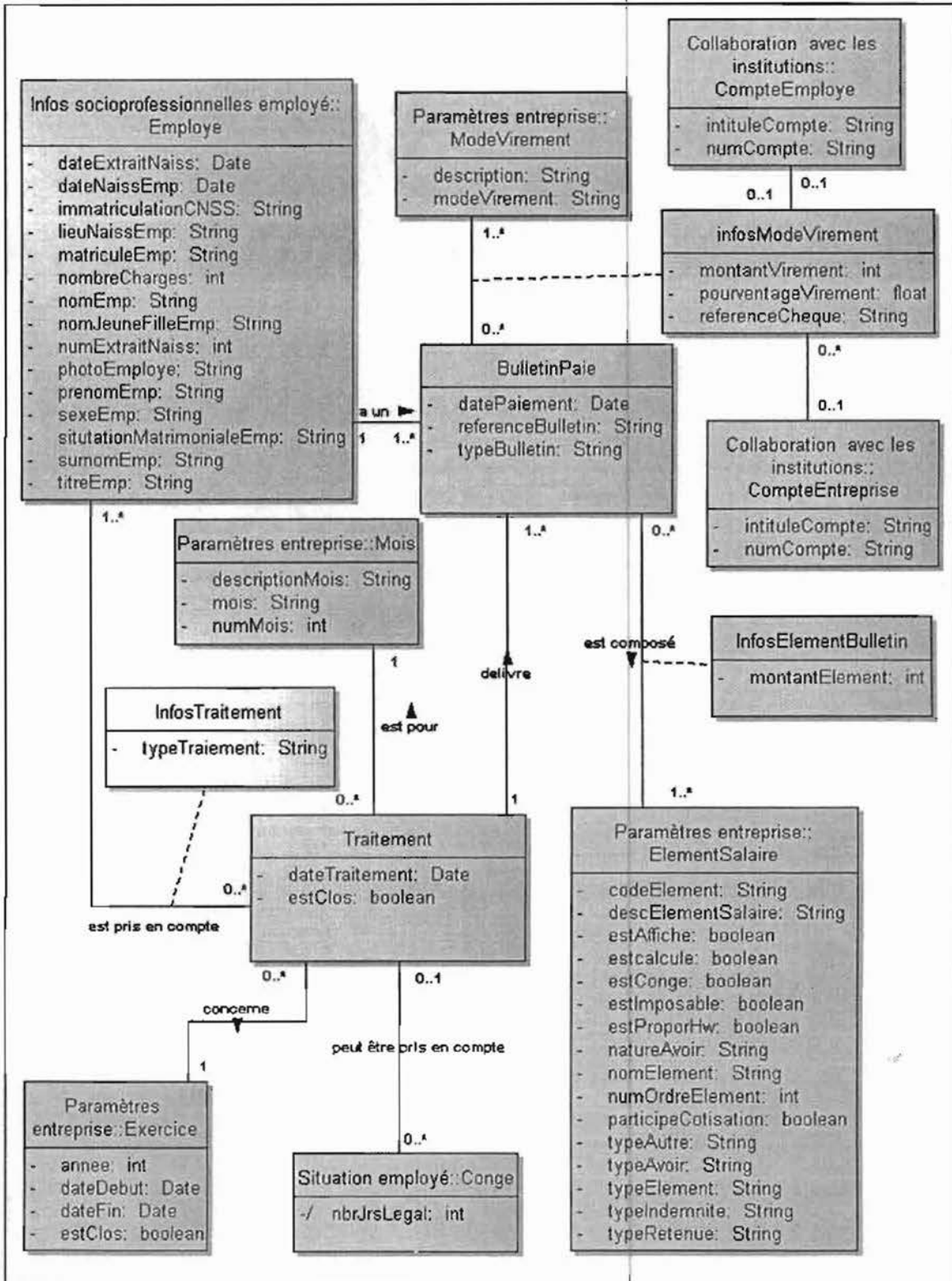


Figure 4.7 : diagramme de classes de la catégorie « Paie »

La catégorie « Profils » (figure 4.8) gère les profils des utilisateurs qui vont se connecter à la plateforme.

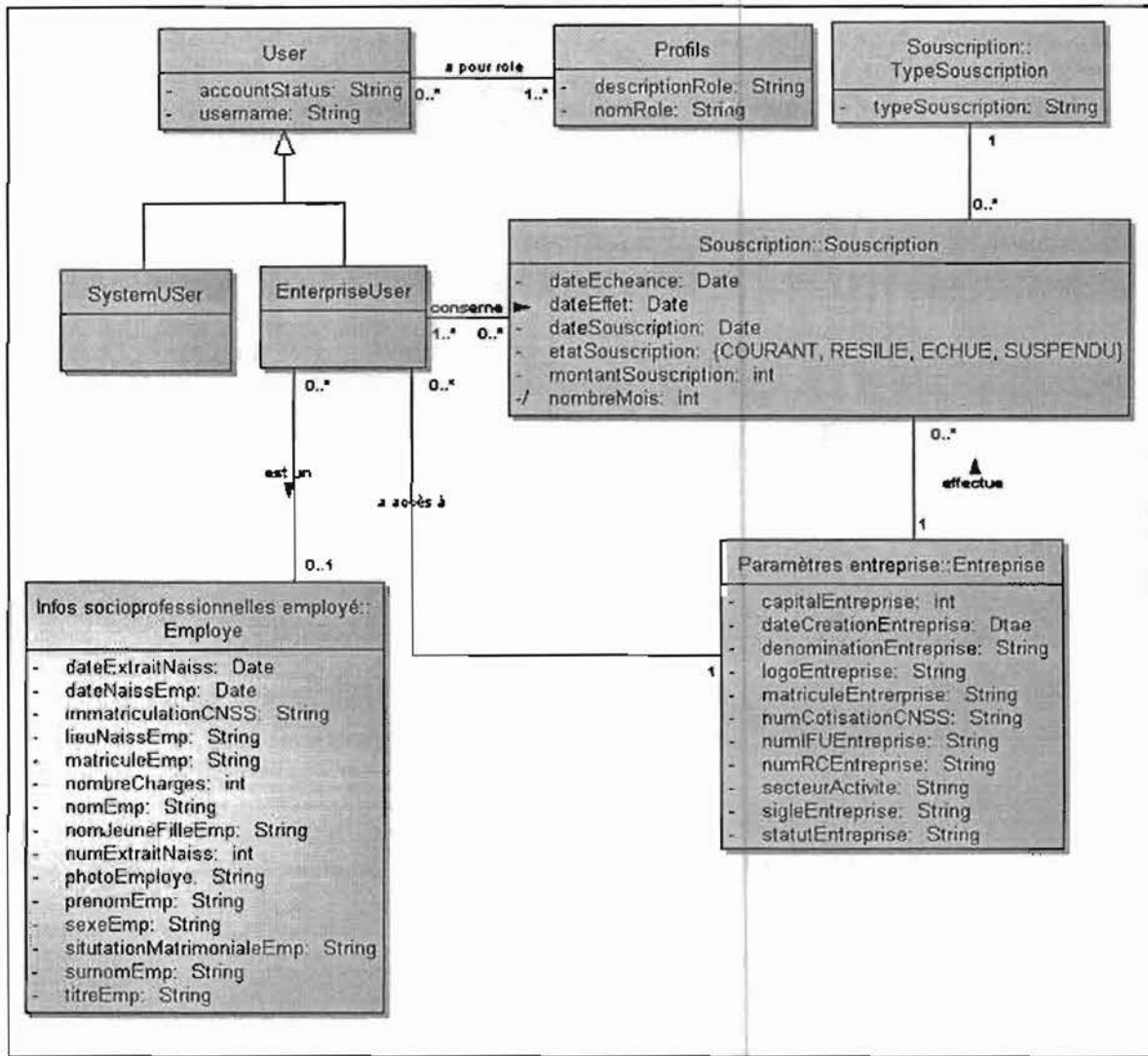


Figure 4.8 : diagramme de classes de la catégorie « Profils »

## 4.2 DEVELOPPEMENT DU MODELE DYNAMIQUE

Le développement du modèle dynamique constitue la troisième activité de l'étape d'analyse. Elle se situe sur la branche gauche du processus. Il s'agit d'une activité itérative fortement couplée avec l'activité de modélisation statique décrite dans la section précédente. L'objectif de cette étape est de présenter les interactions entre les objets du système de sorte à enrichir les modèles statiques notamment les diagrammes de classes. Les diagrammes UML couramment utilisés dans cette activité sont le diagramme de séquence, le diagramme de communication et le diagramme d'états transition. Il ne s'agit pas ici de perdre inutilement le temps à vouloir construire les diagrammes de séquence ou de communication de tous les scénarii, ou encore construire les diagrammes d'états transition de toutes les classes identifiées précédemment. Le but de la manœuvre est de développer le comportement dynamique de la plateforme de sorte à nourrir les modèles statiques. Nous avons donc choisi de vous présenter le diagramme de séquence du scénario « Préparer la paie » (figure 4.9) et le diagramme d'états transition de la classe « Employe » (figure 4.10).

### 4.2.1 Diagramme de séquence : Préparer la paie

Le diagramme de séquence met l'accent sur l'aspect chronologique des communications entre les différents objets du système.

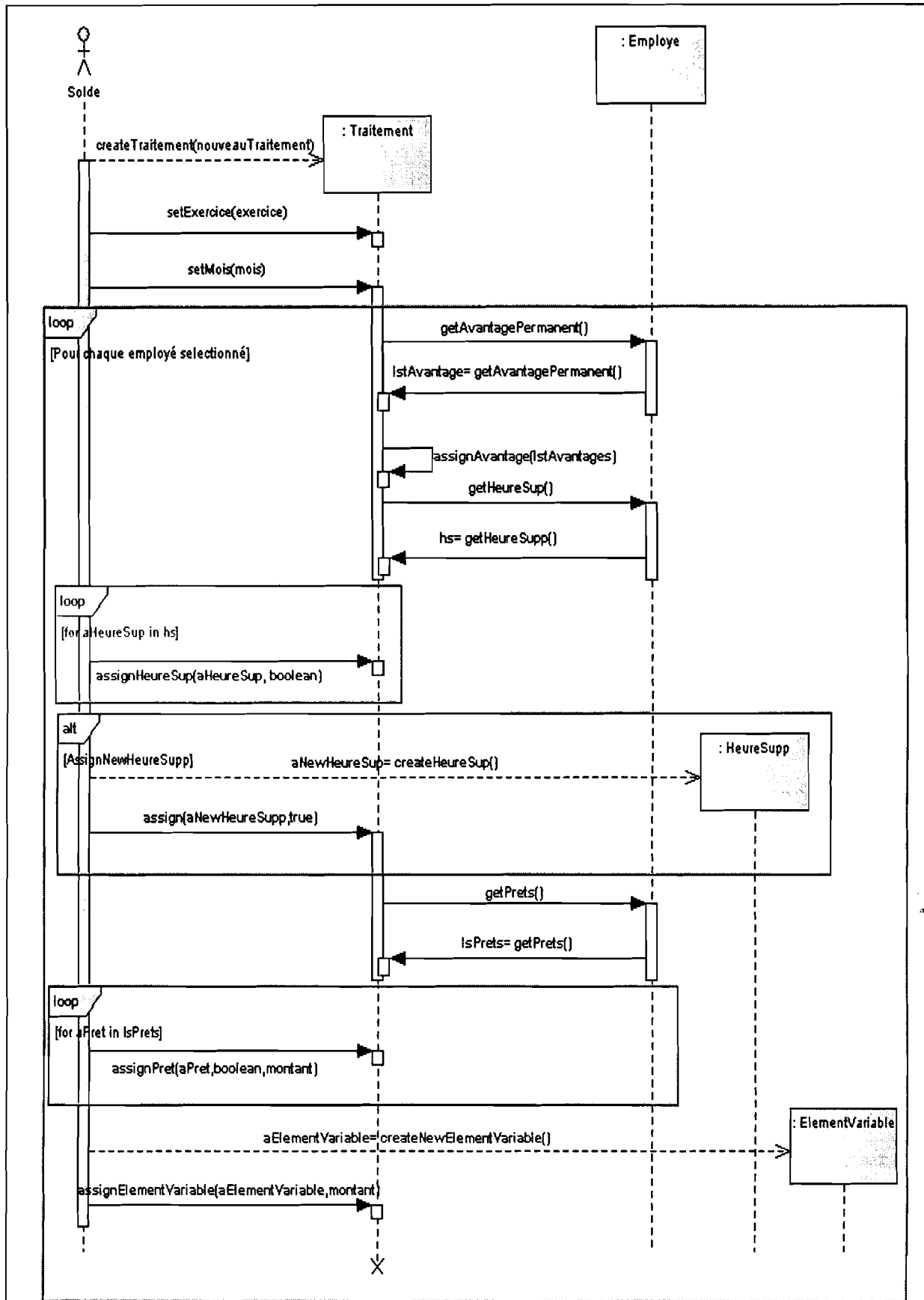


Figure 4.9 : diagramme de séquence objet du cas d'utilisation « préparer la paie »

## 4.2.2 Diagramme d'état : Employé

Le diagramme d'états transition s'intéresse au cycle de vie d'un objet générique au fil de ses interactions avec le reste du monde. De façon générale, un état représente une situation durant la vie d'un objet pendant laquelle, soit il satisfait une certaine condition, soit il exécute une certaine activité ou bien il attend un certain événement.

Le fait d'identifier les différents états d'un employé (figure 4.10) permet de nous rendre compte que le traitement de salaire lié à chaque état est différent des autres.

