

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

**MINISTERE DE LA RECHERCHE SCIENTIFIQUE ET DE L'ENEIGNEMENT
SUPERIEUR**

**UNIVERSITE MOULOU D MAMMERI DE TIZI OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE**



MEMOIRE

DE FIN D'ETUDES



EN VUE DE L'OBTENTION DU DIPLOME DE Master2 EN INFORMATIQUE

Conception et Réalisation d'un service
web pour la gestion des écarts des
stations de service

cas :NAFTAL

Proposé par : Mr

KERBICHE.M

Réalisé par :

M^{elle} DOUNAS FERROUDJA

M^{elle} KADI NACERA

2014_2015

Remerciements

Nous tenons à témoigner notre reconnaissance à DIEU tout puissant, qui nous a aidé et bénis par sa volonté durant toute cette période.

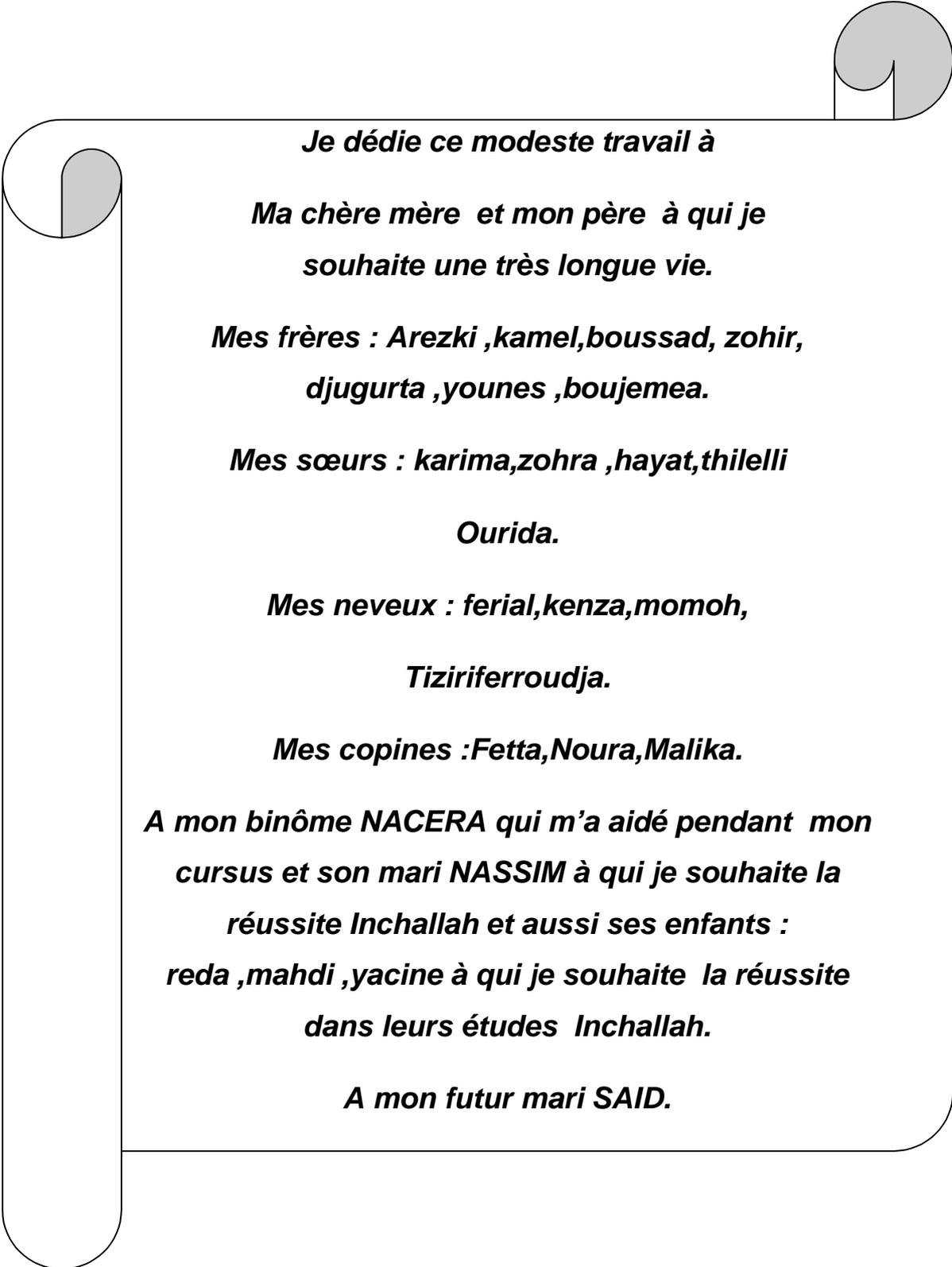
Notre profonde gratitude et sincères remerciements vont à notre promoteur Mr KERBICHE pour sa précieuse assistance, sa disponibilité et l'intérêt qu'elle a manifesté pour ce modeste travail.

Nos plus vifs remerciements vont à tout le personnel de NAFTAL Qui nous ont généreusement aidé durant notre stage surtout le chef de service Mr Chaba Kamel.

Tous nos remerciement aux membres de jury qui nous font l'honneur de juger se modeste travail.

Nos remerciements vont également à tous ceux qui ont répondu aux questions sur des problèmes rencontrés.

Nos remerciements vont aussi à tous ceux qui ont contribué de près ou de loin à la réalisation de notre travail.



Je dédie ce modeste travail à

*Ma chère mère et mon père à qui je
souhaite une très longue vie.*

*Mes frères : Arezki ,kamel,boussad, zohir,
djugurta ,younes ,boujemea.*

Mes sœurs : karima,zohra ,hayat,thilelli

Ourida.

