



Ecole **N**ationale **S**upérieure d'**I**nformatique
et d'**A**analyse des **S**ystèmes



Sofrecom **S**ervices **M**aroc

Mémoire de Projet de Fin d'Études

Pour l'Obtention du Titre

D'Ingénieur d'État en Informatique

Option

Réseaux de Communications

Sujet

Outil de prise de commande Télécom

Soutenu par :

Mlle Imane BOUROUS
M. Mohammed SOUIDI

Sous la direction de :

Mlle Imane NEJJAR (Sofrecom)
M. Ahmed HABBANI (ENSIAS)
M. Mustapha EL FEDDI (Sofrecom)
M. Younes SOUTEH (Sofrecom)

Année Universitaire 2008-2009

Dédicaces

A la mémoire de mon grand père,

A mes très chers parents,

Aucun mot ne pourra exprimer mes sentiments envers vous.

A mon très cher frère Anas,

A ma très chère cousine Houda,

A mes très chères tantes,

A toute ma famille,

Je ne sais comment vous remercier pour tout ce que vous avez fait pour moi.

A Amina et Saad,

Pour les tous instants inoubliables que j'ai passés avec vous, merci d'être les meilleurs amis du monde,

A tous mes amis du lycée Moulay Youssef et de l'ENSIAS,

A tous ceux qui m'aiment,

Je vous remercie pour votre soutien et je vous dédie ce travail...

Imane

Dédicaces

A mes très chers parents

Aucun mot, aucune phrase ne peut exprimer ma gratitude envers vous, merci pour vos efforts, vos sacrifices, vos prières et votre amour. Je ne sais comment vous remercier pour tout ce que vous avez fait pour moi. Que Dieu vous garde et vous procure la joie et la santé.

A mes très chers frères et sœurs

Merci pour votre aide, votre soutien et vos encouragements.

A mon oncle et ma tante

Merci pour votre amour et votre chaleureux accueil.

A tous mes amis

Pour tous les moments inoubliables que j'ai passé avec vous

A l'ENSIAS

Mohammed

Remerciements

Il nous est agréable de nous acquitter d'une dette de reconnaissance auprès de toutes les personnes, dont l'intervention au cours de ce projet, a favorisé son aboutissement.

Nos très chers remerciements vont à M. Ahmed HABBANI, notre encadrant à l'ENSIAS, pour les conseils qu'il nous a prodigués, son judicieux encadrement ainsi que son assistance tout au long du projet.

Nos remerciements les plus sincères vont à Mlle. Imane NEJJAR, M. Mustapha EL FEDDI et M. Younes SOUTEH, nos encadrants au sein de Sofrecom, pour nous avoir encadrés et encouragés tout au long de ce projet, et prodigués leurs directives précieuses et leurs conseils pertinents qui nous ont été d'un appui considérable dans notre démarche.

Nous tenons à remercier tout le personnel de Sofrecom Services Maroc, pour leur soutien et pour leur générosité considérable quant à l'offre de l'information.

Que messieurs les membres du jury trouvent ici l'expression de nos reconnaissances pour avoir accepté de juger notre travail.

Nous tenons également à adresser nos plus sincères remerciements à l'ensemble du corps enseignant de l'ENSIAS, pour avoir porté un vif intérêt à notre formation, et pour nous avoir accordé le plus clair de leur temps, leur attention et leur énergie et ce dans un cadre agréable de complicité et de respect.

Enfin, que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent l'expression de nos remerciements et de notre considération.

Résumé

Le présent document constitue une synthèse de notre projet de fin d'études, effectué au sein de Sofrecom Services Maroc, unité de production Web Source. Ce projet a pour objectif la mise en place d'un outil de prise de commande Télécom basé sur une reformulation fonctionnelle des composants et modules existants de web source commande. Il a aussi été demandé de proposer une architecture SOA du système, en passant par la modélisation des processus métier selon une approche BPM.

Pour se faire, il fallait commencer, en premier lieu, par comprendre la problématique posée et les objectifs à atteindre. En deuxième lieu, Une étude fonctionnelle s'est avérée indispensable pour déterminer les spécifications des différents modules de notre outil. La phase de l'analyse a été abordée par la suite, afin d'élaborer l'ensemble des diagrammes modélisant le système et le processus de prise de commande, puis nous avons entamé une étude technique afin de choisir les outils de développement les plus adéquats à notre architecture SOA. En dernier lieu, nous nous sommes focalisés sur la phase de la réalisation, où l'on a pu établir les différentes couches de notre application.

Au terme de ce projet, les objectifs spécifiés ont été atteints. Des solutions ont été proposées pour chaque module de l'application, et la mise en œuvre de l'architecture SOA a été effectuée.

Mots clés: SOA, BPM, ESB, JMS, J2EE, prise de commande.

Abstract

This document constitutes a synthesis of our project of end of studies, carried out within Sofrecom Services Morocco, production unit Web Source. This project aims at the installation of a tool of order taking Télécom based on a functional reformulation of the components and existing modules of Web source orders, it was also required to propose a SOA architecture of the system, while passing by modeling of the processes trade according to a BPM approach. To be done, it was necessary to start, initially, by understanding the problems posed and the objectives to be reached. In second place, a functional study proved to be essential to determine the specifications of the various modules of our tool, the phase of the analysis was approached thereafter, in order to work out the whole of the diagrams modeling the system and the process of order taking, then we started a technical study in order to choose the most adequate development tools with our SOA architecture. Lastly, we focused ourselves on the phase of the realization, or one could establish the various layers of our environment. At the end of this project, the specified objectives were achieved. Solutions were proposed for each module of the application, and the implementation of the SOA architecture was carried out.

Keywords: SOA, BPM, ESB, JMS, J2EE, process ordering.

Liste des abréviations

Abréviation	Désignation
2TUP	Two Track unified process
API	Application Programming Interface
AS	Application Server
B2B	Business to Business
BPEL	Business Process Execution Language
BPM	Business Process Management
CDDL	Common Development and Distribution License
CORBA	Common Object Request Broker Architecture
CPAL	Common Public Attribution License
DB	DataBase
DCOM	Distributed Component Object Model
EAI	Enterprise Application Integration
EJB	Enterprise Java Bean
ESB	Enterprise Service Bus
FT	France Télécom
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
IDE	integrated development environment
J2EE	Java Enterprise Edition
JAX-R	Java API for XML Registries
JB1	Java Busines Integration
JBPM	Java Business Process Management
JCA	J2EE Connector Architecture
JDBC	Java DataBase Connectivity
JMS	Java Message Service
JNDI	Java Naming Directory Interface
JPA	Java Persistence API
JPDL	jBPM Process Definition Language
JSF	Java Server Faces
LGPL	Lesser General Public License
NMR	Normalized Message Router
OASIS	Organization for the Advancement of Structured Information Standards
OS	Open Source
OW2	ObjectWeb2
P2P	Point To Point
QOS	Quality Of Service
SGBD	Système de Gestion de Base de Données
SI	Système d'Information
SOA	Service Oriented ArchitectureS
SOAP	Simple Object access Protocol
SQL	Structured Query Language
SSM	Sofrecom Services Maroc

TCP	Transmission Transfer Protocol
UC	Use Case
UDDI	Universal Description Discovery and Integration
UDP	<i>User Datagram Protocol</i>
UML	Unified modeling language
W3C	World Wide Web Consortium
WS	Web Service
WSDL	Web Service Description Language
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations

Listes des figures

Figure 1.1 : Organisation de l'ingénierie logicielle de Sofrecom	19
Figure 1.2 : Organigramme de Sofrecom Services Maroc	20
Figure 1.3 : France Télécom assure l'interfaçage entre le client et ses fournisseurs	21
Figure 1.4 : Le système d'information soumis à deux natures de contraintes	26
Figure 1.5 : Le processus de développement en Y	27
Figure 1.6 : Planning du projet	30
Figure 2.1 : L'outil de prise de commande : deux acteurs métiers principaux	35
Figure 2.2 : Architecture logicielle du système	36
Figure 2.3 : Diagramme de contexte de notre système	38
Figure 2.4 : Diagramme de cas d'utilisation Catalogue	40
Figure 2.5 : Diagramme de cas d'utilisation Clients	42
Figure 2.6 : Diagramme de cas d'utilisation Site	43
Figure 2.7 : Diagramme de cas d'utilisation Utilisateurs	44
Figure 2.8 : Diagramme de cas d'utilisation Commandes	46
Figure 2.9 : Diagramme de classes candidates	48
Figure 2.10 : Diagramme de packages de notre système	49
Figure 2.11 : Diagramme de classes catalogue	50
Figure 2.12 : Diagramme de classes client	52
Figure 2.13 : Diagramme de classes Utilisateurs	53
Figure 2.14 : Diagramme d'états de la classe commande	54
Figure 2.15 : Processus de prise de commande	55
Figure 3.1 : SOA à la convergence de solutions	60
Figure 3.2 : publication et découverte des services	61
Figure 3.3 : SOAP, WSDL et UDDI en action	63
Figure 3.4 : La solution ETL	64
Figure 3.5 : Les médiateurs	65
Figure 3.6 : L'Intégration d'applications d'entreprise (EAI)	66
Figure 3.7 : L'Enterprise Service Bus (ESB)	67
Figure 3.8 : Enterprise Service BUS	68
Figure 3.9: Les fondements d'un ESB	69
Figure 3.10: Message multicast	71
Figure 3.11: Communication point à point	72

Figure 3.12: Publish / subscribe	72
Figure 3.13: Architecture multi-étages dans l'environnement J2EE	76
Figure 3.14: Enterprise Java Bean.....	77
Figure 3.15: Java Persistance API.....	78
Figure 3.16: Interface de JBPM	78
Figure 3.17: Intégration de Seam dans une application Web.....	79
Figure 3.18: Etude comparative des ESB OS	81
Figure 4.1: Digramme d'activité de la prise de commande	86
Figure 4.2: Architecture logicielle du système.....	87
Figure 4.3: Architecture logicielle du système.....	87
Figure 4.4: Définition des champs de l'entité catalogue	89
Figure 4.5: Exemple d'arborescence des packages métiers	89
Figure 4.6: Exemple d'arborescence des packages services	90
Figure 4.7: L'interface présentant la liste des clients	90
Figure 4.8: Ajouter une entité	91
Figure 4.9: Ajouter un client	92
Figure 4.10: Processus de prise de commande	94
Figure 4.11: Application JMS	95
Figure 4.12: Communication P2P	95
Figure 4.13: modèle de programmation JMS.....	96
Figure 4.14: Architecture d'une application JMS	96

Liste des tableaux

Tableau 2.1 : Les acteurs du système	37
Tableau 2.2 : Les besoins fonctionnels du système	38
Tableau 2.3 : liste de classes du package « catalogue »	51
Tableau 2.4 : liste de classes du package « client »	52
Tableau 2.5 : liste de classes du package « Utilisateurs »	53
Tableau 2.6 : Les services de prise de commande	56
Tableau 3.1 : Les fonctionnalités d'un ESB	71
Tableau 3.2: principales technologies des ESB open source	80
Tableau 4.1: Description de quelques IHM du système.....	93

Table des matières

Liste des abréviations	7
Listes des figures	9
Liste des tableaux	11
Table des matières	12
Introduction générale.....	15
Chapitre 1	17
Contexte général du projet.....	17
Introduction	18
1.1 Présentation de l'organisme d'accueil.....	18
1.1.1 Vocation et mission de Sofrecom.....	18
1.1.2 Sofrecom Services Maroc	19
1.1.3 Organigramme.....	19
1.2 Présentation générale du projet	20
1.2.1 Présentation du système d'information Web Source	20
1.2.2 Présentation du Web Source Commande	22
1.2.3 Objectif du projet	24
1.2.4 L'approche BPM	24
1.3 Conduite du projet.....	25
1.3.1 Processus de développement.....	25
1.3.2 Processus de modélisation UML.....	27
1.3.3 Planification du projet	28
Conclusion.....	31
Chapitre 2	32
Étude Fonctionnelle.....	32
Introduction	33
2.1 Étude préliminaire	33
2.1.1 Contour fonctionnel du projet	33
2.1.2 L'architecture logicielle du système.....	35
2.2 Capture des besoins fonctionnels	36
2.2.1 Identification des acteurs.....	37
2.2.2 Recueil des besoins fonctionnels.....	37
2.2.3 Diagramme de contexte.....	38

2.3	Spécifications	39
2.3.1	Cas d'utilisation du service fonctionnel Catalogue.....	39
2.3.2	Cas d'utilisation du service fonctionnel Client	41
2.3.3	Cas d'utilisation du service fonctionnel Site.....	43
2.3.4	Cas d'utilisation du service fonctionnel Utilisateur	44
2.3.5	Cas d'utilisation du service fonctionnel Commandes	45
2.3.6	Identification des classes candidates	47
2.4	Analyse.....	48
2.4.1	Développement du modèle statique	48
2.4.2	Développement du modèle dynamique	53
2.5	Processus de prise de commande	54
2.6	Extraction des services	56
	Conclusion.....	57
	Chapitre 3 :.....	58
	Étude technique.....	58
	Introduction	59
3.6	L'architecture orienté services	59
3.6.1	Définition	59
3.6.2	Principe de la SOA	60
3.6.3	Éléments de base d'une architecture SOA	61
3.6.4	Les avantages d'une SOA	63
3.6.5	Les inconvénients d'une SOA.....	63
3.6.6	Implémentations basées sur SOA.....	64
3.7	L'Entreprise Service Bus	67
3.7.1	Définition	67
3.7.2	Schéma d'un ESB.....	68
3.7.3	Eléments d'un ESB	69
3.7.4	L'ESB est basée sur les standards	70
3.7.5	Raisons d'utilisation d'un ESB	70
3.7.6	Les fonctionnalités d'un ESB.....	70
3.7.7	Types de messages supportés par un ESB	71
3.7.8	Les avantages et les inconvénients d'un ESB	72
3.8	Étude Comparative des ESB Open Source	74
3.8.1	Présentation des principaux outils Open Source	74
3.8.2	Critères du Choix d'un outil.....	75

3.8.3	Besoins techniques	75
	Conclusion.....	82
	Chapitre 4 :.....	83
	Réalisation & mise en œuvre.....	83
	Introduction	84
4.1	Conception	84
4.1.1	Conception préliminaire	84
4.1.2	Diagramme d'activité	84
4.2	Architecture Logicielle du Système	87
4.2.1	Couche présentation	88
4.2.2	Couche métier	88
4.2.3	Couche persistance	88
4.2.4	Couche données.....	88
4.3	Étapes de développement	88
4.3.1	Couche persistance	88
4.3.2	Couche métier	89
4.3.3	Couche présentation	90
4.4	Processus de prise de commande	93
4.5	La mise en œuvre de l'ESB.....	94
4.5.1	Java Message Service.....	94
	Conclusion.....	97
	Conclusion générale	98
	Références bibliographiques	99
	Annexes	100
	Annexe A : Glossaire métier	100
	Annexe B : Exemple Catalogue	102
	Annexe C : Plan Assurance Qualité	104

Introduction générale

La mondialisation et les moyens de communication accélèrent la concurrence entre les entreprises. La réduction du temps de travail impose une optimisation des processus de fabrication. Le cycle de vie des produits se réduit, leur développement doit être de plus en plus rapide et les sociétés doivent innover sans cesse.

La mise en place d'une approche SOA fait partie des changements organisationnels indispensables à toute entreprise qui souhaite rester dans la course à l'innovation. En effet, L'architecture SOA procure des avantages considérables en permettant au final aux entreprises de consacrer davantage de ressources et d'argent à l'innovation ainsi qu'à l'offre de nouveaux services de gestion.

A cet égard, et conformément à sa stratégie globale qui vise à améliorer le rendement de ses systèmes, Sofrecom Services Maroc a jugé nécessaire d'organiser des ressources informatiques distribuées dans une solution intégrée, éliminant ainsi les informations isolées et conférant davantage de souplesse à l'entreprise. L'orientation services organise assurément les ressources informatiques en modules, créant des processus métier faiblement couplés qui intègrent des informations provenant de différents systèmes.

C'est dans ce cadre que s'inscrit notre projet de fin d'études qui vise à mettre en place un outil de prise de commande Télécom selon une approche SOA, en passant par la modélisation du processus métier BPM.

Le présent rapport présente en détail notre projet de fin d'études. Il comporte quatre chapitres :

- ✓ Le premier décrit le contexte général du projet, en commençant par une présentation de l'organisme d'accueil, de la problématique traitée et les objectifs visés, suivis de la conduite adoptée pour le déroulement du stage.

- ✓ Le deuxième chapitre comporte deux parties. La première aborde l'étude préliminaire du projet dans laquelle une capture initiale des besoins est effectuée. Nous passons ensuite à la deuxième partie où est exposé l'étude fonctionnelle du projet. Elle présente la capture des besoins fonctionnels et la partie d'analyse.

✓ Le troisième chapitre traite la phase de l'étude technique qui a pour but de proposer une nouvelle architecture logicielle permettant une meilleure structuration des couches de notre application.

✓ Le dernier chapitre expose les différentes étapes de la mise en œuvre du projet à travers deux parties. La première partie présente la conception de l'application, alors que la deuxième partie présente la réalisation et illustre les interfaces du projet réalisé.

A la fin du rapport, nous présentons en complément les différentes annexes traitant le plan d'assurance qualité, un glossaire métier ainsi que d'autres informations nécessaires à la bonne compréhension de notre projet

Chapitre 1

Contexte général du projet

Ce chapitre se propose de présenter l'organisme d'accueil, le système Web Source Commande ainsi que le contexte général du projet, il précise également l'objectif et la conduite de notre projet.

Introduction

Dans le cadre des contrats qui peuvent être passés avec ses clients, France Télécom se présente comme l'intégrateur des solutions en télécommunications pour le compte de chaque signataire d'un contrat. L'outil de prise de commande sert de référentiel unique pour tous les acteurs d'un contrat. Chaque nouvelle affaire ne doit demander qu'un minimum d'adaptation, aussi certaines grandes fonctions de cet outil offrent des possibilités avancées de paramétrage en particulier pour la gestion des commandes.

France Télécom assure ainsi l'interface entre chacun de ses clients signataires d'un contrat, et tous les interlocuteurs nécessaires à la mise en œuvre des solutions proposées dans l'offre de service d'Outsourcing.

Ce chapitre sera consacré à la présentation de Sofrecom Services Maroc (filiale de France Télécom), ainsi que le contexte général et la conduite de notre projet de fin d'étude.

1.1 Présentation de l'organisme d'accueil

1.1.1 Vocation et mission de Sofrecom

Le Groupe Sofrecom est une filiale de France Telecom. Il est spécialisé dans le conseil et les technologies de l'information, et plus particulièrement dans le domaine des télécommunications. Sofrecom, leader du conseil et des technologies de l'information, spécialisé dans le domaine des télécommunications, accompagne les opérateurs - intégré, fixe, mobile, internet - et tous les acteurs de l'industrie des technologies de l'information – autorités de régulation, gouvernements, investisseurs, entreprises - tout au long de leur développement, en Europe, Afrique, Moyen-Orient, Asie et Amérique du Sud. De la définition de leur stratégie à la mise en œuvre de solutions innovantes, Sofrecom dispose d'un ensemble de compétences qui lui permet d'assurer la maîtrise complète de chaque étape d'un projet télécom.

Ingénierie logicielle

Sofrecom apporte maîtrise et compétitivité aux projets d'externalisation de développements logiciels et de Tierce Maintenance Applicative.

Le front office de Sofrecom s'appuie sur ses « software factories » développées dans les filiales : Maroc, Argentine, et Pologne, avec des ressources et des infrastructures dédiées. Quant au back office, il se base sur les compétences des ressources au niveau du

développement (cf. figure 1.1). Cette politique permet une adaptation rapide aux nouvelles technologies et une maîtrise des coûts.

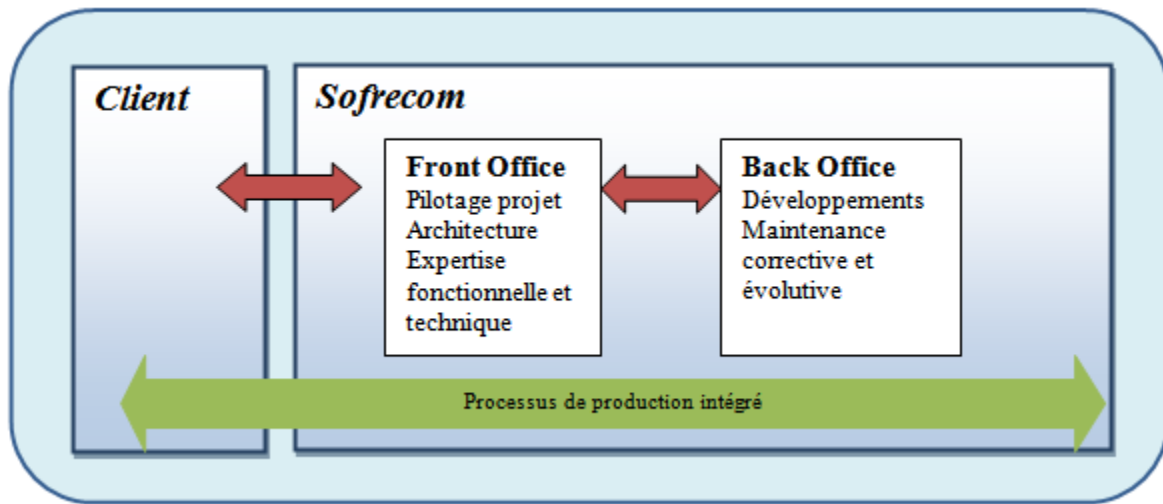


Figure 1.1 : Organisation de l'ingénierie logicielle de Sofrecom

L'ingénierie logicielle répond aux besoins suivants :

- Externalisation de développements logiciels en local, nearshore ou offshore ;
- Intégration et portage de logiciels, de systèmes et d'architecture technique.

1.1.2 Sofrecom Services Maroc

En août 2004, Unilog-Sofrecom est créée comme filiale dédiée à la production logicielle, elle est devenue Sofrecom Services Maroc depuis le 1er mars 2006. Elle répond aux services suivants :

- Ingénierie logicielle pour l'éditeur Sofrecom SA ;
- TMA (Tierce Maintenance Applicative) pour le groupe France Télécom.

Sofrecom Services Maroc est l'un des principaux centres de services de Sofrecom. Elle est conçue et organisée pour offrir un haut niveau des services d'externalisation. Et pour cela elle bénéficie d'une infrastructure technologique de pointe et d'ingénieurs hautement qualifiés.

1.1.3 Organigramme

Sofrecom Services Maroc est organisé en 4 unités de production. Notre stage s'est déroulé au sein de l'unité de production Web Source. Cette unité, comme les autres, est rattachée à la direction de production et de qualité.

La figure 1.2 représente l'organigramme de Sofrecom Services Maroc:

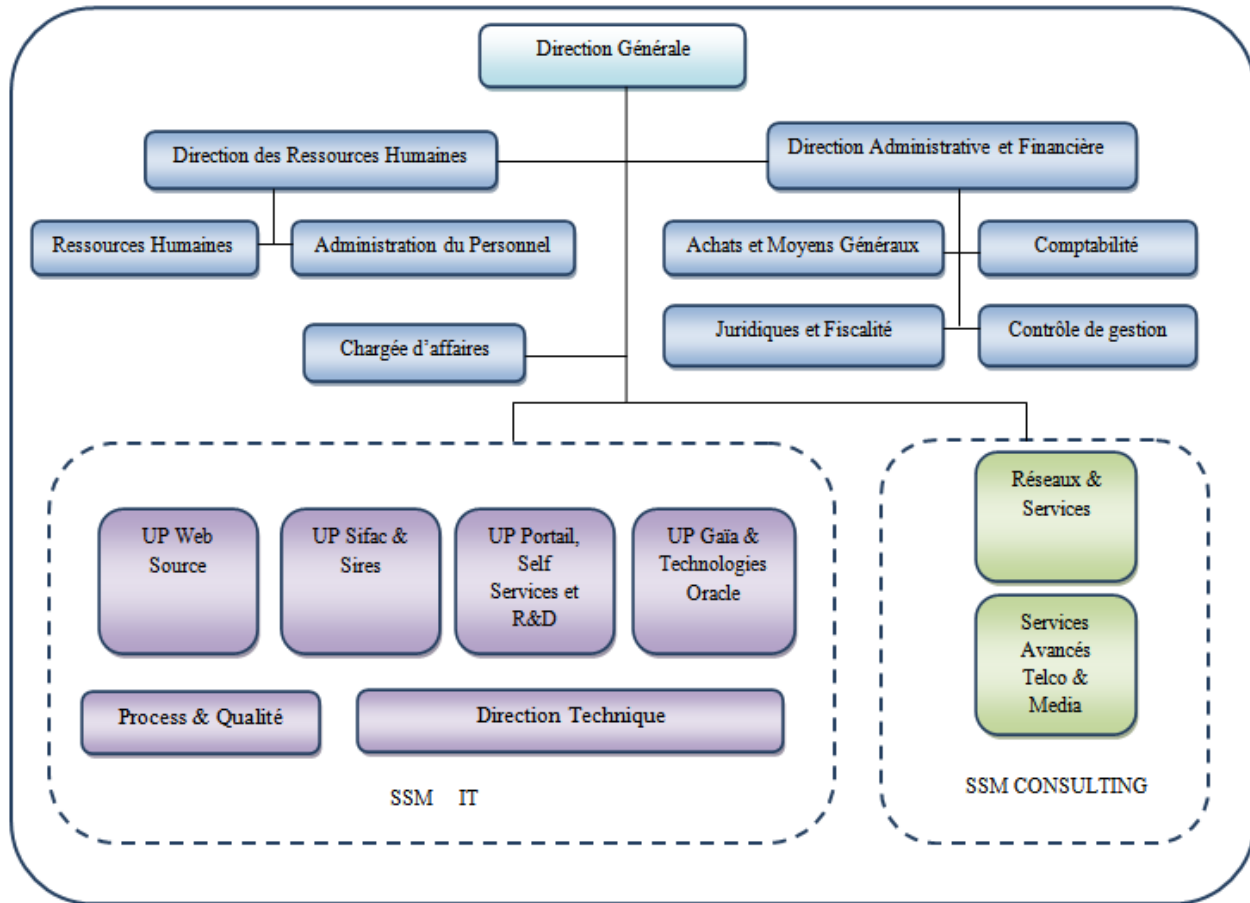


Figure 1.2 : Organigramme de Sofrecom Services Maroc

1.2 Présentation générale du projet

1.2.1 Présentation du système d'information Web Source

Dans le cadre des affaires d'Outsourcing globale ou partielle, ainsi que dans le cadre d'offre sur mesure pour les gros clients de France Télécom, il est contractuellement prévu de fournir un système de prise de commande unique et homogène basé sur un catalogue de produits et services préalablement défini entre le client et France Télécom.

L'outil Web Source constitue un référentiel unique d'administration des ventes partagé entre le Client, France Télécom et les fournisseurs de France Télécom (Filiales tel qu'Orange et Transpac et d'autres fournisseurs hors groupe France Télécom, en particulier pour les services à l'international) (cf. figure 1.3).