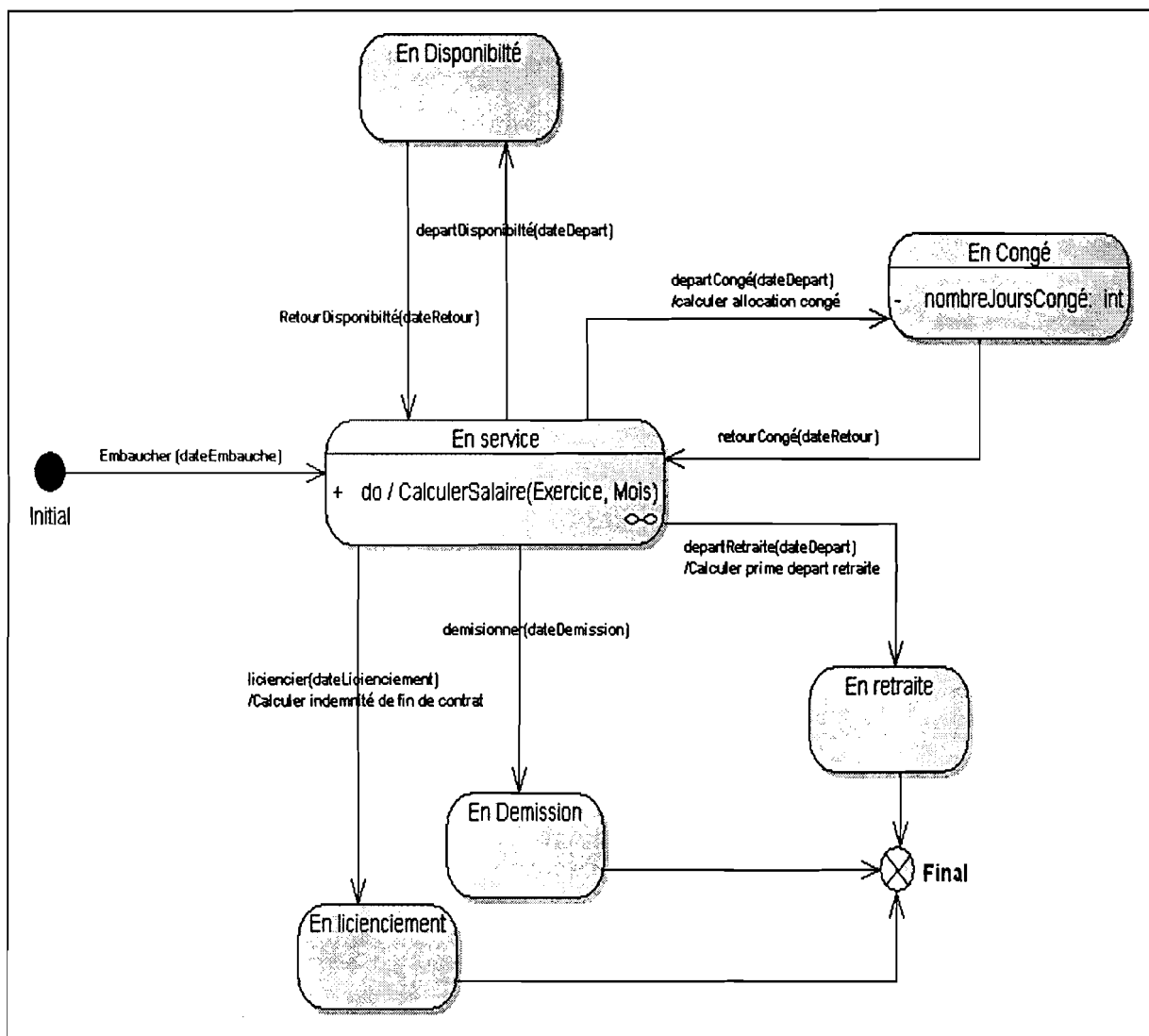
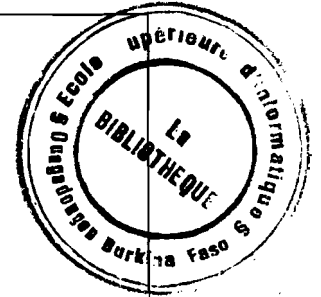


Figure 4.10 : diagramme d'états transition de la classe « Employé »

# 5

## Conception et réalisation



Cette section donne une ébauche architecturale de la plateforme **PSD4GPMS**. Elle vient compléter les étapes d'analyse conceptuelle précédentes, en apportant une description plus détaillée de la plateforme. Conformément aux spécifications de base consignées dans le cahier des charges, la plateforme doit être développée sur toutes ses composantes en environnement Oracle, suivant la technologie Java EE (Java Enterprise Edition). A cette étape, il s'agit de raffiner le résultat des analyses précédentes et d'injecter ce résultat dans l'architecture Java EE.

### 5.1 PRESENTATION DE JAVA EE

Java EE<sup>1</sup> est une plateforme de développement d'applications d'entreprise<sup>2</sup> (multi tiers) s'appuyant sur le langage Java dont les spécifications sont gérées par la société SUN<sup>3</sup>. C'est un ensemble de spécifications coordonnées et pratiques exposant des solutions pour le développement, le déploiement et l'exécution des applications multi tiers centralisées sur un serveur.

On parle aussi de Java EE pour désigner l'ensemble constitué des services offerts et de l'infrastructure d'exécution. Java EE comprend notamment :

- ✓ Les spécifications du serveur d'application (environnement d'exécution) qui définit les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées. Il est ainsi donné à des entreprises tierces de développer des serveurs

<sup>1</sup> C'est le nouveau nom donné à la cinquième version de J2EE (Java 2 Enterprise Edition). On parle maintenant Java EE 5

<sup>2</sup> Applications d'entreprise : applications distribuées au sein de l'entreprise et qui interagissent par l'intermédiaire d'un réseau

<sup>3</sup> Sun Microsystems a été racheté le 20 Avril 2009 par Oracle Corporation.



d'applications conformes à ces spécifications, sans avoir à redévelopper les principaux services ;

- ✓ Des services, au travers d'API<sup>1</sup>, c'est-à-dire des extensions Java indépendantes permettant d'offrir un certain nombre de fonctionnalités standards. Sun fournit une implémentation minimale de ces API appelée Java EE SDK (Java EE Software Development Kit).

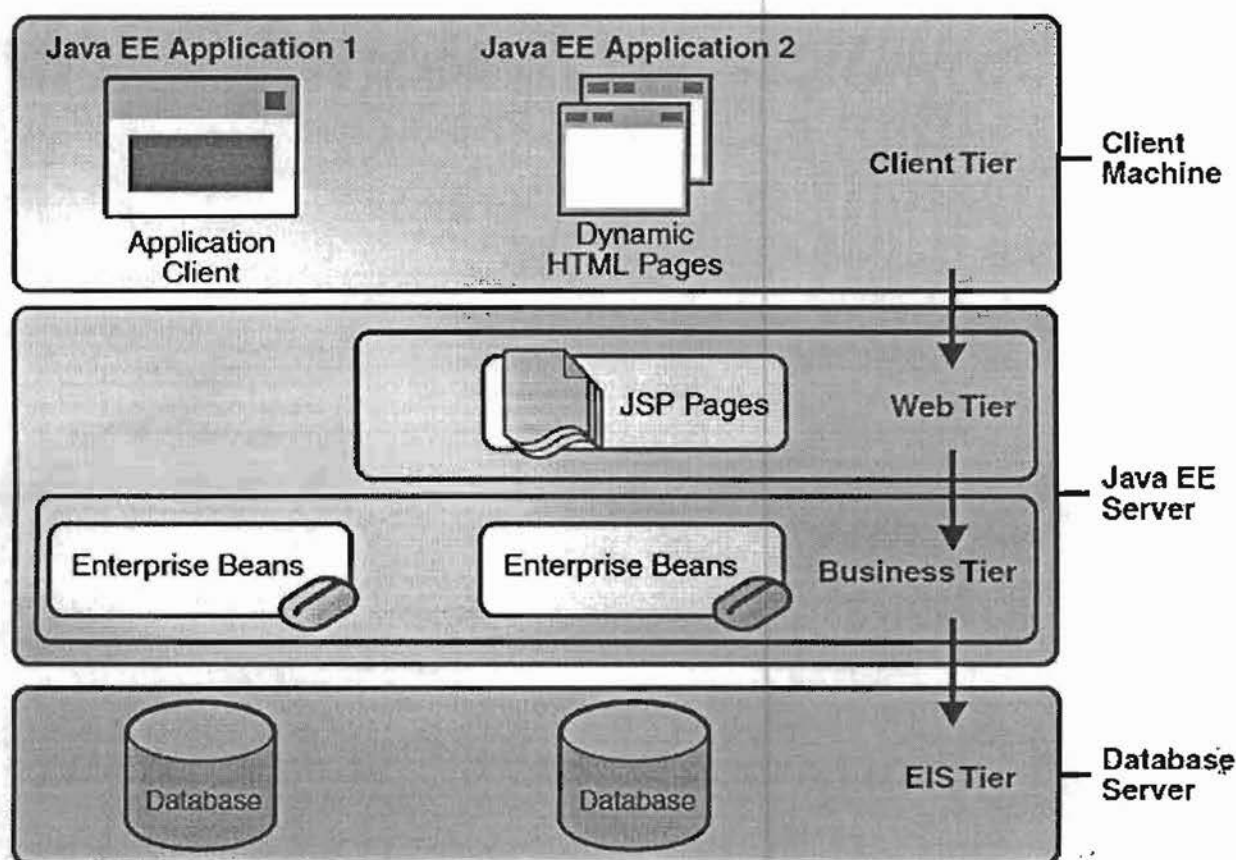


Figure 5.1 : applications Java EE distribuées

Java EE fournit un ensemble de services permettant aux composants de dialoguer entre eux et propose, comme le montre la figure 5.1, une séparation nette de l'application en couches (niveaux) :

<sup>1</sup> API (Application Programming Interface) : interface de programmation définissant la manière dont un composant informatique peut communiquer avec un autre

- ✓ La couche présentation (couche « client tier » de la figure 5.1), déployée sur la machine cliente, correspondant à l'interface homme machine ;
- ✓ La couche métier (couche « Web tier » et couche « business tier » de la figure 5.1) contenant l'essentiel des traitements de données en se basant dans la mesure du possible sur des API existantes ;
- ✓ La couche de données (couche « EIS tier ») qui contient les données de l'entreprise manipulées par l'application.

Java EE fournit notamment les avantages suivants :

- ✓ « Write once, run anywhere » fournit un style de développement basé composant, ce qui va bien avec le processus 2TUP qui est un processus unifié. Les processus unifiés sont basés composants ;
- ✓ La majorité des serveurs d'applications supporte la technologie Java EE, ce qui donne plus de flexibilité pour le déploiement ;
- ✓ Java EE facilite le développement suivant le modèle de conception MVC (Modèle Vue Contrôleur) en séparant nettement les besoins du client (en termes d'interface homme machine) de la logique métier. Le développement suivant le modèle MVC est l'une des meilleures pratiques (best practices) du développement Objet ;
- ✓ Java EE facilite le développement en équipes en permettant le découpage des tâches de développement suivant les talents des développeurs. Ainsi, les designers web pourront s'occuper des pages web, les programmeurs java de la logique métier et ainsi de suite.

Cependant, la plateforme Java EE n'est pas exempt d'inconvénients au nombre desquels on peut citer :

- ✓ La complexité du développement d'applications Java EE ;
- ✓ La nécessité d'assimiler de nombreux concepts avant de pouvoir déployer une application Java EE de qualité ;

- ✓ Les codes « do it yourself » (fais le toi même) ne permettent pas la réutilisation du code ;
- ✓ Une large gamme des codes « do it yourself » sont des tâches communes et doivent être des composants prêts à l'emploi ;
- ✓ L'écriture de beaucoup de lignes de code facilite l'introduction d'erreurs.

Devant la complexité de Java EE et conformément aux spécifications de base consignées dans le cahier des charges préliminaires, nous avons utilisé le framework Oracle ADF 11G (Application Development Framework), qui est un framework pratique et totalement compatible Java EE 5, pour le développement de la plateforme.

## 5.2 PRESENTATION DE ORACLE ADF

De nombreuses tentatives ont été faites pour réduire la complexité de Java EE par la création des frameworks et des abstractions visant à faciliter la tâche du développeur. Le framework Oracle ADF est la contribution d'Oracle à cet effort. Oracle ADF est une couche de productivité complète bâtie sur Java EE, dont le but est de simplifier la création d'applications d'entreprises. Il accélère le développement avec des implémentations de modèles de conception Java EE prêts à l'emploi.

Le framework Oracle ADF [10] est construit sur la technologie Java EE et réduit toute la complexité de celle-ci. Il permet notamment :

- ✓ Plus de réutilisation, moins de codage, donc moins d'erreurs ;
- ✓ De se focaliser totalement sur la gestion de la logique métier et non sur les accessoires ;
- ✓ D'encourager les meilleures pratiques de la plateforme Java EE en implémentant le modèle de conception MVC ;
- ✓ De fournir un environnement flexible, extensible et supportant différents styles de développement.

Oracle ADF est livré gratuitement avec l'éditeur de développement Oracle JDeveloper à partir de la version 10G. La version que nous allons utiliser pour le développement de la plateforme PSD4GPMS est la 11G totalement compatible Java EE 5. La figure 5.2 donne l'architecture simplifiée du framework ADF 11G.

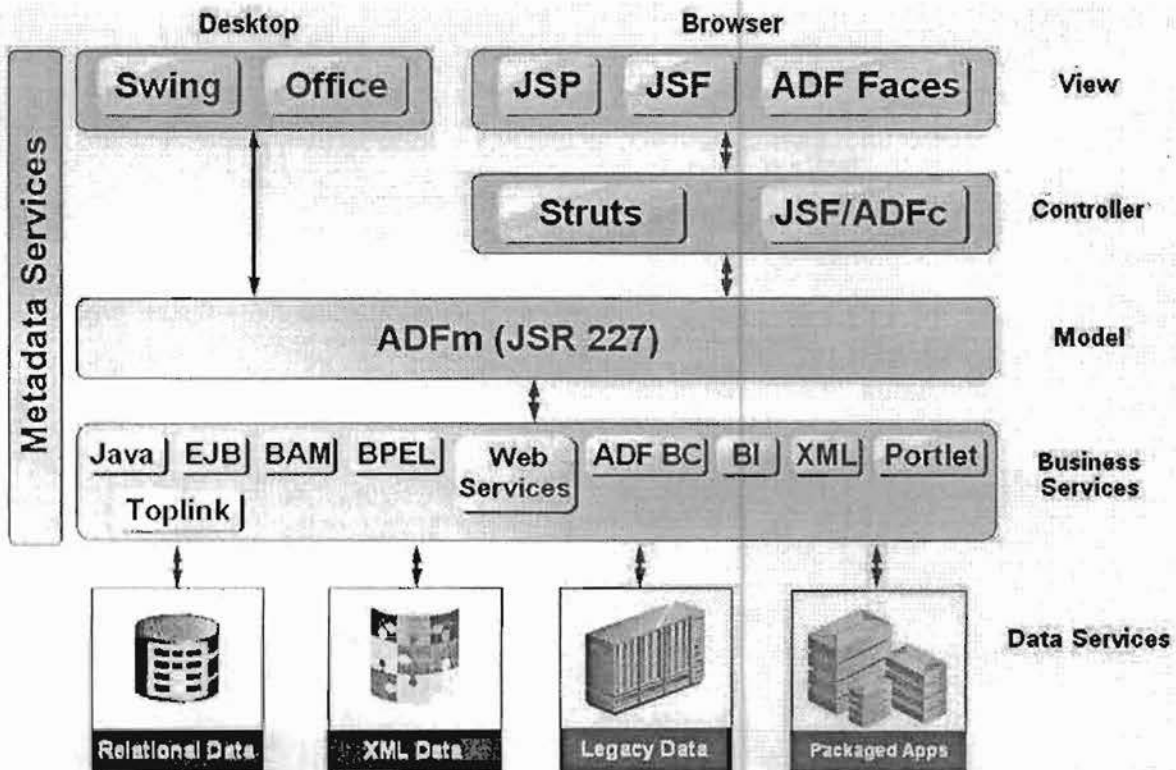


Figure 5.2 : architecture de Oracle ADF 11G

### 5.3 CONCEPTION ET REALISATION DE LA PLATEFORME

Comme Java EE 5 propose une séparation physique de l'application en trois couches : couche présentation, couche métier et couche des données, la conception de la plateforme revient à concevoir chacune de ces couches et à les faire communiquer.