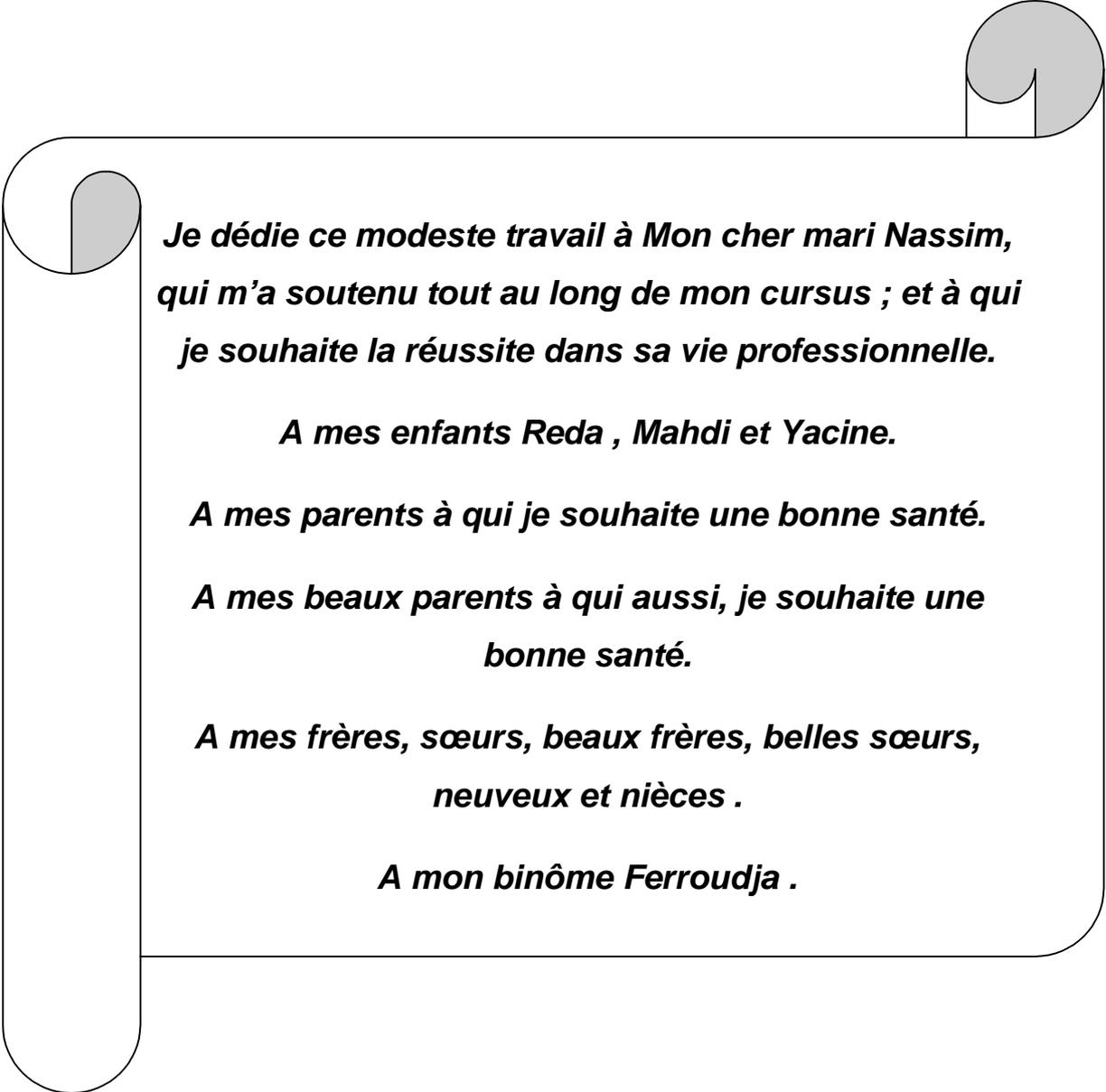
Mes neveux : ferial,kenza,momoh,

Tiziriferroudja.

Mes copines :Fetta,Noura,Malika.

*A mon binôme NACERA qui m'a aidé pendant mon
cursus et son mari NASSIM à qui je souhaite la
réussite Inchallah et aussi ses enfants :
reda ,mahdi ,yacine à qui je souhaite la réussite
dans leurs études Inchallah.*

A mon futur mari SAID.



***Je dédie ce modeste travail à Mon cher mari Nassim,
qui m'a soutenu tout au long de mon cursus ; et à qui
je souhaite la réussite dans sa vie professionnelle.***

A mes enfants Reda , Mahdi et Yacine.

A mes parents à qui je souhaite une bonne santé.

***A mes beaux parents à qui aussi, je souhaite une
bonne santé.***

***A mes frères, sœurs, beaux frères, belles sœurs,
neveux et nièces .***

A mon binôme Ferroudja .

sommaire

Introduction générale

Chapitre I : Les protocoles applicatifs sur internet

I Introduction.....	1
I.1 Les réseaux :	1
I.1.1 Généralité sur les réseaux [GUY PUJOLLE].....	1
I.1.2 Objectifs des réseaux [GUY]	2
I.1.3 Types de réseau [MAG.98].....	2
I.1.3.1 Selon la distance :.....	2
I.1.3.2 Selon la topologie.....	2
c. L'étoile :	4
d. L'anneau:	4
e. Structure de réseau maillée :	5
I.1.4 Les modèles du réseau [CCM 09]:.....	5
I.1.5 Le modèle TCP/IP et ses couches [USER]	6
I.2 Le client /serveur :	8
I.2.1 Introduction	8
I.2.2 Le modèle client /serveur [ARS 09].....	8
I.2.2.1 Définition:	8
I.2.2.2. Middleware	9
I.2.3.Pourquoi le client /serveur :	9
I.2.4 Modes de fonctionnement :.....	10
I.2.5.Fonctionnement d'un système client/serveur : [CCM].....	10
I.2.6 Les protocoles: [GUP 05,DJADJN]	11
Le but du TCP	14
I.2.7 Caractéristique du client/serveur [ASG 08]	16
I.2.8 Types de clients : [GEO 00]:.....	16
I.2.9 Les types des serveurs : [BNOK].....	17
I.2.10 Architecture client/serveur : [WIKIPÉDIA, ORD 99].....	18

I.2.11 Avantages de l'architecture client/serveur : [GEO 00]:.....	21
I.2.12 Inconvénients de l'architecture client/serveur :	21
Conclusion :	21

Chapitre II : Technologie des services web

II Introduction.....	22
II.1 SOA, Services Oriented Architecture [CR 185].....	22
II.1.1 Introduction.....	22
II.1.2 Qu'est-ce que l'Architecture Orienté Services ?.....	23
II.1.3 pourquoi le SOA ?.....	23
II.1.4 Les objectifs de l'architecture orientée SOA.....	23
II.2 Les Web Services [ING.info.93].....	24
II.2.1 Définition.....	24
II.2.2 Historique [CR 185].....	24
II.2.3 Services web=HTTP+SOAP+WSDL+Composants logiciels.....	25
II.2.3.1 SOAP ?le protocole.....	27
II.2.3.1.1 Définition.....	27
II.2.3.1.2 Structure des messages SOAP.....	27
II.2.3.2 UDDI, l'annuaire universel.....	28
II.2.3.2.1 Présentation.....	28
II.2.3.2.2 Architecture et Fonctions [CR 185].....	28
II.2.3.2.3 Contenu de l'annuaire.....	29
II.2.3.3 WSDL, méta-informations [ING .inf].....	30
II.2.3.3.1 Présentation.....	30
II.2.3.3.2. Éléments d'une définition WSDL.....	30
II.2.4 Pourquoi les services web ?.....	31
II.2.5. Modèle d'interaction des services web.....	33
II.2.5.1 Scénario général de fonctionnement des services web.....	33
II.2.5.2 Etapes d'exécution des services web [CR 185].....	35
II.3. Apport de xml.....	35
II.3.1 Les technologies XML.....	36
II.3.2 XML et services distribués.....	36
II.3.3 XML et le remote procedure Call.....	37
II.3.4 XML et l'architecture des systèmes d'information.....	38
II.3.5 XML et l'intégration d'applications.....	39

II.4. Architecture des services web	42
II.4.1 Principes de développement des services web.....	46
Conclusion.....	47

Chapitre III : Etude de l'existant

III Introduction	48
III.1 Présentation de NAFTAL	48
III.1.1 Historique de NAFTAL	48
III.1.2 Les missions de NAFTAL	49
III.1.3 Organisation de NAFTAL.....	49
III.1.4 Branche commercialisation NAFTAL	52
III.2 Présentation du district commercial de Tizi-Ouzou.....	52
III.2.1 Missions et attributions du district commercial de Tizi-Ouzou.....	53
III.2.2 Organisation du district commercial de Tizi-Ouzou	53
III.2.3 Organisation du département commercial.....	58
Conclusion.....	58