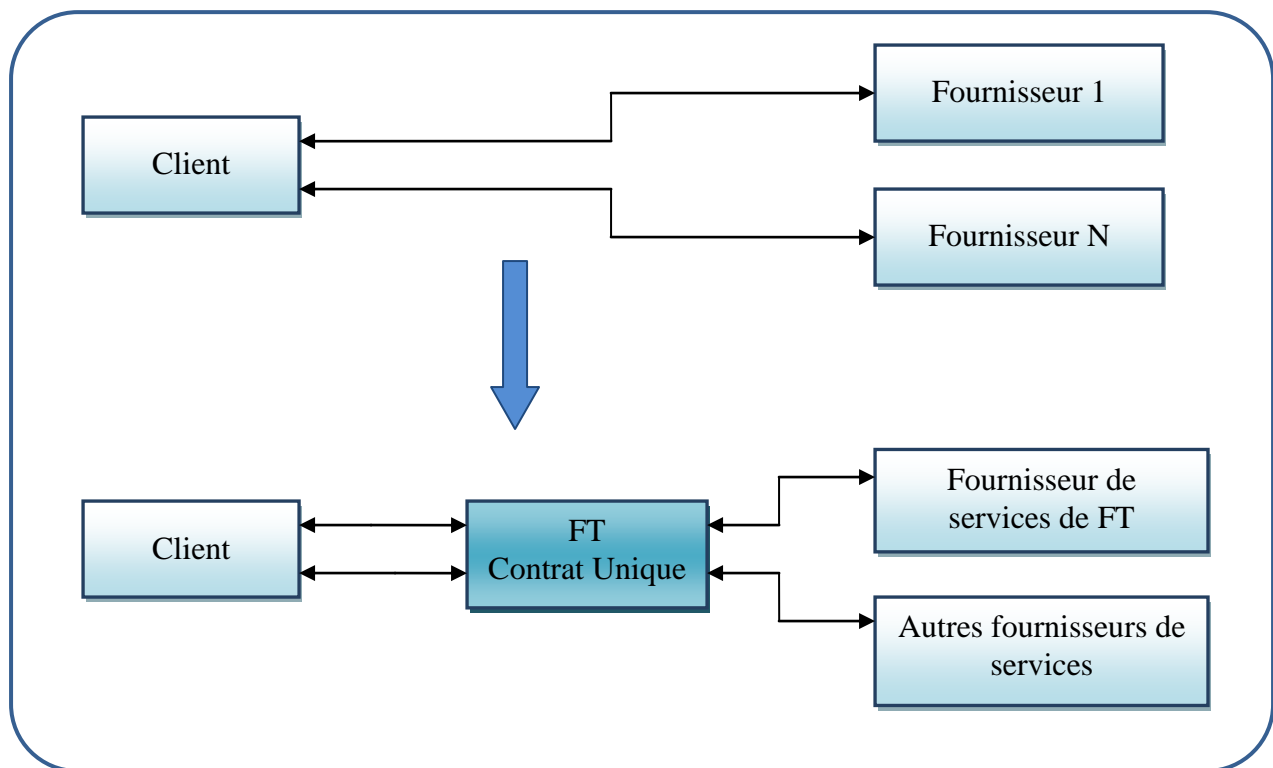


Figure 1.3 : France Télécom assure l'interfaçage entre le client et ses fournisseurs

C'est un outil fortement paramétrable, en particulier, en termes de création de catalogue de services, de processus de commande, de contrôle d'accès, de devise et de langue (IHM multilingue et possibilité d'avoir des catalogues multilingues etc.). L'outil doit s'adapter au processus de commande du client (étapes de validation interne entre autre).

Les utilisateurs des grands comptes France Télécom et les fournisseurs accèdent à l'application par Internet (Browser WEB). Les administrateurs accèdent à l'application pour son paramétrage à l'aide d'une interface Client serveur (Java Swing).

Web Source est amené à dialoguer avec les systèmes d'information des clients ou des fournisseurs dans le cadre du processus de prise de commande et celui de facturation (Le nom Web Source, signifie l'outsourcing grâce au réseau Web). Le besoin des clients et des fournisseurs est la cohabitation entre Web Source et leur système d'e-procurement existant (application EP externe) pour des raisons d'harmonisation interne de leurs Systèmes d'Information respectifs.

Web Source englobe la prise de commandes, la facturation, l'extraction de flux d'usages et un outil complet d'aide à la décision.

Le client grand compte de France Télécom passe ses commandes sur le système « Web Source Commande », ce dernier alimente le module de Facturation « Web Source Facturation », nommé aussi « E-Fact ». Ces deux modules alimentent tous les deux l'« infocentre » qui se charge du « reporting » et permet de produire les annexes des factures. Le module « Web Source Facturation », est aussi alimenté par le module « E-Flex », module se chargeant de la gestion des flux externes. Ainsi « E-Fact » prépare les factures et les renvoie à l'un des deux modules d'émission de factures selon leurs complexités.

Ainsi, les principales fonctions du système Web Source sont donc :

FRONT-OFFICE :

- Prise de commande à partir du catalogue ou du parc de service ;
- Suivi de la réalisation de la commande ;
- Consultation du parc de service ;
- Gestion de l'organisation client ;
- Gestion des droits des utilisateurs.

BACK-OFFICE :

- Chargé de la réception et le traitement des commandes par les fournisseurs.

ADMINISTRATION :

- Création des contrats et clients ;
- Création des utilisateurs et des groupes utilisateurs ;
- Paramétrage des catalogues client ;
- Paramétrage des processus de commande client.

1.2.2 Présentation du Web Source Commande

C'est un référentiel unique, partagé par tous les acteurs, il comprend essentiellement:

- Les catalogues de services ;
- Les commandes ;
- Le parc de services ;
- Les clients et leur organisation interne (entité, centres de coût et les sites).

Il permet une gestion optimisée du parc des services télécoms de l'entreprise :

- Par une vision temps réel et « On-line » de l'exhaustivité du parc de services Télécom ;
- Par un suivi temps réel et « On-line » de l'état d'avancement de toutes les commandes de services vers tous les fournisseurs FT ;
- Par une vision cohérente et homogène par tous les acteurs ;
- Par un processus de commandes qui respecte les circuits de décision de l'entreprise (Gestion de Workflow intégrée) ;
- Satisfaire à la plupart des affaires, l'objectif étant que chaque nouvelle affaire ne demande qu'un minimum d'adaptation logicielle ;
- Gérer au travers du catalogue, les produits et services standards des opérateurs ainsi que les services spécifiques proposés au travers de catalogues dédiés ;
- Gérer les commandes au travers des interfaces harmonisées, qui intègrent les fonctions multilingues ;
- Satisfaire à toutes les organisations rencontrées chez les clients et les fournisseurs, au travers du Workflow.

L'outil Web Source Commande permet :

- Aux acteurs chez le Client de :
 - Prendre les commandes (saisie, dépôt, autorisation, etc.) ;
 - Gérer l'organisation (Centres de coûts, etc.) ;
 - Gérer le référentiel des sites ;
 - Gérer les affectations des services sur les centres de coûts ;
 - Gérer les habilitations des utilisateurs du client (ils paramètrent leur Workflow).
- Aux acteurs chez France Télécom de :
 - recevoir les commandes par entité ;
 - Valider (ou reformuler) les commandes, et prononcer la mise en service.

L'outil Web Source Commande possède deux modules :

- Le module d'administration et de paramétrage : accessible via une interface client lourd, permettant aux différentes directions d'affaires de France Télécom d'effectuer certaines opérations comme la création des contrats, des catalogues et des clients, l'initialisation des habilitations et l'association des opérations sur un service au processus dédié (paramétrage du Workflow) ;

- Le module de la gestion des commandes et d'exécution des processus accessible via le web aux utilisateurs clients et aux utilisateurs fournisseurs.

1.2.3 Objectif du projet

Dans le cadre de la capitalisation sur les applications d'outsourcing développés au sein de Sofrecom, la direction technique désire mettre en place un outil de prise de commande basé sur une reformulation fonctionnelle des composants et modules existants de web source commande.

Web source commande étant destiné aux clients Grand Compte de France Télécom (Axa France, ST Electronics), l'outil de prise de commande sera plutôt destiné pour les particuliers et les clients petits compte de France Télécom.

L'objectif est de proposer aussi une architecture SOA pour l'outil de prise de commande, en passant par la modélisation des processus métier selon une approche BPM.

Le choix du concept SOA est la solution pour concevoir une architecture à base de services, permettant d'améliorer la flexibilité, en offrant un couplage faible entre les modules (services).

Le cahier de charges de notre projet, se constituait des étapes suivantes :

- Analyse et modélisation d'un nouveau catalogue ;
- Analyse et modélisation des autres entités métiers ;
- Analyse et modélisation des processus de la commande ;
- Identification des services de bases et services de haut niveau ;
- Proposition d'une nouvelle cinématique basée sur des processus métiers ;
- Modélisation des processus de la commande avec JBPM ;
- Mettre en œuvre une architecture SOA accessible via une plateforme de services sécurisée ;
- Réalisation d'un prototype opérationnel du système.

1.2.4 L'approche BPM

On appelle « BPM » (Business Process Management) l'approche consistant à modéliser informatiquement les processus métiers de l'entreprise, aussi bien dans leur aspect applicatif qu'humain.

L'objectif de cette démarche est d'aboutir à une meilleure vue globale de l'ensemble des processus métiers de l'entreprise et de leurs interactions afin d'être en mesure de les optimiser et, dans la mesure du possible, de les automatiser au maximum à l'aide d'applications métiers.

Le cycle de vie d'une démarche BPM peut globalement être décomposé de la manière suivante :

- Etude de l'entreprise en analysant ses objectifs et son organisation afin d'être en mesure de décomposer l'ensemble de son activité en processus métiers ;
- Modélisation des processus métiers, c'est-à-dire représenter informatiquement un modèle le plus proche possible de la réalité ;
- Implémentation de la solution : mise en œuvre d'une solution de BPM, reliée au système d'information de l'entreprise (applications et bases de données) ;
- Exécution : il s'agit de la phase opérationnelle où la solution de BPM est mise en œuvre ;
- Pilotage, consistant à analyser l'état des processus à travers des tableaux de bords présentant les performances des processus ;
- Optimisation, c'est-à-dire proposer des solutions permettant d'améliorer les performances des processus métiers.

1.3 Conduite du projet

1.3.1 Processus de développement

2TUP signifie « 2 Track Unified Process ». C'est un processus qui répond aux caractéristiques du Processus Unifié. Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise [Roques & al, 03]. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes « 2 Track » signifie littéralement que le processus suit deux types de chemins. Il s'agit des « chemins fonctionnels » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système d'information (cf. figure 1.4).

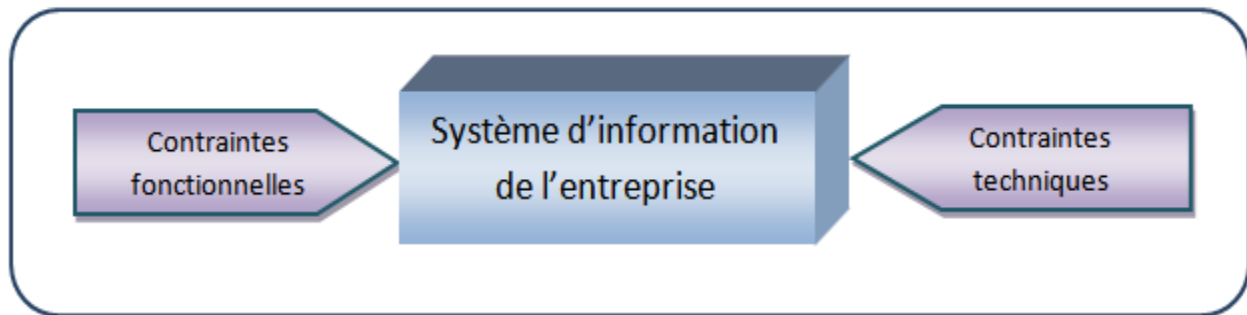


Figure 1.4 : Le système d'information soumis à deux natures de contraintes

La branche gauche (contraintes fonctionnelles) : capitalise la connaissance du métier de l'entreprise. Elle constitue généralement un investissement pour le moyen et le long terme.

Les fonctions du système d'information sont en effet indépendantes des technologies utilisées.

Cette branche comporte les étapes suivantes :

- La capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs ;
- L'analyse.

La branche droite (architecture technique) : capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent l'être en effet indépendamment des fonctions à réaliser.

Cette branche comporte les étapes suivantes :

- La capture des besoins techniques ;
- La conception générique.

La branche du milieu : à l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches. Cette fusion conduit à l'obtention d'un processus en forme de Y (cf. figure 1.5).

Cette branche comporte les étapes suivantes :

- La conception préliminaire ;
- La conception détaillée ;
- Le codage ;

- L'intégration.

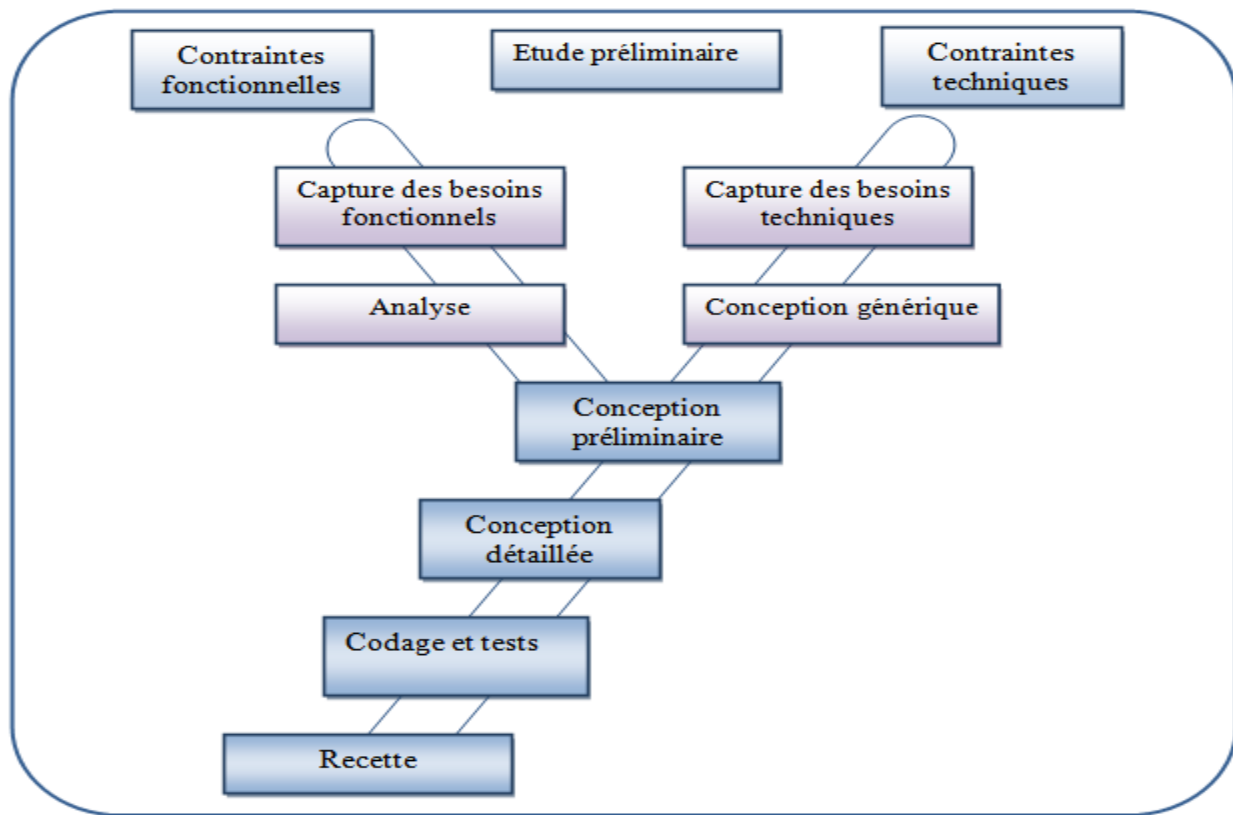


Figure 1.5 : Le processus de développement en Y

1.3.2 Processus de modélisation UML

Le processus 2TUP s'appuie sur UML tout au long du cycle de développement, car les différents diagrammes de ce dernier permettent de part leur facilité et clarté, de bien modéliser le système à chaque étape.

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue.

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage, c'est pour ça qu'UML est présenté parfois comme une méthode alors qu'il ne l'est absolument pas.

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la définition des besoins jusqu'au codage.

Voici une présentation rapide des différents diagrammes UML qui vont être utilisés tout au long du projet :

Le diagramme des cas d'utilisation : représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est normalement utilisé lors des étapes de capture des besoins fonctionnels et techniques.

Le diagramme d'activités : représente les règles d'enchaînement des activités et actions dans le système. Il peut être assimilé comme un algorithme mais schématisé.

Le diagramme de packages : présent depuis UML 2.0, ce diagramme modélise des catégories cohérentes entre elles, pour un souci de partage des rôles.

Correspond à l'étape de modélisation des différents scénarios d'un cas d'utilisation.

Le diagramme de classes : sûrement l'un des diagrammes les plus importants dans un développement orienté objet. Sur la branche fonctionnelle, ce diagramme est prévu pour développer la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet.

Le diagramme de séquence : représente les échanges de messages entre objets, dans le cadre d'un fonctionnement particulier du système.

Le diagramme d'états : représente le cycle de vie d'un objet. Il spécifie les états possibles d'une classe et leur enchaînement. Ce diagramme est utilisé lors des étapes d'analyse et de conception.

1.3.3 Planification du projet

Conduire un projet, c'est assurer le pilotage d'un processus de changement avec des ressources dédiées en optimisant les compétences, l'organisation, les systèmes et les outils de conduite.

Une approche managériale réactive, souple et systématique pour mener à bien des changements importants, complexes, ciblés sur le but à atteindre. Il y a trois niveaux de gestion du projet : la gestion de la production, des ressources et du temps. Afin de satisfaire à ce dernier critère qu'est la gestion du temps, il est nécessaire d'établir un planning prévisionnel. Sachant que la ressource

affectée à ce projet est l'élève ingénieur et que le présent planning est réajusté au fur et à mesure de l'avancement du projet.

Le stage débute le 16 février et prendra fin le 16 juin, ce qui implique une durée totale de 85 J*H.

La première étape de notre stage a été d'analyser les besoins fonctionnels du système en étudiant les spécifications de Web source commande, cette étape nous a permis de dégager les besoins fonctionnels du projet, en parallèle nous avons effectué des recherches sur l'architecture SOA et l'ESB afin de choisir l'environnement de développement le plus adéquats aux exigences techniques de notre application.

La seconde étape a été de proposer une nouvelle modélisation du système, pour ceci nous avons rédigé les dossiers de spécifications fonctionnelles des différentes entités métier de l'outil de prise de commande et nous avons aussi modélisé le processus Workflow de la prise de commande, puis la dernière étape a été de mettre en œuvre l'environnement et d'implémenter la première version de notre application.

Ci-dessous la planification détaillée de notre projet de fin d'étude :

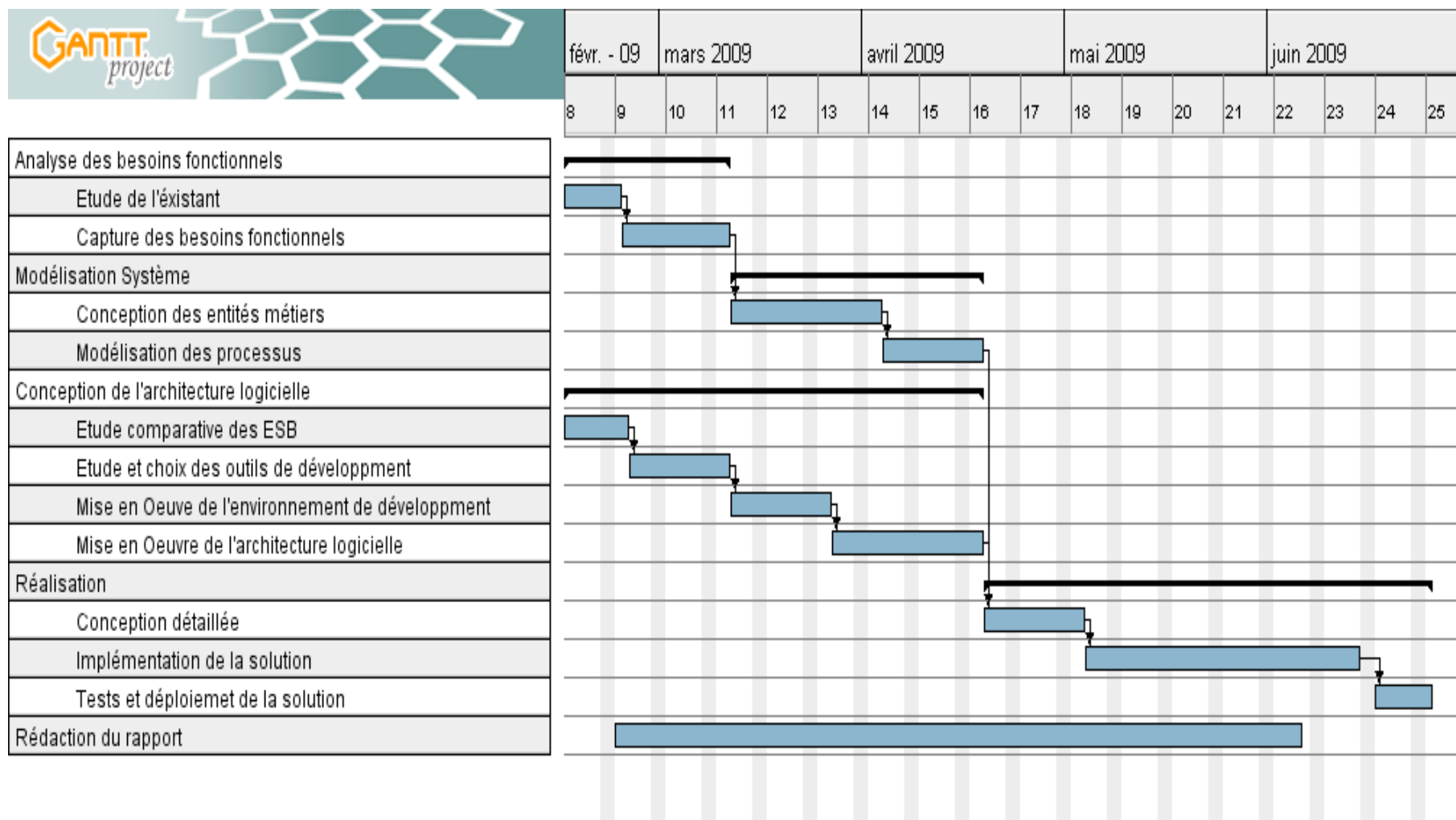


Figure 1.6 : Planning du projet

Conclusion

Ce chapitre a été le point de départ pour l'élaboration du projet, dans la mesure où il décrivait son contexte général, en présentant successivement l'organisme d'accueil Sofrecom Services Maroc, les objectifs généraux à atteindre, ainsi que la démarche et les étapes de sa mise en œuvre.

Le chapitre suivant sera réservé à l'analyse et les spécifications de notre projet. Il sera ainsi consacré à une étude préliminaire où l'on répertoriera nos besoins fonctionnels, puis nous entamerons l'étude fonctionnelle des différents modules de l'outil de prise de commande.

Chapitre 2

Étude Fonctionnelle

Ce chapitre comprend un sous chapitre pour l'étude préliminaire du projet, qui consiste à effectuer un premier repérage des besoins fonctionnels du projet.

Il a pour but de présenter l'étude fonctionnelle afin de mieux cerner les exigences fonctionnelles du système

Introduction

L'étude préliminaire ou la Pré-étude est la toute première étape du processus 2TUP. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou des diagrammes très simples. Elle prépare les activités plus formelles de capture des besoins fonctionnels et de capture techniques.

Ce chapitre sera consacré à l'étude préliminaire de notre projet, puis nous nous focaliserons sur une étude fonctionnelle détaillée des différents modules de notre système.

2.1 Étude préliminaire

2.1.1 Contour fonctionnel du projet

L'outil de prise de commande est une application de gestion commerciale pour les affaires d'Outsourcing de France Telecom.

L'outil de prise de commande permet aux Clients de France Telecom (FT) la prise de commande et le suivi centralisé des produits et services sur des catalogues qu'ils ont préalablement définis dans le cadre d'un contrat d'Outsourcing, ce contrat est un accord entre un Client et un Fournisseur de services concernant des services définis avec des prix qui leur sont associés avec un niveau de service.

L'outil de prise de commande permet notamment la gestion :

- Des clients ;
- Des utilisateurs du système ;
- Des catalogues de services, des prestations, des produits et leurs options;
- Des sites d'installation ;
- Des commandes.

2.1.1.1 *Le catalogue*

La prise de commande se fait via des catalogues offrant un ensemble d'éléments commercialisés et qui sont regroupés par leurs natures ou leurs technologies (voir Annexe B)

Les catalogues s'organisent d'une manière hiérarchique :

- Les classes (ex : Téléphonie, VoIP, IPTV)

Elles regroupent un ensemble d'éléments (services, produits, prestations) ayant la même technologie ou bien la même nature.

- Les services (ex : Service Téléphonie Mobile)

Un service est un élément commandable du catalogue. Chaque service possède des éléments de tarifications ainsi que des options obligatoires et des options facultatives.

- Les produits (ex : Antenne Wimax, Modem routeur Wimax, Carte PCMCIA)

Chaque produit à un élément de tarification, des options obligatoires et des options facultatives.

- Les prestations (ex : Maintenance)
- Les offres (ex : Téléphone Illimité vers opérateur, Abonnement Internet Haut Débit)

C'est un package regroupant un ensemble de services, de produits et éventuellement des prestations à un tarif spécial.

- Les options (ex : Carte SIM, Débit 256 kbps)

Une option peut être obligatoire ou facultative, il existe une notion de compatibilité entre les options (l'option « Débit 256 kbps » est incompatible avec l'option « Débit 512 kbps »).

2.1.1.2 La prise de commande

La commande débute lors de l'envoi de la demande par le client vers le fournisseur. Elle s'arrête lors de la mise en œuvre commerciale, c'est à dire contractuelle, du ou des services (produits ou prestations) demandés. La prise de commande implique deux acteurs métier principaux : le client et le fournisseur. Le déroulement d'une commande s'effectue de la manière suivante:

- Le Client : initialise un groupement de commandes, choisit l'ensemble des détails et des options de l'élément à commande, puis envoie la commande au fournisseur.
- Le Fournisseur : quand il reçoit la commande, s'il trouve que les données saisies par le client ne sont pas satisfaisantes, il demande une reformulation, sinon il étudie la faisabilité de la commande, puis procède à l'installation du service commandé, puis à la facturation.

La figure suivante (cf. figure 2.1) inventorie les rôles du client et du fournisseur :

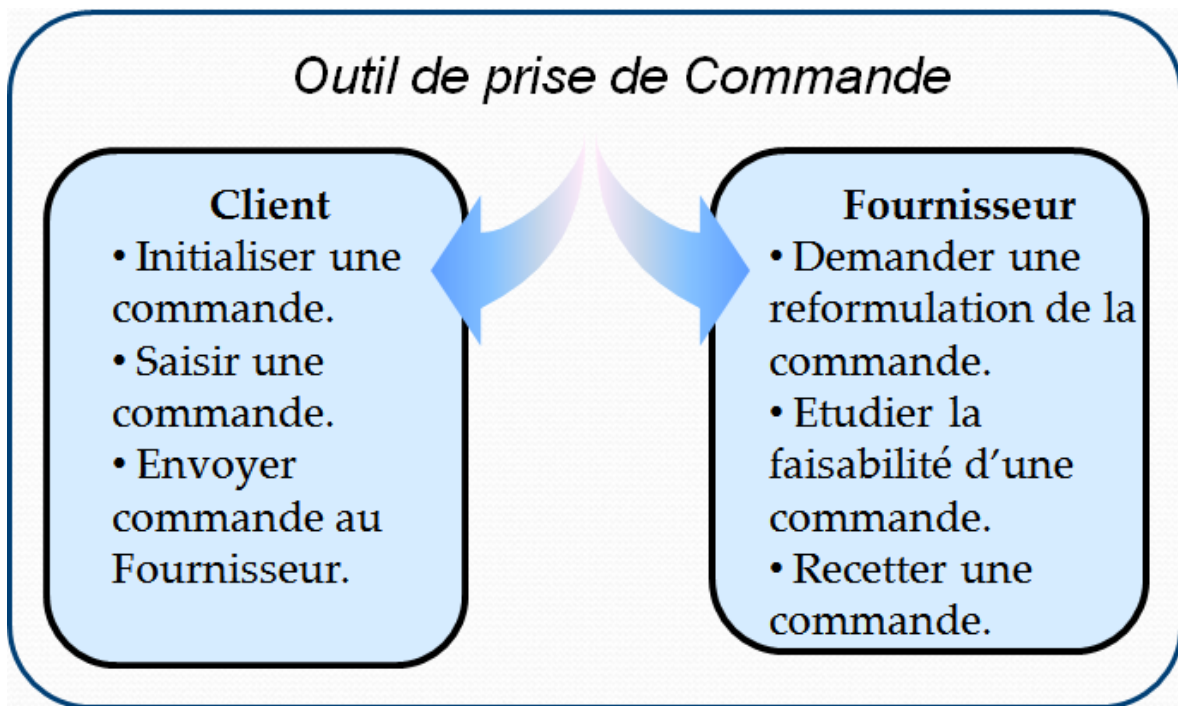


Figure 2.1 : L'outil de prise de commande : deux acteurs métiers principaux

Afin de répondre aux objectifs cités ci-dessus, le projet doit inclure les modules suivants :

- Un module qui permet la gestion des catalogues des services et des produits pouvant faire le sujet d'une commande via l'outil de prise de commande ;
- Un module permettant la gestion des clients ainsi que la gestion de leurs organisations;
- Un module pour la gestion des sites sur lesquels seront installés les services commandés par les clients ;
- Un module dédié à la gestion des utilisateurs du système incluant également la gestion des rôles et des droits ;
- Un module pour la gestion de commandes, les processus de prise de commande doivent être modélisés selon une approche BPM.