#### 5.3.1 Conception de la couche des données

La couche des données constitue une composante essentielle dans l'architecture Java EE. En effet, elle héberge toutes les données de l'entreprise manipulées par l'application. La conception de cette couche consiste à :

- ✓ Définir le mécanisme de stockage c'est à dire sous quelle forme les instances seront sauvegardées sur un support physique ;
- ✓ Définir la couche d'accès aux données c'est à dire comment se réalise la transformation du modèle de stockage en modèle mémoire ;
- ✓ Evaluer le volume et l'évolution des données qui seront stockées ;
- ✓ Envisager des procédures de sécurité.



### 5.3.1.1 Mécanisme de stockage

Il existe aujourd'hui plusieurs modes de stockage possibles : système de fichiers, base de données relationnelle, base de données objet, base de données XML. Cependant, pour des raisons de performance, de sécurité et même de maturité, la plateforme PSD4GPMS exploitera une base de données relationnelle. Le système de gestion de bases de données retenu est *Oracle Database 10G Release 2 Enterprise Edition*, décrit plus en détail dans la section *Déploiement et sécurité*.

### 5.3.1.2 Volume et évolution des données

Il est important et utile d'avoir une idée sur le volume et l'évolution des données de la base. Ces indicateurs aident souvent dans le choix du système de gestion de bases de données à utiliser. Le choix de *Oracle Database* comme base de données nous enlève ce souci de volume et d'évolution. Néanmoins, pour cette version de la plateforme, nous estimons un nombre maximal de vingt (20) souscriptions. La plateforme gèrera donc les données d'au plus vingt (20) entreprises exploitant chacune des données de l'ordre de 20 Gigabytes par an. Un mécanisme de restitution des données est prévu pour les entreprises désirant résilier définitivement leur contrat.

### 5.3.1.3 Sécurité des données

Sécuriser une application, revient dans la plupart des cas, à protéger les données qu'elle manipule. Pour ce faire, un ensemble de procédures de sécurité et de mesures de protection des données a été mis en place ; ces procédures sont détaillées dans la section *Déploiement et sécurité*.

### 5.3.1.4 Mise en œuvre de la base de données.

Le processus 2TUP retenu étant orienté objet, et la base de données exploitée par la plateforme étant relationnelle, nous devons établir des correspondances afin d'extraire les modèles relationnels des diagrammes de classes UML. Nous avons utilisé le tableau 5.1 pour faire le rapprochement entre les modèles de classes UML et les modèles relationnels.

**Tableau 5.1** : correspondances entre concepts objets et concepts relationnels

| Modèle Objet                   | Modèle relationnel                          |
|--------------------------------|---|
| Classe                         | Table                                       |
| Attribut de type simple        | Colonne                                     |
| Attribut de type composé       | Colonnes ou clé étrangère                   |
| Instance                       | T-uplet                                     |
| OID (Object unique identifier) | Clé primaire                                |
| Association                    | Clé étrangère ou table de liens             |
| Héritage                       | Clé primaire identique sur plusieurs tables |

### 5.3.1.5 Accès aux données

La conception de la couche d'accès aux données dans l'architecture Java EE, est basée sur les frameworks de mappage objet/relationnel. Dans le cadre du développement de notre plateforme, nous utilisons le framework de mappage objet relationnel Oracle Toplink pour la persistance des données. Nous en parlerons plus en détail un peu plus loin dans cette section.

## 5.3.2 Conception de la couche métier et de la couche présentation

Le framework Oracle ADF 11G que nous implémentons pour la réalisation de la plateforme, encourage les meilleures pratiques de la plateforme Java EE en implémentant le modèle de conception MVC. Les modèles de conception décrivent des organisations pratiques de classes objets qui résultent souvent d'une conception empirique, le concepteur objet tentant de faciliter la réutilisation et la maintenance du code.

Comme le montre la figure 5.3, le modèle MVC sépare les données (le modèle), l'interface homme machine (la vue) et la logique de contrôle (le contrôleur) d'un objet. Le modèle définit l'accès aux données et l'implémentation des services métiers. La vue représente l'interface utilisateur, c'est-à-dire ce avec quoi l'utilisateur interagit. Elle n'effectue aucun traitement, elle se contente simplement d'afficher les données que lui fournit le contrôleur. On peut donc avoir plusieurs vues qui présentent les données d'un même modèle c'est-à-dire plusieurs façons de présenter les mêmes données.

Le contrôleur gère l'interface entre le modèle et la vue. Il va interpréter la requête de cette dernière pour lui envoyer la vue correspondante. Il effectue la synchronisation entre le modèle et les vues.

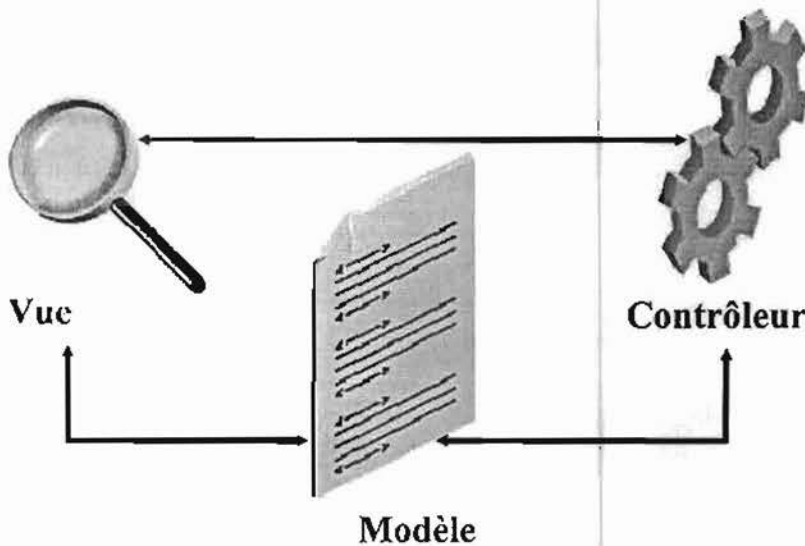


Figure 5.3 : modèle MVC

### 5.3.2.1 Présentation de Oracle JDeveloper

Le développement de la plateforme nécessite un environnement de développement intégré (EDI). Oracle JDeveloper est un EDI Java gratuit produit par Oracle dont l'objectif est principalement de faciliter le développement des applications métiers. Pour cela il intègre les plus récents standards industriels en termes de développement Java, XML, Web services et SQL. L'utilisation de JDeveloper 11G offre les avantages suivants :

- ✓ JDeveloper 11G est livré avec le framework ORACLE ADF (décrit plus haut) et permet donc l'utilisation de celui-ci ;

- ✓ JDeveloper contient de nombreux assistants et outils pour simplifier le développement avec Oracle ADF ;
- ✓ JDeveloper intègre de nombreux outils qui permettent de gérer l'ensemble du cycle de développement de l'application.

### 5.3.2.2 Les modules de l'application

Avec l'outil Oracle JDeveloper, le choix d'une technologie aide considérablement dans la production de l'application. Avec la prise en compte des recommandations d'Oracle, il a été convenu de développer la plateforme suivant la technologie web [JSF, EJB, TopLink] du framework Oracle ADF en suivant le design pattern MVC. Cette technologie sépare l'application en deux projets distincts : le projet « Model » et le projet « viewController ».

Le projet « Model » assure d'une part, la persistance des données avec le framework de mappage objet/relationnel Oracle Toplink, et d'autre part, l'encapsulation et l'exposition des services métier de l'entreprise dans les *Entreprise Java Beans type session*.

Le projet « ViewController » implémente le « V » et le « C » du modèle MVC. Il est assuré par le framework Oracle ADF Faces (partie intégrante de Oracle ADF), en collaboration avec le standard JSF pour la gestion de la vue et de la logique applicative.

#### 5.3.2.2.1 Présentation et réalisation du projet Model

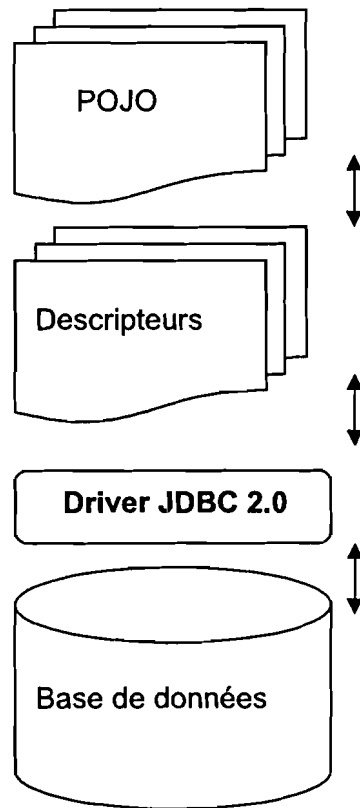
Ce projet assure la persistance des données avec le framework de mappage objet/relationnel Oracle TopLink et de l'encapsulation des services métiers avec les sessions beans.

##### 5.3.2.2.1.1 Persistance des objets avec Oracle Toplink

Oracle recommande de développer le projet « Model » en utilisant le framework de mappage objet/relationnel Oracle Toplink pour la persistance des données et Enterprise Java Beans type session pour l'implémentation des services métiers.

Oracle Toplink est un framework de gestion de la persistance objet/relationnel en Java, membre de la famille de produits Oracle Fusion Middleware. Il permet notamment de stocker des objets Java et des EJB (Enterprise Java Beans) dans des bases de données relationnelles. Oracle Toplink supporte tous les serveurs de bases de données possédant un driver JDBC 2.0.





**Figure 5.4 :** architecture simplifiée du framework Toplink

La liaison entre une donnée dans la base de données et une donnée dans l'application s'appelle « mapping ». Toplink utilise, comme le montre la figure 5.4, des descripteurs pour établir cette liaison. Du côté de l'application, Toplink utilise des classes simples qu'on appelle POJO (Plain Old Java Object). Ce concept de POJO a pour but de représenter une idée avec le meilleur design possible. Il a souvent été opposé aux EJB jusqu'à la version 3 qui les utilise. C'est donc une classe simple et indépendante de tout système ou framework. Les POJO correspondent aux JavaBeans. Toutefois, on n'emploie plus ce nom à cause de son ambiguïté avec les Enterprise JavaBeans.

### 5.3.2.2.1.2 Implémentation des services métier avec les EJB

#### Micro processus de conception logique

Afin d'aboutir à l'implémentation des services métier avec les EJB, les classes provenant de l'analyse doivent passer par le micro processus de conception logique consistant à concevoir les classes, les associations, les attributs et les opérations.

En effet, les classes qui proviennent de l'analyse ne sont pas toujours conformes aux possibilités du langage d'implémentation. Dans certains cas, une analyse orientée objet est

réalisée dans un langage non objet. Concevoir les classes, c'est aussi en introduire de nouvelles soit pour prendre en charge des responsabilités purement techniques, soit pour décharger une classe d'analyse de certains aspects techniques.

La conception des associations est nécessaire étant donné que l'association est un concept inconnu de la plupart des langages orientés objets. Elle se transforme en attribut ou en tableau d'attributs suivant sa multiplicité.

La conception des attributs consiste principalement à définir le type des attributs identifiés en analyse. Bien que la plupart des attributs se satisfassent des types de base Java, certains attributs d'analyse correspondent à une structure de données qu'il est nécessaire de préciser. Concevoir les attributs, c'est également spécifier leur visibilité et leur mode d'accès. Par défaut, un attribut est privé et ce principe reste invariable dans notre conception. La prise en compte des propriétés UML {changeable}, {readOnly} ou {frozen} est donc implicitement gérée par des opérations d'accès. Enfin, concevoir les attributs, c'est spécifier les méthodes qui servent à la mise à jour des attributs ;

Enfin la conception des opérations constitue la dernière étape du micro processus de conception logique, le but de la manœuvre ici étant de concevoir des POJO pour les Sessions Beans, nous avons repoussé cette activité dans la conception des façades EJB.

### **Implémentation des services métier avec les EJB**

Les EJB sont des composants Java portables, réutilisables et déployables, qui peuvent être assemblés pour créer des applications. Ils s'exécutent dans un conteneur EJB qui va leur fournir des services tels que les transactions et la persistance. La technologie Java EE a spécifié trois types d'EJB : les EJB type « session », les EJB type « entity » et les EJB type « message-driven ».

Les EJB type « *session* » sont des applications coté serveur permettant de fournir un ou plusieurs services à plusieurs applications clientes. Ces EJB constituent les briques de la logique métier d'une application. Ils traitent les données et effectuent les opérations liées à la logique métier de l'entreprise ;

Les EJB type « *entity* » forment la passerelle applicative entre la logique applicative et les sources de données. Ils permettent une abstraction quasi complète du stockage des données, permettant à l'application de rendre persistantes ou de charger les données en mémoire de manière totalement transparente ;

Les EJB type « message-driven » permettent le traitement des messages asynchrones et la communication entre les différents composants Java EE.