Chapitre IV : Analyse et conception

IV Introduction :	59
IV.1 Phase d'Initialisation :	59
IV.2 Phase d'Elaboration :	59
IV.2.1 Identification des acteurs :	60
IV.2.3 Représentation des diagrammes de cas d'utilisation :	61
IV.2.4 Définition des itérations :	63
IV.3 Construction :.....	63
IV.3.1 Réalisation de l'itération 1	63
IV.3.1.1 Diagrammes de séquence pour itération 1:l.....	63
IV.3.1.2 Diagrammes de classe pour l'itération 1	67
IV.3.2 Réalisation de l'itération 2	67
IV.3.2.1 Diagrammes de séquence pour itération 2:	67
IV.3.2.2 Diagrammes de classe pour l'itération2	69
IV.3.3 Réalisation de l'itération 3 :.....	70
IV.3.3.1 Diagrammes de séquence pour itération 3:	70

IV.3.3.2 Diagrammes de classe pour l'itération 3	74
Conclusion :	75

Chapitre V : Réalisation

V Introduction	76
V.2 Les langages de programmation :	79
❖ Java EE	79
V.3 les Itérations :	80
V.3.1 Itération I :	80
V.3.2 Itération II :	80
V.3.3 Itération III:	80
V.4 Présentation de quelques interfaces :	81
V.4.1 Présentation d'interface page d'accueil :	82
V.4.2 Présentation d'interface authentification :	82
V.3.3 Présentation d'interface d'un produit:	83
V.3.4 Présentation d'interface d'ajouter un produit:	83
Conclusion :	84

Conclusion générale

Bibliographie

Annexes

Langage de Modélisation UML.....	01
Présentation de RUP.....	17
Présentation SGBD & EasyPHP.....	21

Liste des Figures

Figure I.1 :Topologie en bus.....	03
Figure I.2 :Topologie en arbre	03
Figure I.3 :Topologie en étoile	04
Figure I.4 :Topologie en anneau.....	04
Figure I.5 :Topologie en maillée.....	05
Figure I.6 :Représentation des couches du modèles OSI.....	05
Figure I.7 :Représentation des couches du modèle TCP/IP et les couches correspondante du modèle OSI.....	07
Figure I.8 :Fonctionnement d'un système client serveur.....	10
Figure I.9 :Le client serveur à deux niveaux.....	18
Figure I.10 :Le client serveur à trois niveaux.....	19
Figure I.11 :Le client serveur à multi niveaux.....	20
Chapitre II :	
Figure II.1 :Structure d'un message SOAP.....	27
Figure II.2 ; Découverte des fonctions d'un service	29
Figure II.3 : Entité composant un annuaire.....	30
Figure II.4 : Modèle d'interaction des services.....	34
Figure II.5 : Place de SOAP dans la pile de communication entre services web.....	37
Figure II.6 : XML comme langage commun de communication entre composants du SI.....	38
Figure II.7 : Niveaux D'intégration EAI dans le S.I.....	39
Figure II.8 : Intégration d'un SI à l'aide des technologie EAI.....	42
Figure II.09 : Les couches des services web.....	42
Figure II.10 : Cadre architecturale des services web.....	43
Figure II.11 : Architecture d'implémentation des services web.....	47
Chapitre III :	
Figure III-1 : Organigramme de l'entreprise NAFTAL [1].....	51
Figure III-2 : Organigramme du district commercial de Tizi-Ouzou [1].....	54
Figure III-3 :Organigramme du Département commercial [1].....	58

Chapitre IV :

Figure IV.1: diagramme de contexte.....	60
Figure IV.2 : Diagramme de cas d'utilisation.....	62
Figure IV.3 :Diagramme de séquence du cas d'utilisation : « Authentification ».....	64
Figure IV.4 :Diagramme de séquence du cas d'utilisation : «Modification du mot de passe»....	66
Figure IV.5 : Diagramme de classes pour l'itération 1.....	67
Figure IV.6 :Diagramme de séquence du cas d'utilisation : «Ajouter un chef de station ».....	68
Figure IV.7 : Diagramme de classes pour l'itération2.....	69
Figure IV.8 :Diagramme de séquence du cas d'utilisation : «Saisie des valeurs des écarts ».....	70
Figure IV.9 :Diagramme de séquence du cas d'utilisation : «Supprimer une vente ».....	72
Figure.IV.10 :Diagramme de classes pour l'itération 3.....	74

Chapitre V :

Figure V.1: L'interface de Hibernate.....	76
Figure V.2 : L'interface EasyPHP	77
Figure V.3 : L'interface PhpMyAdmin.....	77
Figure V.4 : L'interface de Netbeans	78
Figure V.5 : L'interface de JEE.....	79
Figure V.6 :Interface page d'accueil.....	82
Figure V.7 : Interface authentification.....	82
Figure V.8 : Interface d'un produit.....	83
Figure V.5 :Interface ajouter un produit.....	83

Les tableaux

Chapitre II

Tableau II.1 : Principaux travaux de standardisation.....	44
---	----

Introduction générale :

La révolution technologique a bouleversé profondément le fonctionnement des entreprises des nations industrialisées notamment l'Europe occidentale, Etats-Unis et le Japon. Elle a contribué à leurs progrès.

La mondialisation nous impose à notre tour de se mettre petit à petit aux normes internationales afin de favoriser les échanges nord-sud, ce qui impose nos entreprises à informatiser leurs services pour qu'elles soient efficaces et réactives.

L'informatisation des services permet aux entreprises une bonne maîtrise ainsi qu'une meilleure compréhension des masses croissantes d'informations de toute nature dont elles disposent. Elles seront ainsi ouvertes à des perspectives totalement inédites en matière de gestion de l'information car désormais l'information devient dynamique.

Dans le cadre de notre stage à l'entreprise NAFTAL, et après une étude de leur fonctionnement, nous avons diagnostiqué certaines anomalies liées au problème de non informatisation de leurs services auxquelles nous avons apporté des solutions.

En effet, parmi les tâches étudiées, on citera : la gestion des écarts au niveau de chaque station qui se déroule de la manière suivante : A la fin de chaque mois, l'importation des documents se fait encore sous support magnétique, ce qui engendre des pertes de données en cas d'endommagement du support.

De plus, la consolidation des données, qui se fait par le chef de service réseaux, peut engendrer des erreurs lors de la saisie des données et une perte de temps pouvant ralentir le mouvement de l'entreprise.

La solution envisagée, sera de mettre en œuvre une application web de gestion, qui permettra à chaque chef de station de gérer les ventes ainsi que les écarts enregistrés sur sa station, et de permettre au chef de service réseaux de visualiser ces données et d'y mettre des commentaires sur ces écarts, pour enfin être visualisés par le chef de département qui apportera des solutions aux éventuels problèmes rencontrés.

Pour atteindre cet objectif, nous avons structuré notre mémoire en cinq chapitres :

Chapitre I : Les protocoles applicatifs sur internet .La première partie est consacré à la présentation de quelques généralités sur : les réseaux et c/s.

Chapitre II : Technologie des services web .Dans cette partie nous allons présenter quelques définitions des services web et leurs descriptions.

Chapitre III : Etude de l'existant. Nous allons présenter l'organisme d'accueil, en précisant ses différentes directions ainsi que leurs fonctionnalités.

Chapitre IV : Analyse et conception. Ce chapitre est dédié à la présentation de la démarche de modélisation pour le développement de notre application et ceci en étudiant quelques diagrammes de cas d'utilisation introduit par le langage UML.

Chapitre V : Réalisation. Dans cette dernière partie nous allons présenter les outils utilisés ainsi que quelques interfaces de l'application.