Afin d'assurer un système modulaire, où l'on peut commodément remplacer un composant par un autre et réutiliser des composants, nous avons opté pour une architecture SOA.

2.1.2 L'architecture logicielle du système

Pour mettre en œuvre un outil de prise de commandes télécom s'appuyant sur une architecture SOA, nous avons proposé pour le système l'architecture modélisée sur la figure ci-dessous (cf.figure 2.2) :

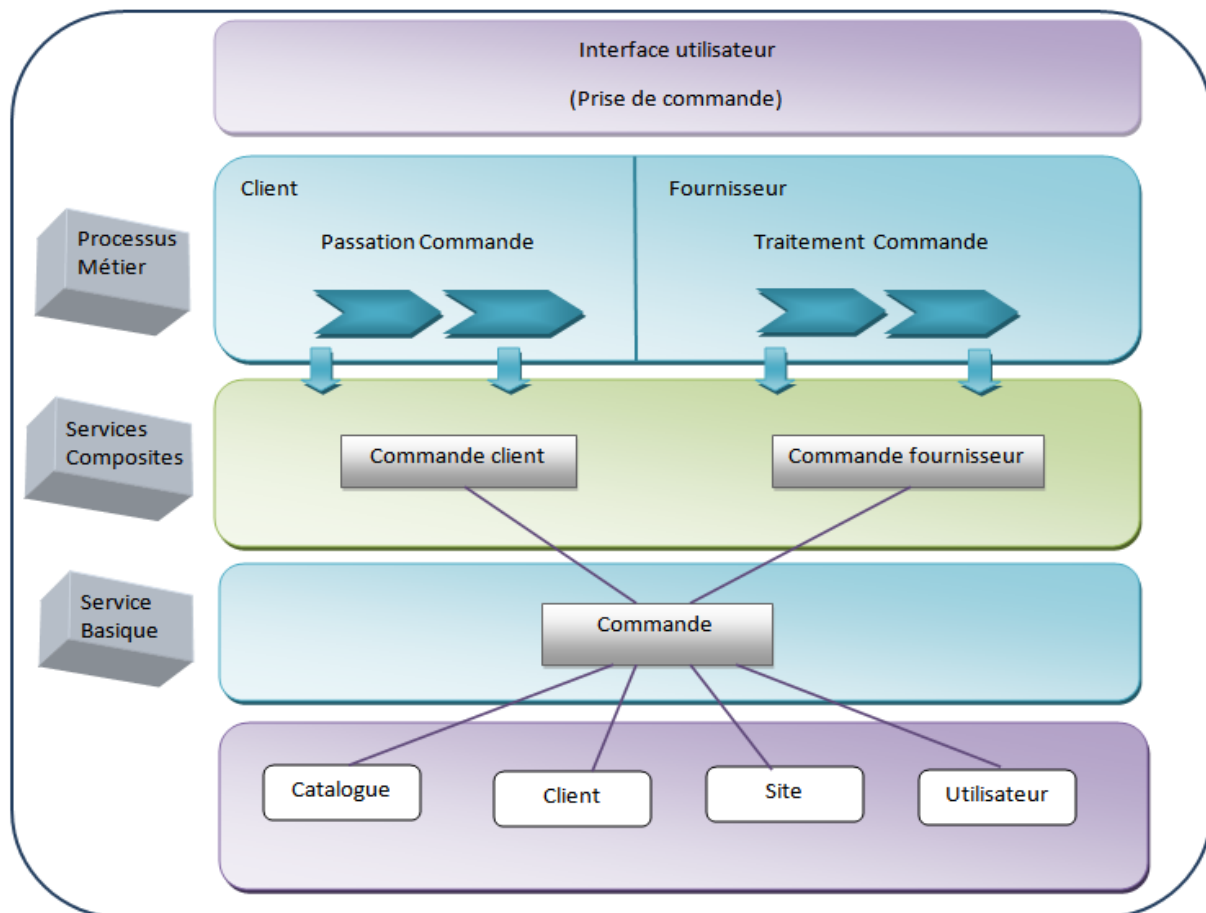


Figure 2.2 : Architecture logicielle du système

L'administrateur aura accès aux modules d'administration pour effectuer la gestion des catalogues, des clients, des sites et des utilisateurs. Les données sont récupérées de ces modules via la couche des services basiques représentées par le service « Commande ». Ce service est appelé dans la couche service composite par les services « Commande Client » et « Commande Fournisseur », qui à leur tour sont invoqués par le processus de prise de commande. Ce processus est déclenché via l'interface utilisateur accessible aux clients et aux fournisseurs.

2.2 Capture des besoins fonctionnels

Dans l'optique de concevoir une application qui soit modulaire, facilement extensible et orientée objet, le formalisme UML s'est imposé comme l'outil le plus adéquat pour la modélisation de ce projet. En effet, UML permet de mener la phase d'analyse et de conception tout en bénéficiant de la puissance et de la simplicité de l'orienté objet. UML est également le formalisme adapté au processus de développement choisi.

La capture des besoins fonctionnels a pour objectifs:

- L'identification des cas d'utilisation du système par ses acteurs ;
- La description des cas d'utilisation ;
- L'organisation des cas d'utilisation ;
- L'identification des classes candidates des modèles d'analyse.

2.2.1 Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

Les acteurs du système identifiés dans un premier temps ont les rôles suivants (cf. tableau 2.1):

Acteur	Type	Description de son rôle
Super Administrateur	Humain	C'est un acteur particulier du système. Il en existe qu'un seul et il est créé automatiquement à l'initialisation du système. Son rôle est la création des administrateurs et des groupes d'utilisateurs.
Administrateur Client	Humain	Il est responsable de la gestion des utilisateurs clients, des groupes utilisateurs et des sites.
Administrateur Fournisseur	Humain	Il est responsable de la gestion des utilisateurs fournisseurs et des groupes utilisateurs.
Utilisateur Fournisseur	Humain	Un acteur des fonctionnalités du système mises à sa disposition par l'interface utilisateur de type web.
Utilisateur Client	Humain	Un acteur des fonctionnalités du système mises à sa disposition par l'interface utilisateur de type Web.

Tableau 2.1 : Les acteurs du système

2.2.2 Recueil des besoins fonctionnels

Un recueil des besoins fonctionnels a permis de dégager les fonctionnalités essentielles du système (cf. tableau 2.2) :

Fonctionnalités	Description
Gestion des Catalogues	Cette fonctionnalité permet de gérer l'ensemble des services, produits ou prestations pouvant être commandés dans l'outil de prise de commande.
Gestion des Clients	Cette fonctionnalité permet de gérer les clients (entités juridiques) habilités à passer des commandes sur un ensemble de catalogues.
Gestion des Sites	Cette fonctionnalité permet de gérer les sites ou notamment les localisations permettant d'acheminer ou de déployer un service, un produit ou bien une prestation.
Gestion des utilisateurs	Cette fonctionnalité permet de gérer les utilisateurs de notre système.
Gestion de la prise de Commande	Permet la gestion des étapes de la prise de commande.

Tableau 2.2 : Les besoins fonctionnels du système

2.2.3 Diagramme de contexte

Les rôles des acteurs identifiés précédemment peuvent être représentés de façon synthétique sur un diagramme (cf. figure 2.3) appelé diagramme de contexte :

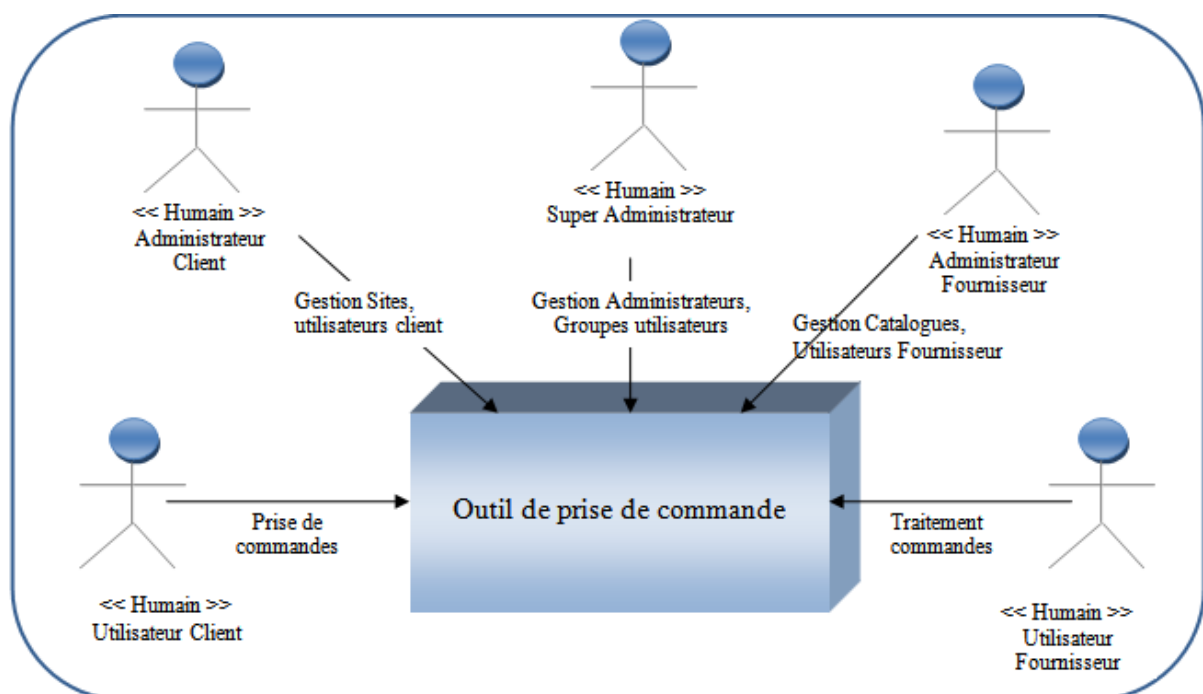


Figure 2.3 : Diagramme de contexte de notre système

2.3 Spécifications

2.3.1 Cas d'utilisation du service fonctionnel Catalogue

Un catalogue représente l'ensemble des services, des produits, des prestations ou des offres pouvant être commandés dans l'outil de prise de commande. Le catalogue est composé des items suivants :

- Les classes : elles regroupent les éléments du catalogue ayant la même nature ou bien la même technologie. Une classe appartient à un seul catalogue ;
- Les éléments : ils représentent les composants d'une classe et peuvent faire référence à des produits, des services ou des prestations. Les éléments gérés dans le catalogue peuvent évoluer dans le temps : modification de caractéristiques, de tarif, ...etc. Ces changements sont tracés dans le catalogue par un système de "versionning", ainsi un élément peut être en version « en cours » ou bien en version « archivé ». Un élément du catalogue est rattaché à une seule classe ;
- Les services : un service représente l'un des composants de base du catalogue qui est commandé par l'utilisateur, il peut être décliné en des options facultatives et des options obligatoires. Un service peut être associé à des produits obligatoires et/ou des produits facultatifs ;
- Les produits : c'est un composant matériel, il représente l'un des composants de base du catalogue qui est commandé par l'utilisateur, il peut être décliné en des options facultatives et des options obligatoires. Un produit peut être associé à des produits obligatoires et/ou des produits facultatifs. Un produit peut être associé à plusieurs services et/ou à plusieurs prestations ;
- Les prestations : c'est une intervention de maintenance ou bien une assistance. Une prestation représente l'un des composants de base du catalogue qui est commandé par l'utilisateur. Une prestation peut être associée à des produits obligatoires et/ou des produits facultatifs ;
- Les options : un composant faisant parti d'un service ou d'un produit. Il peut être facultatif comme il peut être de base. Une option ne peut être rattachée qu'à un seul élément ;
- Les offres : c'est un package regroupant des services, des produits et/ou des prestations.

Comme représenté sur le diagramme ci-dessous (cf. figure 2.4), l'administrateur peut notamment effectuer la gestion des catalogues, la gestion des classes, la gestion des services, la gestion des produits, gestion des prestations et la gestion des offres.

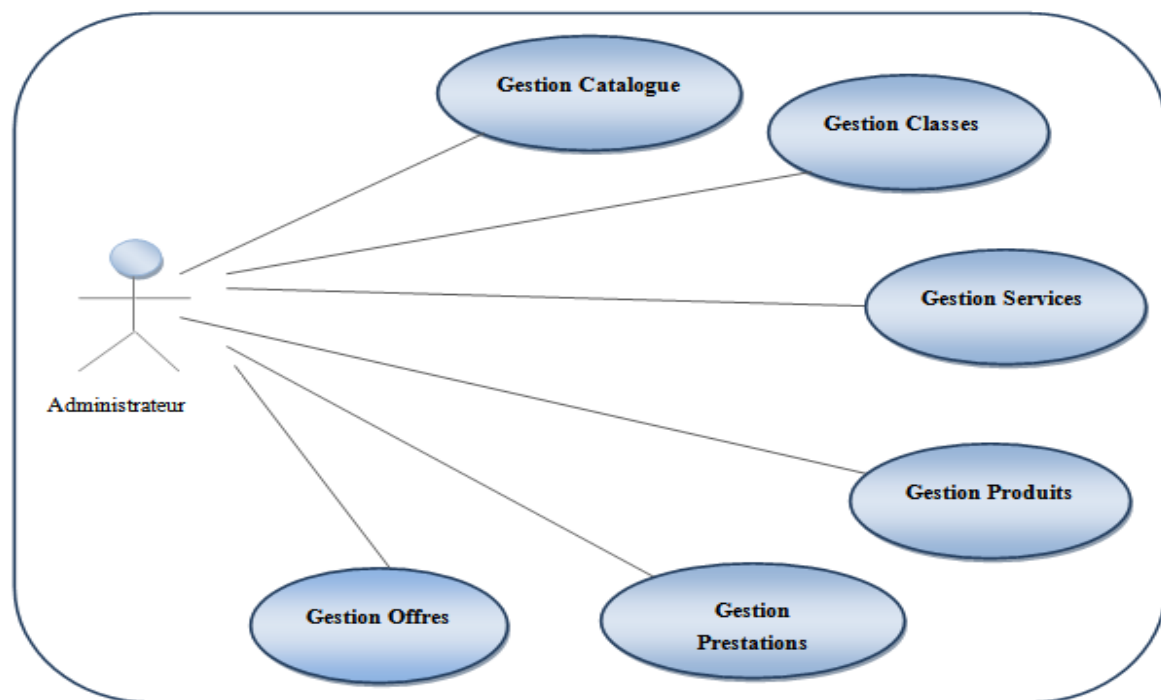


Figure 2.4 : Diagramme de cas d'utilisation Catalogue

UC Gestion Catalogue : description textuelle du cas d'utilisation

Nom du cas d'utilisation : Gérer Catalogue

Acteurs : Administrateur.

Contexte de déclenchement: l'acteur gère les catalogues d'un contrat.

Pré-conditions : l'acteur s'est connecté au système

Description du UC : permet à l'acteur de gérer les catalogues à savoir : l'ajout, la modification et la suppression du catalogue.

Post-conditions : la gestion des catalogues est effectuée.

Scénario nominal :

1. Connexion au système ;
2. Création d'un catalogue ;
3. Modification d'un catalogue ;
4. Suppression d'un catalogue.

UC Gestion Services : description textuelle du cas d'utilisation

Nom du cas d'utilisation : Gérer les Services**Acteurs :** Administrateur**Contexte de déclenchement :** l'acteur gère les services d'une classe d'un catalogue**Pré-conditions :** l'acteur s'est connecté sur le système**Description du UC :** la gestion des services d'une classe de catalogue permet l'ajout, la suppression et la modification des services.**Post-conditions :** la gestion des services est effectuée.**Règles de gestion :**

- Un service appartient à une seule classe du catalogue ;
- Un service a des options obligatoires et des options facultatives ;
- Un service peut avoir des produits obligatoires et des produits optionnels.

Scénario nominal

1. Sélection d'un catalogue ;
2. Le système verrouille le catalogue sélectionné ;
3. Sélection d'une classe de service et définition du type « Service » de l'élément ;
4. Création du service ;
5. Modification du service ;
6. Suppression du service ;
7. Modification de l'état d'un service.

2.3.2 Cas d'utilisation du service fonctionnel Client

Un client est une entité juridique désignant une nouvelle affaire sur le système. Chaque client est attaché à un seul et unique catalogue, il possède une raison sociale et il est associé à une et une seule adresse. Le client est responsable de la facturation des commandes de ses entités, et il en possède plusieurs. Une entité est une organisation du client pour laquelle on peut commander des services. Cette organisation peut être décrite d'une façon hiérarchique. L'entité juridique ou du premier niveau du client est créée automatiquement par le système à la création de ce dernier, et elle est l'entité mère de toutes les entités du client. Par défaut, cette entité possède :

- comme date d'ouverture, la date d'ouverture du client ;
- comme nom, la raison sociale du client ;
- comme code, le code du client.

Une entité n'est liée qu'à un seul client, et l'entité mère est directement liée à ce dernier et ne possède pas d'entité supérieure. Les autres entités sont rattachées l'une à l'autre selon leurs niveau hiérarchique.

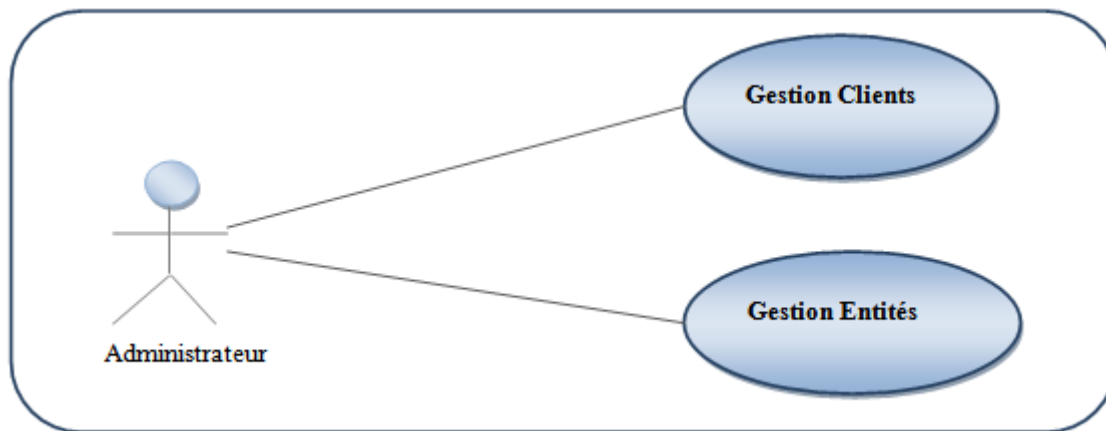


Figure 2.5 : Diagramme de cas d'utilisation Clients

Comme représenté sur le diagramme ci-dessus (cf. figure 2.5), les grands cas d'utilisations offerts par le module Client sont la gestion des clients et la gestion des entités. Voici un exemple de description du UC gestion Client :

Nom du cas d'utilisation : Gérer Client

Acteurs : Administrateur

Contexte de déclenchement : l'acteur gère les clients d'un catalogue.

Pré-conditions : l'acteur s'est connecté au système. Il a sélectionné un catalogue.

Description du UC : permet à l'acteur de gérer les clients à savoir : l'ajout, la modification et la suppression du client.

Post-conditions : la gestion des clients est effectuée.

Règles de gestion :

- Un client est associé à un et un seul catalogue ;
- Un client a obligatoirement une entité de premier niveau.

Scénario nominal :

1. Connexion au système ;
2. Sélection d'un catalogue ;
3. Création, modification ou suppression d'un client.

2.3.3 Cas d'utilisation du service fonctionnel Site

Un site est une localisation qui permet d'acheminer ou de déployer un service, un produit ou bien une prestation, il peut être le site d'installation d'un service, le site de livraison d'un produit, ou le site de maintenance ou d'une assistance d'une prestation.

Un site est caractérisé par un pays, une ville, un code postal, une voie et éventuellement un quartier et un complément d'adresse (cf. figure 2.6).

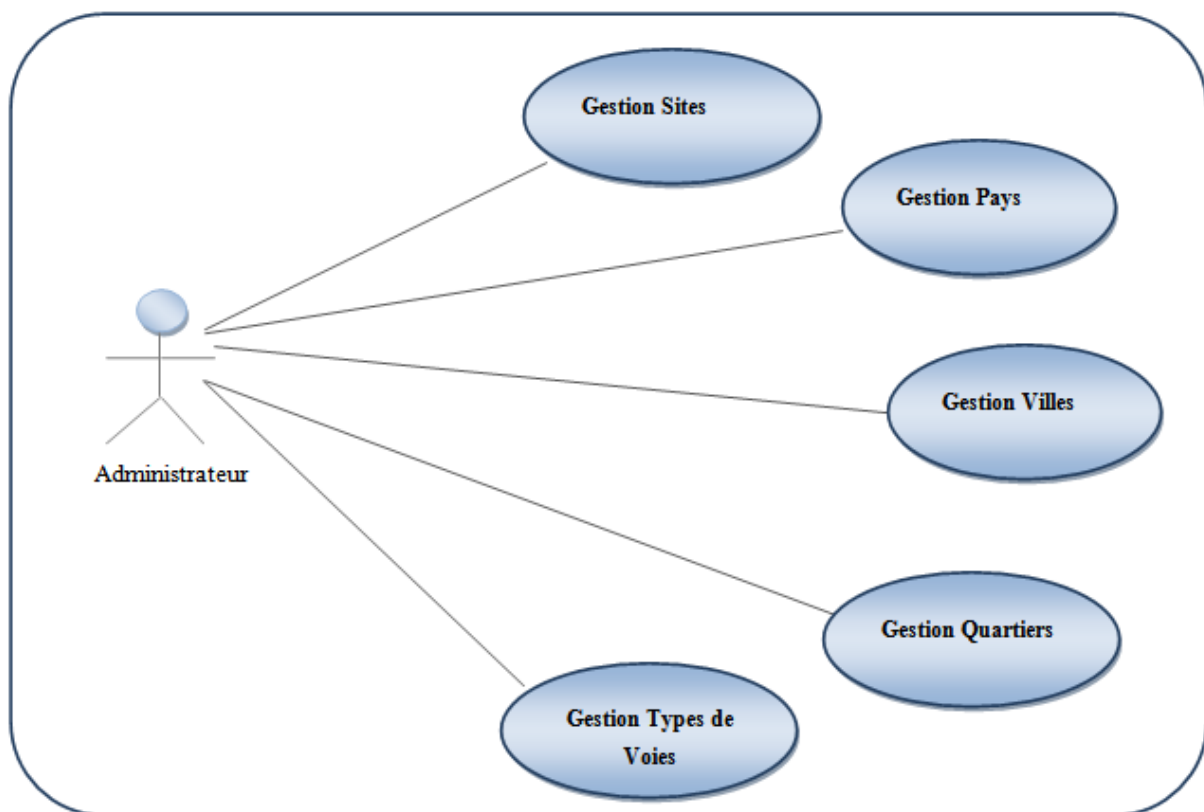


Figure 2.6 : Diagramme de cas d'utilisation Site

Voici une description du cas d'utilisation du UC gérer Pays

Nom du cas d'utilisation : Gérer Pays

Acteurs : Administrateur

Contexte de déclenchement : l'acteur gère les pays.

Pré-conditions : l'acteur s'est connecté au système.

Description du UC : permet à l'acteur de gérer les pays à savoir : l'ajout, la modification et la suppression de pays du système.

Post-conditions : la gestion des pays est effectuée.

Scénario nominal :

1. Connexion au système.
2. Création d'un pays ;
4. Modification d'un pays ;
5. Suppression d'un pays.

2.3.4 Cas d'utilisation du service fonctionnel Utilisateur

On appelle « Utilisateur », les utilisateurs du système. Ils disposent chacun dans le système d'un enregistrement porteur d'information de contrôle et d'autorisation. Les utilisateurs « Utilisateur » accèdent à l'application via une IHM WEB. Ces utilisateurs sont associés à un ou plusieurs groupes d'utilisateurs.

Dans le système sont définis les administrateurs, qui ont pour rôle l'administration des catalogues et des utilisateurs, les utilisateurs clients ont pour rôle la prise de commandes ainsi que la consultation des catalogues, les utilisateurs fournisseurs sont responsables du traitement des commandes et un superviseur qui a pour rôle la gestion des administrateurs et des groupes des utilisateurs. Il n'y a qu'un seul superviseur dans le système et il est créé automatiquement à la mise en œuvre de l'outil.

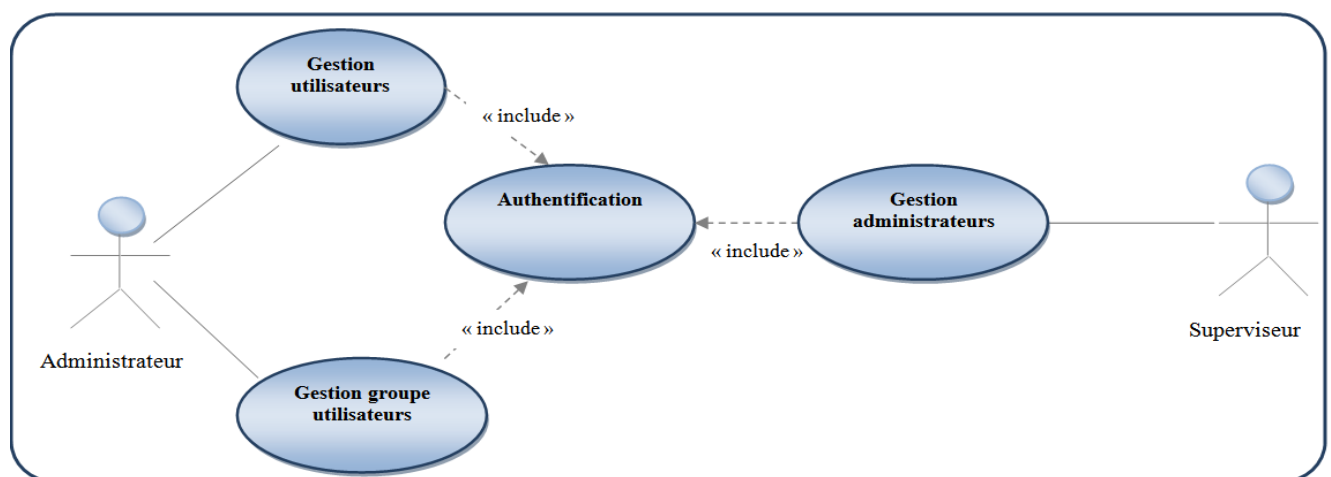


Figure 2.7 : Diagramme de cas d'utilisation Utilisateurs

Cette figure (cf. figure 2.7) présente les cas d'utilisations du module de gestion des utilisateurs : l'administrateur doit s'authentifier pour pouvoir accéder aux modules de gestions des utilisateurs et des groupes d'utilisateurs.

UC Gestion Administrateurs : description textuelle du cas d'utilisation

Nom du cas d'utilisation : Gérer les utilisateurs administrateurs client ou administrateurs fournisseur

Acteurs : Super Administrateur

Contexte de déclenchement : l'acteur gère les administrateurs du système

Pré-conditions : l'acteur s'est connecté au système.

Description du UC : permet à l'acteur la gestion des administrateurs à savoir : l'ajout, la modification et la suppression de ces utilisateurs.

Post-conditions : la gestion des administrateurs est effectuée

Contraintes IHM : la gestion des administrateurs de type « Client » et « Fournisseur » est accessible à partir du menu principal du module d'administration. l'interface graphique de la gestion des administrateurs, propose la liste de tous les administrateurs de type « Client » et « Fournisseur ». Cette liste peut être filtrée par type d'administrateurs.