**Tableau 5.2 : façades EJB de la plateforme PSD4GPMS**

| Façade               | Façade(s) utilisée(s)  | Description  |
|----------------------|--|--|
| ParamEtatiquesFacade |  | Fournit tous les services relatifs aux paramètres en vigueur de l'Etat qui entrent dans le cadre du traitement des salaires.   |
| ParamEntreprise      |  | Fournit tous les services nécessaires à la gestion des paramètres de l'entreprise, dans le cadre du traitement des salaires. Cette façade permet à chaque entreprise d'avoir son propre contexte dans le logiciel. |
| InfosEmployesFacade  | ParamEntreprise  | Fournit tous les services nécessaires à la gestion des informations relatives à la paie sur les employés de l'entreprise.  |
| PaieFacade           | ParamEtatiquesFacade<br>ParamEntreprise<br>InfosEmployesFacade | Contient le noyau de la paie. Elle fournit principalement les services nécessaires au calcul des salaires des employés.  |
| AdminFacade          |  | Fournit des services liés à la souscription des entreprises tierces et à la gestion des utilisateurs.  |

Les meilleures pratiques recommandent d'implémenter les EJB type « session » selon le modèle de conception « façade ». Ce modèle fournit une interface d'accès uniforme à un ensemble de systèmes. L'application utilise donc un ensemble de systèmes sans réellement en avoir conscience et sans savoir les difficultés d'utilisation que chaque sous-système peut induire. Le tableau 5.2 détaille les différentes façades que nous avons implémentées. Ces façades sont déduites des catégories obtenues en analyse.

### 5.3.2.2.2 Présentation du projet ViewController

Par défaut JDeveloper nomme ce projet « ViewController » pour signifier que ce projet prend en compte le « V » et le « C » du modèle MVC. Le « ViewController » est lié au projet « Model » et exploite les services que celui-ci lui expose. Le contrôle est effectué

essentiellement par le contrôleur JSF *FacesServlet*. La couche présentation de la plateforme est donc assurée par les technologies JSF et Oracle ADF Faces.

#### 5.3.2.2.1 JSF

JSF [11] (Java Server Faces) est un framework de développement d'applications Web en Java permettant de respecter le modèle d'architecture MVC et basé sur des composants côté présentation. Java Server Faces permet :

- ✓ Une implémentation du modèle MVC ;
- ✓ un mappage entre l'HTML et l'objet (le code java étant séparé du code html) ;
- ✓ l'utilisation d'un ensemble de composants riches et réutilisables ;
- ✓ une liaison simple entre les actions côté client de l'utilisateur (eventlistener) et le code Java côté serveur ;
- ✓ Création de nouveaux composants graphiques.

JSF implémente le modèle MVC2 (figure 5.5) qui est basé sur MVC. Ainsi, en regardant de près, l'on constate qu'une application JSF est divisée en trois éléments : la vue, le modèle de navigation et la logique applicative.

#### Modèle

Avec JSF, afin de séparer le code Java de la description des pages, le concept de bean managé (managed bean) a été introduit. Un bean managé est un *JavaBean*<sup>1</sup> dont le cycle de vie est géré par JSF. Les beans managés sont définis dans le fichier de configuration « faces-config.xml » et peuvent être accédés à travers les « Expression Language » dans les pages JSF.

---

<sup>1</sup> Un bean managé est un bean respectant la spécification *JavaBean* :

- ✓ La classe doit posséder au moins un constructeur vide ;
- ✓ Les attributs de la classe ne doivent pas être publics ;
- ✓ Chaque attribut de la classe doit posséder les méthodes *getter* et *setter*.

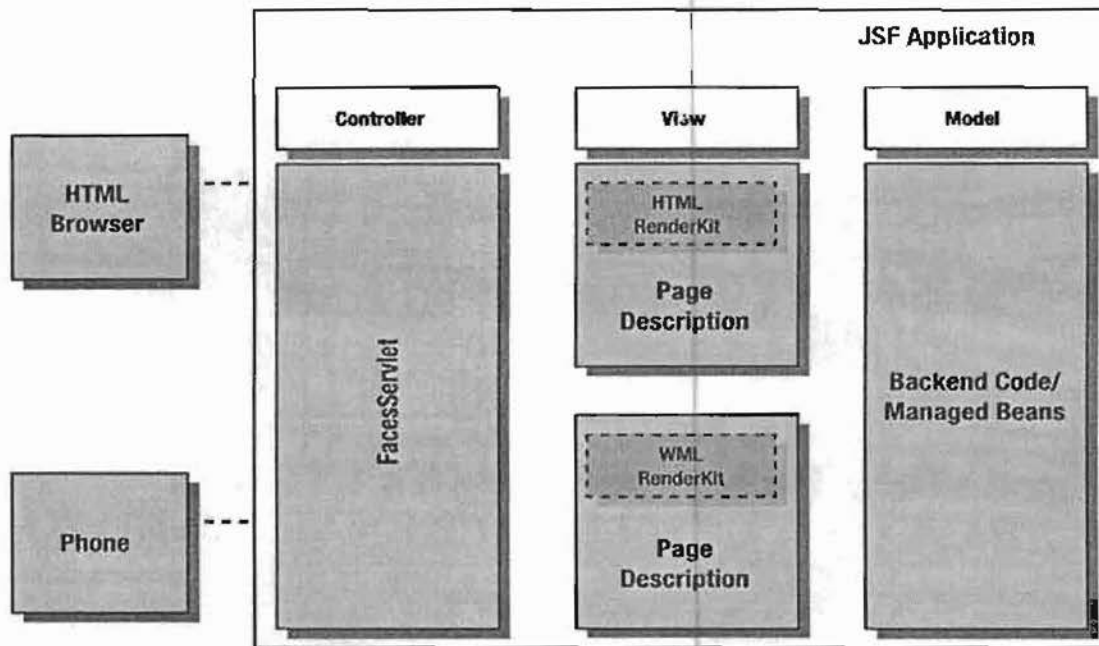


Figure 5.5 : architecture MVC2 avec JSF.

### Le contrôleur

Le contrôle est assuré par un ensemble de servlets dont le plus important qui est *FacesServlet* (figure 5.5). Le contrôleur *FacesServlet* prend en entrée une requête http provenant d'un client, la traite parfois avec l'aide du modèle et retourne une réponse http au client.

### Le fichier de configuration JSF « faces-config.xml »

Toute application JSF nécessite un fichier de configuration qui s'appelle par défaut « faces-config.xml ». Ce fichier est placé dans le répertoire WEB-INF au même niveau que le fichier descripteur de déploiement « web.xml » et décrit essentiellement les principaux éléments suivants : les beans managés (managed bean), les règles de navigation entre les pages, les ressources éventuelles suite à des messages, la configuration de la localisation des ressources, etc. Le tableau 5.3 montre un extrait du fichier de configuration de la plateforme.

**Tableau 5.3** : extrait du fichier de configuration faces-config.xml

## Extrait du fichier de configuration faces-config.xml

```
...
<!-- Enregistrement du menu-->
<managed-bean>
  <managed-bean-name>MainMenu</managed-bean-name>
  <managed-bean-class> org.apache.myfaces.trinidad.model.XMLMenuModel </managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
  <managed-property>
    <property-name>createHiddenNodes</property-name> <value>>false</value>
  </managed-property>
  <managed-property>
    <property-name>source</property-name> <property-class>java.lang.String</property-class>
    <value>/WEB-INF/mainMenu.xml</value>
  </managed-property>
</managed-bean>
<!-- Règles de navigation-->
<navigation-rule>
  <navigation-case>
    <from-outcome>home</from-outcome>
    <to-view-id>/app/psd4gpms.jsp</to-view-id>
    <redirect/>
  </navigation-case>
  <navigation-case>
    <from-outcome>entreprise</from-outcome>
    <to-view-id>/app/rh/entreprise/entreprise.jsp</to-view-id>
    <redirect/>
  </navigation-case>
  ...
</navigation-rule>
...
```

### 5.3.2.2.2 Oracle ADF Faces Rich Client

Oracle ADF Faces Rich Client, connu sous le nom de Oracle ADF Faces<sup>1</sup>, est un ensemble de composants JSF qui intègrent les fonctionnalités AJAX (Asynchronous JavaScript And XML), permettant ainsi le rafraîchissement partiel des pages de l'application web. La dernière version de Oracle ADF Faces combine les techniques de développement AJAX et la technologie JSF pour produire des Applications Internet Riches (RIA pour Rich Internet Application). ADF Faces intègre les fonctionnalités AJAX, facilitant donc le développement des applications RIA avec JSF. Le framework Oracle ADF Faces fournit plus de cent (100) composants RIA, incluant les tables hiérarchiques, les tables sous forme d'arbre, les menus en arbre, des boîtes de dialogues, etc.

## 5.4 PRESENTATION DU DES CAPTURES D'ECRANS

Pour donner un aperçu concret du travail réalisé durant le stage, nous allons présenter quelques écrans de la plateforme PSD4GPMS.



<sup>1</sup> Oracle ADF Faces utilise comme base fondamentale *Apache MyFaces Trinidad*.

### 5.4.1 Connexion à la plateforme

L'authentification est obligatoire pour tout utilisateur désireux accéder à la plateforme. Pour des raisons de sécurité, le nombre de tentatives d'accès à la plateforme est limité à trois (03) sur un poste donné. Au delà de trois (03) tentatives sans succès, la plateforme est verrouillée sur ce poste pour une durée paramétrable par l'administrateur central de la plateforme.



Figure 5.6 : page de connexion à la plateforme

### 5.4.2 La page d'accueil

Après une connexion réussie, l'utilisateur accède à la page d'accueil qui lui présente un contenu relatif à son profil et au type d'utilisateur qu'il est. Le menu est donc filtré en conséquence. Typiquement, les utilisateurs « entreprise » atterrissent sur l'entreprise à laquelle ils appartiennent. Les acteurs « cabinet », quant à eux peuvent se connecter à plusieurs entreprises simultanément. La figure 5.7 montre la page d'accueil, selon la vision du développeur.





Figure 5.7 : page d'accueil de la plateforme

### 5.4.3 Configuration de l'entreprise

La figure 5.8 présente la page de configuration de l'entreprise qui est accessible par l'acteur « entreprise » RH ou par l'acteur cabinet correspondant. Cette page permet d'accéder aux différentes pages de définition des paramètres de l'entreprise connectée : éléments de salaire, classement et grille salariale, services, fonctions, etc.

Entreprise: ICOMG  
 Matricule: 902021PMS280M  
 Dénomination: International Consulting and Management Group  
 Installée: BURKINA FASO  
 Statut: SARL  
 Capital: 1000000000  
 Date de création: 01/01/2005  
 Immatriculation CNSS: 1235653666P  
 Numéro IFU:  
 Numéro RC:  
 Entreprise mère:

Prendre en compte l'ancienneté dans le calcul des salaires

Ville: Ouagadougou  
 Adresse cadastrale: Rue 1521 Ouid  
 Adresse postale: 01 BP 13405 Ouagadougou  
 Téléphone: +226 50 351 456  
 Fax:  
 Telex:  
 Email: infos@icomg.com  
 Site web: www.icomg.com

| Banque/Caisse          | Intitulé compte | Numéro compte |
|------------------------|-----------------|---------------|
| Aucun numéro de compte |                 |               |

Copyright © ICOMG 2008. Tous droits réservés. A propos de GPMMS

Figure 5.8 : page d'accueil pour la configuration de l'entreprise connectée

#### 5.4.4 Eléments de salaire

Cette page présente, comme le montre la figure 5.9, les éléments de salaire de l'entreprise connectée tout en donnant la possibilité de les modifier ou d'en ajouter. Elle est accessible par l'utilisateur RH ou par tout acteur « cabinet » correspondant.

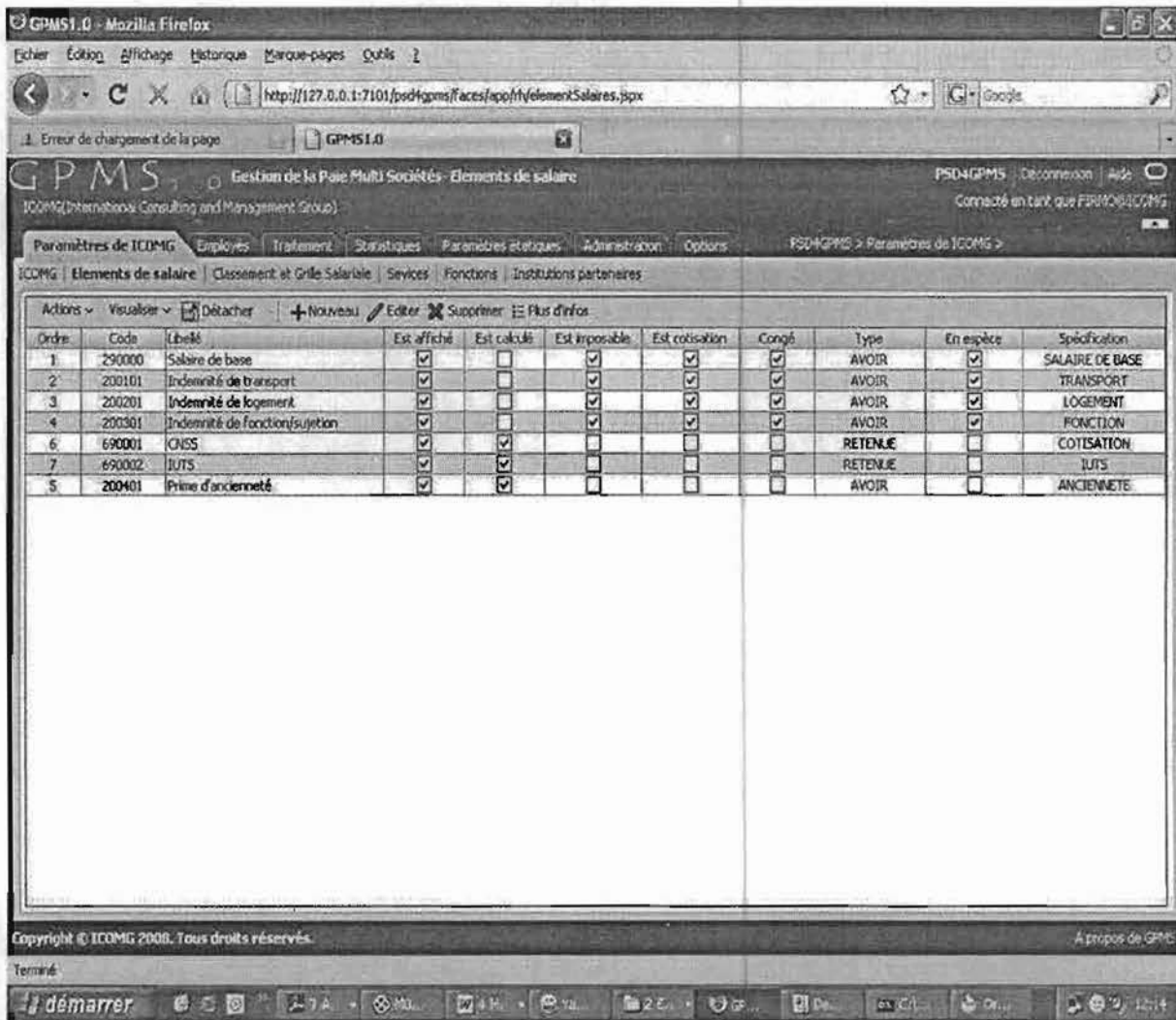


Figure 5.9 : page listant les éléments de salaire de l'entreprise connectée

### 5.4.5 Employés

Cette page (figure 5.10) présente les employés de l'entreprise connectée. Elle est accessible aux utilisateurs « entreprise » RH et Solde. Elle donne accès aux pages de définition des informations salariales des employés : avantages permanents, avantages variables, heures supplémentaires effectuées, prêts et avances sur salaire, etc. Elle offre aussi un mécanisme de filtrage des employés par recherche avec mots clefs.