I Introduction

Un réseau est un ensemble de moyens matériels et logiciels, géographiquement dispersés, destinés à offrir un service comme les réseaux téléphoniques ou à assurer le transport des données. Les réseaux devront véhiculer toute sorte d'information avec des contraintes temporelles plus au moins fortes avec des volumes extrêmement variés.

I.1 Les réseaux :

I.1.1 Généralité sur les réseaux [GUY PUJOLLE].

a) **Un Réseau informatique:** Un réseau informatique est un ensemble de composants matériels ou logiciels, permettant d'assurer un service de communication.

b) **Un protocole :** c'est un ensemble de règles et de messages assurant un service de communication.

c) **Nœud :** c'est une station de travail, une imprimante, un serveur ou toute entité pouvant être adressée par un numéro unique. L'unicité de l'adresse est garantie par le constructeur d'une carte réseau qui donne un numéro unique ne pouvant être changé par une personne.

d) **Serveur :** ordinateurs qui fournissent des ressources partagées aux utilisateurs par un serveur de réseau.

Et on a quatre types de serveurs qui sont : Serveur messagerie, Serveur Tel net, Serveur FTP, Serveur WWW (World Wide Web).

e) **Paquet :** c'est la plus petite unité d'information pouvant être envoyé sur le réseau. Un paquet contient en générale l'adresse de l'émetteur, l'adresse du récepteur et les données à transmettre.

f) **Trame :** c'est une unité structurée de bit

g) **Clients :** ordinateurs qui accèdent aux ressources partagées fournies par un serveur de réseau.

h) **Support de connexion :** la façon dont les ordinateurs sont reliés entre eux.

i) **Données partagées :** fichiers accessibles sur les serveurs du réseau Imprimantes et autres périphériques partagés : fichiers, imprimantes ou autres éléments utilisés par les usagers du réseau.

j) **Ressources diverses :** autres ressources fournies par le serveur

I.1.2 Objectifs des réseaux [GUY]

Les réseaux permettent :

- Le partage de fichiers.
- Partage de périphériques.
- La communication entre personnes (grâce au courrier électronique, la discussion en directe...)
- Le transfert de données en général (réseaux informatique)
- Le partage d'applications (compilateurs, système de gestion de base de données).
- Le transfert de la parole, de la vidéo et des données.
- Gain du temps et du temps

I.1.3 Types de réseau [MAG.98]

I.1.3.1 Selon la distance :

a. Les LAN (Local Area Network)

Un LAN ou réseau local, est un réseau qui relie des ordinateurs et des périphériques situés à proximité les uns des autres, par exemple dans un même bâtiment de 10 mètres à 1 km de distances. C'est le type de réseau le plus répandu dans les entreprises. Les LAN ne comportent généralement pas plus de 100 ordinateurs.

b. Les MAN (Métropolitain Area Network)

Un MAN ou un réseau métropolitain, est une série de réseaux locaux, il relie des ordinateurs situés dans une même ville ou une même agglomération jusqu'à 10 km.

c. Les WAN (Wide Area Network)

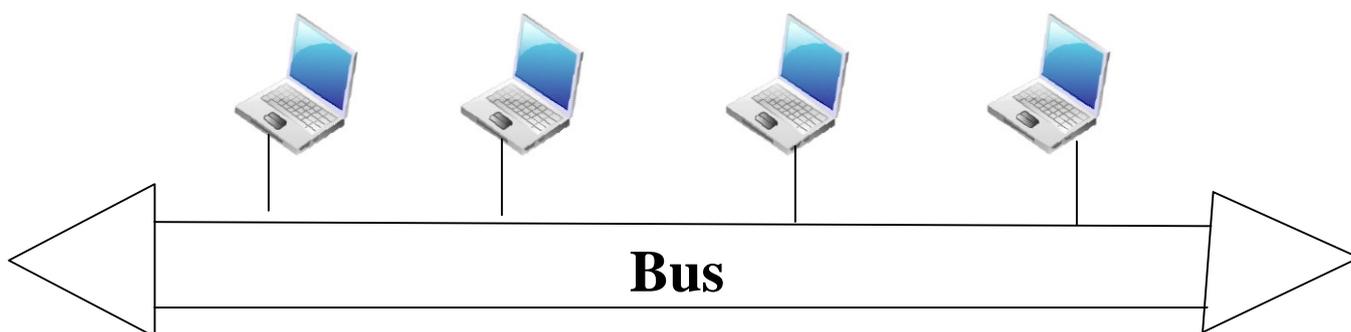
Un WAN ou réseau étendu, sert à relier des LAN et des MAN. Les réseaux qui composent un WAN peuvent être situés dans un même pays ou être dispersés dans le monde.

I.1.3.2 Selon la topologie : indique comment le réseau doit être conçu et organisé tant au niveau physique qu'au niveau logique.

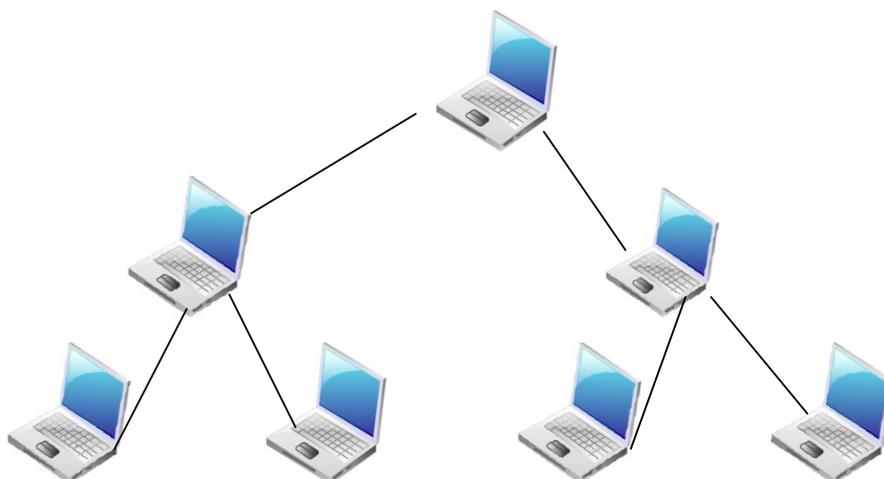
- Le niveau physique : correspond aux éléments matériels tel que les ordinateurs, les câbles et les connecteurs qui constituent le réseau, c'est à ce niveau que l'on détermine à quel endroit les différents ordinateurs doivent être placés et comment les différents composants du réseau doivent être reliés les uns aux autres.
- Le niveau logique : détermine comment les informations doivent circuler au sein du réseau.

a. Structure du réseau en bus :

Sur un réseau en bus, les ordinateurs sont reliés par un même câble rectiligne ininterrompu, chaque extrémité du câble possède un bouchon de terminaison qui empêche le signal d'être renvoyé dans l'autre sens quand il arrive à l'extrémité de ce câble, ce qui pourrait provoquer des interférences. Sur un réseau en bus un seul ordinateur peut transférer des informations au même moment. Lorsqu'un ordinateur envoie des informations, celles-ci parcourent l'ensemble du câble, l'ordinateur de destination doit ensuite récupérer les informations à partir du câble.

**Figure I.1 : Topologie en bus.****b. Structure de réseau hiérarchique :**

Aussi connu sous le nom de réseau en arbre, il est divisé en niveaux. Le sommet, de haut niveau, est connectée à plusieurs nœuds de niveau inférieur, dans la hiérarchie. Ces nœuds peuvent être eux-mêmes connectés à plusieurs nœuds de niveau inférieur. Le tout dessine alors un arbre, ou une arborescence. Le point faible de ce type de topologie réside dans l'ordinateur "père" de la hiérarchie qui, s'il tombe en panne, interdit alors toute communication entre les deux moitiés du réseau.