Scénario nominal

1. Connexion au système ;
3. Création d'un utilisateur administrateur client ou administrateur fournisseur ;
4. Modification d'un utilisateur administrateur client ou administrateur fournisseur ;
5. Suppression d'un utilisateur administrateur client ou administrateur fournisseur.

2.3.5 Cas d'utilisation du service fonctionnel Commandes

Une commande est un acte juridique qui contractualise une demande, pour un service du catalogue, d'un client vers un fournisseur. Les commandes prises à travers le système sont dématérialisées. La conséquence est qu'à tout moment, le client et le fournisseur partagent la même vision de cette commande.

La commande débute lors de l'envoi d'une demande par le client vers le fournisseur. Elle s'arrête lors de la mise en œuvre commerciale, c'est à dire contractuelle, du ou des services demandés. Les commandes, ainsi « recettées » dans le système alimentent le parc des services.

Toute commande fait partie d'un groupement de commande, qui va être envoyé au fournisseur.

Description détaillée des cas d'utilisations

Comme représenté sur le diagramme ci-dessous (cf. figure 2.8), un utilisateur effectue une nouvelle commande, un groupement de commandes est automatiquement initialisé puisque chaque commande fait partie d'un groupement, ensuite l'utilisateur recherche un service dans le catalogue ou dans le parc (regroupant les services déjà commandés), affecte la commande à une entité, renseigne les détails et les sites d'installations, et avant de terminer le groupement de commandes, il consulte la synthèse de la commande réalisée.

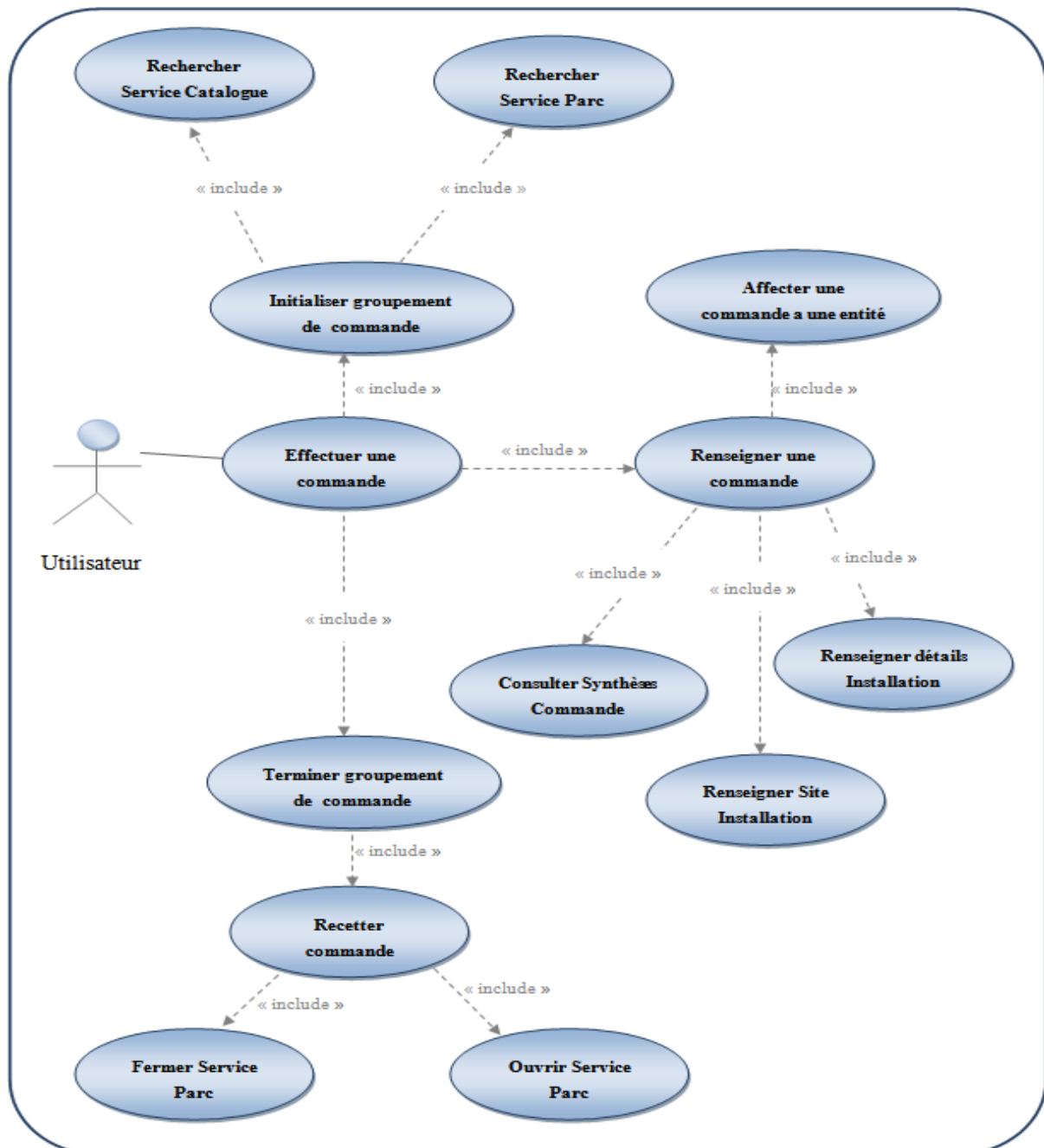


Figure 2.8 : Diagramme de cas d'utilisation Commandes

Description textuelle du cas d'utilisation : Effectuer une nouvelle commande

Nom du cas d'utilisation : Effectuer une nouvelle commande

Acteurs : Utilisateur ou Administrateur Client

Contexte de déclenchement : à partir de plusieurs situations, l'acteur peut déclencher des commandes de nouveaux services catalogue ou parc :

- Par lancement d'une opération de commandes d'ouverture à partir du menu de sa page d'accueil,
- Par sélection à partir de sa boîte de réception (ouverture d'une étape),
- Par sélection d'un message au niveau de la page d'accueil pour faire évoluer le groupement de commandes correspondant laissé en attente.

Pré-conditions : l'acteur s'est authentifié

Description du UC : cette fonctionnalité permet à l'acteur d'effectuer des commandes de nouveaux services à partir du catalogue ou à partir du parc.

Post-conditions : le ou les nouveaux services sont « recettés », et les services sont actifs dans le parc.

Règles de gestion : les commandes sont créées dans un groupement de commandes.

Pour une commande d'ouverture catalogue, l'acteur sélectionne un service (ou produit ou prestation) à commander à partir du catalogue.

Pour une commande d'ouverture parc, l'acteur sélectionne un service (ou produit ou prestation) à commander à partir du parc actif.

La commande est créée physiquement en base métier, uniquement lorsque la première étape est pré terminée et que le groupement est terminé.

Pour une commande d'ouverture, l'acteur peut sélectionner un service à commander à partir du catalogue ou à partir du parc actif.

2.3.6 Identification des classes candidates

Le diagramme des classes candidates est un premier diagramme de classes obtenu à partir de l'identification des cas d'utilisation. C'est un premier travail d'analyse qui permet une première abstraction du système sous forme d'objets et de classes.

Comme l'explique la figure ci-dessous (cf. figure 2.9), un utilisateur Client effectue une commande, cette dernière concerne un service, un produit, une prestation ou une offre et elle est rattachée à un site d'installation.

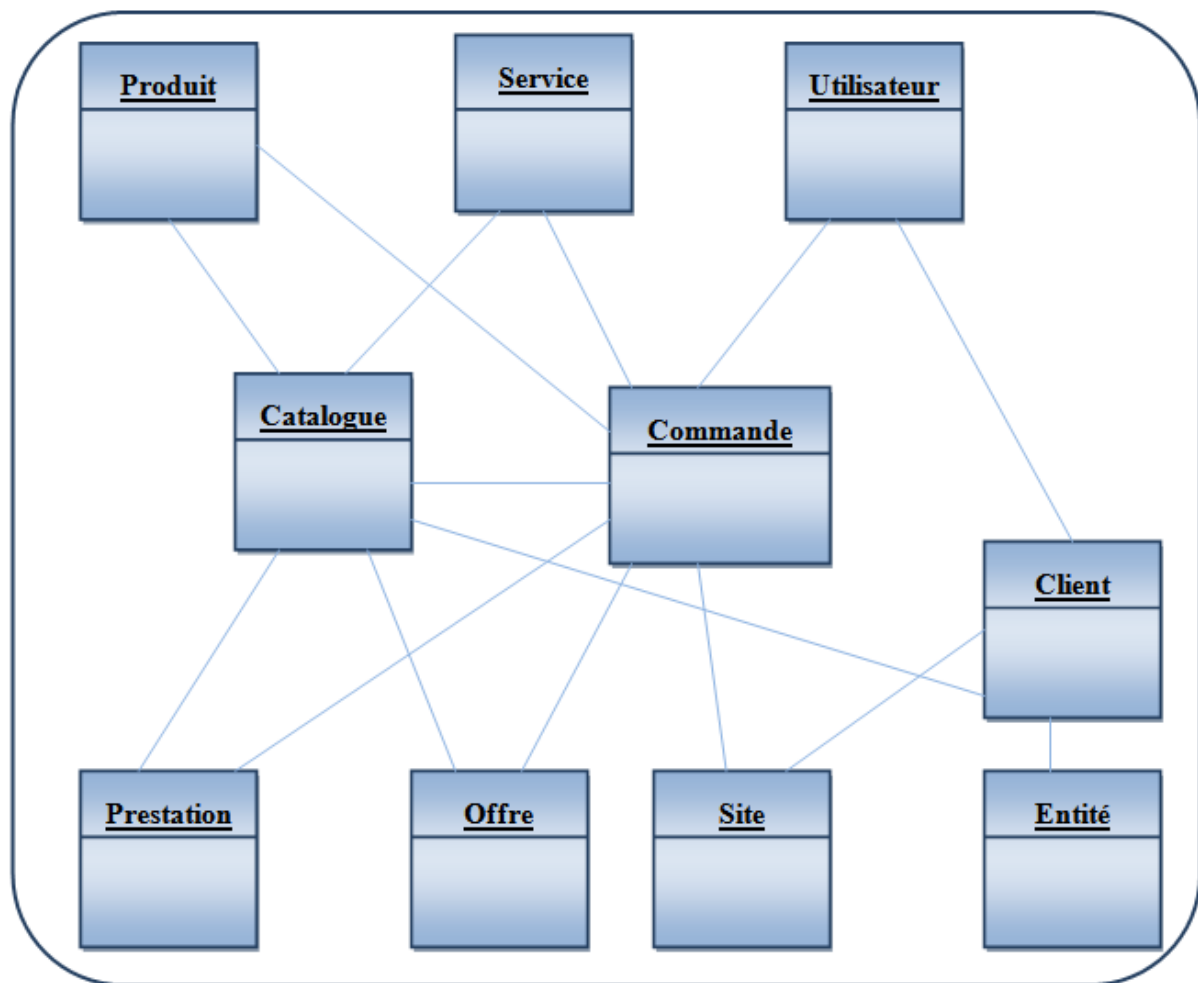


Figure 2.9 : Diagramme de classes candidates

2.4 Analyse

2.4.1 Développement du modèle statique

Le développement du modèle statique constitue la première étape d'analyse. Le diagramme de classe est organisé en package avant de le découper pour étudier les classes établies sommairement. Il est construit en détaillant, complétant et optimisant le diagramme des classes candidates.

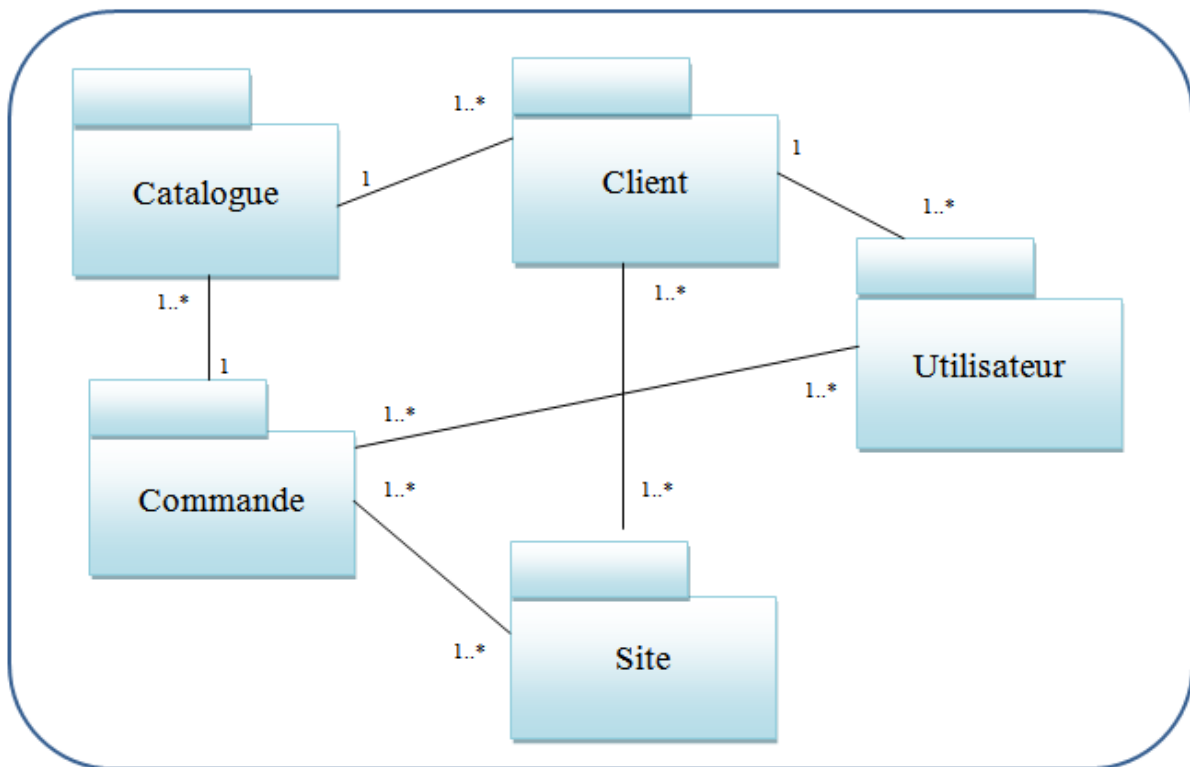


Figure 2.10 : Diagramme de packages de notre système

Afin d'expliquer le présent diagramme (cf. figure 2.10) de classe, il est nécessaire de le diviser selon les packages fonctionnels cités lors de l'étude fonctionnelle.

2.4.1.1 Package « catalogue »

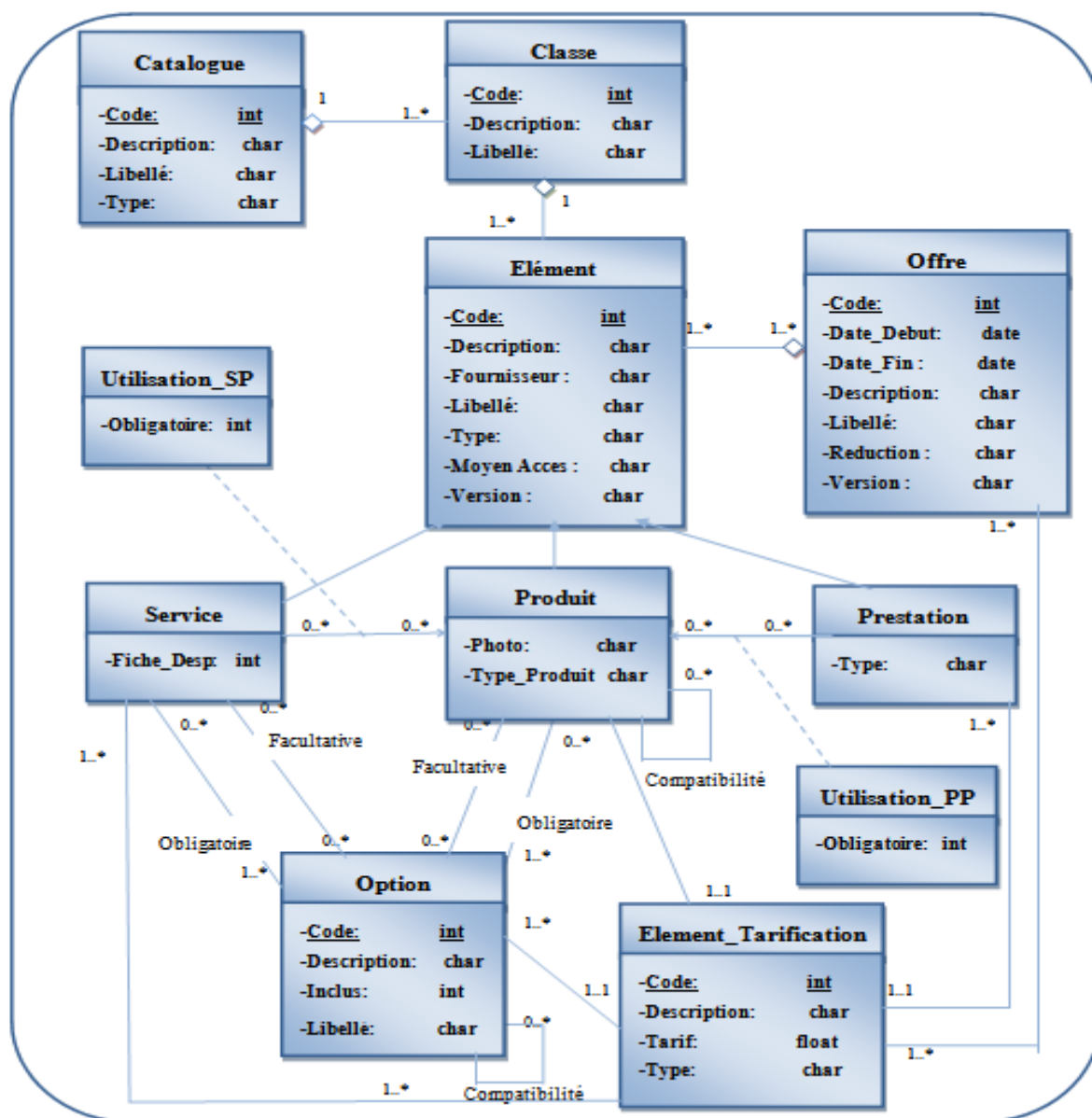


Figure 2.11 : Diagramme de classes catalogue

Les classes du package catalogue sont :

Classe	Description
Catalogue	L'ensemble des services, des produits, des prestations ou des offres pouvant être commandés dans l'outil de prise de commande.
Classe	Une classe permet de regrouper des éléments et des offres de même technologie ou de même nature.

Elément	Un élément fait parti des composants d'une classe. Il peut désigner un produit, un service ou bien une prestation.
Offre	C'est un package regroupant des services, des produits et (ou) des prestations.
Option	Un composant faisant parti d'un service ou d'un produit. Il peut être facultatif comme il peut être de base.
Prestation	Une intervention de maintenance ou bien une assistance. Une prestation peut être associée à des produits obligatoires et/ou des produits facultatifs.
Produit	Un produit représente tout composant matériel pouvant être commandé par l'utilisateur à travers l'outil de prise de commande peut être décliné en des options incluses, des options facultatives. Un produit peut être directement sujet d'une commande comme il peut faire partie des services commandés.
Service	Un service représente tout composant immatériel du catalogue pouvant être commandé par l'utilisateur à travers l'outil de prise de commande. Il peut être décliné en des options incluses, des options facultatives et (ou) des produits.

Tableau 2.3 : liste de classes du package « catalogue »

Ce package représente l'ensemble des composants pouvant être commandés dans l'outil de prise de commande. Un catalogue est composé de classes, Un élément peut être un produit, un service ou une prestation. Les services et les produits peuvent être déclinés en des options obligatoires et des options facultatives.

2.4.1.2 Package « Client »

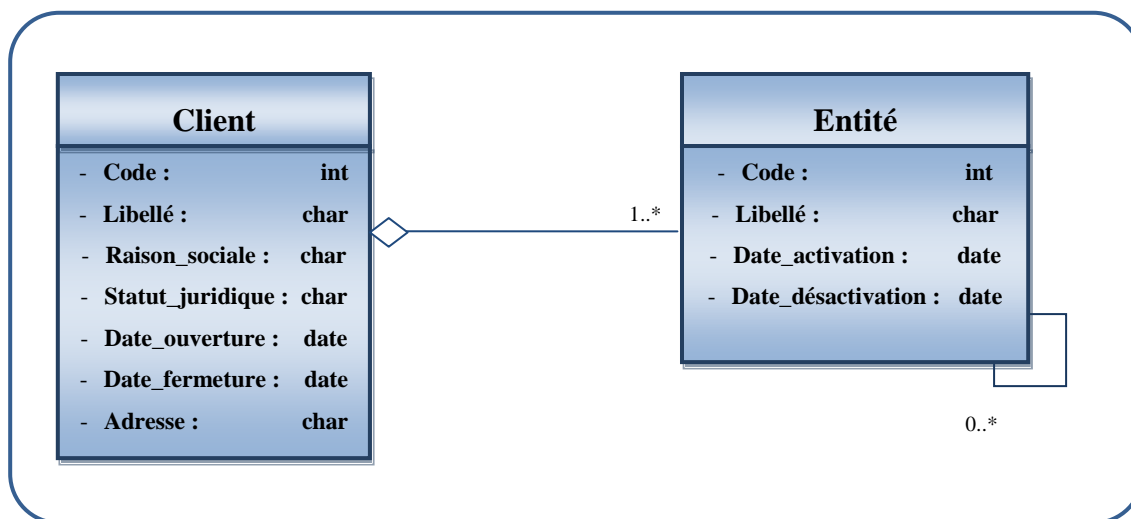


Figure 2.12 : Diagramme de classes client

Ce package regroupe les clients et leurs entités ayant habilités d'effectuer une prise de commande dans l'outil (cf. figure 2.12). Un client a obligatoirement une entité du premier niveau qui est créé automatiquement lors de la création de ce dernier.

Quant aux autres entités, elles sont rattachées à leur entité de niveau hiérarchique supérieur.

Les classes du package « client » sont (cf. tableau 2.4):

Classe	Description
Client	Représente une nouvelle affaire sur le système.
Entité	Représente l'organisation du client pour laquelle on peut commander des services, des produits et des prestations.

Tableau 2.4 : liste de classes du package « client »

2.4.1.3 Package « Utilisateurs »

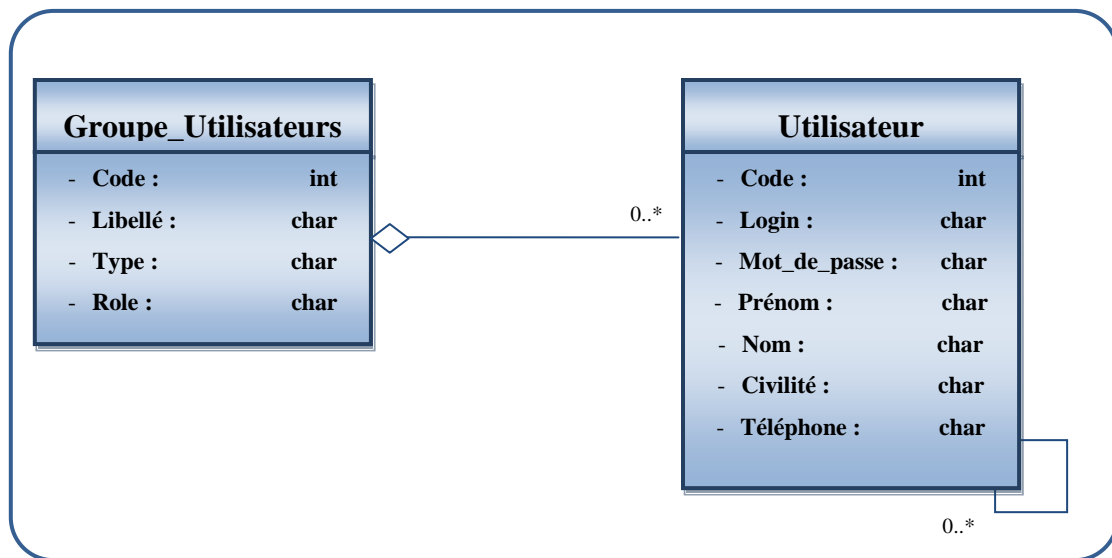


Figure 2.13 : Diagramme de classes Utilisateurs

Ce package (cf. figure 2.13) regroupe l'ensemble des utilisateurs de notre système, ils sont regroupés dans des groupes d'utilisateurs (cf. tableau 2.5).

Classe	Description
Groupe d'utilisateurs	Contient des regroupements des utilisateurs ayant les mêmes droits.
Utilisateurs	Cette classe contient la liste des utilisateurs du système.

Tableau 2.5 : liste de classes du package « Utilisateurs »

2.4.2 Développement du modèle dynamique

Le développement du modèle dynamique constitue la deuxième activité de l'étape d'analyse, Il s'agit d'une activité itérative, fortement couplée avec la modélisation statique.

2.4.2.1 Diagramme d'états

La vue locale d'un objet, décrivant comment il réagit à des événements en fonction de son état courant, est représentée graphiquement sous forme d'un diagramme d'états. Un objet passe par une succession d'états durant son existence. Un état a une durée finie, variable selon la vie de l'objet, en particulier en fonction des événements qui lui arrivent.

La figure suivante (cf. figure 2.14) représente le diagramme d'états de la classe commande :

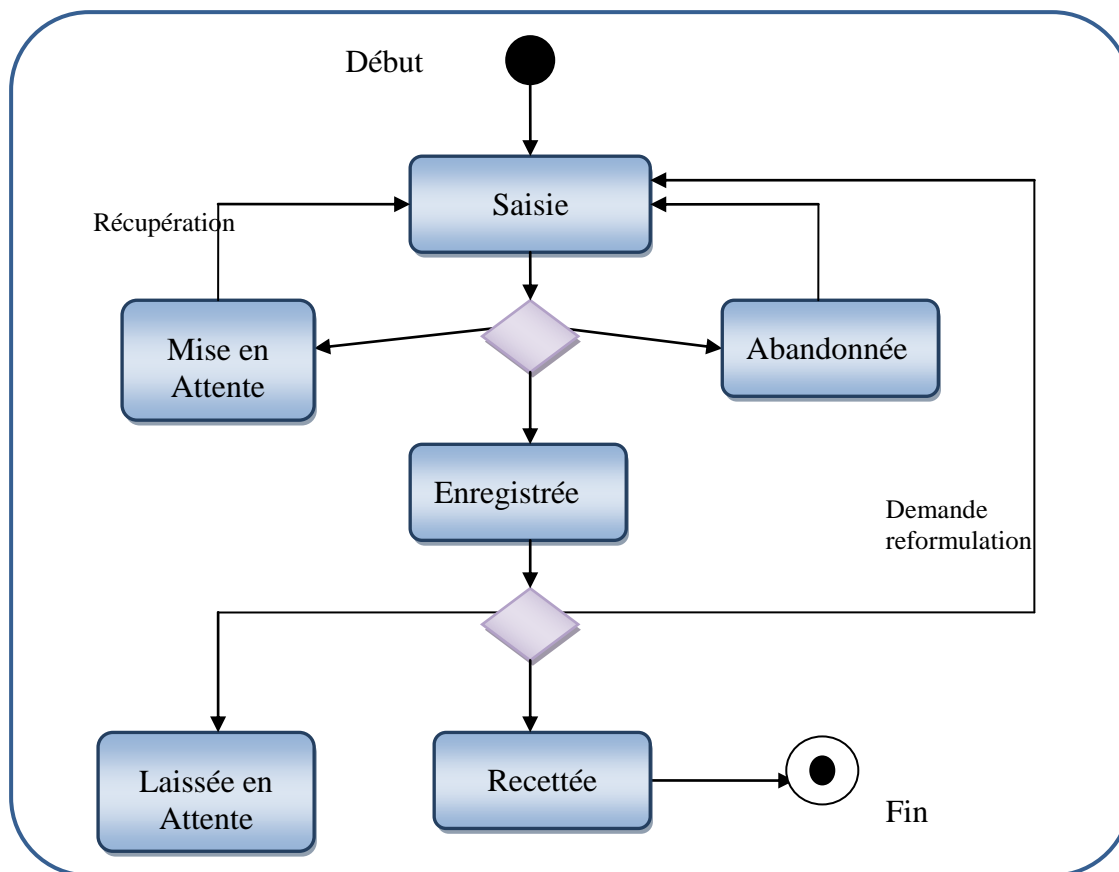


Figure 2.14 : Diagramme d'états de la classe commande

Comme illustré dans la figure précédente (cf. figure 2.14), la commande passe par plusieurs états : elle peut être saisie par le client, puis soit abandonnée, mise en attente ou enregistrée au cas où elle est envoyée au fournisseur, et dans cette étape, elle peut soit être recettée, soit laissée en attente si le fournisseur désire la traiter plus tard.