The screenshot shows the GPMS1.0 web application interface. The browser address bar indicates the URL: http://127.0.0.1:7101/psd4gpm/frames/app/rh/employe/employees.jsp. The application title is 'GPMS1.0 Gestion de la Paie Multi Sociétés - Liste'. The main content area displays a table of employees with the following data:

| Matricule        | Nom et prénom(s)        | Date de naissance | Lieu de naissance | Sexe     | Date d'embauche | Charges | Immatriculation CNSS | Type d'employé | Statut de l'employé |
|------------------|-------------------------|-------------------|-------------------|----------|-----------------|---------|----------------------|----------------|---------------------|
| 9211KWM506850092 | PITROIPA PAUL           | 03/11/2008        | Village           | Masculin | 01/10/2005      | 5       | 164081109010K        | Cadre          | SERVICE             |
| IG05070M6207001  | KONE MOUSSA             | 22/11/1965        | Ouagadougou       | Masculin | 01/01/2005      | 5       | 162071090001T        | Cadre          | SERVICE             |
| IG0507Y80011010  | YABRE S. T Bertrand     | 01/12/1980        | Ouagadougou       | Masculin | 01/01/2007      | 0       | neant                | Cadre          | SERVICE             |
| 110147           | KABORE Tong-Noma Firmin | 11/10/1982        | Ouagadougou       | Masculin | 01/01/2005      | 0       | neant                | Cadre          | SERVICE             |

Figure 5.10 : page listant les employés de l'entreprise connectée

### 5.4.6 Ajout d'un nouvel employé

Cette page (figure 5.11) permet d'ajouter un nouvel employé à la liste des employés de l'entreprise connectée. Elle est accessible à l'utilisateur « entreprise » RH ou à tout acteur « cabinet » ayant ce profil.

**Identification de l'employé**

Matricule: 11047 Titré: M. Nom: KABORE  
 Jeune fille: Prénom(s): Tong-Nona Firmin Date naissance: 11/10/1982  
 Lieu naissance: Dégoula Sexe: Masculin Type d'employé: Cadre  
 Nombre charges: 0 Date embauche: 01/01/2005 Immatriculation CNSS: 2122-2121-1212  
 Situation matrimoniale: CELIBATAIRE Date extrait: Numéro extrait:  
 Service: Direction technique Fonction: Concepteur Nationalité: burkinabé

**Classement de l'employé**

Visualiser Classement: ALL

| Code Element | Element de salaire     | Montant |
|--------------|------------------------|---------|
| 290000       | Salaires de base       | 250000  |
| 200101       | Indemnité de transport | 100000  |

**Avantages permanents propres à l'employé**

Visualiser + / -

| Code Element | Element de salaire     | Montant avantage |
|--------------|------------------------|------------------|
| 290000       | Salaires de base       | 700 000          |
| 200101       | Indemnité de transport | 75 000           |
| 200201       | Indemnité de logement  | 50 000           |

**Adresse de l'employé**

Photo

Ville: Ouagadougou  
 BP: 01 BP 13024 Ouaga 01  
 Tel mobile: 70 15 89 43  
 Tel Bureau:  
 Email: kafito@gmail.com  
 Site web: www.kafito.com

**Comptes bancaires de l'employé**

Visualiser + / -

| Banque/Caisse  | Intitulé compte | Numéro compte   |
|----------------|-----------------|-----------------|
| BANK OF AFRICA | KABORE T Firmin | 0-210-131-000-2 |

Enregistrer Annuler

Figure 5.11 : page d'ajout d'un nouvel employé

### 5.4.7 Traitement des salaires

Cette page (figure 5.12) contient essentiellement les liens suivants :

- ✓ Nouveau traitement permettant d'aller à la page de préparation d'une nouvelle paie ;
- ✓ Liste des traitements listant les traitements déjà effectués ;
- ✓ Bulletins de paie listant les bulletins de paie disponibles par traitement.

La page est accessible à l'utilisateur « entreprise » Solde et à tout acteur « cabinet » ayant ce profil.



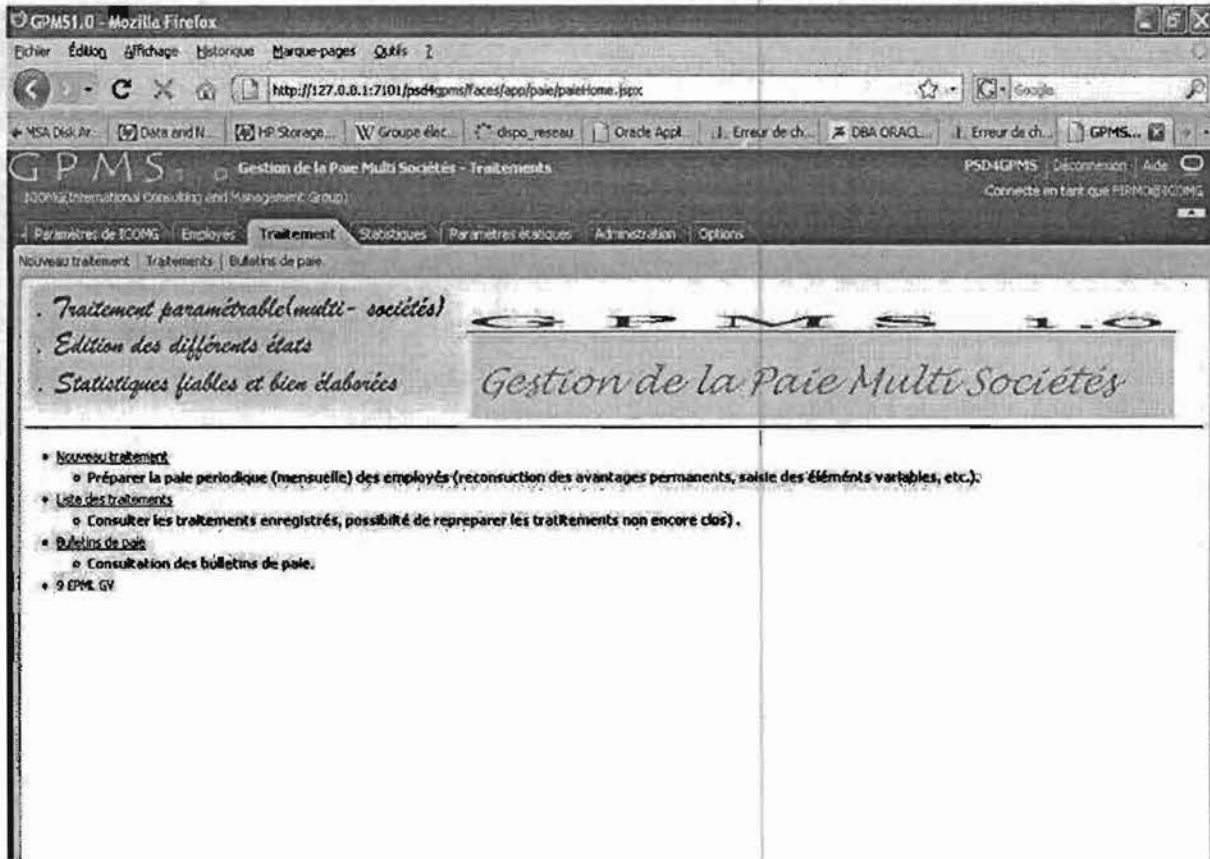


Figure 5.12 : page d'accueil pour effectuer des traitements

### 5.4.8 Liste des traitements

Cette page, accessible à l'utilisateur « entreprise » solde et à l'acteur « cabinet » correspondant, liste, comme la figure 5.13, les différents traitements de salaire effectués par l'entreprise connectée. Les traitements non encore clos peuvent être ré-ouverts pour des paramétrages éventuels.

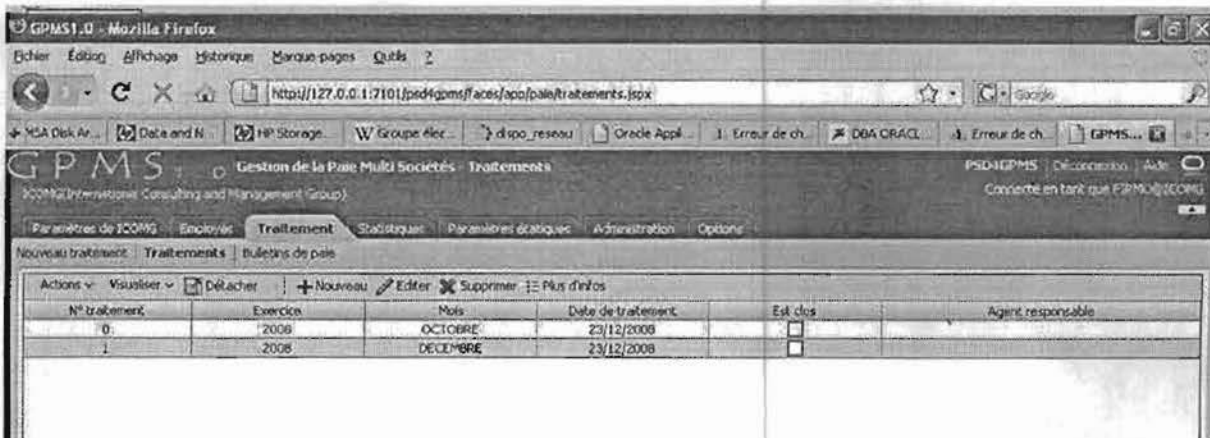


Figure 5.13 : page listant les traitements de salaire effectués de l'entreprise connectée

### 5.4.9 Nouveau traitement

Cette page permet d'effectuer un nouveau traitement. Elle est accessible à l'utilisateur « entreprise » Solde et à tout acteur « cabinet » correspondant. Elle n'est pas complètement achevée car elle ne prend en compte que les avantages permanents. La gestion des prêts, des heures supplémentaires et des éléments variables est en train d'y être intégrée.

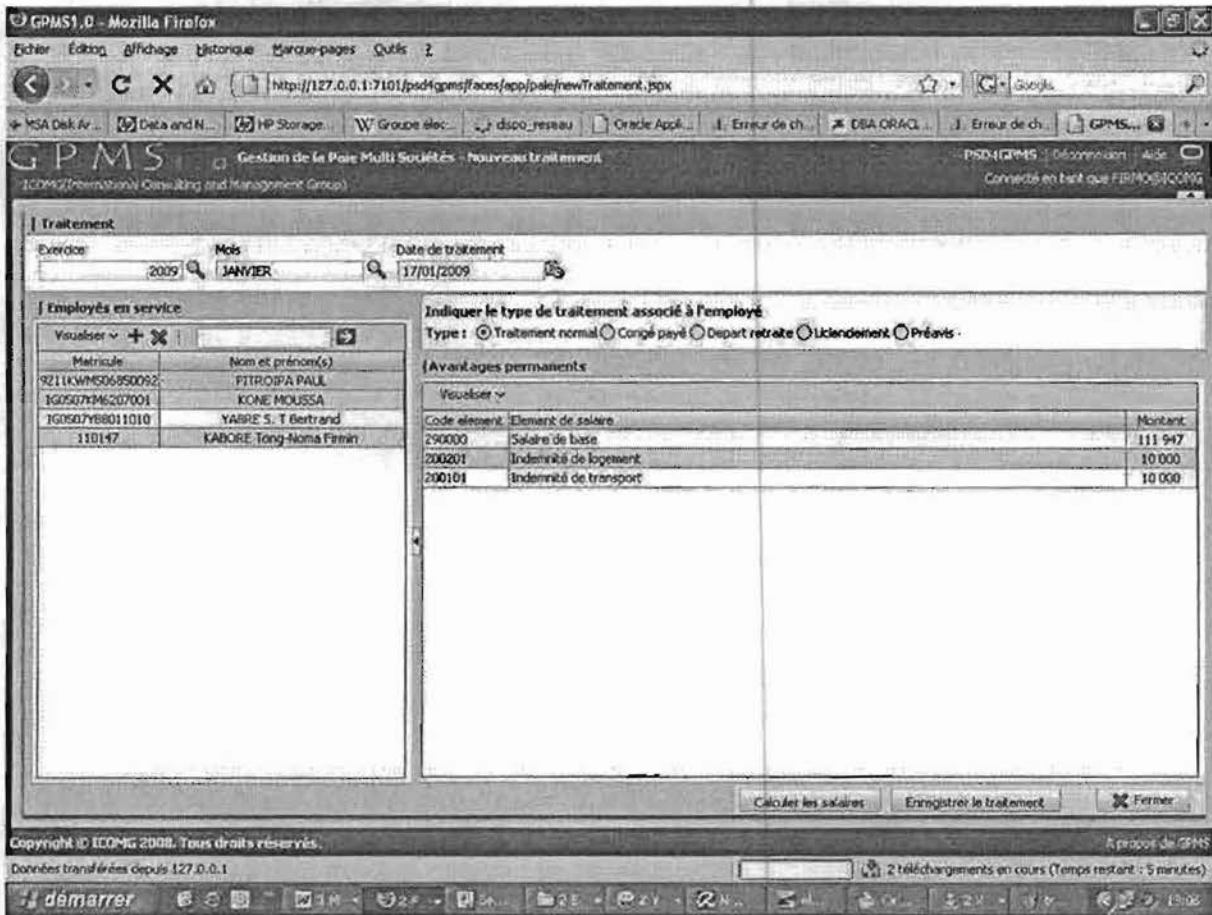


Figure 5.14 : page pour un nouveau traitement de salaire

### 5.4.10 Bulletins de paie

Cette page (figure 5.15) liste les bulletins de paie pour un traitement donné. Elle est accessible aux utilisateurs « entreprise » RH et Solde et aux acteurs « entreprise » ayant ces profils.