**Figure I.2 : Topologie en arbre.**

c. L'étoile :

Dans un réseau en étoile, la forme physique du réseau ressemble à une étoile. Une image est plus parlante :

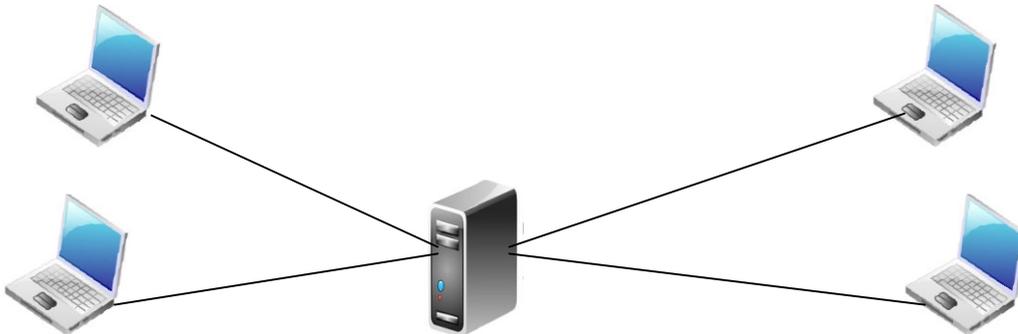


Figure I.3 : Topologie en étoile

d. L'anneau:

Un réseau en anneau a la forme d'un anneau. Cependant, la topologie physique d'un réseau en anneau est le bus. L'information circule dans une seule direction. La topologie de type bus possédait un problème de collision de données : 2 machines ne doivent pas échanger des données en même temps, sinon elles s'entrechoquent. Le principe est assez simple : une machine connectée au réseau possède un jeton virtuel. Ce jeton, c'est une autorisation de communiquer. Une fois que la machine a transmis ce qu'elle voulait, elle passe le jeton à la machine suivante, et ainsi de suite. Si le détenteur du jeton n'a rien à dire, il le passe au suivant.

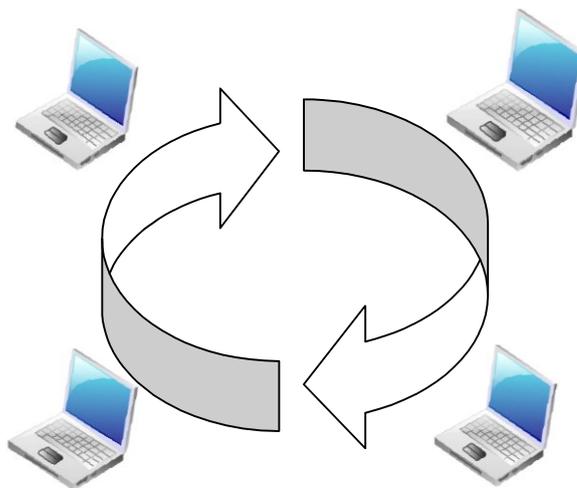


Figure I.4: Topologie en anneau.

e. Structure de réseau maillée :

Dans une topologie maillée, chaque ordinateur est connecté à chacun des autres par un câble séparé. Son principal avantage, est sa capacité de tolérance de panne. En effet, lorsqu'un câble se rompt, il existe de nombreux autres itinéraires routés. Cette topologie est parfois très coûteuse.

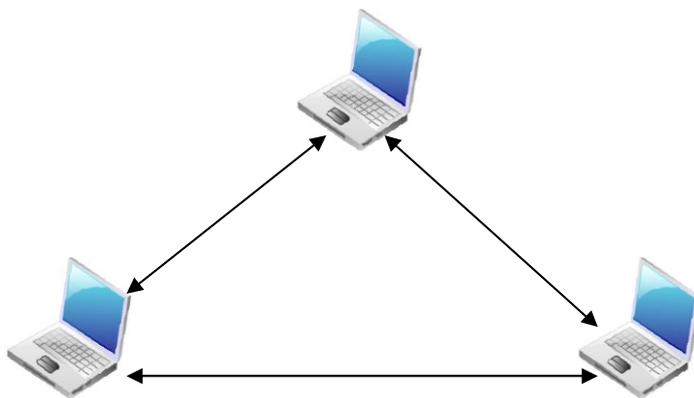


Figure I.5: Topologie en maillée.

I.1.4 Les modèles du réseau [CCM 09]:

Norme définie par l'ISO (International Standardisation Organisation) pour permettre l'interconnexion des systèmes hétérogènes. L'architecture du modèle de référence d'interconnexion des systèmes ouverts (ou modèle de référence OSI) est hiérarchisée en sept couches. Un système ouvert est un ordinateur, un terminal, un réseau, n'importe quel équipement respectant cette norme et donc apte à échanger des informations avec d'autres équipements hétérogènes et issus de constructeurs différents. Les 7 couches du modèle OSI.

L'OSI est un modèle de base normalisé par l'International Standard Organisation (ISO).

Application	<input type="checkbox"/>	Couche Application	7	Couche Application	<input type="checkbox"/>	Donnée
	<input type="checkbox"/>	Couche Présentation	6	Couche Présentation	<input type="checkbox"/>	Donnée
	<input type="checkbox"/>	Couche Session	5	Couche Session	<input type="checkbox"/>	Donnée
Transport des données	<input type="checkbox"/>	Couche Transport	4	Couche Transport	<input type="checkbox"/>	Donnée
	<input type="checkbox"/>	Couche Réseau(Network)	3	Couche Réseau(Network)	<input type="checkbox"/>	Paquet
	<input type="checkbox"/>	Couche liaison de données (Data Link)	2	Couche liaison de données (Data Link)	<input type="checkbox"/>	Trames
	<input type="checkbox"/>	Physique(Physical)	1	Couche Physique(Physical)	<input type="checkbox"/>	BIT
		Support de communication				

Figure I. 6 : Représentation des couches du modèle OSI

1. La couche physique : fournit les moyens mécaniques, électriques, fonctionnels et procéduraux nécessaires à l'activation, au maintien et à la désactivation des connexions physiques destinées à la transmission de bits entre deux entités de liaison de données.

2. La couche liaison de données : fournit les moyens fonctionnels et procéduraux nécessaires à l'établissement, au maintien et à la libération des connexions de liaison de données entre entités du réseau. Elle détecte et corrige, si possible, les erreurs dues au support physique et signale à la couche réseau les erreurs irrécupérables. Elle supervise le fonctionnement de la transmission et définit la structure syntaxique des messages, la manière d'enchaîner les échanges selon un protocole normalisé ou non.

3. La couche réseau : assure toutes les fonctionnalités de relai et d'amélioration de services entre entité de réseau, à savoir : l'adressage, le routage, le contrôle de flux et la détection et correction d'erreurs non réglées par la couche 2.

4. La couche transport : assure un transfert de données transparents entre entités de session et en les déchargeant des détails d'exécution. Elle a pour rôle d'optimiser l'utilisation des services de réseau disponibles afin d'assurer au moindre coût les performances requises par la couche session.

5. Les couches session, présentation et application : constituent les couches hautes du modèle OSI (Open System Interconnexion) et offrent des services orientés vers les utilisateurs alors que les couches basses sont concernées pas la communication fiable de bout en bout. Elles considèrent que la couche transport fournit un canal fiable de communication et ajoutent des caractéristiques supplémentaires pour les applications.