2.5 Processus de prise de commande

Toute opération de prise de commande lancée au niveau de l'outil de prise de commande se traduit par l'initialisation d'un processus. Ce processus décrit l'ensemble des actions nécessaires pour réaliser l'opération. L'objectif de ce processus est de passer une commande d'un nouveau service, produit ou prestation du catalogue. Pour ce faire, l'utilisateur client doit remplir un formulaire concernant la commande en question, ce formulaire est envoyé au fournisseur.

Lors de la réception de la synthèse de la commande, le fournisseur procède à son analyse, au cas où ce formulaire est complet, le fournisseur entame l'étude de faisabilité de la commande pour passer à l'étape de réalisation et d'installation du service sur le site concerné.

La figure ci-dessous (cf. figure 2.15) représente le processus de prise de commande :

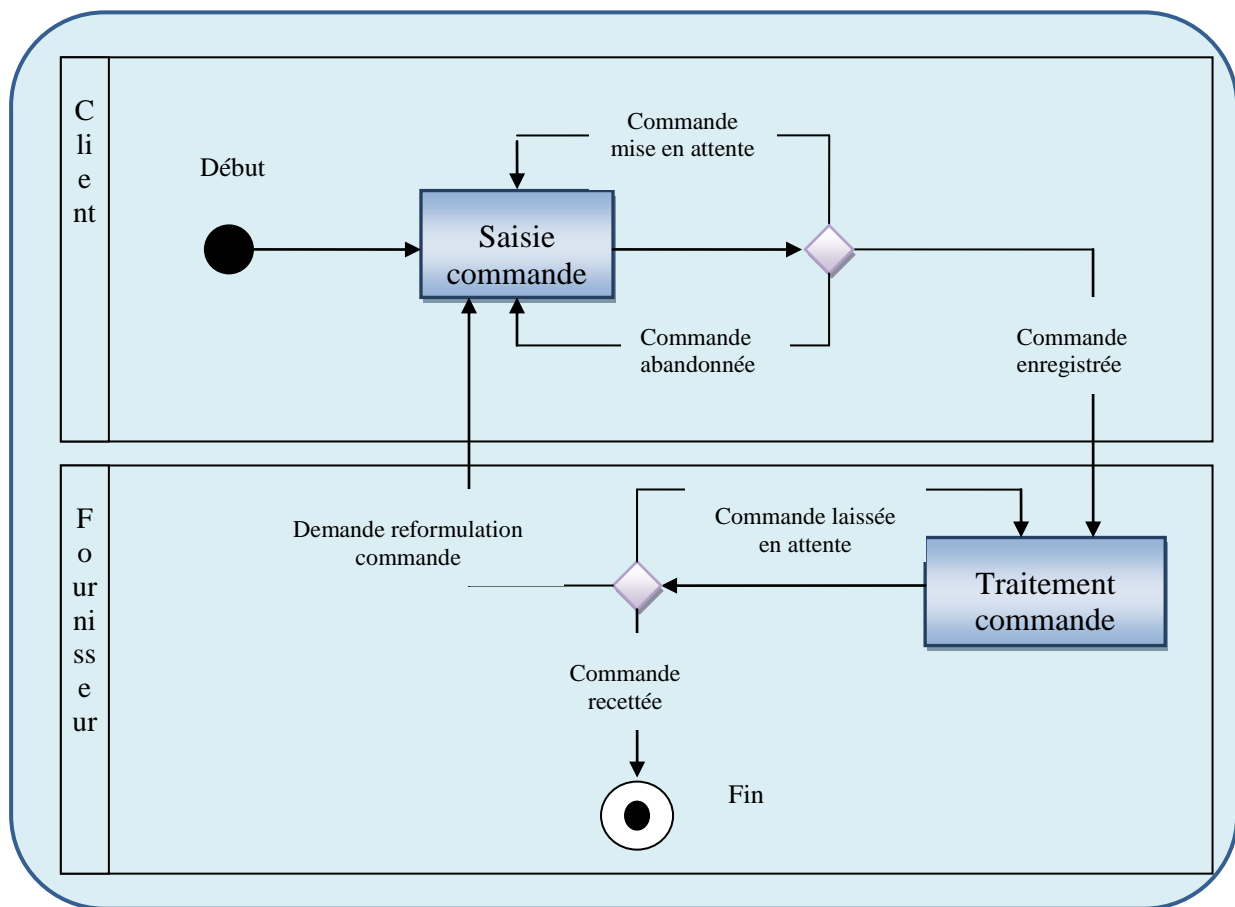


Figure 2.15 : Processus de prise de commande

Description textuelle du processus de prise de commande

Identifiant : Prise Commande.

Acteurs métier : Client, Fournisseur.

Résumé : l'objectif de ce processus est de passer une commande portant sur un nouveau service, pour cela le client doit remplir un formulaire permettant la saisie des informations concernant la commande en question. En envoyant ce formulaire par mail, le fournisseur procède à son analyse, au cas où ce formulaire est complet, le fournisseur entame l'étude de faisabilité de la commande pour passer à l'étape de réalisation et d'installation du service sur le site concerné.

Événements métier reçus : lancement d'une opération de commandes d'ouverture à partir du menu de la page d'accueil.

« Prise Commande » est déclenché à chaque fois que le client lance cette opération.

Pré-conditions :

- Le Client a un contrat avec France Télécom ;
- Le client s'est authentifié sur le système.

Description :

L'objectif de ce processus métier est d'effectuer des commandes de nouveaux services à partir du catalogue.

Post-conditions

- Le ou les nouveaux services sont « recettés » ;
- Les services sont actifs dans le parc.

2.6 Extraction des services

Cette partie décrit les services élaborés pour notre outil de prise de commande (cf. tableau 2.6). Les différents services se présentent comme suit :

Service	Utilisateurs	Description
Commande	Applications Externes Client	Le service « Commande » est un service Basique, il permet aux utilisateurs du système de passer une commande, de consulter les informations la concernant, de l'enregistrer, de la traiter et de la recetter. Ce service se charge également de la vérification des informations saisies lors de la passation de la commande et du changement des versions de la commande. Le service Commande est un service basic puisqu'il ne fait pas appel à d'autres services intermédiaires pour effectuer les opérations citées auparavant.
Commande Client	Applications Externes Client	Le service « Commande_Client » est un service Composite, il permet au client de passer une commande, de consulter les informations la concernant et de l'enregistrer, de l'abandonner ou de la mettre en attente. Le service Commande_Client est un service composite puisqu'il fait appel à d'autres services intermédiaires pour effectuer les opérations citées auparavant.
Commande Fournisseur	Applications Externes Client	Le service « Commande_Fournisseur » est un service Composite, il permet au fournisseur de traiter une commande, de la recetter ou bien de la mettre en attente. Le service Commande_Fournisseur est un service composite puisqu'il fait appel à d'autres services intermédiaires pour effectuer les opérations citées auparavant.

Tableau 2.6 : Les services de prise de commande

Conclusion

Au terme de ce chapitre, nous avons présenté l'étude préliminaire du projet, qui sert à avoir un premier regard sur les besoins fonctionnels et techniques du projet. Nous avons présenté également l'étude fonctionnelle du projet réalisé en respectant le processus de développement 2TUP. Ainsi, nous avons traité la branche gauche du modèle, où nous prêtons attention à la capture des besoins fonctionnels, traduits en spécifications, projetés ensuite en langage UML à travers les diagrammes des cas d'utilisation. Nous avons ensuite présenté les différents diagrammes de classes de nos principaux modules, et enfin nous avons explicité le processus de la prise de commande.

Dans le chapitre suivant, nous allons détailler l'étude technique du projet, à savoir les exigences requises par l'outil, l'architecture logicielle adoptée, et les Frameworks techniques utilisés.

Chapitre 3

Étude technique

Après avoir élaboré tous les diagrammes détaillant les fonctionnalités de l'application, vient la phase de l'étude technique. Cette phase a pour but de proposer une architecture orientée services (SOA) offrant plus de flexibilité et assurant le couplage faible entre les applications. Parmi les implémentations d'une telle architecture on trouve l'Enterprise Service Bus (ESB) que nous avons choisie pour réaliser notre projet. Dans le présent chapitre, nous présentons l'architecture orientée services et une étude comparative entre les différentes solutions open source de l'Enterprise service bus.

Introduction

Le choix de l'architecture technique est un facteur déterminant dans la performance de notre application. Nous avons opté pour une architecture orientée services qui est une réponse très efficace aux problématiques que rencontrent les entreprises en termes de réutilisabilité, d'interopérabilité et de réduction de couplage entre les différents systèmes qui implémentent leurs systèmes d'informations. Maintenant, le concept de SOA recouvre des aspects et des domaines très variés qui dépassent d'une certaine manière le domaine initial qu'est l'architecture logicielle. En effet l'un des points forts de cette approche a été de répondre à un fort besoin d'interaction entre les entités logicielles qui composent une application finale. L'approche SOA propose un mécanisme d'assemblage souple à l'aide de services, indépendant d'une technologie particulière et permet de définir les besoins métiers (les services) de l'application.

3.1 L'architecture orienté services

Depuis quelques années, la notion d'Architecture Orientée Services (SOA) s'est rapidement répandue et a été largement acceptée par l'industrie du logiciel grâce en particulier à la montée en puissance de l'Internet.

3.1.1 Définition

L'architecture orientée services (notée SOA pour Services Oriented Architecture) n'est pas une technologie ou un produit. C'est une façon de penser et de concevoir le Système d'Informations. C'est une architecture logicielle s'appuyant sur un ensemble de services simples. Son objectif est de décomposer une fonctionnalité en un ensemble de fonctions basiques, appelées services, fournies par des composants et de décrire le schéma d'interaction entre ces services qui interagissent et communiquent entre eux via des messages (figure 3.1).

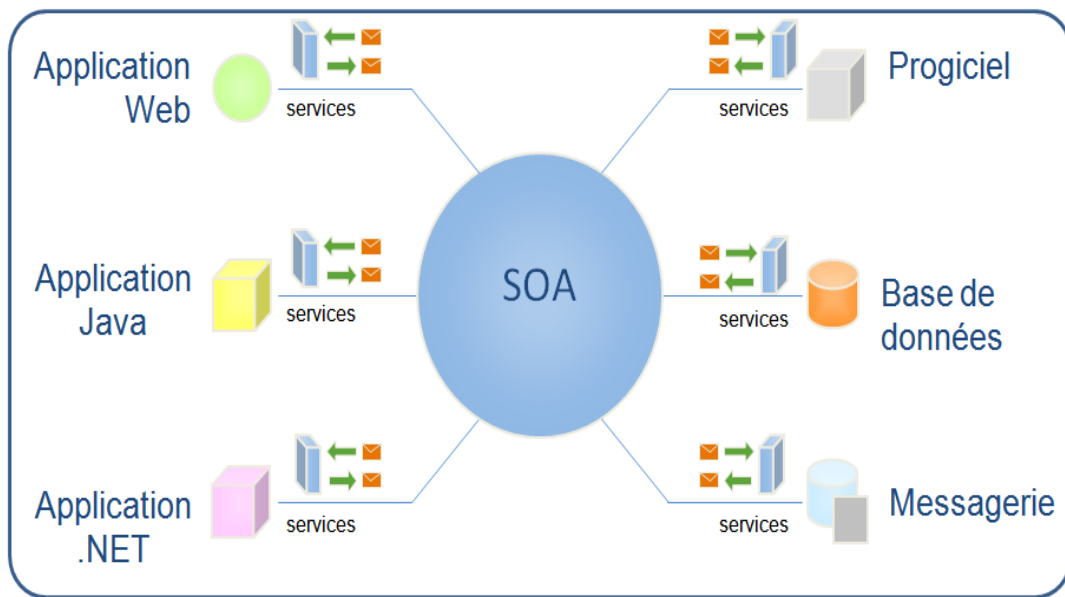


Figure 3.1 : SOA à la convergence de solutions

3.1.2 Principe de la SOA

L'architecture orientée service se base sur les principes suivants:

- **Diviser pour régner**

Substituer la découpe strictement applicative par une structuration en composants plus réduits et potentiellement plus simples à faire évoluer.

- **Alignement métier**

Construire et organiser le système à partir des réalités métiers, qui doivent se retrouver dans ses constituants.

- **Neutralité technologique**

Assurer une indépendance totale entre les interfaces et les implémentations. L'élément qui utilise un service ne doit pas être contraint ni par la technologie d'implémentation, ni par sa localisation (potentiellement distribuée).

- **Mutualisation**

Favoriser la réutilisation de services métiers par plusieurs lignes métiers ou applications et permettre la construction de services de haut niveau par combinaison de services existants.

- **Automatisation des processus métier**

Isoler la logique des processus métiers sur des composants dédiés qui prennent en charge les enchaînements et les échanges de flux d'information.

▪ Échanges orientés Document

Les informations échangées par les services possèdent une structure propre, guidée par les besoins métiers. On privilégie la transmission de contenus complets et utilisables au profit d'accès direct aux structures de type objet ou relationnel.

3.1.3 Éléments de base d'une architecture SOA

Il n'existe pas de spécifications officielles d'une architecture SOA, néanmoins les principales notions fédératrices que l'on retrouve dans une telle architecture sont les suivantes (figure 3.2) :

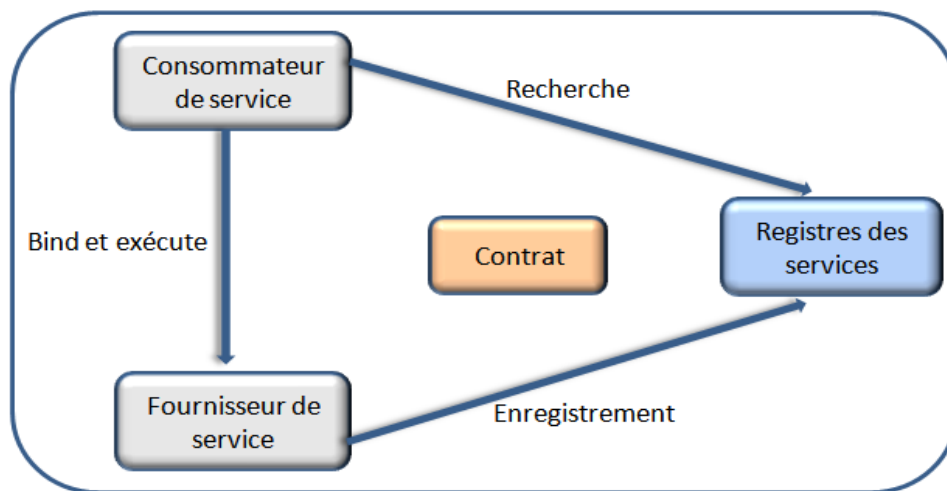


Figure 3.2 : publication et découverte des services

3.1.3.1 Le service :

Le service est l'unité atomique d'une architecture SOA. Une application est un ensemble de services qui dialoguent entre eux par des messages. Le couplage entre services est un couplage lâche et les communications peuvent être synchrones ou asynchrones.

Le service peut:

- Être codé dans n'importe quel langage ;
- S'exécuter sur n'importe quelle plate-forme (matérielle ou logicielle).

Le service doit :

- Offrir un ensemble d'opérations dont les interfaces sont publiées ;
- Être autonome (disposer de toutes les informations nécessaires à son exécution) ;
- Respecter un ensemble de contrats (règles de fonctionnement).

3.1.3.2 La description du service

La description du service consiste à décrire les paramètres d'entrée du service, le format et le type des données retournées. Le principal format de description de services est le WSDL.

- **WSDL** (figure 3.3)

Le Web Services Description Language (noté WSDL) est un langage normalisé par le W3C(World Wide Web Consortium) reposant sur la notation XML (eXtensible Markup Language) permettant de décrire les services web. Le WSDL permet ainsi de décrire l'emplacement du service web ainsi que les opérations (méthodes, paramètres et valeurs de retour) que le service propose.

3.1.3.3 Les Services Web

Les services web (notée WS pour Web Services) représentent un mécanisme de communication entre applications distantes à travers le réseau indépendamment de tout langage de programmation et de toute plate-forme d'exécution, et ceci en :

- Utilisant un protocole de transport comme http (HyperText Transfer Protocol) ;
- Employant une syntaxe basée sur la notation XML pour décrire les appels de fonctions distantes et les données échangées ;
- Organisant les mécanismes d'appel et de réponse.

3.1.3.4 La publication et la découverte des services (figure 3.2)

La publication consiste à publier dans un registre les services disponibles aux utilisateurs, tandis que la notion de découverte recouvre la possibilité de rechercher un service parmi ceux qui ont été publiés. Le principal standard utilisé est UDDI.

- **UDDI** (figure 3.3)

L'Universal Description Discovery and Integration (noté UDDI), normalisé par l'OASIS (Organization for the Advancement of Structured Information Standards), est un annuaire de services fondé sur XML qui vise à décrire une manière standard de publier et d'interroger les services web proposés par un réseau, il permet de localiser sur ce dernier le service Web recherché.

3.1.3.5 L'invocation

L'invocation représente la connexion et l'interaction du client avec le service. Le principal protocole utilisé pour l'invocation de services est SOAP.

- **SOAP** (figure 3.3)

Simple Object Access Protocol (noté SOAP) est un protocole permettant des appels de procédures à distances s'appuyant principalement sur le protocole HTTP et sur le XML. Il

permet de définir les mécanismes d'échanges d'information entre des clients et des fournisseurs de services web.

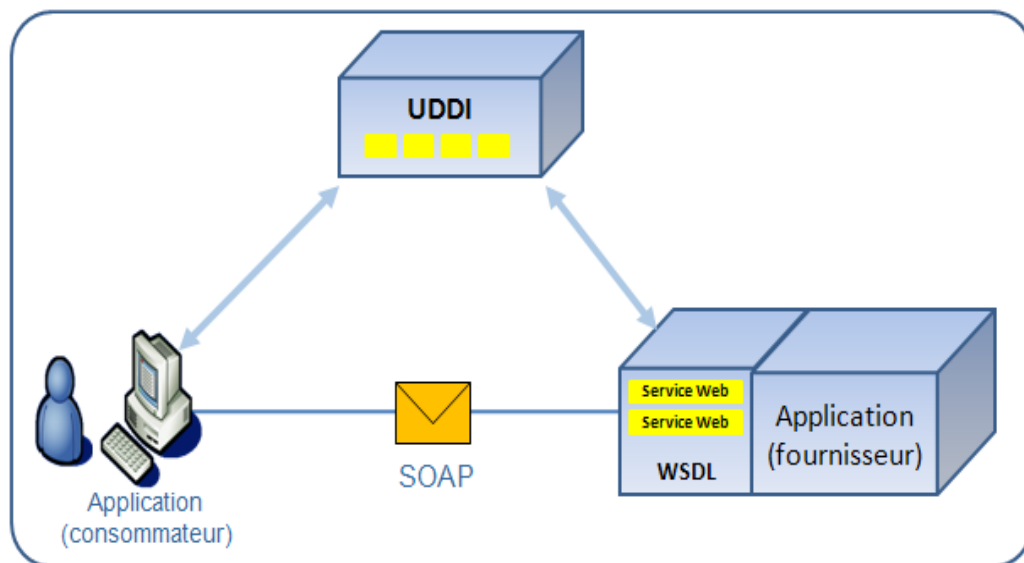


Figure 3.3 : SOAP, WSDL et UDDI en action

3.1.4 Les avantages d'une SOA

Une architecture orientée services permet d'obtenir tous les avantages d'une architecture client-serveur et notamment :

- Améliorer la rapidité ainsi que la productivité des développements ;
- Une modularité permettant de remplacer facilement un composant (service) par un autre ;
- Une réutilisabilité possible des services ;
- De plus grande tolérance aux pannes ;
- Une maintenance facile ;
- De meilleures possibilités d'évolution.

3.1.5 Les inconvénients d'une SOA

La SOA représente des avantages évidents liés à l'intégration et la création de nouveaux services, toutefois elle n'est pas épargnée de quelques inconvénients qui sont :

- Les coûts de conception et de développement initiaux plus conséquents ;
- La nécessité d'appréhender de nouvelles technologies ;
- L'existant non SOA dans les entreprises ;
- Les performances réduites pour des traitements simples (couches supplémentaires).

3.1.6 Implémentations basées sur SOA

En termes de technologies, les SOA ne sont pas étroitement liées à un éditeur ou une mouvance particulière. Il est ainsi possible d'envisager l'implémentation de projets respectant les caractéristiques des SOA, aussi bien sous une technologie propriétaire qu'avec des solutions Open source. Depuis le début des années 90, les implémentations SOA se sont succédées avec l'objectif de permettre une architecture de plus en plus fluide et de plus en plus simple. Parmi lesquelles on trouve :

3.1.6.1 ETL (figure 3.4)

Extraire-Transformer-Charger (noté ETL pour Extract-Transform-Load) est une technologie informatique qui assurent la synchronisation, la consolidation et la propagation des données entre deux systèmes disparates (souvent des bases de données). Elle extrait les données d'un système, pour mettre à jour, après un transcodage adéquat, un autre système.

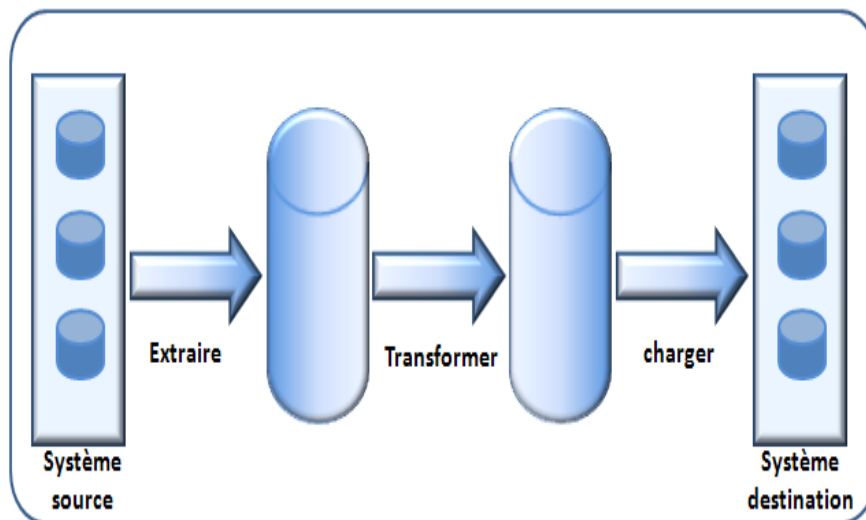


Figure 3.4 : La solution ETL

L'approche ETL ne répond pas aux besoins du projet, elle reste exclusivement centrée sur les données, et échoue en conséquence à résoudre les problématiques de l'intégration et de processus métiers.

3.1.6.2 Le middleware (figure 3.5)

Les solutions middleware fournissent une infrastructure technique assurant la médiation entre deux ou plusieurs systèmes. Leur rôle est d'assurer le transport d'un message d'un système vers un autre, avec un niveau de couplage plus ou moins important. Exemple : CORBA (Common Object Request Broker Architecture)

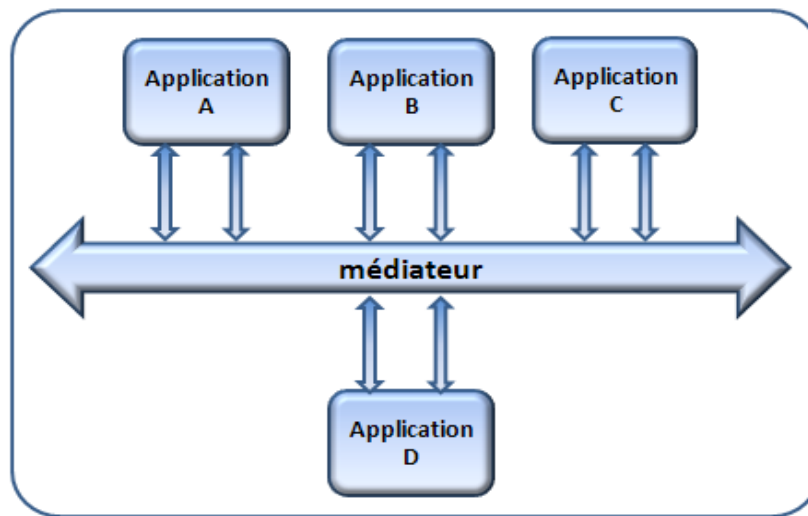


Figure 3.5 : Les médiateurs

Les middlewares ont une sémantique asynchrone : le client construit un message et le transmet au middleware, qui se charge de le router vers les systèmes cibles. La garantie de livraison du message est confiée au médiateur. Les middlewares offrent en outre des capacités de routage souvent limitées, qui obligent à des efforts de configuration importants, chaque route doit être explicitement définie, rendant leur mise en œuvre difficile à grande échelle, ils restent des solutions très techniques. Ils permettent certes à la fois la propagation des données et l'intégration des traitements, mais la sémantique des échanges y reste fondamentalement point-à-point.

3.1.6.3 Les EAI (figure 3.6)

L'Intégration d'application d'entreprise (notée EAI pour Enterprise Application Integration) est une architecture intergicielle permettant à des applications hétérogènes de gérer leurs échanges tirant parti des deux précédentes approches. Les EAI proposent une architecture dans laquelle un composant central assure la médiation physique entre le client et sa cible. Ce composant central prend à son compte l'ensemble des problématique techniques de bas niveau (localisation, disponibilité, communication, transcodage, sécurité). Ils permettent aussi d'assurer la transformation des données afin de limiter le couplage fonctionnel entre les systèmes, et d'appliquer des règles de routage.

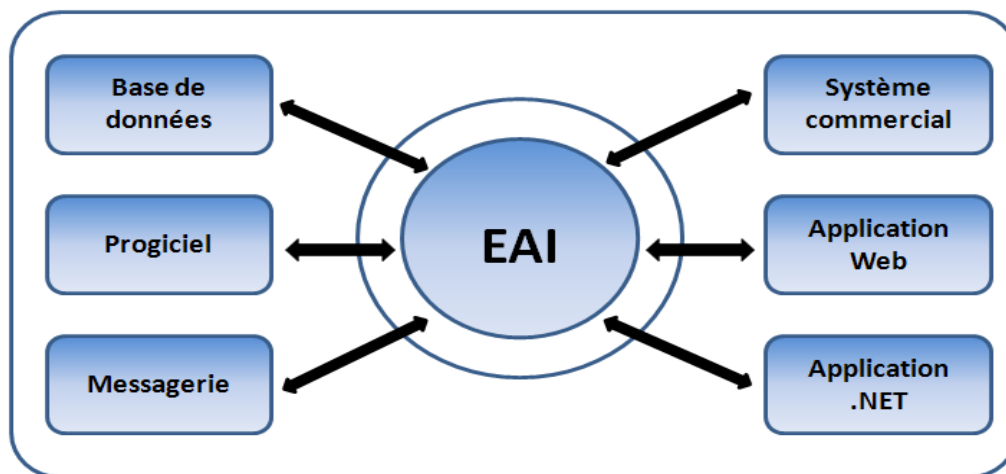


Figure 3.6 : L'Intégration d'applications d'entreprise (EAI)

Malgré leurs évidentes qualités, les solutions d'EAI souffrent de leur caractère très propriétaire :

- Le protocole utilisé pour les échanges et le transport des messages au sein d'un EAI est propriétaire ;
- La technologie interne aux EAI est propriétaire. Ainsi, l'accès aux applications se fait par l'intermédiaire de connecteurs spécifiques à chaque éditeur ;
- Les formats et encodages de données utilisés dans les EAI sont propriétaires.
- L'utilisation des EAI dans notre projet s'avère extrêmement coûteuse,

Les EAI sont des solutions d'intégration propriétaires qui emprisonnent les entreprises, interdisent la flexibilité, empêchent la réutilisation et s'avèrent extrêmement coûteuse.