The screenshot shows the 'Bulletins de paie' page in the GPMS1.0 application. The page is divided into several sections:

- Navigation:** Includes 'Paramètres de ICOMG', 'Employés', 'Traitement', 'Statistiques', 'Paramètres étatiques', 'Administration', and 'Options'.
- Bulletins de Paie Table:**

| Traitement    | Date traitement |
|---------------|-----------------|
| OCTOBRE 2008  | 23/12/2008      |
| DECEMBRE 2008 | 23/12/2008      |
- Employés Table:**

| Matricule        | Nom et prénom(s)        |
|------------------|-------------------------|
| 110147           | KABORE Tong-Nama Firmin |
| 9211KWM506950092 | PITROIPA PAUL           |
| IG0507KM5207001  | KONE MOUSSA             |
| IG0507Y68011010  | YABRE S. T Bertrand     |
- Employee Details:**

Employé : MOUSSA KONE (IG0507KM5207001)  
 Type traitement : normal  
 Nombre de charge : 5
- Salary Breakdown Table:**

| Code Élément | Élément de salaire             | Retras  | Avoir   |
|--------------|--------------------------------|---------|---------|
| 290000       | Salaires de base               |         | 700 000 |
| 200101       | Indemnité de transport         |         | 75 000  |
| 200201       | Indemnité de logement          |         | 175 000 |
| 200301       | Indemnité de fonction/sujetion |         | 75 000  |
| 690001       | CNSS                           | 33 000  |         |
| 690002       | ILTS                           | 179 676 |         |
- Summary Totals:**

|              |                |             |             |
|--------------|----------------|-------------|-------------|
| SALAIRE BRUT | BASE IMPOSABLE | SALAIRE NET | NET A PAYER |
| 1 025 000    | 787 000        | 812 324     | 812 324     |

Figure 5.15 : page de visualisation des bulletins de paie





# 6

## Déploiement et sécurité

La plateforme ainsi conçue devant être déployée et accessible sur Internet, la sécurité se doit d'être au cœur de nos préoccupations. Cette section présente l'architecture de déploiement et les ressources matérielles et logicielles nécessaires à la mise en place effective et sécurisée de la plateforme.

### 6.1 ARCHITECTURE PRELIMINAIRE DE DEPLOIEMENT

Conformément aux exigences du cahier des charges, la plateforme PSD4GPMS sera déployée sur toutes ses composantes en environnement Oracle. Pour ce faire, il a été retenu en accord avec le comité de pilotage, le serveur d'application Oracle Weblogic Server 10.3 comme serveur d'application et Oracle Database 10G comme serveur de base de données.

#### 6.1.1 Oracle Weblogic Server 10.3

Un serveur d'applications est un conteneur d'applications permettant à des utilisateurs connectés en réseau d'accéder à tout ou partie d'une application hébergée par celui-ci en exemplaire unique. Le serveur d'application est une infrastructure incontournable dans toute application Java EE. La plateforme Java EE retenue nous impose l'utilisation d'un serveur d'application compatible Java EE. La majorité des serveurs d'applications le sont, ce qui donne plus de flexibilité pour le déploiement et la maintenance de la plateforme. La plateforme PSD4GPMS sera déployée niveau applicatif sur le serveur d'application **Oracle Weblogic Server 10G Release 3 (10.3)**. Oracle Weblogic Server 10.3 est un serveur d'application supportant le clustering, édité par Oracle depuis janvier 2008. Il est le leader mondial des serveurs d'application Java EE.

### 6.1.2 Oracle Database 10G

La plateforme PSD4GPMS exploitera une base de données relationnelle **Oracle Database 10G<sup>1</sup> Enterprise Edition**. Oracle Database 10G est conçue pour le « Grid Computing » (calcul distribué) qui est une technologie permettant de regrouper les serveurs de telle sorte qu'ils ne forment qu'une seule et même entité plus puissante. Oracle Database 10G est un Système de Gestion de Base de Données (SGBD) qui supporte aussi bien le modèle relationnel pur que celui orienté objet. Il a notamment les atouts suivants :

- ✓ Il est le plus puissant du marché et évolue très rapidement ;
- ✓ Il supporte le clustering ;
- ✓ Il est très robuste et résolument orienté Internet ;
- ✓ Il prend en charge la machine virtuelle Java (Java Virtual Machine) ;
- ✓ Il est très portable ;
- ✓ Il gère l'interopérabilité aussi bien entre les bases de données Oracle existantes, qu'entre des bases de données non Oracle (SQL Server, Informix, DB2 ...) ;
- ✓ Il gère efficacement les données et respecte bien les standards ;
- ✓ Il a une édition « express » téléchargeable gratuitement sur Internet ;
- ✓ Il est plébiscité par les entreprises comme la base de données la plus sécurisée du marché.

### 6.1.3 Diagramme préliminaire de déploiement

La plateforme doit être accessible sur Internet pour permettre à des entreprises tierces de s'y abonner pour traiter leur paie en ligne. Son déploiement et son exploitation exigent donc au minimum trois (03) postes de travail comme le montre la figure 6.1:

---

<sup>1</sup> Le « G » pour Grid Computing (l'utilisation de plusieurs machines pour exécuter une même tâche)

- ✓ Une machine sur laquelle sera installé le serveur d'applications **Oracle Weblogic Server 10.3** ;
- ✓ Une machine sur laquelle sera installé le serveur **Oracle Database 10G** ;
- ✓ Un poste de travail disposant d'un client http pour se connecter à la plateforme.

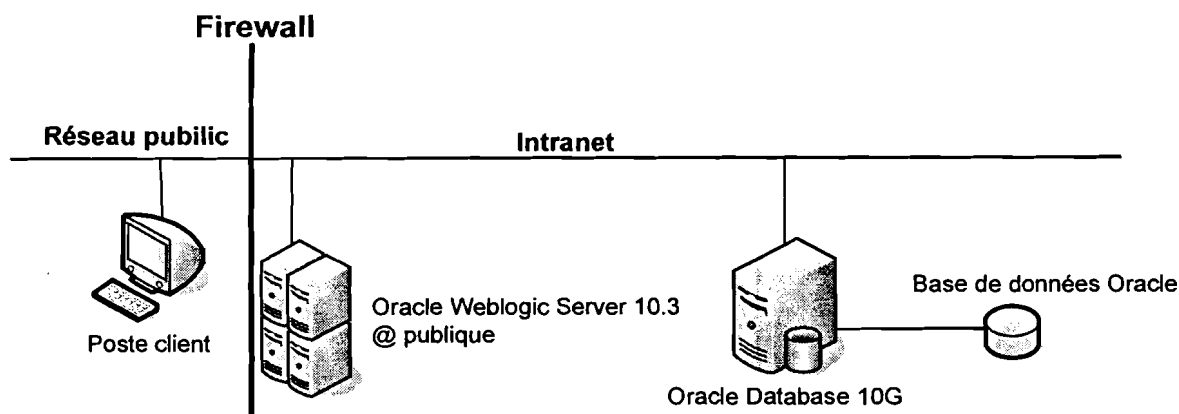


Figure 6.1 : diagramme préliminaire de déploiement

Les recommandations techniques minimales pour ces postes de travail sont consignées dans le tableau 6.1.

Tableau 6.1 : configuration minimale des postes de la plateforme

| Machine                       | Caractéristiques   |
|-------------------------------|--|
| Client http                   | <b>Navigateurs :</b><br>✓ Internet explorer 7 ou version supérieure ;<br>✓ Mozilla Firefox 2.8 ou version supérieure.<br><b>Java Script : Activé</b>   |
| Oracle Database 10G R2        | <b>Système d'exploitation :</b> Oracle Enterprise Linux 5 <sup>1</sup><br><b>Processeur :</b> Pentium IV, fréquence supérieure à 3 GHz<br><b>Disque dur :</b> capacité supérieure à 80 To<br><b>Mémoire RAM :</b> capacité supérieure à 2 Go |
| Oracle Weblogic Server 10G R3 | <b>Système d'exploitation :</b> Oracle Enterprise Linux 5<br><b>Processeur :</b> Pentium IV, fréquence supérieure à 3 GHz<br><b>Disque dur :</b> capacité supérieure à 80 Go<br><b>Mémoire RAM :</b> capacité supérieure à 2 Giga octets     |

<sup>1</sup> **Oracle Enterprise Linux 5 :** Redhat Enterprise Linux avec les paquetages et les paramètres nécessaires à l'installation des logiciels Oracle (base de données, serveur d'applications). Il a un coût de support inférieur à celui de Redhat Enterprise Linux.

La plateforme ainsi déployée court les principaux risques suivants :

- ✓ Toute panne matérielle ou logicielle d'un des serveurs (serveur d'application ou serveur de base de données) entraîne l'indisponibilité de la plateforme ;
- ✓ Les données échangées entre les entreprises souscrites et la plateforme (mot de passe, salaire des employés, etc.) sont en clair et peuvent donc être interceptées, modifiées ou même détruites par n'importe quel pirate ;
- ✓ Les mots de passe interceptés peuvent être utilisés par n'importe quel pirate pour se connecter à la base de données pour faire des malversations ;
- ✓ L'intranet n'étant pas aussi protégé, il est aussi accessible de l'extérieur et court donc tous les risques sus cités ;

Cette liste des risques n'étant pas exhaustive, il faut établir un plan de sécurité pour réduire au maximum les risques encourus par la plateforme.

## 6.2 PROCEDURES DE SECURITE

La sécurité informatique, d'une manière générale, consiste à assurer que les ressources matérielles et logicielles d'une organisation sont uniquement utilisées dans le cadre prévu. La sécurité peut s'évaluer suivant plusieurs critères dont les principaux sont :

- ✓ **La disponibilité** garantissant l'accessibilité de l'information au moment voulu aux personnes autorisées ;
- ✓ **L'intégrité** garantissant l'exactitude l'information ;
- ✓ **La confidentialité** qui restreint l'information aux personnes autorisées.

La mise en place d'une politique de sécurité consiste à passer en revue tous ces critères.

### 6.2.1 Conception d'une solution haute disponibilité

En informatique, il est très difficile de fournir un service 24H/24. Il y a toujours des perturbations suite notamment à des pannes matérielles ou logicielles. La conception d'une

solution haute disponibilité consiste donc à réduire au maximum les temps d'arrêt et de permettre une reprise immédiate après panne. Pour ce faire, il faut :

- ✓ Etudier et prévoir toutes les causes possibles d'arrêt (arrêts prévus et arrêts imprévus) de la plateforme ;
- ✓ Trouver une solution pour chaque cause énumérée.

Les causes possibles d'arrêt d'un système sont très nombreuses et variées. Il faut donc se concentrer sur les plus importantes et les plus fréquentes afin de les résoudre rapidement. On peut recenser les causes suivantes :

- ✓ La coupure d'alimentation électrique ou la panne du boîtier d'alimentation électrique;
- ✓ Les pannes de disque dur ;
- ✓ Destruction des données due aux mauvaises manipulations des utilisateurs ;
- ✓ Panne du réseau interne ;
- ✓ Panne du système d'exploitation ;
- ✓ Destruction ou disparition d'un serveur ;
- ✓ Déstabilisation du système par les virus informatiques ;
- ✓ Destruction du site de production (sinistre).

La technique généralement utilisée pour garantir la disponibilité d'un système est d'identifier les points de défaillance unique et de les rendre redondants. Cependant, il a été convenu avec le comité de pilotage, de l'acquisition de serveurs *HP Proliant* pour le déploiement de la plateforme (applicatif et base de données). La gamme de serveurs *HP Proliant* étant conçue avec une redondance matérielle au niveau de ces points (port réseau, boîtier d'alimentation électrique, processeur, etc.), nous assurons ainsi pour notre solution la haute disponibilité.

### **6.2.1.1 Redondance au niveau de l'alimentation électrique**

La continuité d'une alimentation électrique de parfaite qualité est essentielle pour une solution haute disponibilité. Les constructeurs de serveurs y ont suffisamment pensé en rendant redondant le boîtier d'alimentation électrique. En effet les serveurs *HP Proliant*

disposent d'au moins deux boîtiers d'alimentation (avec un système de commutation automatique) qui peuvent être branchés directement sur des sources d'alimentation différentes. Pour assurer donc la continuité de l'alimentation électrique, il a été convenu avec le comité de pilotage de l'acquisition d'un onduleur grande autonomie et d'un groupe électrogène. Ainsi, seul l'onduleur est en marche pendant la fourniture normale du courant. Quand une coupure intervient, le groupe électrogène est immédiatement démarré pour permettre le basculement automatique à l'épuisement de l'onduleur.

### 6.2.1.2 Solution aux problèmes de disques durs

Un défi aussi important rencontré dans la conception d'une solution haute disponibilité est de lutter contre les pannes de disques durs. En effet, le disque dur étant l'élément sensible de tout serveur, s'il tombe en rade, l'information stockée est perdue et aucun traitement ne peut se poursuivre. Pour remédier aux problèmes de disque dur et assurer la disponibilité de l'information, les concepteurs ont très souvent recours à la technologie **RAID** (Redundant Array of Inexpensive Disks). Le principe du RAID repose sur l'idée de combiner plusieurs disques pour former une unité de stockage appelée grappe. Il a principalement pour but d'étendre la capacité de stockage, d'introduire une notion de redondance (pour autoriser une plus grande sécurisation des données) et améliorer les performances.