I.1.5 Le modèle TCP/IP et ses couches [USER]

La suite **TCP/IP** est l'ensemble des protocoles utilisés pour le transfert des données sur Internet. Elle est souvent appelée **TCP/IP**, d'après le nom de deux de ses protocoles : TCP (Transmission Control Protocol) et IP (Internet Protocol).TCP/IP, comme son nom l'indique, est en fait constitué de deux protocoles TCP et IP.

TCP (Transmission Control Protocol) se situe au niveau transport du modèle OSI, il s'occupe donc d'établir une liaison virtuelle entre deux ordinateurs. Au niveau de l'ordinateur émetteur, TCP reçoit les données de l'application dans un buffer, les sépare en datagrammes

pour pouvoir les envoyer séparément, l'ordinateur distant (qui utilise le même protocole) à la réception doit émettre un accusé de réception, sans celui-ci, le datagramme est réémis. Au niveau de l'ordinateur récepteur, TCP réassemble les datagrammes pour qu'ils soient transmis à l'application dans le bon ordre.

IP (Internet Protocol) assure l'acheminement de chaque paquet sur le réseau en choisissant la route la plus appropriée. Pour pouvoir s'y retrouver IP va de pair avec un système d'adressage qui identifie de manière unique les réseaux traversés ainsi que chaque entité d'un réseau (appelé aussi nœud: ordinateur, routeur, ...).

La relation entre TCP et IP et la suivante, TCP fait passer à IP un datagramme accompagné de sa destination, IP ne s'occupe pas de l'ordre d'expédition, c'est TCP qui s'occupe de tout remettre en ordre, il se contente de trouver la meilleure route possible.

Souvent les termes « datagrammes » et « paquet » semblent identiques. En fait, on parle de datagramme lorsqu'il est question de TCP (couche 4 de l'OSI), le datagramme est l'unité de données. On parle de paquet pour les couches réseaux (3 IP) et liaison (2 et 1), c'est est une réalité physique, on peut les voir circuler sur le réseau. Généralement, un paquet contient seulement un datagramme, si bien que concrètement, il y a peu de différence entre les deux.

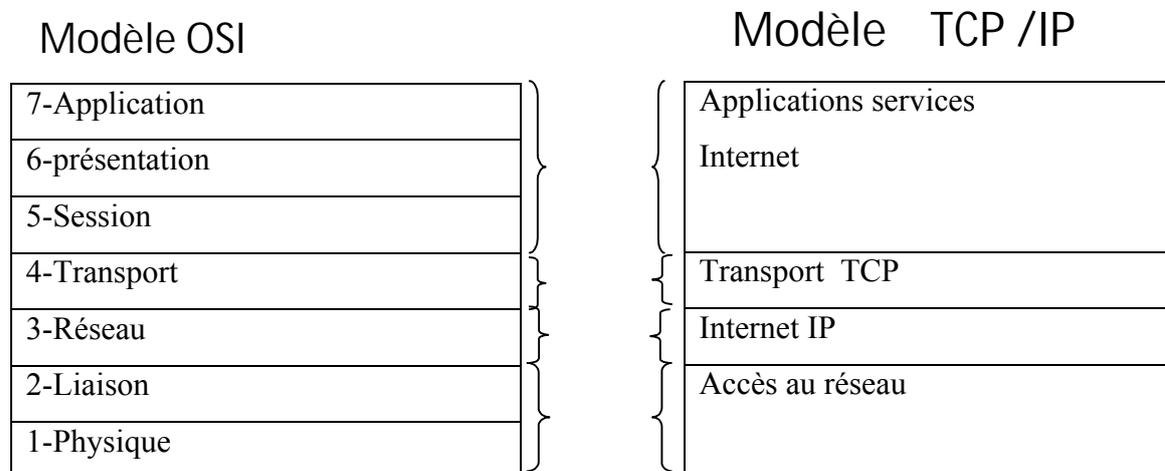


Figure I. 7 : Représentation des couches du modèle TCP/IP et les couches correspondante du modèle OSI.

I.2 Le client /serveur :

I.2.1 Introduction

Ce chapitre propose ensemble de l'architecture client-serveur et définit les notions de base indispensable à la compréhension du modèle.

L'enjeu pour les entreprises consiste à réaliser une intégration de l'informatique personnelle dans le système informatique d'entreprise avec les objectifs suivant :

-Tout utilisateur dans l'entreprises doit pouvoir accéder à toute information utile à sa tâche des lors que cet accès est autorisé par les règle de confidentialité et de sécurité vigueur.

-L'accès doit être instantané et doit pouvoir être fait à partir de n'importe quel poste de travail.

-L'accès à l'information doit avoir lieu par une interface que possible.

Une solution est apportée dans les années 90 : le modèle client /serveur (c /s). Une application est bâtie selon le modèle Client/serveur lorsqu'elle est composée de deux parties, coopérant l'une avec l'autre à la réalisation d'un même traitement. La première partie, appelée processeur client, est installée sur un poste de travail alors que la second, appelée processus serveur, est implantée sur un ordinateur (ou même des ordinateurs, éventuellement situés dans des lieux géographiques différents) chargé de rendre le service. Dans un environnement que le serveur, c'est un des principaux atouts de ce modèle. Donc, le Client/serveur est un mode de dialogue entre client et serveur, ou le client est un processus qui demande l'exécution de services au serveur qui accomplit ces services, et envoie en retour des réponses.

I.2.2 Le modèle client /serveur [ARS 09]

I.2.2.1 Définition: L'architecture informatique client/serveur met en œuvre un ou plusieurs ordinateurs appelés clients, exécutant un programme applicatif communiquant avec un ordinateur distant appelé serveur, qui traite leurs requêtes. Le client/serveur signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en terme de capacités d'entrée-sortie, qui leur fournit des services réseau à un autre ordinateur.

Généralement, la notion de serveur recouvre une activité de stockage d'informations. Mais l'un des rôles du serveur est de réagir à une requête en provenance d'un ordinateur. Cette requête concerne l'utilisation d'une ressource ou la récupération d'informations. En réponse à une requête, le serveur se charge d'assurer le partage de la ressource demandée ou initiée le

transfert d'information vers le client .Le rôle de l'ordinateur client se borne généralement au lancement des requêtes et à l'affichage des informations obtenues.

I.2.2.2. Middleware

Un des composants clé de l'architecture client/serveur est le middleware qui est simplement un logiciel assurant la médiatisation entre clients et serveurs dans le cadre D'architectures de système hétérogènes. En d'autres termes c'est un ensemble des services logiciels construit au-dessus d'un protocole de transport afin de permettre l'échange des requêtes et des réponses associées entre clients et serveurs de manière transparente, permettant de cacher l'hétérogénéité des composants mis en jeu (réseaux, SGBD...).

I.2.3. Pourquoi le client /serveur :

Le concept du client/serveur s'impose dès lorsque l'on souhaite de décomposer l'exécution d'une application et faire en sorte que différentes machines au sens large (matérielle, logicielles) participent à l'exécution de l'application par opposition aux techniques de centralisées sur mainframe ou tout se fait au niveau du serveur central.

Les intérêts du Client/serveur sont multiples :

- **Exploitation** : il est alors facile de l'utiliser (prépare des entrées, comprendre les sorties d'un système ou d'un composant)
- **Flexibilité** : il est facile de modifier un système ou un composant ;
- **Interopérabilité** : les systèmes ou les composants peuvent échanger et utiliser plus facilement les informations échangées;
- **Mise à l'échelle** : il est alors facile de mettre à l'échelle le système ou le composant selon les dimensions du problème à résoudre.