3.1.6.4 Les ESB (figure 3.7)

Les Enterprise Service Bus (noté ESB) sont les héritiers directs des EAI, ils reprennent les caractéristiques architecturales de ses derniers et se concentrent sur les fonctions d'interconnexion et de médiation, et s'appuient pour cela sur un ensemble de standards orientés vers les services, qui facilitent les aspects opérationnels du processus métier, réduisent le temps, l'effort et les coûts de l'intégration d'applications.

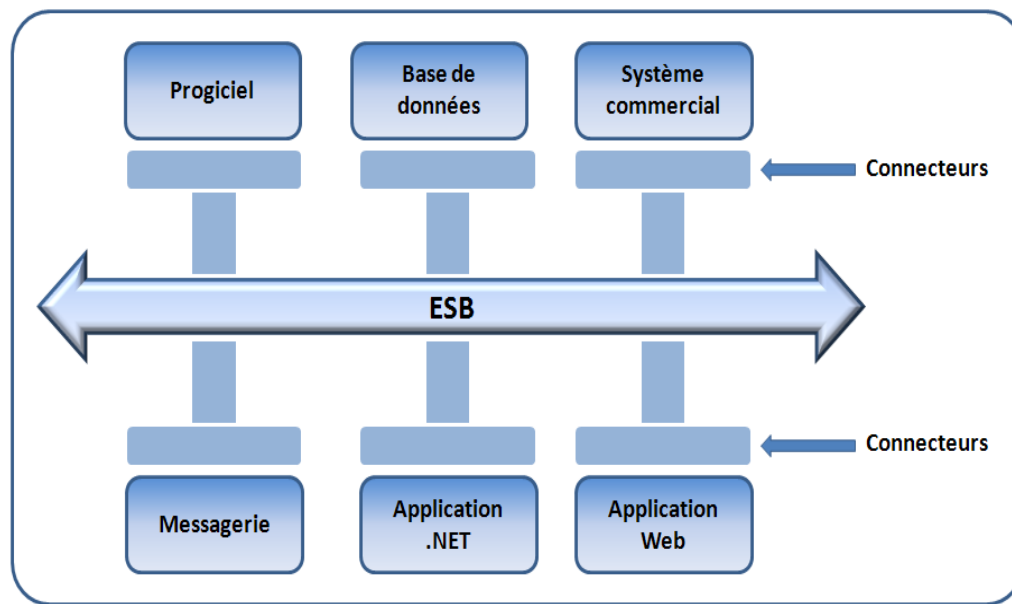


Figure 3.7 : L'Enterprise Service Bus (ESB)

Les ESB sont aujourd'hui la technologie d'intégration et de médiation inter-applicative privilégiée pour la mise en œuvre d'une architecture orientée service.

Nous avons choisis d'utiliser l'enterprise service bus car il propose une intégration complètement distribuée basée sur les standards offrant plus de flexibilité, de portabilité, et assure le couplage lâche entre les applications

3.2 L'Enterprise Service Bus

3.2.1 Définition

Un Enterprise Service Bus (figure 3.8) est une solution d'intégration implémentant une architecture totalement distribuée, son but est de permettre à des applications hétérogènes ou provenant de différents constructeurs de communiquer indépendamment de la plate-forme, des langages de programmation ou de la structure des données échangées. C'est un Middleware des architectures SOA s'appuyant sur des standards comme XML et les Services Web.

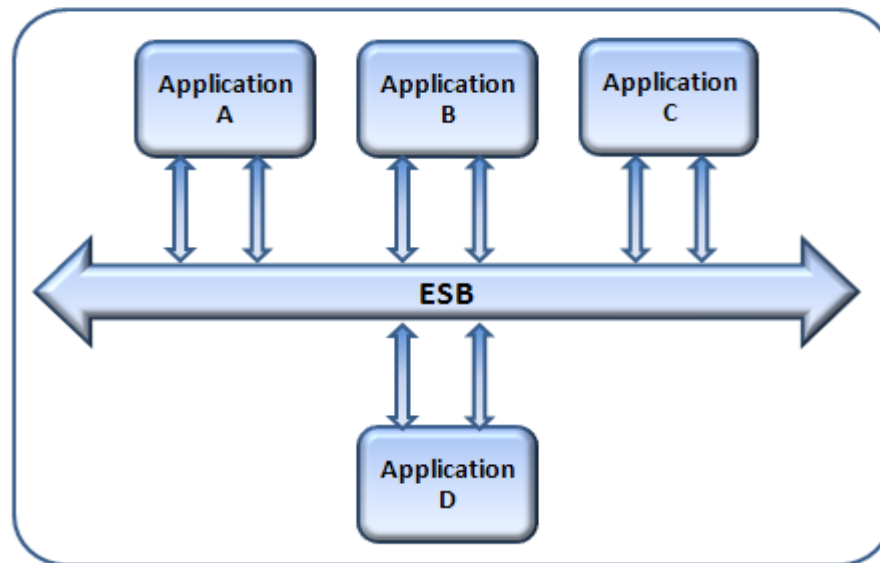


Figure 3.8 : Enterprise Service BUS

La notion de distribution est centrale pour un ESB. En effet, par essence les applications à intégrer sont réparties sur différentes machines. Par la mise en œuvre de ce principe de distribution, le « bus » de l'ESB devient virtuel, les données de configuration et d'administration étant alors distribuées sur les extrémités de l'ESB.

3.2.2 Schéma d'un ESB

Un ESB n'invente aucun nouveau concept, par contre il utilise les technologies standards existants sur le marché, rassemblées d'une façon à construire une architecture orientée services. Il insiste en revanche sur les interfaces standards des services web entre ses composants et met en place :

- **Un serveur d'intégration compatible services web** : une application désirant solliciter un composant présent sur le réseau, ou exposer ses services sur le réseau, le fera au travers d'un serveur d'intégration, conçu à cet effet. Les aspects de qualité de service liés à l'exposition du service sur le réseau seront déclarés par l'application, et gérés de manière transparente par le serveur d'intégration.
- **Des interfaces standardisées**, entre les applications et le serveur d'intégration: une application réalisée à l'aide d'un langage donné peut invoquer les services du serveur d'intégration. La demande de ces services se fait à l'aide d'une API standard liée au langage utilisé, ce qui isole l'application de toute API propriétaire d'un serveur d'intégration.

- **Un protocole de communication standard services web**, intégrant nativement les notions de qualité de service indispensables à l'intégration d'applications à grande échelle (confidentialité, intégrité, non répudiation, livraison garantie, etc).
- **Un conteneur de services web et d'orchestration intégré au serveur d'intégration**, afin de supporter divers composants relatifs à la logique d'intégration (transformation de message, enrichissement, gestion de processus longs, etc). Ces composants sont standard services web, ce qui les isole des aspects propriétaires des conteneurs.

3.2.3 Eléments d'un ESB

L'Enterprise Service Bus possède comme son nom l'indique une architecture en bus, ce qui fait de l'ESB une solution hautement distribuée. L'ESB se base sur quatre fondements (figure 3.9) :

- Le **Middleware Orienté Message** (noté MOM pour Middleware Oriented Message) qui permet l'échange de messages de manière asynchrone entre les différentes applications et offre une garantie de livraison. Ainsi chaque message est déposé sur une file d'attente avant d'être consommé par le destinataire.
- La **transformation** des informations qui circule dans le bus. Cette transformation s'effectue au niveau de l'ESB pour garantir l'interopérabilité entre les applications.
- Le **routage intelligent** ou le routage basé sur le contenu. c'est-à-dire déduire la destination du message. Pour cela l'ESB se base sur le contenu du message et les règles qui ont été définies.
- Les **services Web** qui permettent d'interfacer les applications avec le bus.

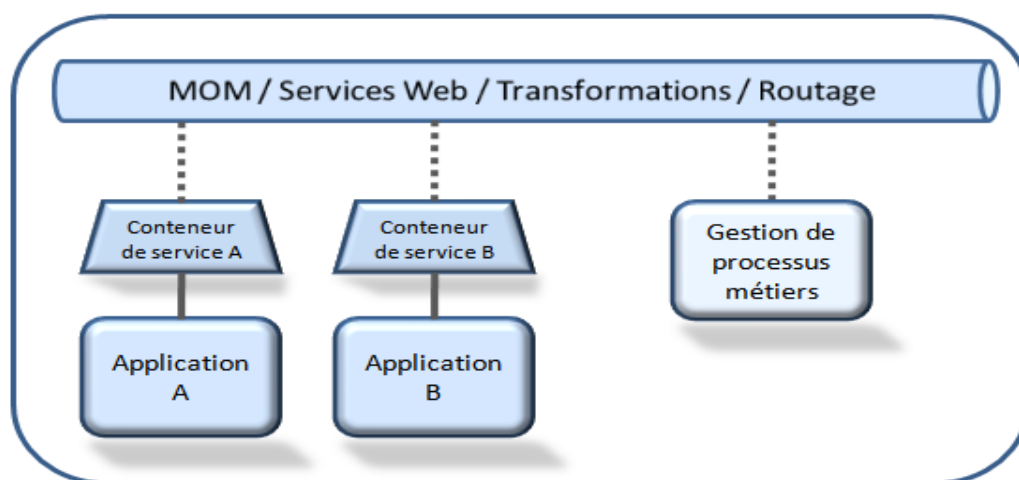


Figure 3.9: Les fondements d'un ESB

3.2.4 L'ESB est basée sur les standards

Les ESB sont basées sur des standards reconnus, ce qui facilite leur interopérabilité et l'interconnexion des applications et des systèmes d'informations de deux entreprises même différents. Les messages qui circulent dans le bus peuvent être transmis au bus du partenaire. A terme, les aspects techniques de transaction, sécurité, etc. seront supportés complètement par la mise en œuvre des normes des services web spécifiques à chaque problématique.

3.2.5 Raisons d'utilisation d'un ESB

Une solution d'ESB est destinée à intégrer les applications selon un principe de couplage lâche, où les applications peuvent évoluer et fonctionner, dans une certaine mesure, indépendamment les unes des autres. Pour mettre en œuvre ce principe, l'ESB a répondu à un ensemble de besoins et de contraintes :

- **Nécessité d'intégration des applications :** Il faut avoir un besoin clair pour faire intégrer des applications de l'entreprise comme :

- Rapports en temps réel
- Réduction du temps de mise des produits sur le marché

- **Environnement hétérogène :** Quand on est obligé à traiter beaucoup de technologies et de protocoles différents, alors c'est mieux de mettre en place une solution centralisée pour faire face à ces défis.

- **Réduction du coût total :** Les entreprises cherchent à diminuer le coût total des entretiens à faire en mesure de satisfaire tous les nouveaux produits. une architecture centralisée peut aider à réduire ces coûts.

3.2.6 Les fonctionnalités d'un ESB

L'ESB offre un ensemble de fonctionnalités, Le tableau suivant (cf. tableau 3.1) synthétise les principales fonctionnalités que l'on peut y attendre :

Fonctionnalité	Description
Connectivité	Supporte de multiples protocoles de transport synchrone ou asynchrone. Il faut voir l'ESB comme un "super-connecteur". Son rôle est de se connecter à tout type de ressources ou fournisseurs de services. C'est grâce à cela qu'il permet une exposition en couplage lâche de fournisseurs de services.
Routage	Permet d'effectuer le routage des messages basé sur des règles (de contenu, de contexte, etc.). Idéalement ce routage s'appuie sur un annuaire, sur un registre de services et éventuellement sur un ensemble de règles.

Médiation	Adapte le format des messages et le protocole, effectue des transformations des messages entre l'appelant et l'appelé.
Exposition de services	Transforme en service(s) un composant ou un traitement d'une application
Agrégation simple de services	Effectue des agrégations simples de services de niveau N pour construire des services de niveau N+1. Si l'agrégation est complexe on utilise un moteur d'orchestration
Traitement d'événements complexes	Permet la création des règles de corrélation et de jointure d'événements.
Contrat de services	Permet la définition des contrats de services : niveau de QoS (quality of service), gestion des priorités, sécurisation des messages (signature et cryptage), garantie de la livraison et de l'intégrité des messages.
Supervision et audit	Offre des fonctions d'audit, de traçabilité, de mesure, d'administration et d'exploitation qui permettent le suivi des traitements. Selon la qualité de l'implémentation, cette fonctionnalité sera déléguée à un outil tiers.

Tableau 3.1 : Les fonctionnalités d'un ESB

3.2.7 Types de messages supportés par un ESB

L'ESB se définit comme un framework de messages destiné à gérer la communication entre des services et des applications utilisant des fournisseurs de transport et des technologies de messages disparates. Les types de messages qu'un ESB peut supporter sont :

- **Multicast** : Envoi des messages à plusieurs destinataires (figure 3.10)

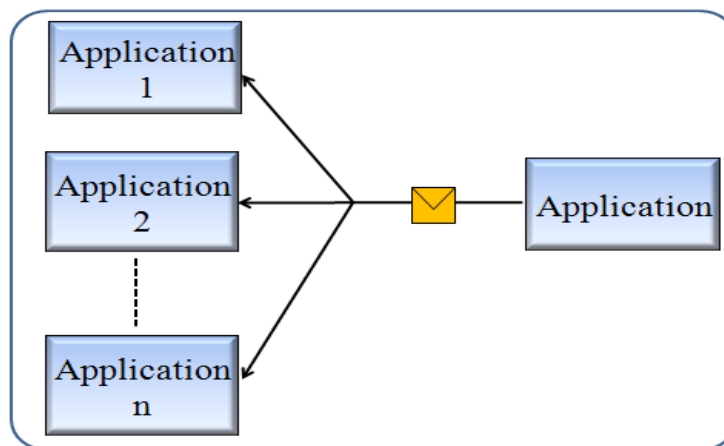


Figure 3.10: Message multicast

- **Point to point (P2P)** : ce protocole est basé sur le concept de queue de messages, d'envoyeurs et de receveurs. Chaque message est adressé à une queue spécifique et les clients

consomment leurs messages. Ce protocole doit être utilisé pour s'assurer qu'un seul client consommera le message (figure 3.11).

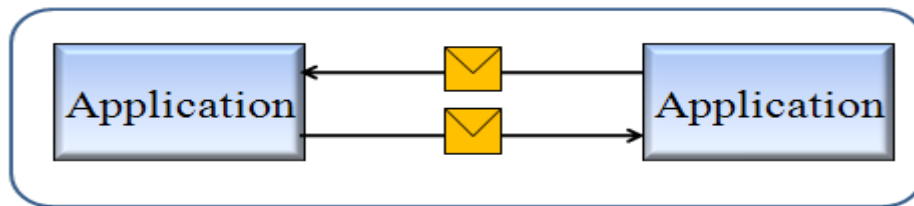


Figure 3.11: Communication point à point

▪ **Publish/subscribe** : les clients sont anonymes et adressent les messages selon certaines rubriques. Le système distribue le message à l'ensemble des souscripteurs puis l'efface (figure 3.12).

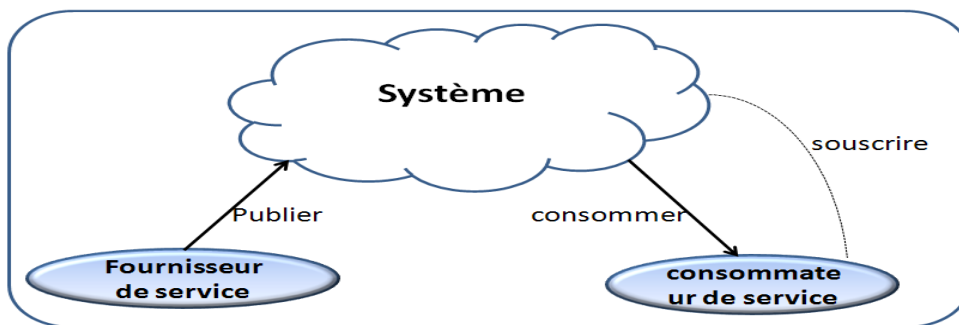


Figure 3.12: Publish / subscribe

- **Synchrone ou asynchrone** : un client consomme les messages de façon :
 - **Synchrone** :
 - L'émetteur attend que le récepteur ait lu le message
 - Le récepteur qui attend un message est bloqué jusqu'à son arrivée
 - **Asynchrone** :
 - L'émetteur n'est pas bloqué en attente de la réception
 - Le récepteur a un rythme autonome de réception

3.2.8 Les avantages et les inconvénients d'un ESB

3.2.8.1 Les avantages

▪ **Flux centralisés** : toutes les interfaces de l'ESB convergent vers un bus central, qui traite et redistribue les flux vers les applications enregistrées ;

- **Coût de migration des interfaces :** Lors du changement d'une des applications interfacées (migration, changement de produit), peu de modifications sont nécessaires. Seuls le connecteur, le mappage ou la collaboration spécifique à l'application doivent être modifiés ;
- **Utilisations des standards du web :** Les ESB se concentrent sur les fonctions d'interconnexion et de médiation, et s'appuient pour cela sur un ensemble de standards, parmi lesquels on trouve les web services, XML,...etc. rendant ainsi les applications indépendantes des langages de programmation et des plates formes ;
- **Tolérance aux pannes :** si une partie du système d'information tombe en panne cela ne va pas influencer le fonctionnement de tout le système ;
- **Maintenabilité :** capacité à corriger les erreurs de certains modules sans effet sur les autres ;
- **Sécurité :** confidentialité, cryptage des messages et contrôle d'accès.

3.2.8.2 Les inconvénients

- **Coût initial :** Le coût de mise en place de l'infrastructure est assez élevé. Mais il se réduit grandement au fur et à mesure de l'ajout de nouveaux flux ;
- **Complexité de mise en œuvre :** un ESB nécessite de très grande connaissance en services web et un savoir faire techniques assez riche ;
- **Un ESB seul ne permet pas de mettre en place du BPM** (Business Process Management). L'ESB prend en charge l'agrégation de services ou l'enrichissement de données mais il ne faut pas le considérer comme un orchestrateur de processus ;
- De part sa position centrale, **l'ESB peut aussi devenir un goulet d'étranglement.** Plus les médiations sont complexes plus les pertes de performances peuvent augmentées.

Les ESB reprennent les grands principes de l'EAI, mais l'utilisation poussée de standards rend leur coût de licence beaucoup plus abordable, ils offrent l'interopérabilité des applications et permettent aux compétences et aux acquis en logiciels d'être facilement transférés dans les projets d'intégration.

3.3 Étude Comparative des ESB Open Source

Durant notre stage à Sofrecom Services Maroc, nous étions emmené à faire une étude comparatives entre les ESB open source afin de choisir une solution qui va convenir le mieux à notre applications. Le choix de l'outil dépend de plusieurs contraintes techniques que nous allons étudier dans ce qui suit.

3.3.1 Présentation des principaux outils Open Source

L'étude que nous avons effectuée, s'est focalisée sur les ESB open source des plus grands éditeurs de solutions informatiques sur le marché. Nous avons pris les quatre outils suivants :

- **JBossESB** de JBoss

JBoss Enterprise Service Bus est un ESB acquis, mis en Open Source et développé par JBoss depuis mi 2006. Le logiciel est un composant de la stratégie SOA de JBoss. La solution est issue de l'ESB Rosetta acquis par JBoss en juin 2006. Les fonctionnalités clés de JBoss ESB intègrent la prise en charge de plusieurs services de messagerie, un moteur de transformation pour pallier aux différents formats de données, un registre de service, un référentiel d'événements, un service de notification, etc.

- **OpenESB** de Sun Microsystems

OpenESB est un projet Open Source pour créer une plate-forme d'intégration métiers et SOA. Un grand nombre de ses composants est en cours de développement dans la communauté OpenESB : l'accès aux systèmes d'ESB et d'autres plates-formes SOA, transformation des messages, etc.

- **PETALS** d'EBM WebSourcing

Le projet PETALS s'inscrit dans l'initiative ESB lancée par ObjectWeb en juin 2004 afin de fédérer les efforts autour du projet open source dans le domaine du middleware d'intégration. PETALS est un bus d'entreprise orienté services hautement distribué. Il a été construit sur les spécifications JBI (Java Business Integration). Le projet est dirigé par EBM Websourcing. Les principaux objectifs du projet sont de proposer une implémentation d'une plateforme JBI prenant en compte les problématiques d'environnements d'intégration hautement distribués.

- **Mule** d'Ippon Technologies

Mule est à la fois un ESB et un EAI, servant de plate-forme pouvant gérer les échanges entre de multiples applications utilisant des technologies disparates. Mule supporte plus de 30

standards, protocoles et technologies. Le projet open source est supporté par la société Mule Source Inc.

3.3.2 Critères du Choix d'un outil

Afin de choisir la meilleure solution qui va répondre le mieux aux besoins de notre application, des critères ont été fixé pour faire valoir les caractéristiques de chaque outil. Le choix a été dicté principalement par :

- L'adéquation des possibilités de l'outil aux besoins d'intégration (connecteurs, cohérence avec l'existant de l'établissement, possibilités d'orchestration, ...) et au niveau de criticité à supporter;
- Le prix d'acquisition et de mise en œuvre;
- Les compétences disponibles sur le marché ou en interne, et donc le niveau de maîtrise possible de la technologie;
- La pérennité de la solution et le pouvoir de l'éditeur de continuer à améliorer son outil pour avoir la garantie de satisfaire les besoins future de l'application.
- L'existence de standards pour l'échange de messages.

3.3.3 Besoins techniques

Pour développer notre projet, nous avons opté pour les nouvelles technologies de la plate forme J2EE (Java Enterprise Edition), allant du mappage et la persistance des données, jusqu'à la représentation des interfaces utilisateurs, nous présentons dans ce qui suit les technologies qu'on va utiliser dans notre projet :

3.3.3.1 La technologie J2EE

La technologie Java 2 Entreprise Edition (J2EE) définit la norme de développement des applications d'entreprise distribuées et multi tiers. Elle permet de simplifier les applications d'entreprise en les basant sur des composants modulaires normalisés tout en fournissant un ensemble complet de services et en gérant automatiquement un grand nombre de détails sur le comportement des applications. Elle est composée de plusieurs technologies qui sont regroupées en trois catégories : composant, service et communication.

J2EE définit une architecture multi-tiers basée sur des composants (figure 3.13). Avec ce modèle, les applications peuvent interagir facilement avec le système d'information et les composants implantant la logique d'entreprise sur des serveurs d'applications.

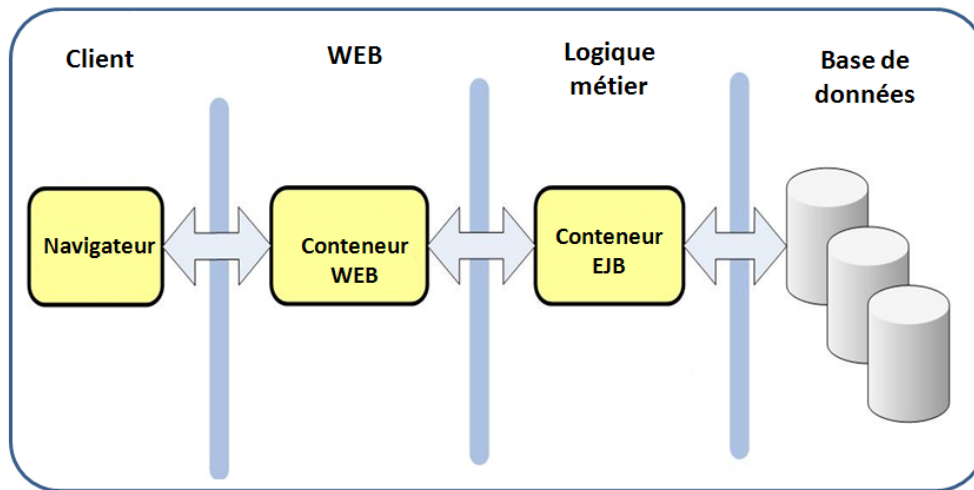


Figure 3.13: Architecture multi-étages dans l'environnement J2EE

L'architecture J2EE consiste en trois parties :

- « Components » : les composants qui prennent en charge la présentation et la logique métier ;
 - « Containers » : fournit un contexte aux composants ;
 - « Connectors » : les connecteurs fournissent l'accès aux sources de données de l'entreprise.
- Cet environnement supporte trois types de composants d'applications, dont les définitions seront présentées par la suite, JSF, JPA, EJB.

3.3.3.2 Java server faces

Java Server Faces (abrégé en JSF) est un framework Java basé sur les composants, pour le développement d'applications Web et fait partie du standard Java EE. La première version a vu le jour en 2003. JSF offre à la fois un ensemble de composants graphiques/visuels pour composer les interfaces utilisateur et une architecture côté serveur qui permet de lier ces interfaces avec des classes/traitement Java.

Les buts de JSF sont :

- De faciliter le développement des pages web grâce à un framework basé sur les composants ;
- De fournir un framework simple pour gérer ces composants, leurs interactions et leurs cycles de vie ;
- Faciliter la validation des formulaires ;
- Faciliter l'internationalisation des sites ;

- Automatiser au maximum la génération de code du côté client afin de ne pas avoir à se soucier des différences entre chaque navigateur.

3.3.3.3 Enterprise Java Bean

Les EJB (Enterprise Java Bean) sont des composants tournant sur des serveurs d'applications EJB (figure 3.14) qui permettent d'offrir des services de données. De plus, le Framework EJB sur le serveur assure la sécurité, la persistance, le support réseau et la gestion des transactions aux composants.

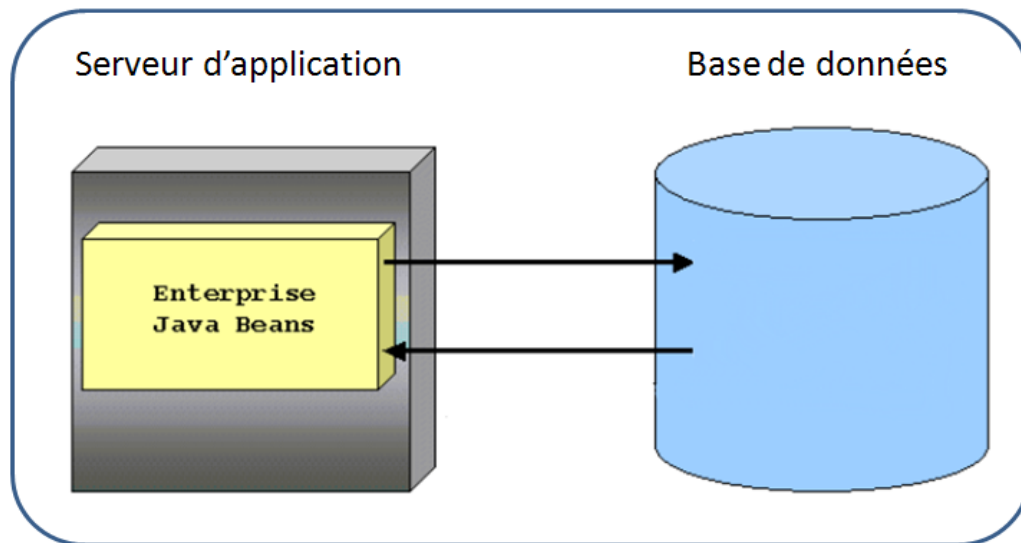


Figure 3.14: Enterprise Java Bean

3.3.3.4 JPA :

Java Persistence API (abrégé en JPA) [Patricio, 08] est une API de persistance des données (figure 3.15), qui fait partie de la spécification EJB3. Cette API réalise la fusion des travaux sur Hibernate avec la continuité des spécifications EJB précédentes 2.0 et 2.1.

Les avantages de JPA sont :

- Utilisation possible de l'héritage des objets ;
- Souplesse d'écriture du code de persistance ;
- Possibilité de déploiement dans ou hors un conteneur EJB ;
- Facilité de tester unitairement les requêtes ;
- Proche des concepts Hibernate.

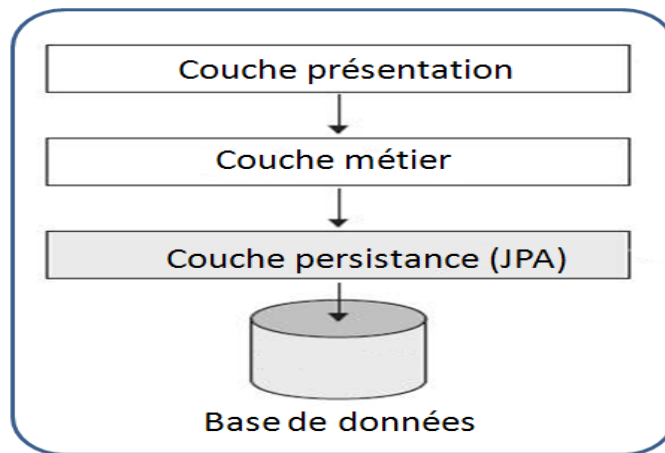


Figure 3.15: Java Persistence API

3.3.3.5 SGBD MySQL

MySQL [MySQL-Server] est un système de gestion de base de données (SGBD) de type SQL (Structured Query Language) créé en 1995 par une société suédoise. Il est aujourd'hui développé et distribué par la société commerciale MySQL. MySQL est un serveur de base de données SQL très rapide, multithreads, multiutilisateurs, stable et très fiable. Il est connu dans le monde des logiciels open source comme étant le plus rapide de sa catégorie. Il est également facile à utiliser.