Les principes de base les plus couramment utilisés dans la technologie RAID sont :

- ✓ **Le striping** qui permet de dispatcher les données sur l'ensemble des disques de la grappe. Ce système n'est pas sécurisé: si un disque de la grappe tombe en panne, c'est l'ensemble des données de la grappe qui est perdu. Le but principal du striping est de permettre une vitesse de transfert élevée et d'outrepasser la limite de la capacité des disques durs imposée par la technologie.
- ✓ **Le mirroring (duplication)** qui consiste à dupliquer les informations et à les copier sur chacun des disques de la grappe. Ce système est donc sécurisé: si un disque de la grappe tombe en panne, les informations resteront sauvegardées sur les disques encore opérationnels. Ce système est qualifié de redondant.

Les disques assemblés selon la technologie RAID peuvent être utilisés de différentes façons, appelés niveau RAID. Une solution RAID peut être matérielle ou logicielle. Pour la

plateforme PSD4GPMS, il a été convenu d'implémenter le RAID 10 (1+ 0) matériel au niveau du stockage de la base de données. Le RAID 10 consiste à faire du mirroring sur du striping. Cette combinaison permet d'avoir une vitesse de lecture accrue sur des données redondantes.

### **6.2.1.3 Disponibilité du réseau interne (Intranet)**

Le fonctionnement de la plateforme nécessite un réseau interne opérationnel. Pour assurer une haute disponibilité du réseau interne, nous allons avoir recours à une technologie appelée « bonding channel ». Le bonding channel est une solution de redondance de cartes réseau permettant de configurer une seule interface virtuelle pour mapper plusieurs interfaces physiques réseaux. De cette façon, si la carte réseau active tombe en rade, l'une des autres cartes encore opérationnelles prend immédiatement le relais et devient la carte active.

### **6.2.1.4 Politique de sauvegarde et de restauration des données**

Quelle que soit la qualité des moyens de défense mis en œuvre (physique ou logique), les données peuvent être altérées sciemment ou accidentellement. Dans une solution haute disponibilité, les données doivent être disponibles à tout moment lorsqu'on en a besoin, et doivent être conservées afin de pouvoir être récupérées le moment voulu. Pour ce faire, il a été adopté la procédure de sauvegarde suivante :

- ✓ Une sauvegarde complète du système en fin de semaine ;
- ✓ Une sauvegarde différentielle (incrémentale) tous les soirs ;
- ✓ Une assurance que les sauvegardes sont bien effectuées et conservation de ces sauvegardes hors du site de production.

Pour les entreprises en fin de contrat, une exportation des données, en utilisant l'outil export/import d'Oracle (Data Pump) sera utilisée afin de leur restituer leurs données.

### **6.2.1.5 Protection contre les virus informatiques**

Les plus grandes sources de danger d'un système informatique sont les virus informatiques. Ces petits programmes malveillants s'auto reproduisent et se répandent grâce aux réseaux et aux supports amovibles comme les clés USB, les CDROM, etc. Cela peut avoir pour cause la

déstabilisation du système et la destruction des données. Ce risque augmente avec la connexion Internet. La technique pour la protection contre les virus informatiques est de recourir à l'utilisation des antivirus. Pour ce faire, nous préconisons l'installation d'un antivirus récent et régulièrement mis à jour sur l'ensemble des postes de la plateforme.

#### 6.2.1.6 Clustering des serveurs

Les techniques sus citées s'appliquent essentiellement sur les composants des serveurs. Cependant, les serveurs eux-mêmes peuvent disparaître ou être détruits. Il faut donc songer à les rendre aussi redondants. La technologie en vogue qui permet de rendre des machines redondantes est le clustering.

Un cluster est un groupe de serveurs (appelés nœuds) indépendants qui coopèrent pour fonctionner comme un système unique :

- ✓ Les nœuds interconnectés agissent comme un serveur unique ;
- ✓ Le logiciel de gestion du cluster masque sa structure ;
- ✓ Les disques sont accessibles en lecture et en écriture par tous les nœuds du cluster ;
- ✓ Les nœuds sont généralement dotés du même système d'exploitation.

Un cluster se décrit donc comme un ensemble logique de serveurs qui garantissent une haute disponibilité des ressources et une répartition des charges de traitement. Il fournit les avantages suivants :

- ✓ **La haute disponibilité** puisque qu'il faut que tous les nœuds du cluster tombent en panne pour rendre le système indisponible ;
- ✓ **Performance** étant donné que la performance de la plateforme est taillée sur mesure ;
- ✓ **Evolutivité** car on peut ajouter ou retirer des nœuds selon le besoin.

##### 6.2.1.6.1 Clustering des serveurs de base de données

Pour qu'une application puisse fonctionner correctement en environnement cluster, il faut qu'elle supporte le clustering. Oracle Real Application Cluster 10G (Oracle RAC) est une



option de Oracle Database 10G permettant aux bases de données Oracle de fonctionner en environnement cluster.

Oracle RAC permet donc à plusieurs machines de travailler ensemble sur les mêmes données en parallèle de façon à réduire significativement le temps de traitement des requêtes et de garantir la disponibilité des données.

Pour la mise en œuvre d'Oracle RAC 10G, il a été convenu avec le comité de pilotage de l'acquisition de :

- ✓ Une baie partagée *HP StorageWorks 1000 Modular Smart Array* ;
- ✓ Trois (03) serveurs *HP Proliant DL 585 G5*.

La figure 6.2 montre l'architecture RAC (Real Application Cluster) avec les composants sus-cités.

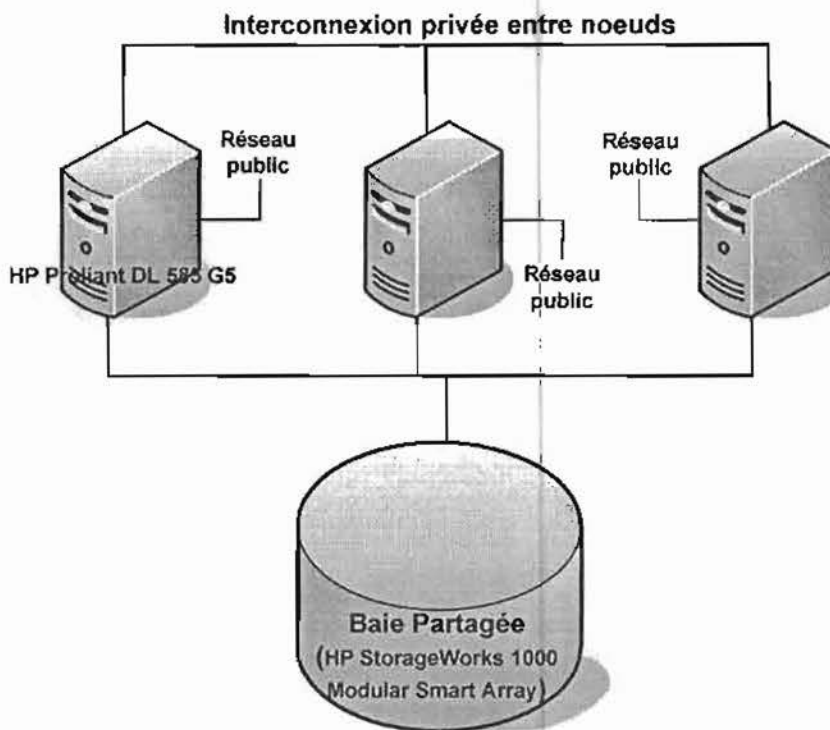


Figure 6.2: Architecture RAC (Real Application cluster)

### 6.2.1.6.2 Clustering du serveur d'application

Oracle Weblogic Server supporte deux types de clustering :

- ✓ « **L'Application FailOver** » correspondant au clustering actif/passif. Typiquement, quand une composante du cluster, exécutant une certaine tâche tombe en panne, une copie de cette composante poursuit et achève l'exécution de la tâche ;
- ✓ Le « **Load Balancing** » où tous les nœuds coopèrent ensemble pour l'exécution d'une même tâche. Ce type de cluster nécessite l'intervention d'un composant matériel ou logiciel appelé load balancer (équilibreur de charge) qui va fédérer la charge entrante entre les différents nœuds du cluster. C'est le clustering dans son sens le plus complet.

Pour offrir une grande disponibilité à la plateforme PSD4GPMS, nous proposons le clustering en mode load balancing. Pour ce faire, il a été convenu avec le comité de pilotage de l'acquisition d'une solution load balancing matérielle et de trois (03) serveurs *HP Proliant HP ML*. La figure 6.3 présente l'architecture de base recommandée par Oracle de Weblogic Server Cluster.

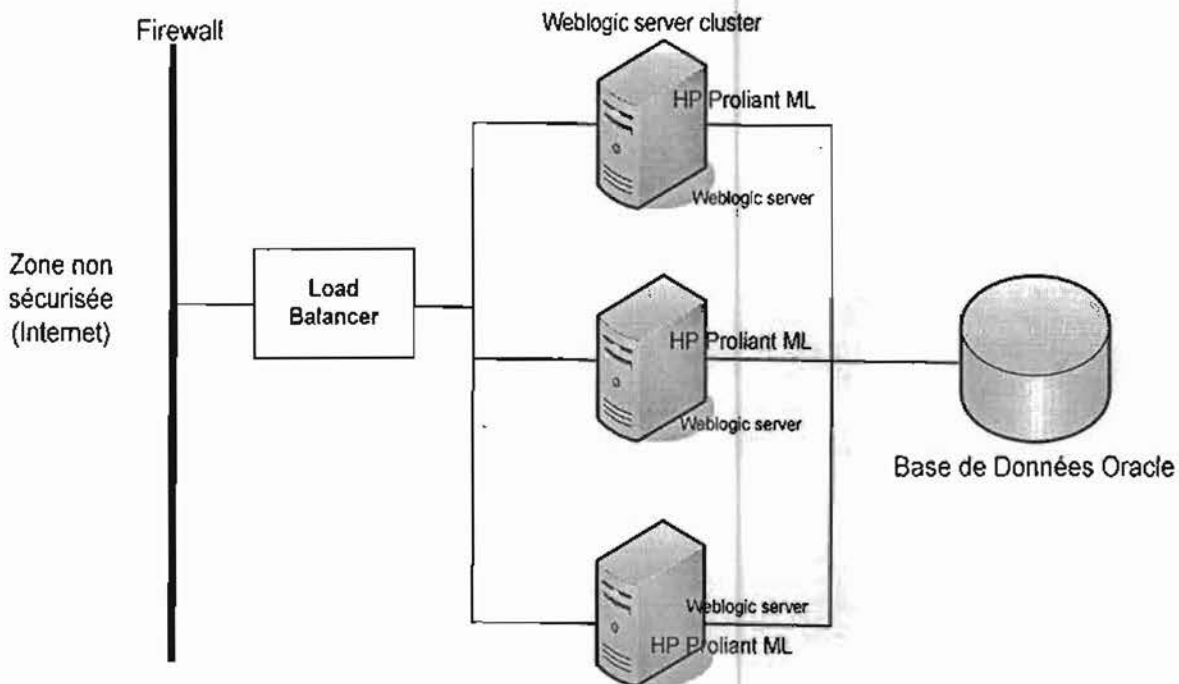


Figure 6.3 : Weblogic Server Cluster (architecture de base)

### 6.2.1.7 Clonage du site de production

Si toutefois le site de production venait à disparaître à cause d'un sinistre quelconque, cela entraînera la destruction des clusters et toutes les données non sauvegardées. Il faut donc cloner le site de production.

L'outil Oracle Data Guard facilite le clonage des sites de production et assure un basculement automatique vers le site cloné en cas de problème au niveau du site de production.

Oracle Data Guard et Oracle Real Application Cluster constituent le fondement de l'architecture de disponibilité maximale. Cette architecture est la plus complète pour réduire au maximum les périodes d'arrêt. Cependant, pour des raisons de coût, il a été convenu avec le comité de pilotage de ne pas s'en servir.

### 6.2.2 Intégrité et confidentialité des données

Pour assurer l'intégrité et la confidentialité de l'information, nous adopterons les techniques suivantes :

- ✓ La segmentation du réseau en utilisant la technique des DMZ (Zone démilitarisée) ;
- ✓ La mise en œuvre du protocole https (HTTP + SSL) ;
- ✓ La mise en œuvre d'Oracle Virtual Private Database (VPD) ;
- ✓ La sensibilisation des utilisateurs ;
- ✓ Les techniques de sécurité dans le code.

#### 6.2.2.1 Segmentation du réseau

La segmentation du réseau a pour but de réduire au maximum, par la technique des DMZ, les ressources accessibles de l'extérieur. La DMZ est une collection logique de matérielles et de services accessibles depuis l'extérieur. Pour le déploiement de la plateforme, il a été convenu de segmenter le réseau suivant les trois (03) zones suivantes comme le montre la figure 6.4 :

- ✓ Le réseau interne (Intranet) ;
- ✓ La zone démilitarisée (DMZ) ;
- ✓ Le réseau externe (Internet).

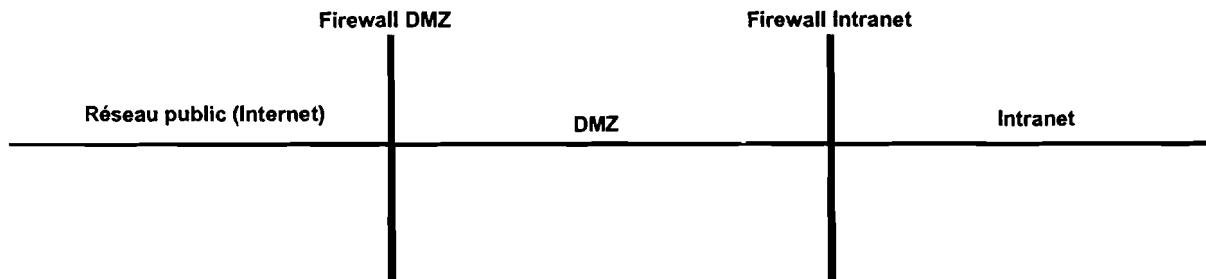


Figure 6.4 : segmentation du réseau

Le tableau 6.2 donne la répartition des machines de la plateforme sur les différentes zones.