Ces intérêts engendrent un certain nombre d'avantage pour l'entreprise :

- Réduire les couts: Le cout de fonctionnement en client/serveur peut s'avérer plus élevé qu'un système fonctionnant sur une architecture centrée serveur (mainframes) avec des client/serveur introduit un facteur de complexité et d'exploitation, mais l'arrivée de la version Internet du modèle comme modèle dominant.
- permettre le partage des données par plusieurs utilisateurs à partir de différents postes de travail.
- Décharger le réseau, puisque seules les requetés et les résultats correspondants sont véhiculés.

- Possibilité d'utiliser des outils non disponibles ou non adaptés à la machine serveur tels que les tableurs et les logiciels multimédia.
- Permettre une évolutivité des moyens informatiques.

I.2.4 Modes de fonctionnement :

- **Le mode connecté**

Un « canal de communication » se crée entre le client et le serveur (à la demande du client), et les échanges (ordres, accusés de réception, données...) transitent par ce canal, ou alors via d'autres canaux ouverts pour l'occasion

Exemple: IMAP, FTP...

- **Le mode « datagramme » ou « échanges de paquets »**

Il s'agit souvent de systèmes plus simples, où le client envoie sa requête (un ordre). Dans un paquet, et le serveur lui répond dans un ou plusieurs paquets.

ex: TFTP (Trivial File Transfert Protocole), HTTP

I.2.5. Fonctionnement d'un système client/serveur : [CCM]

Un système client/serveur fonctionne selon le schéma suivant :

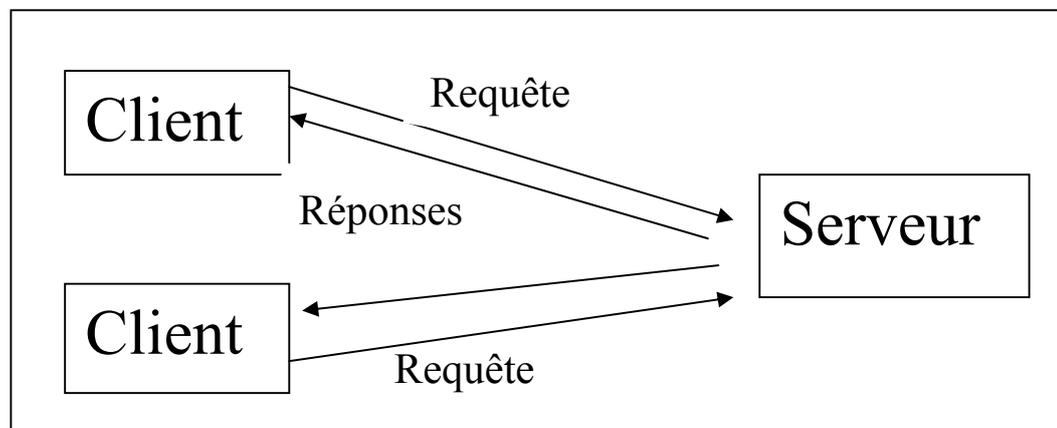


Figure I.8 : Fonctionnement d'un système client/serveur

- **Serveur** : c'est un programme offrant un service sur un réseau (par extension, machine offrant un service).
- **Client** : On appelle logiciel client un programme qui utilise le service offert par un serveur. Le client envoie une requête et reçoit la réponse. Le client peut-être raccordé par une liaison temporaire.
- **Requête** : C'est un message transmis par un client à un serveur décrivant l'opération à Exécuter.

- **Réponse** : C'est un message transmis par un serveur à un client suite à l'exécution d'une opération contenant les paramètres de l'opération.

I.2.6 Les protocoles: [GUP 05,DJADJN]

Qu'est-ce qu'un protocole:

Un protocole est une méthode standard qui permet la communication entre deux machines, c'est à dire un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau.

Il en existe plusieurs selon ce que l'on attend de la communication :

- **HTTP**: Le protocole HTTP est une implémentation de type client serveur des plus simples qui soit. Le client est généralement votre navigateur web (mozilla, konqueror...) mais l'objectif de cet article est de vous donner toutes les bases pour écrire vos propres clients. Celui-ci envoie une requête à un serveur (apache ...) qui lui répond. C'est un protocole en mode texte, généralement utilisé sur une connexion TCP sur le port 80, dédié au transfert de ressource. Il n'est pas dédié à l'échange de fichier même si dans la pratique c'est souvent le cas avec les sites web.
- **TCP/IP** : Le TCP/IP est une suite de protocoles. Le sigle TCP/IP signifie «Transmission Control Protocol/Internet Protocol». Il provient des noms des deux protocoles majeurs de la suite de protocoles, c'est-à-dire les protocoles TCP/IP. Le TCP/IP représente .D'une certaine façon l'ensemble des règles de communication sur internet et se base sur la notion adressage IP, c'est-à-dire le fait de fournir une adresse IP à chaque machine du réseau afin de pouvoir acheminer des paquets de données. La suite de protocole TCP/IP a été créée elle est conçue pour répondre à un certain nombre de critères parmi lesquels :
 - Le fractionnement des messages en paquets ;
 - L'utilisation d'un système d'adresses ;
 - L'acheminement des données sur le réseau (routage) ;
 - Le contrôle des erreurs de transmission de donnée.

La suite de protocole TCP/IP est utilisée pour faire communiquer des réseaux reliant des ordinateurs de différents types de réseau. C'est sur cette suite de protocoles que repose le fonctionnement de l'Internet. Elle contient les protocoles suivants : HTTP, FTP, ARP, ICMP, TCP, UDP, IP, SMTP, TELNET...

- **LDAP** : Le protocole LDAP définit la méthode d'accès aux données sur le serveur au niveau du client, et non la manière de laquelle les informations sont stockées.

Donc il fournit :

- le protocole permettant d'accéder à l'information contenue dans l'annuaire,
 - un modèle d'information définissant le type de données contenues dans l'annuaire,
 - un modèle de nommage définissant comment l'information est organisée et référencée,
 - un modèle fonctionnel qui définit comment on accède à l'information
 - un modèle de sécurité qui définit comment les données et accès sont protégés
 - un modèle de duplication qui définit comment la base est répartie entre serveurs
 - des APIs pour développer des applications clientes,
- **SSH (ou Secure *SHell*)** : est un protocole servant à créer une connexion sécurisée entre deux systèmes. Dans le protocole SSH, un ordinateur client établit une connexion avec un ordinateur serveur.

SSH offre les garanties de sécurité suivantes :

- Après avoir effectué une connexion initiale, le client peut s'assurer de se connecter au même serveur lors des sessions suivantes.
- Le client peut transmettre ses données d'authentification au serveur, telles que son nom d'utilisateur et son mot de passe, en format crypté.
- Toutes les données envoyées et reçues pendant la connexion sont transférées de façon chiffrée, ce qui les rend extrêmement difficiles à déchiffrer et à lire.