3.3.3.6 JBPM

Java Business Process Management (abrégé JBPM) [JB-JBPM] est une plate-forme d'exécution des processus allant de la gestion des processus métier (BPM) et le workflow jusqu'à l'orchestration des services. C'est un logiciel libre développé par JBoss. Il permet la gestion de flux d'information ainsi que la coordination entre biens et personnes. Il permet de poser graphiquement les différentes étapes qui composent un processus avec des états, des transitions, et des affectations de tâches (figure 3.16).

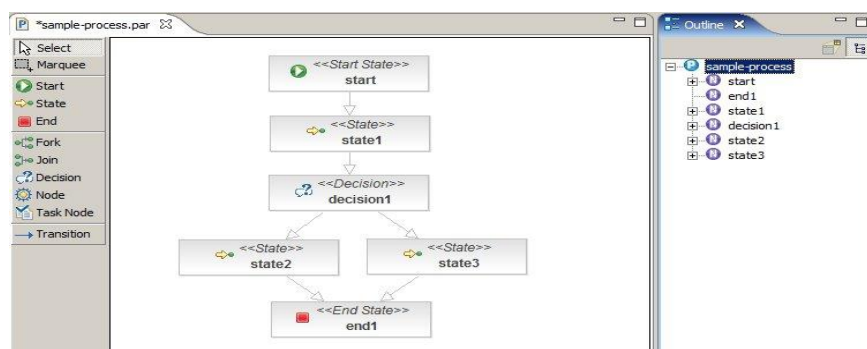


Figure 3.16: Interface de JBPM

3.3.3.7 JBossSEAM

Java Enterprise Edition 5 propose deux technologies de « haut niveau » pour développer des applications Web : Java Server Faces pour la partie présentation et EJB 3.0/JPA pour la partie métier. Avant, ces deux technologies coexistent plus qu'elles ne coopèrent : il n'existait aucune API permettant de les coupler aisément. C'est à cet exercice que s'est livrée une équipe de JBoss, le résultat est un produit spécifique JBoss appelé « SEAM » [JB-SEAM] [Kittoli, 09] (figure 3.17).

Seam est une API d'intégration qui intègre d'abord et avant tout Java Server Faces et EJB 3.0/JPA. Elle permet aussi d'intégrer JPBPM (solution de Workflow JBoss). L'idée fondamentale de « Seam » est d'utiliser les POJO EJB 3.0 comme bean JSF. Les expressions « JSF Expression Language » atteindront alors des méthodes et propriétés publiques de l'EJB. Ainsi les deux mondes sont réunis : la couche de présentation manipule des objets disposant directement de la panoplie des services proposés par la norme EJB, en particulier la gestion des transactions et la sécurité.

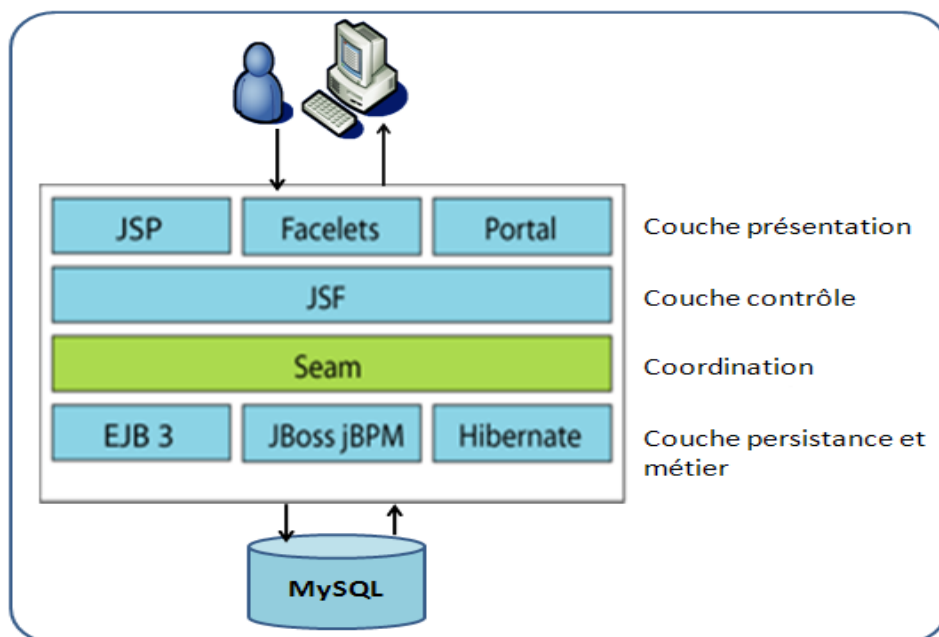


Figure 3.17: Intégration de Seam dans une application Web

3.3.3.8 Eclipse

Eclipse [Eclipse] est un environnement de développement intégré dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques. Eclipse utilise énormément le concept de modules nommés "plug-ins" dans son architecture. D'ailleurs, hormis le noyau de la plate-forme nommé "Runtime", tout le reste de la plate-forme est développé sous la forme de plug-ins. Ce concept permet de fournir un mécanisme

pour l'extension de la plate-forme et ainsi fournir la possibilité à des tiers de développer des fonctionnalités qui ne sont pas fournies en standard par Eclipse.

Après avoir défini les besoins techniques, nous avons relevé les Framework, les API de programmations, les éditeurs, les serveurs d'applications et les protocoles de communication, etc. Qu'utilisent chaque ESB afin de s'assurer que ses technologies répondent aux besoins de notre application.

Le tableau suivant récapitule les différents aspects techniques de cette étude (cf. tableau 3.2) :





ESB caractéristiques	JBoss ESB 	Open ESB 	Mule 	etals 
Version	4.5 GA	2.0	2.2.0	2.3
Serveur d'application	JBoss AS	Glassfish	Apache Tomcat	JONAS
Orchestration	BPEL	BPEL module project	BPEL	BPEL
BPM	JBPM/JPDL	BPEL Designer	JBPM	Orchestra/Intalio
IDE	Eclipse/JBdevstudio	Netbeans	Eclipse	Eclipse
Support EJB 3	✓	✓	✓	✓
Intégration	JB/JCA	JB/JCA	JB/JCA	JB/JCA
Web services	✓	✓	✓	✓
Annuaire Web service	UDDI/JAX-R	UDDI/JAX-R	UDDI	UDDI
Transport	JMS (JBossMessaging) SOAP/HTTP(S)/(S)FTP/ email/file/DB	JMS/HTTP/FTP SOAP /file/DB	JMS/HTTP/FTP SOAP/SMTP/file	JMS/SOAP/HTTP/ email/file
Transformation	Smooks/XSLT	XSLT	Smooks/XSLT	XSLT
Routage	Jboss rules	NMR	NMR	NMR
Documentation	Exemples/guide utilisateur	Exemples/guide utilisateur	Exemples/guide utilisateur	Exemples/guide utilisateur

Tableau 3.2: principales technologies des ESB open source

Nous avons comparé les outils de chaque ESB avec les besoins technique du projet (figure 3.18), et nous avons constaté que Mule et Petals n'offrent pas une architecture totalement

open source, certains de leurs outils sont payants et requièrent des licences pour pouvoir les utiliser. Nous avons continué notre étude avec JBossESB et OpenESB, et on a trouvé que ce dernier ne supporte pas le JBPM et SEAM qui sont des technologies JBoss, et que son IDE est Netbeans qui n'est pas totalement extensible et utilise presque juste les produits de SUN. Par contre JBoss utilise Eclipse qui est un outil de développement très ouvert auquel on peut lui ajouter d'autres fonctionnalités et framework.

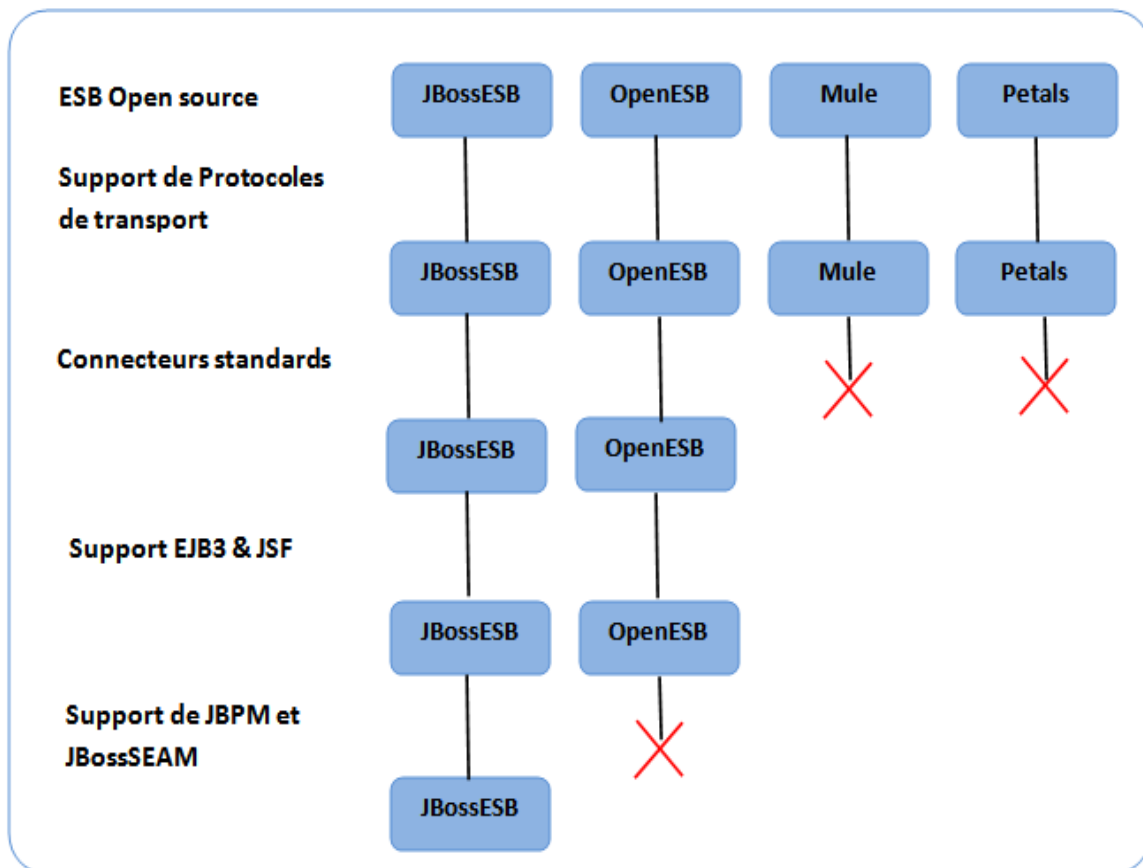


Figure 3.18: Etude comparative des ESB OS

Nous avons choisis de réaliser notre projet avec JBossESB [JBossESB], car il répond aux exigences techniques du projet, et offre une boîte à outils complète allant de l'IDE jusqu'à le serveur d'application avec une garantie de compatibilité (produits du même constructeur).

Conclusion

Dans ce chapitre, nous avons exposé la synthèse de notre étude comparative sur les ESB open source. Nous avons tout d'abord commencé par une présentation de l'architecture orientée services, puis une présentation de l'Enterprise Service Bus. Ensuite, nous avons effectué une étude comparative entre les ESB open Source les plus populaires, nous avons déterminé un ensemble de critères sur lesquelles s'est basée cette étude comparative. Cette dernière nous a menés aux choix de JBossESB suite à notre constat de la compatibilité de ses outils et leurs performances.

Après la constitution du dossier technique, et après avoir justifié notre choix, nous avons commencé la mise en œuvre du projet qui fera l'objet du chapitre suivant.

Chapitre 4

Réalisation & mise en œuvre

Après avoir mené à bien les phases précédentes, passant par l'étude comparative, ainsi que l'étude fonctionnelle et technique de l'application, l'étape suivante de notre démarche pour ce projet est la phase de mise en œuvre.

Dans ce dernier chapitre, nous allons présenter la phase de développement des différents modules issus de l'étude fonctionnelle. Puis on va entamer la partie de la mise en œuvre de l'ESB.

Introduction

Pour implémenter l'architecture logicielle proposée de notre solution, nous avons opté pour le modèle MVC (Modèle Vue Contrôleur), car il répond au besoin de la séparation en couches proposée par notre architecture, et de l'indépendance entre l'aspect présentation et celui technique, chose qui rendra le système plus flexible, plus évolutif et facile à maintenir.

4.1 Conception

La conception du système consiste à reprendre le modèle d'analyse à la lumière des choix technologiques. Son objectif est de reprendre le modèle d'analyse et le refaire d'une façon plus raffinée pour préparer la phase de réalisation.

4.1.1 Conception préliminaire

La conception préliminaire explique comment organiser un modèle de conception au vu des regroupements d'analyse et des couches logicielles d'architecture. Elle permet de passer de l'analyse objet à la conception.

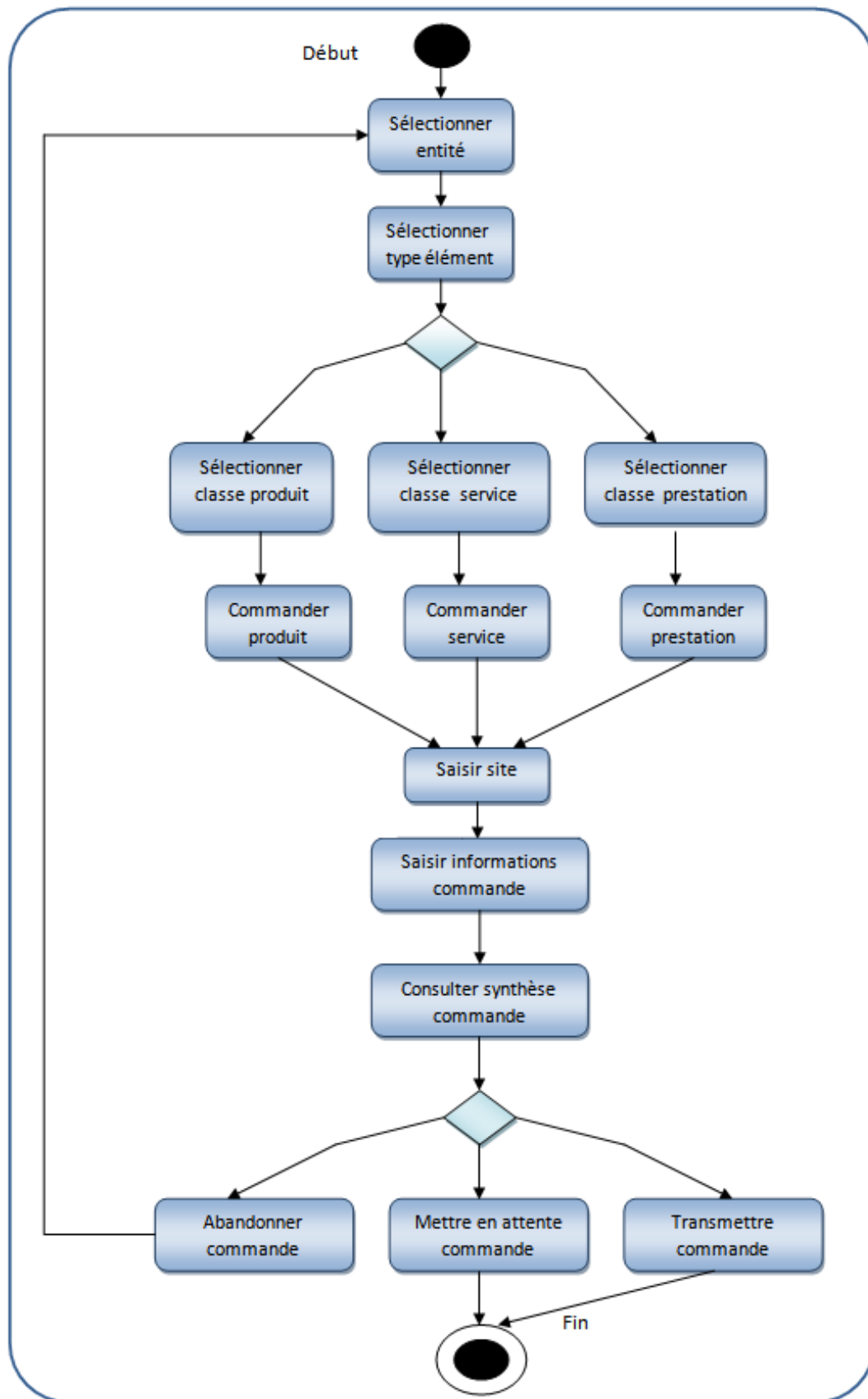
La conception préliminaire s'appuie sur les points de vue de spécification fonctionnelle et structurelle de l'analyse, mais également sur les frameworks de la conception technique. Le choix judicieux de ces frameworks allège le travail et permet de se focaliser sur les vues de l'application.

4.1.2 Diagramme d'activité

Un diagramme d'activité permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Lorsque l'utilisateur client désire effectuer une nouvelle commande, il sélectionne tout d'abord l'entité du client, puis le type de l'élément à commander (qui peut être un produit, un service ou une prestation), lors du choix du type de l'élément, l'utilisateur client est redirigé à la page de la commande, au cas où il veut commander un service, il choisit les options et les produits facultatifs de ce dernier, puis il saisit le site et la date d'installation. Dès lors, il peut consulter la synthèse de la commande effectuée et la transmettre au fournisseur

La figure suivante représente le diagramme d'activité de la prise de commande (figure 4.1):

**Figure 4.1: Digramme d'activité de la prise de commande**

4.2 Architecture Logicielle du Système

Parmi les différentes façons de structurer une architecture, la mieux adaptée pour notre projet est l'approche par couches (figure 4.2). Une couche est une division logique et horizontale d'un système qui fournit une abstraction particulière du système aux couches supérieures. Chaque couche possède des responsabilités spécifiques. Dans une structuration par couches, les couches inférieures prennent en charge des responsabilités qui offrent un socle de base pour les couches supérieures, permettant d'abstraire celles-ci de problématiques qui ne les concernent pas.

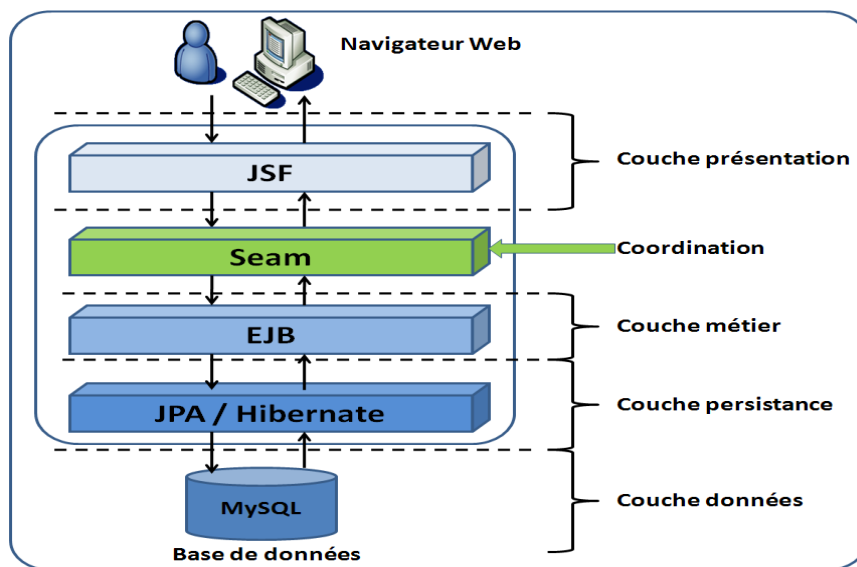


Figure 4.2: Architecture logicielle du système

Seam joue le rôle de coordinateur entre JSF et EJB. Il permet de relier la couche présentation Avec les fonctionnalités de la couche métier (figure 4.3)

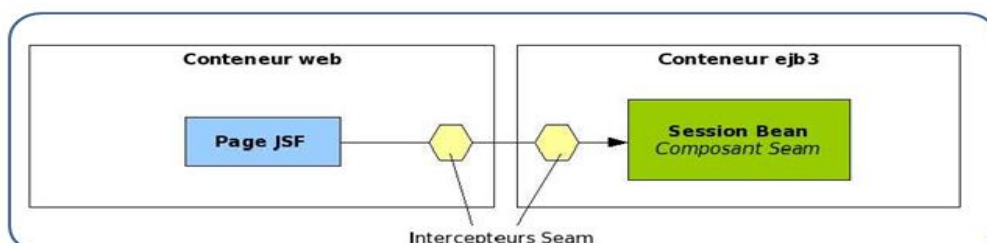


Figure 4.3: Architecture logicielle du système

4.2.1 Couche présentation

Cette couche concerne l'interaction entre notre système et les requêtes de l'utilisateur à travers les interfaces Homme-Machine développées en utilisant JSF. Ces interfaces permettent à l'utilisateur de piloter l'application et d'en recevoir des informations.

4.2.2 Couche métier

Cette couche implémente les algorithmes " métier " de l'application. Elle est indépendante de toute forme d'interface avec l'utilisateur. C'est généralement la couche la plus stable de l'architecture. Elle ne change pas si on change l'interface utilisateur ou la façon d'accéder aux données nécessaires au fonctionnement de l'application.

4.2.3 Couche persistance

Cette couche s'occupe de l'accès aux données persistantes au sein du SGBD MySQL. Elle utilise une API fournie avec Java permettant de se connecter à la base de données.

4.2.4 Couche données

Cette couche est responsable du stockage physique de données. Elle permet le stockage des objets de la couche métier dans une base de données MySQL. Le modèle physique de cette base de données est déduit à partir du diagramme de classe du système.

4.3 Étapes de développement

La mise en œuvre d'une application basée sur le framework Seam suit une démarche différente des autres applications J2EE, car elle prend en compte l'architecture applicative particulière de Seam et dépend des concepts généraux de ce framework.

4.3.1 Couche persistance

Le développement de notre application commence par la définition des entités de la base de données avec les EJB3 et JPA. Cette phase consiste à définir les différents champs et contraintes de chaque entité. Exemple : le client (figure 4.4).


```

@Entity
@Table(name = "catalogue", catalog = "catalogue")
public class Catalogue implements java.io.Serializable {
    private static final long serialVersionUID = 1770526091311037461L;
    private int catalogueId;
    private String description;
    private String libelle;
    private Integer type;
    private int etat;

    public Catalogue() {

```

Figure 4.4: Définition des champs de l'entité catalogue

4.3.2 Couche métier

Cette deuxième couche comporte les beans des différents packages métier du système. Ces beans sont les implémentations des classes métiers du système.

L'arborescence des packages métiers est la suivante (figure 4.5):

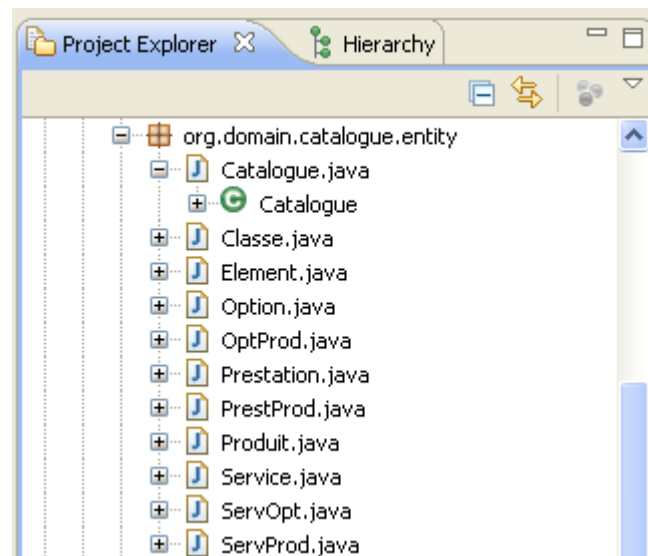


Figure 4.5: Exemple d'arborescence des packages métiers

Chaque package de l'arborescence comporte les beans correspondant aux classes lui appartenant. Ces beans sont appelés par les composants services (figure 4.6) pour répondre aux requêtes des utilisateurs.

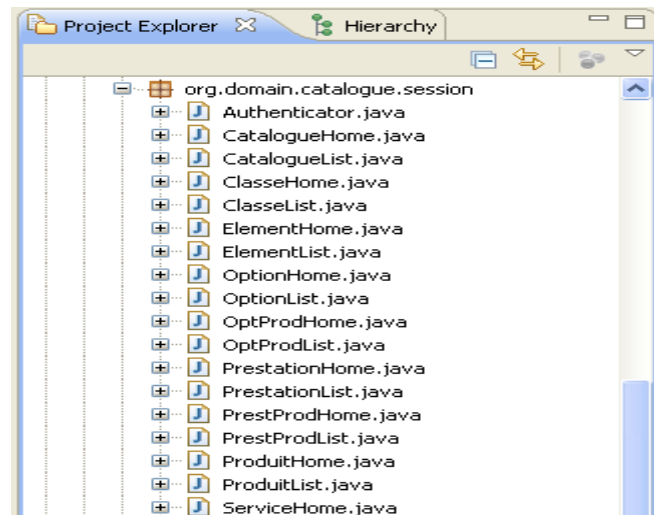


Figure 4.6: Exemple d'arborescence des packages services

Chaque package contient les services permettant de réaliser les opérations demandées par l'utilisateur.

4.3.3 Couche présentation

Pour augmenter les performances de l'interface graphique, nous avons utilisé Seam qui utilise diverses technologies de Java Server Faces (Ajax4JSF, Richfaces, MyFaces). Il permet de les combiner et les faire coordonner pour créer des interfaces beaucoup plus ergonomiques et beaucoup plus rapides

Voici quelques aperçus de des interfaces utilisateur (figures 4.7, 4.8, 4.9)

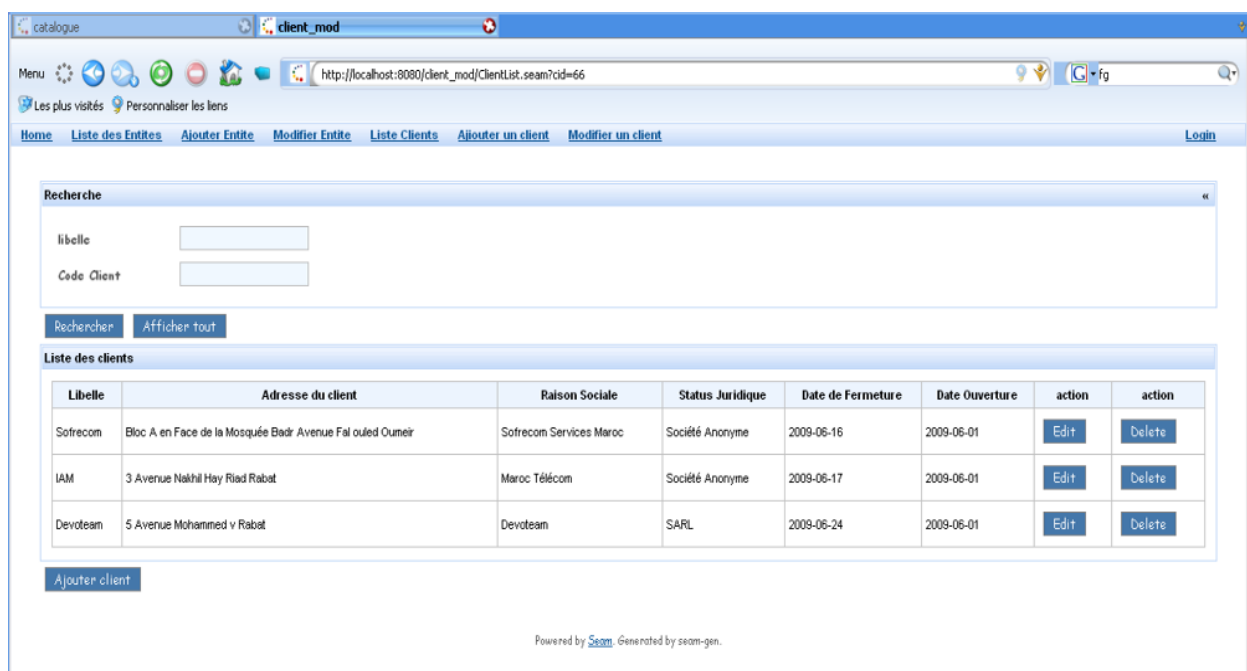


Figure 4.7: L'interface présentant la liste des clients

Sur cette interface, un utilisateur peut faire une recherche parmi les clients, modifier un client. Quelques fonctionnalités comme la recherche, s'effectuent aux niveaux de la page. Pas besoin de recharger toute la page, ce qui augmente la rapidité de l'application.