Tableau 6.2 : Répartition des machines dans les réseaux

| Zone                     | Machines                                    |
|--------------------------|---|
| Intranet                 | RAC d'Oracle Database 10G R2                |
| Zone démilitarisée (DMZ) | Cluster d'Oracle Weblogic Server 10G R3     |
| Réseau public (Internet) | Client (disposant d'un navigateur Internet) |

La segmentation du réseau, comme le montre la figure 6.4, nécessite l'intervention de deux firewalls<sup>1</sup> (pare-feu) :

- ✓ Le **Firewall Intranet** placé à l'entrée de l'Intranet pour n'accepter que les paquets provenant du serveur Weblogic Server installé dans la zone démilitarisée ;
- ✓ **Firewall DMZ** acceptant les connexions venant de l'extérieur. Cependant le port et la destination des paquets doivent être correctement configurés.

### 6.2.2.2 Mise en œuvre de SSL

La segmentation du réseau ainsi faite, permet de réduire les ressources accessibles de l'extérieur et de restreindre l'accès à ces ressources. Cependant la communication entre le client et la plateforme reste en clair sans aucun chiffrement. La plateforme est donc toujours vulnérable. Pour pallier à cela, nous implémentons le protocole HTTP/SSL pour toute communication entre la plateforme et l'extérieur (réseau public).

<sup>1</sup> Un firewall est un dispositif matériel ou logiciel placé à l'entrée d'un réseau et qui va filtrer tous les paquets entrants et sortants de ce réseau.

Secure Socket Layer (SSL) est un système qui permet d'échanger des informations entre deux ordinateurs de façon sécurisée. SSL assure notamment les trois propriétés suivantes :

- ✓ **La confidentialité** qui rend impossible l'espionnage de l'information échangée ;
- ✓ **L'intégrité** qui rend impossible le truquage de l'information échangée ;
- ✓ **L'authenticité** qui permet de s'assurer de l'identité du programme, de la personne ou de l'entreprise avec laquelle on communique.

SSL a été développé par Netscape et est intégré depuis 1994 pratiquement dans tous les navigateurs. Il est un complément à TCP/IP et permet notamment de sécuriser n'importe quel programme ou protocole utilisant TCP/IP.

La mise en œuvre de SSL nécessite une PKI (Public Key Infrastructure), les PKI étant gérées par des sociétés externes chargées de vérifier la validité des identités (certificats) des serveurs. La liste des PKI (les plus connus) est incluse dans les navigateurs. Pour la plateforme PSD4GPMS, nous allons utiliser Oracle Certificate Authority, livré gratuitement avec le kit de Weblogic Server 10G.3.

### **6.2.2.3 Mise en œuvre de Oracle Virtual Private Database**

La base de données Oracle Database 10G R2 dispose d'une option intéressante qu'on peut utiliser pour renforcer la sécurité au niveau de la base de données. Il s'agit de l'option VPD (Virtual Private Database). Oracle Virtual Private Database est une technique d'ajout dynamique de clause WHERE aux ordres SQL (Structured Query Language) sur les objets de la base (tables) permettant ainsi d'assurer une sécurité niveau ligne. Nous avons donc activé cette option afin de créer une « base de données virtuelle » pour chaque entreprise souscrite.

### **6.2.2.4 Sensibilisation des utilisateurs**

Devant l'avancée technologique permettant de verrouiller correctement les systèmes d'information, les attaquants n'ont plus recours qu'à la naïveté de l'homme. L'homme constitue donc aujourd'hui le maillon le plus faible en termes de sécurité de tout système d'information. La mise en place d'une politique de sécurité forte passe donc par la sensibilisation des utilisateurs du système d'information. La sensibilisation des utilisateurs a pour objectif :

- ✓ De développer une « culture sécurité » au sein de l'entreprise afin qu'ils ne soient plus le maillon faible du système ;
- ✓ De faire appel aux utilisateurs pour renforcer la sécurité du système d'information.

Pour la plateforme PSD4GPMS, les utilisateurs seront sensibilisés aux techniques suivantes :

- ✓ Mots de passe suffisamment robustes (éviter les mots de passe simples et évidents) ;
- ✓ Déconnection immédiate après avoir fini de se servir de la plateforme ;
- ✓ Avertissement contre la technique du social engineering ;

### 6.2.2.5 Gestion des profils utilisateurs

La confidentialité des données passe aussi par une gestion correcte des profils d'utilisateurs. Il faut canaliser l'information et situer la responsabilité des utilisateurs. Nous avons donc défini des rôles pour restreindre la responsabilité de chaque utilisateur dans le système. Ainsi, chaque utilisateur n'accèdera qu'aux données auxquelles il a droit et n'effectuera que les traitements qui lui sont autorisés. Un journal d'historisation permettra de retracer les différentes connexions afin d'identifier les responsables de chaque opération.

### 6.2.2.6 Authentification obligatoire

L'étape de connexion est obligatoire pour tout utilisateur de la plateforme. Avant d'avoir accès à toute fonctionnalité du système, les utilisateurs doivent d'abord s'authentifier à l'aide de leur nom d'utilisateur et leur mot de passe. Cette obligation de connexion est prise en charge par le servlet **oracle.security.jps.ee.http.JpsFilter**.

Devant cette obligation de connexion, les mots de passe restent donc les remparts utilisés pour accéder à l'information. De ce fait, les mots de passe sont chiffrés avec un algorithme irréversible avant leur inscription sur disque. Lorsqu'un utilisateur se connecte, le mot de passe qu'il fournit est immédiatement chiffré et comparé à la valeur stockée. Et pour plus de sécurité, les mots de passe seront régulièrement modifiés.

### 6.2.3 Conclusion

La sécurité d'un système n'est jamais acquise définitivement ; elle se vit au quotidien. Elle ne repose pas uniquement sur les outils de sécurité mais également sur une stratégie, une organisation et des procédures cohérentes. Cela nécessite une structure de gestion adéquate dont la mission est de gérer, mettre en place, valider, contrôler et faire comprendre à l'ensemble des acteurs de la structure, l'importance de la sécurité. Cela se fera par l'affichage de notes d'information et des travaux d'entretien sur la sécurité.

### 6.4 ARCHITECTURE FINALE DE DEPLOIEMENT DE PSD4GPMS

Au regard des solutions de sécurité que nous avons proposées, voici l'architecture finale de déploiement de la plateforme.

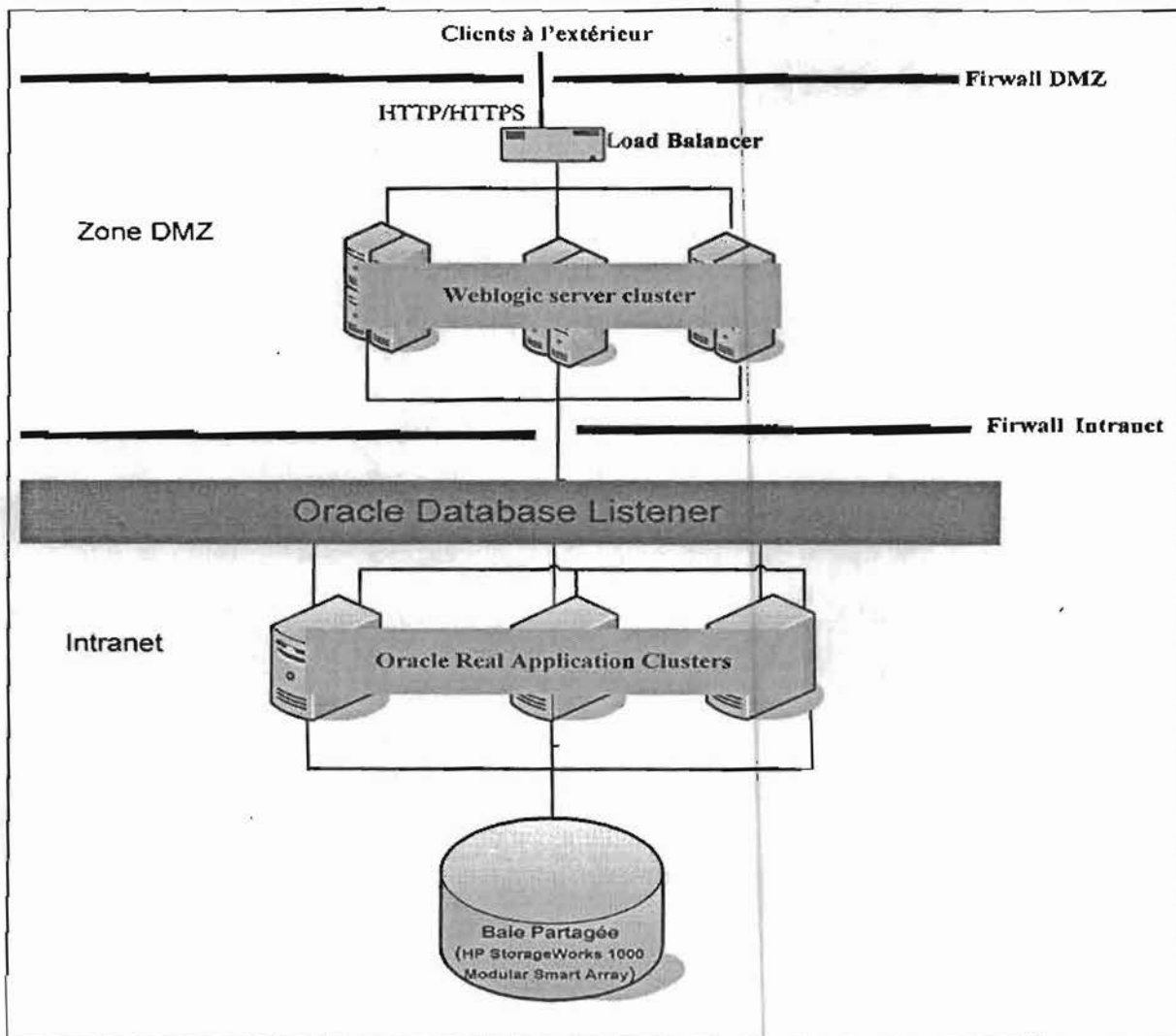


Figure 6.5 : architecture finale de déploiement de la plateforme

# Conclusion

Le logiciel que nous avons décrit tout au long de ce document est le fruit d'un stage pratique de conception de quatre (04) mois au sein du cabinet ICOMG. Ce stage s'est déroulé selon les grandes lignes suivantes :

- ✓ Une phase d'insertion qui nous a permis de nous intégrer au sein du cabinet et d'appréhender les contours de notre projet ;
- ✓ Une phase d'analyse conceptuelle qui nous a permis de dégager les besoins fonctionnels et opérationnels que doit satisfaire la plateforme. Elle nous a conduits notamment à la construction des modèles fonctionnelle, statique et dynamique de la plateforme ;
- ✓ Et enfin, une phase de conception détaillée et de réalisation qui a aboutit à la plateforme opérationnelle PSD4GPMS.

En termes d'atouts, ce stage nous a permis :

- ✓ D'approfondir nos connaissances théoriques vues en classe et de les mettre en pratique ;
- ✓ D'appréhender de nouveaux concepts ;
- ✓ Et de s'imprégner des réalités du monde professionnel.

En termes de perspectives, nous comptons à moyen ou à long terme :

- ✓ Finaliser complètement la plateforme PSD4GPMS ;



- ✓ Etendre progressivement la plateforme aux autres pays du monde ;
- ✓ Intégrer le traitement indiciaire des salaires ;
- ✓ Externaliser les règles de calcul des éléments de salaires ;
- ✓ Etendre PSD4GPMS à d'autres domaines comme la gestion des ressources humaines.

PSD4GPMS doit être vue comme un ensemble de services de traitement de paie en ligne payables au mois ou à l'usage. Elle s'inscrit donc dans une nouvelle tendance de l'informatique connue sous le sigle SaaS (Software as a Service).

## Bibliographie

- [1] ROQUES Pascal, VALLEE Franck. *ML 2 en action – De l'analyse de besoins à la conception*. 4ème édition. Eyrolles, 2007, 382 p. (Architecte logiciel).
- [2] ROQUES Pascal. *UML 2 par la pratique*. 6ème édition. Eyrolles, 2008, 368 p. (Noire).
- [3] SRIGANESH Patel Rima, BROSE Gerald, SILVERMAN Micah. *Mastering Enterprise Java Beans 3.0*. 1<sup>ère</sup> édition. Wiley, 2006, 686 p.
- [4] SUPINFO Laboratoire des technologies SUN. *EJB 3 – Des concepts à l'écriture du code. Guide du développeur*. 2ème édition. Dunod, 2008, 368 p. (InfoPro)
- [5] ORACLE CORPORATION.  
*Oracle Fusion Middleware : Web User Interface Developer's Guide for Oracle Application Development Framework 11g Release 1, 2008* [en ligne].  
Disponible sur : <<http://www.oracle.com/technology/documentation/jdev.html>>  
(Consulté le 13/08/2008)
- [6] ORACLE CORPORATION  
*Oracle Fusion Middleware : Fusion Developer's Guide for Oracle Application Development Framework 11g Release 1, 2008* [en ligne].  
Disponible sur : <<http://www.oracle.com/technology/documentation/jdev.html>>  
(Consulté le 15/08/2008)

- [7] ORACLE CORPORATION  
*Oracle Fusion Middleware : Developer's Guide for Oracle TopLink 11g Release 1, 2008* [en ligne].  
Disponible sur : <<http://www.oracle.com/technology/documentation/jdev.html>>  
(Consulté le 18/08/2008)
- [8] JENDROCK Eric, BALL Jennifer, GARSON Debbie et al.  
*The Java EE 5 Tutorial For Sun Java System Application Server 9.1*, [en ligne].  
Disponible sur : <<http://java.sun.com/javaee/5/docs/tutorial/doc/>>  
(Consulté le 20/08/2008)
- [9] ORACLE CORPORATION  
*Oracle Weblogic Server : Using clusters 10G Release 3(10.3), 2008* [en ligne].  
Disponible sur : <<http://e-docs.bea.com/wls/docs103/pdf.html>>  
(Consulté le 18/08/2008)
- [10] ORACLE CORPORATION  
*Oracle Fusion Middleware : Oracle Application Development Framework*, [en ligne]  
Disponible sur : <<http://www.oracle.com/technology/products/adf/index.html>>  
(Consulté le 20/08/2008)
- [11] Sun Microsystems  
*Java Platform, Enterprise Edition (Java EE) : JavaServer Faces Technology*,  
[en ligne]. Disponible sur : <<http://java.sun.com/javaee/jaserverfaces/>>  
(Consulté le 20/08/2008)