The screenshot shows a web application interface with the following sections:

- Choisir un catalogue:** A table with columns 'libelle', 'dateActivation', and 'action'. It lists three catalogues: 'Catalogue Packet8', 'Téléphonie Mobile', and 'Connexion Internet'.
- Rechercher un client:** A search form with a 'Code Client' input field, 'Rechercher' and 'Afficher tout' buttons, and a table of results with columns: 'Code Client', 'Libelle Client', 'Adresse du client', 'Raison Sociale', 'Status Juridique', 'Date de Fermeture', 'Date Ouverture', and 'action'.
- Choisir une entité mère:** A table with columns 'libelle', 'dateActivation', 'dateDesactivation', and 'action'. It lists three entities: 'Sofrecom', 'Unité de production Websource', and 'Direction Ressources Humaines'.
- Ajouter une nouvelle Entité:** A form for adding a new entity. It includes fields for 'Entité Père' (Sofrecom), 'Code Entité' (112), 'libelle' (Unité de production Sifac & Sires), 'dateActivation' (06/03/2009), and 'dateDesactivation' (06/10/2009). There is also a calendar widget for the date and a 'Save' button.

Figure 4.8: Ajouter une entité

Sur cette interface, un utilisateur ajoute une entité et l'associe à client, elle lui permet de sélectionner le catalogue voulu, puis sélectionner un client appartenant à ce catalogue, de sélectionner l'entité mère, et enfin de saisir les différentes informations de la nouvelle entité.

Choisir un catalogue

libelle	dateActivation	action
Catalogue Packet0	catalogue contenant des produit et de service internet	Select
Téléphonie Mobile	Ce catalogue contient les produits et les services de Téléphonie mobile de France Télécom	Select
Connexion Internet	Ce catalogue contient les services de connexion Internet	Select

Add Client

Catalogue: Téléphonie Mobile

Code Client *: 236

libelle *: Capgemini

raisonSociale *: Capgemini

statusJuridique *: Société Anonyme

dateFermeture *: 06/11/2009

adresse *:

* Champ Obligatoire

Save Cancel

Powered by Seam. Generated by seam-gen.

Figure 4.9: Ajouter un client

Sur cette interface, un utilisateur peut ajouter un client en l'associant à un catalogue. L'utilisateur sélectionne le catalogue voulu puis saisie les informations du nouveau client.

Voici un tableau (cf. tableau 4.1) de description de quelques interfaces IHM de notre application :

Vue IHM	Description
Commander un service	<p>Cette IHM permet à l'utilisateur client de consulter les éléments de tarification du service qu'il désire commander et il peut également sélectionner les options facultatives et les produits facultatifs, déterminer la quantité du service commandé etc.</p> <p>Il est à noter, que les options facultatives sélectionnées doivent être compatibles entre elles, idem pour les produits facultatifs, dès lors, quand l'utilisateur coche un produit facultatif, les produits incompatibles à ce dernier ne sont plus commandables (ils ne peuvent plus être cochés).</p>

Consulter le catalogue	<p>Cette Interface permet à l'utilisateur ou l'administrateur de consulter le catalogue, le catalogue est constitué de classes comportant des services, des produits et des prestations.</p> <p>Il y deux façons de consultation :</p> <ul style="list-style-type: none"> - Soit par la sélection d'un catalogue où une liste de classes associées à ce catalogue apparait, et lorsque l'on choisit la classe, les différents services, produits et prestations sont affichés. On peut alors consulter l'élément souhaité. - Soit par la recherche directe d'un service, un produit ou une prestation, en tapant uniquement le libellé de l'élément que l'on désire consulter.
Consulter la synthèse d'une commande	<p>Cette IHM contient toutes les données saisies par l'utilisateur lors de la prise d'une commande : l'élément commandé, éventuellement ses options obligatoires et facultatives, le(s) site(s) d'installation ou de livraison, la date d'installation ou de livraison.</p> <p>Cette IHM comporte également, trois boutons : envoyer la commande au fournisseur, mettre en attente la commande (pour une éventuelle modification après ou pour continuer la saisie ultérieurement) ou bien carrément abandonner la commande (dans ce cas la commande n'est pas enregistré et l'utilisateur est redirigé à la page de la sélection d'entité).</p>

Tableau 4.1: Description de quelques IHM du système

4.4 Processus de prise de commande

Nous avons utilisé le JBPM pour modéliser le processus de prise de commande La figure ci-dessous (figure 4.10) montre le processus métier sous forme d'un automate avec des états et des transitions :

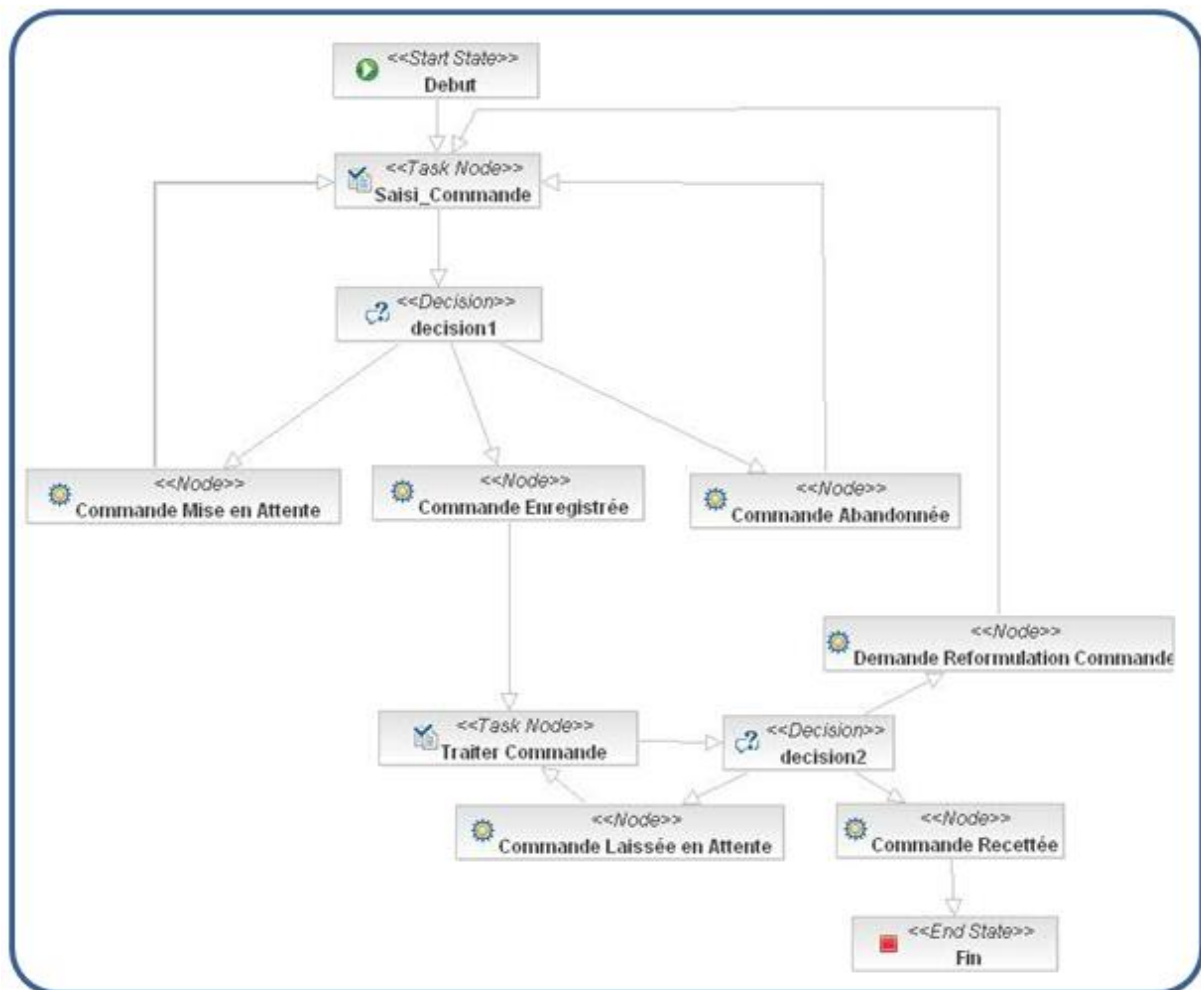


Figure 4.10: Processus de prise de commande

4.5 La mise en œuvre de l'ESB

Dans la deuxième phase du développement de notre projet, nous avons mis en place JBossESB [JB-ESB-PG, 09] pour permettre aux différents modules d'échanger des messages et de tourner sur une architecture orientée service. Pour faire circuler les messages dans le bus de l'ESB, nous avons utilisé l'API standard Java Message Service.

4.5.1 Java Message Service

L'interface de programmation Java Message Service (notée JMS) [JB-Messaging] permet d'envoyer et de recevoir des messages de manière asynchrone entre applications ou composants Java (figure 4.11). Un client peut également recevoir des messages de façon synchrone dans le mode de communication point à point.

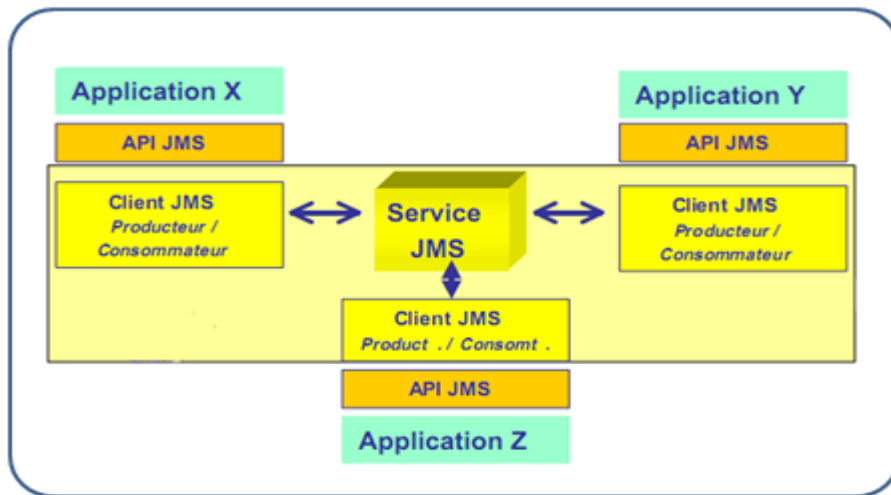


Figure 4.11: Application JMS

Nous avons utilisé le modèle point à point pour échanger les messages dans le bus, le producteur publie les messages dans une file (queue) et le client lit les messages de la file (figure 4.12). Dans ce cas le producteur connaît la destination des messages et poste les messages directement dans la file du client.

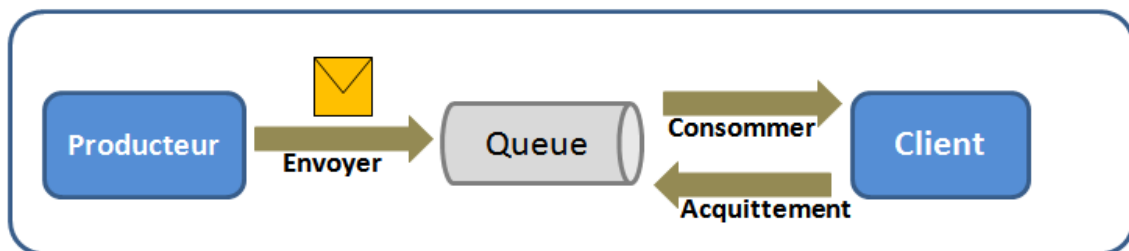


Figure 4.12: Communication P2P

Un client s'enregistre auprès d'un écouteur de message (listener). Lorsqu'un message arrive, JMS le délivre en invoquant la méthode `onMessage` du listener.

4.5.1.1 Programmation JMS

Pour qu'une communication entre deux applications puisse s'établir, l'application cliente exécute la séquence d'opération suivante (figure 4.13):

- Elle recherche un objet « `ConnectionFactory` » dans un répertoire en utilisant l'API JNDI (Java Naming Directory Interface) ;
- Elle utilise l'objet « `ConnectionFactory` » pour créer une connexion JMS, il obtient alors un objet « `Connection` » ;

- Elle utilise l'objet « Connection » pour créer une ou plusieurs sessions JMS, il obtient alors des objets « Session » ;
- Elle utilise le répertoire JNDI pour trouver un ou plusieurs objets « Destination » ;
- Elle peut alors, à partir d'un objet « Session » et des objets « Destination », de créer les objets « MessageProducer » et « MessageConsumer » pour émettre et recevoir des messages.

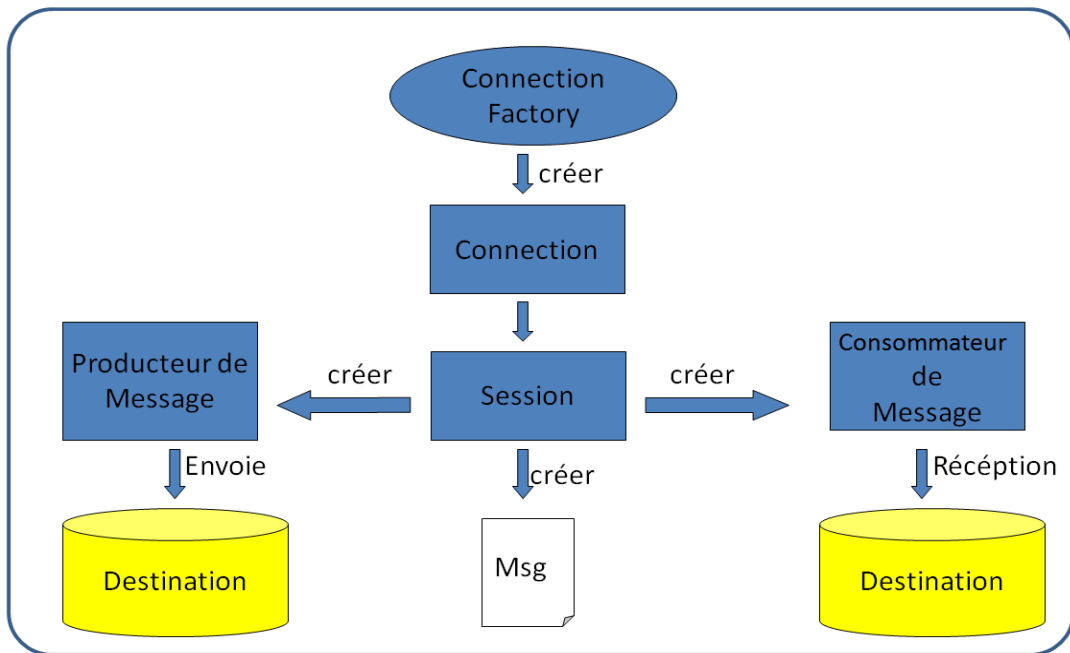


Figure 4.13: modèle de programmation JMS

Les étapes et les objets caractérisant une application JMS sont résumés dans la figure 4.14.

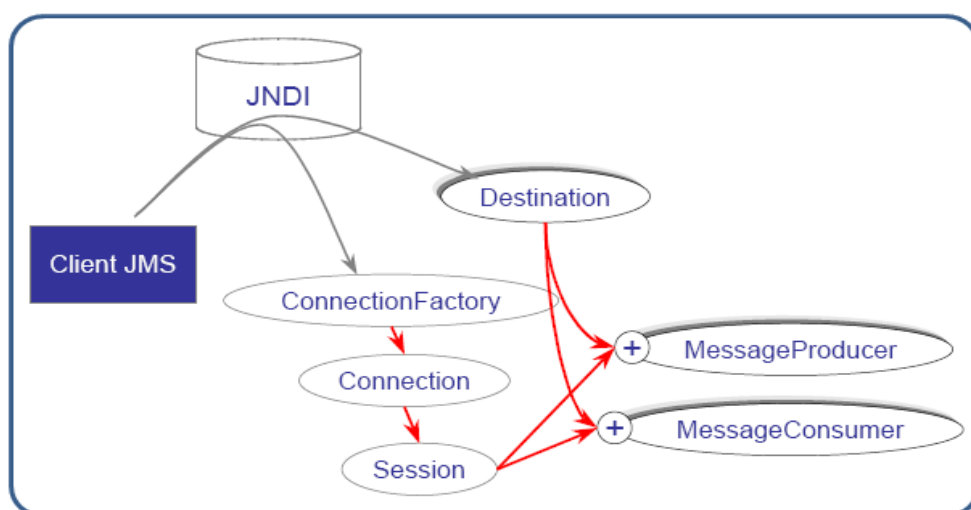


Figure 4.14: Architecture d'une application JMS

Conclusion

Dans ce chapitre, nous avons décrit l'architecture applicative détaillée du projet, nous avons défini la démarche de mise en œuvre, et nous avons présenté quelques interfaces de l'application, puis nous avons présenté le processus de prise de commande et le mécanisme de l'échange des messages entre les applications avec le standard JMS nécessaires pour la mise en place de l'ESB.

Conclusion générale

Notre projet, effectué au sein de Sofrecom Services Maroc, consistait en la conception et la réalisation d'un outil de prise de commande Télécom selon une approche SOA. Les services SOA sont en effet aisément réutilisables et peuvent être rapidement assemblés sous la forme de nouvelles applications composites.

Pour réaliser ce projet, nous avons commencé par une analyse détaillée des besoins requis. Ensuite, nous avons construit une modélisation du système en utilisant le langage UML, au terme de l'étude fonctionnelle. Puis, nous avons proposé une nouvelle architecture logicielle tout en choisissant les Frameworks techniques nécessaires à la réalisation du projet. Enfin, nous avons abordé la phase de la réalisation qui consiste à la mise en œuvre de l'architecture proposée.

Des tests ont été réalisés et ont permis de valider le bon fonctionnement des modules développés.

Durant la période de notre stage, nous avons eu l'occasion de renforcer plusieurs notions telles que la gestion optimale du temps et l'esprit du travail en groupe. Le stage était aussi pour nous une occasion de rencontrer plusieurs ingénieurs informaticiens qui travaillent sur des projets différents, et de découvrir la nature du travail dans une société des services informatiques. La réalisation du projet nous a aussi permise de raffiner notre capacité de conception et de renforcer nos compétences en matière de technologie J2EE.

Toutefois, d'autres perspectives restent en vue pour l'amélioration de l'Outil de Prise de Commande Telecom tels que l'intégration des applications des fournisseurs pour leur permettre de proposer des services directement aux clients, et d'élargir leurs secteurs d'activités.

Bibliographie

[Maesano & al, 03] Libero Maesano, Christian Bernard et Xavier Le Galles, Services Web avec J2EE et .NET, Paris, EYROLLES, 2003, 1087 p.

[Patricio, 08] Anthony PATRICIO, Java Persistence et Hibernate, Paris, EYROLLES, 2008, 388 p.

[Roques & al, 03] Pascal Roques, Franck Vallée, UML 2 en action : De l'analyse des besoins à la conception J2EE, Paris, EYROLLES, 2003, 386 p.

Webographie

[Eclipse] *Eclipse* [Logiciel], v3.4, Disponible sur <http://www.eclipse.org>, 27 mai 09

[JBossESB] *JBoss ESB* [Logiciel], v4.5, Disponible sur <http://www.jboss.org/jbossesb>, 27 mai 09

[JB-ESB-PG, 09] JBoss Inc, *JBossESB Programmers Guide, 2009*, [En ligne] Date de dernière mise à jour : le 2 Décembre 2008 Disponible sur :

<http://www.jboss.org/jbossesb/docs/4.5.GA/manuals/html/ProgrammersGuide.html>, 30 Mai 09

[JB-JBPM] *JBoss SEAM* [Logiciel], v3.2.6, Disponible sur <http://www.jboss.org/jbossjbp/>, 30 mai 09

[JB-Messaging] *JBoss Messaging* [Logiciel], v1.4.4, Disponible sur : <http://www.jboss.org/jbossmessaging/>, 30 mai 09

[JB-SEAM] *JBoss SEAM* [Logiciel], v2.0, Disponible sur <http://seamframework.org/>, 30 mai 09

[Kittoli, 09] Samson Kittoli, *Seam - Contextual Components, 2009*, [En ligne] Date de dernière mise à jour : le 15 Décembre 2008 Disponible sur :

http://docs.jboss.com/seam/2.1.1.GA/reference/en-US/html_single/, 30 Mai 09

[MySQL-Server] *MySQL Server* [Logiciel], v5.0, Disponible sur <http://dev.mysql.com/downloads/>, 27 mai 09

Annexe A : Glossaire métier

Terme métier	Explication
<i>Catalogue</i>	L'ensemble des services, des produits, des prestations ou des offres pouvant être commandés dans Web Source Commande
<i>Classe</i>	Une classe permet de regrouper des éléments et des offres de même technologie ou de même nature.
<i>Élément</i>	Un élément fait parti des composants d'une classe. Il peut désigner un produit, un service ou bien une prestation.
<i>Offre</i>	C'est un package regroupant des services, des produits et (ou) des prestations.
<i>Option</i>	Un composant faisant parti d'un service ou d'un produit. Il peut être facultatif comme il peut être de base.
<i>Prestation</i>	Une intervention de maintenance ou bien une assistance. Une prestation peut être associée à des produits obligatoires et/ou des produits facultatifs.
<i>Produit</i>	<p>Un produit représente tout composant matériel pouvant être commandé par l'utilisateur à travers Web Source commande. Il peut être décliné en des options incluses, des options facultatives.</p> <p>Un produit peut être directement sujet d'une commande comme il peut faire partie des services commandés.</p>
<i>Service</i>	Un service représente tout composant immatériel du catalogue pouvant être commandé par l'utilisateur à travers Web Source commande. Il peut être décliné en des options incluses, des options facultatives et (ou) des produits.
<i>Client</i>	<p>Il a le droit juridique et contractuel de passer des commandes sur un catalogue associé à un Contrat.</p> <p>Il participe à la prise de commande et à la validation de la commande. Il consulte le parc client et le catalogue des services.</p>
<i>Fournisseur</i>	Le fournisseur intervient sur certaines étapes du processus de commande et met en service le service commandé.

<i>Commande</i>	Une contractualisation d'une demande, pour un service du catalogue, d'un client vers un fournisseur.
<i>Site</i>	Un site est une localisation qui permet d'acheminer ou de déployer un service, un produit ou bien une prestation, il peut être le site d'installation d'un service, le site de livraison d'un produit, ou le site de maintenance ou d'une assistance d'une prestation.
<i>Boîte de réception</i>	Permet à l'utilisateur l'accès aux commandes de service en cours pour une visualisation des commandes en cours de traitement ainsi que de leurs états.
<i>Parc</i>	Le parc est l'ensemble des services installés et facturés sur les différents sites.

Annexe B : Exemple Catalogue

Service : forfait origami

Catalogue : Téléphonie mobile Orange

Classe de service : téléphonie mobile

Options obligatoires

- Abonnement 24 mois
- Appels illimités vers 3 numéros 24h24,7j/7
- 10 SMS/mois inclus
- Alerte par SMS en cas de dépassement du forfait

Options facultatives

- Forfait 1h/mois
- Forfait 2h/mois
- Forfait 5h/mois
- Service roaming vers l'international
- Internet 3G+

Produit obligatoire: Iphone 3G

Options obligatoires

- Ecouteurs stéréo avec micro
- Adaptateur secteur USB
- Documentation
- Chiffon de nettoyage
- Outil d'extraction de la carte SIM

Options facultatives

- film de protection écran Belkin
- câble allume-cigare
- câble USB Apple
- chargeur secteur Apple
- station d'accueil Apple
- kit piéton stéréo Apple
- rallonge allume-cigare 2 sorties
- adaptateur de voyage multi-pays

- kit mains libres Bluetooth à installer, sans antenne
- kit avec câble ipod/iPhone, sans installation Orange
- kit avec câble ipod/iPhone, avec installation Orange
- copieur de cartes SIM (et USIM)

Annexe C : Plan Assurance Qualité

Le présent projet tend à mettre en place un outil léger de prise de commandes télécom, afin de contrôler la qualité du projet en question, il est nécessaire de mettre en place un Plan Assurance Qualité incluant le planning du projet, en partant de la phase d'étude préliminaire jusqu'à la phase de mise en production de l'outil.

▪ Objectif du Plan Assurance Qualité

Le plan d'assurance qualité est l'ensemble des activités préétablies et systématique mises en œuvre dans le cadre du système qualité, et démontrées en tant que besoin, pour donner la confiance appropriée en ce qu'une entité satisfera aux exigences pour la qualité.

L'assurance qualité a pour mission de fiabiliser chaque étape du processus d'une activité allant de la prise de commande en passant par la mise sur le marché, le service après-vente et jusqu'au soutien après la vente.

La démarche d'assurance qualité consiste à prévenir systématiquement et méthodiquement tout dysfonctionnement source de non qualité. C'est le passage d'une logique curative à une logique préventive des erreurs.

▪ Responsabilités du Plan Assurance Qualité

Le suivi du PAQ et du contrôle de son application est la tâche de l'encadrant et responsable qualité de Sofrecom. Quand à la réalisation du présent PAQ, cette tâche relève de la responsabilité du stagiaire élève ingénieur. Ainsi nous répartissons les responsabilités du PAQ selon le tableau suivant :

Intervenants	Edition	Validation	Suivi du plan	Réalisation
Imane BOUROUS	X			X
Mohammed SOUIDI	X			X
Younes SOUTEH		X	X	
Mustapha EL FEDDI		X	X	

▪ Organisation du projet

Le projet se déroule en collaboration entre l'ENSIAS et Sofrecom selon la répartition suivante:

Coté Sofrecom :

	Personne	Rôle
Responsables techniques	Mustapha EL FEDDI	Architecte Logiciel
	Younes SOUTEH	Architecte Logiciel
Responsables opérationnels	Imane NEJJAR	Experte Fonctionnelle

Coté ENSIAS :

Personne	Rôle
Ahmed HABBANI	Professeur encadrant
Imane BOUROUS	Stagiaire élève ingénieur
Mohammed SQUIDI	Stagiaire élève ingénieur

▪ Objectifs qualité du projet

En termes de qualité les résultats de notre projet doivent satisfaire les critères suivants :

- Une modularité permettant de remplacer facilement un composant (service) par un autre.
- Une réutilisabilité possible des composants.
- De meilleures possibilités d'évolution.
- Une plus grande tolérance aux pannes.
- Une maintenance facilitée.

▪ Activités d'assurance et de contrôle de la qualité

Chaque membre de l'équipe projet est tenu de respecter les dispositions décrites dans le PAQ et de vérifier l'adéquation du produit (document ou code) avec les spécifications du cahier des charges du projet. D'une autre part, les responsables du projet (nos encadrants) veillent à ce que ces dispositions soient respectées.

▪ Planification du projet

Conduire un projet, c'est assurer le pilotage d'un processus de changement avec des ressources dédiées en optimisant les compétences, l'organisation, les systèmes et les outils de conduite.

Une approche managériale réactive, souple et systématique pour mener à bien des changements importants, complexes, ciblés sur le but à atteindre. Il y a trois niveaux de gestion du projet : la gestion de la production, des ressources et du temps. Afin de satisfaire à ce dernier critère qu'est la gestion du temps, il est nécessaire d'établir un planning prévisionnel. Sachant que la ressource affectée à ce projet est l'élève ingénieur et que le présent planning est réajusté au fur et à mesure de l'avancement du projet.

Le stage débute le 16 février et prend fin le 16 juin, ce qui implique une durée totale de 85 J*H selon le planning suivant :