Puisque le protocole SSH chiffre tout ce qu'il envoie et reçoit, il peut être utilisé pour sécuriser des protocoles non sûrs. Grâce à la technique de retransmission de port, un serveur SSH peut être employé pour sécuriser des protocoles non sûrs,

- **SMTP(*Simple Mail Transfer Protocol*, traduisez *Protocole Simple de Transfert de Courrier*)** : est le protocole standard permettant de transférer le courrier d'un serveur à un autre en connexion point à point. Il s'agit d'un protocole fonctionnant en mode connecté, encapsulé dans une trame TCP/IP. Le courrier est remis directement au serveur de courrier du destinataire. Le protocole SMTP fonctionne grâce à des commandes textuelles envoyées au serveur SMTP (par défaut sur le port 25). Chacune des commandes envoyées par le client (validée par la chaîne de caractères ASCII *CR/LF*, équivalent à un appui sur la touche entrée) est suivie d'une réponse du serveur SMTP composée d'un numéro et d'un message descriptif.

- **Le protocole FTP :** permet d'échanger un fichier à la fois, dans les deux sens entre la machine client (celle qui a initié la connexion, donc la machine appelante) et la machine serveur (celle qui fournit le service FTP, donc la machine appelée). Le protocole FTP permet aussi d'autres actions telles que la création et la suppression de répertoires (à condition qu'il soit vide), le listage des fichiers, la suppression et le renommage de fichiers, etc.
- **Le protocole Telnet :** est un protocole standard d'Internet permettant l'interfaçage de terminaux et d'applications à travers Internet. Ce protocole fournit les règles de base pour permettre de relier un client (système composé d'un affichage et d'un clavier) à un interpréteur de commande (côté serveur).

Le protocole Telnet s'appuie sur une connexion TCP pour envoyer des données au format ASCII codées sur 8 bits entre lesquelles s'intercalent des séquences de contrôle Telnet. Il fournit ainsi un système orienté communication, bi-directionnel (half-duplex), codé sur 8 bits facile à mettre en œuvre.

Le protocole Telnet repose sur trois concepts fondamentaux :

- Le paradigme du terminal réseau virtuel (NVT, *Network Virtual Terminal*) ;
 - Le principe d'options négociées ;
 - Les règles de négociation.
-
- **Protocole NNTP (Network News Transfer Protocol):** NNTP (Network News Transfert Protocol) est le protocole d'échange des news ou forums de discussions à travers Usenet (nom donné au réseau logique constitué des serveurs de news disséminés sur la planète). Il assure l'échange des news entre les serveurs et également la communication entre serveur et postes clients aussi bien pour la lecture que pour l'écriture de Messages. Ainsi, lorsqu'un utilisateur poste un article dans un groupe de news, il est dans un premier temps déposé sur le serveur de news auquel le poste client est relié. Puis, ce serveur va réexpédier cet article aux différents serveurs auxquels il est relié, qui eux-mêmes procéderont de la sorte. Ainsi, en quelques heures un message posté à Angers peut se retrouver sur un serveur de news en Australie. Mais, ce processus de diffusion systématique, n'est pas assuré pour tous les groupes de news existant au niveau mondial, car chaque serveur de news n'assure le relai que de certains

groupes. En effet, tout serveur de news fixe pour chaque groupe la durée de conservation des messages sur ses disques durs.

- **Le protocole HTTP (HyperText Transfer Protocol)** est le protocole le plus utilisé sur Internet depuis 1990. La version 0.9 était uniquement destinée à transférer des données sur Internet (en particulier des pages Web écrites en HTML). La version 1.0 du protocole (la plus utilisée) permet désormais de transférer des messages avec des en-têtes décrivant le contenu du message en utilisant un codage de type MIME.

Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format HTML)

- **TCP (qui signifie Transmission Control Protocol)**, soit en français: Protocole de Contrôle de Transmission) est un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle (c'est-à-dire le protocole IP). Lorsque les données sont fournies au protocole IP, celui-ci les encapsule dans des datagrammes IP, en fixant le champ protocole à 6 (Pour savoir que le protocole en amont est TCP...). TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission. Les caractéristiques principales du protocole TCP sont les suivantes :

- TCP permet de remettre en ordre les datagrammes en provenance du protocole IP
- TCP permet de vérifier le flot de données afin d'éviter une saturation du réseau
- TCP permet de formater les données en segments de longueur variable afin de les "remettre" au protocole IP
- TCP permet de multiplexer les données, c'est-à-dire de faire circuler simultanément des informations provenant de sources (applications par exemple) distinctes sur une même ligne
- TCP permet enfin l'initialisation et la fin d'une communication de manière courtoise.

Le but du TCP

Grâce au protocole TCP, les applications peuvent communiquer de façon sûre (grâce au système d'accusés de réception du protocole TCP), indépendamment des couches inférieures. Cela signifie que les routeurs (qui travaillent dans la couche Internet) ont pour seul rôle

l'acheminement des données sous forme de datagrammes, sans se préoccuper du contrôle des données, car celui-ci est réalisé par la couche transport (plus particulièrement par le protocole TCP).

Lors d'une communication à travers le protocole TCP, les deux machines doivent établir une connexion. La machine émettrice (celle qui demande la connexion) est appelée client, tandis que la machine réceptrice est appelée serveur. On dit qu'on est alors dans un environnement Client-Serveur. Les machines dans un tel environnement communiquent en mode connecté, c'est-à-dire que la communication se fait dans les deux sens.

Pour permettre le bon déroulement de la communication et de tous les contrôles qui l'accompagnent, les données sont encapsulées, c'est-à-dire qu'on ajoute aux paquets de données un en-tête qui va permettre de synchroniser les transmissions et d'assurer leur réception.

Une autre particularité de TCP est de pouvoir réguler le débit des données grâce à sa capacité à émettre des messages de taille variable, ces messages sont appelés segments.

Le protocole **ARP** a un rôle phare parmi les protocoles de la couche Internet de la suite TCP/IP, car il permet de connaître l'adresse physique d'une carte réseau correspondant à une adresse IP, c'est pour cela qu'il s'appelle Protocole de résolution d'adresse (en anglais ARP signifie Address Resolution Protocol).

Chaque machine connectée au réseau possède un numéro d'identification de 48 bits. Ce numéro est un numéro unique qui est fixé dès la fabrication de la carte en usine. Toutefois la communication sur Internet ne se fait pas directement à partir de ce numéro (car il faudrait modifier l'adressage des ordinateurs à chaque fois que l'on change une carte réseau) mais à partir d'une adresse dite logique attribuée par un organisme: l'adresse IP.

Ainsi, pour faire correspondre les adresses physiques aux adresses logiques, le protocole ARP interroge les machines du réseau pour connaître leur adresse physique, puis crée une table de correspondance entre les adresses logiques et les adresses physiques dans une mémoire cache.

Lorsqu'une machine doit communiquer avec une autre, elle consulte la table de correspondance. Si jamais l'adresse demandée ne se trouve pas dans la table, le protocole ARP

émet une requête sur le réseau. L'ensemble des machines du réseau vont comparer cette adresse logique à la leur. Si l'une d'entre-elles s'identifie à cette adresse, la machine va répondre à ARP qui va stocker le couple d'adresses dans la table de correspondance et la communication va alors pouvoir avoir lieu.

I.2.7 Caractéristique du client/serveur [ASG 08]

- **Service** : le modèle client/serveur est bâti sur une relation entre des processus tournants sur des machines séparées. Le processus serveur est fournisseur de service. Le processus client est un consommateur de services. Le modèle établit ainsi une répartition claire des fonctions à partir de la notion de service.
- **Partage des ressources** : un serveur peut traiter plusieurs clients en même temps et contrôle leur accès aux ressources.
- **Transparence à la localisation** : les processus serveur et client peuvent résider sur la même machine ou par l'intermédiaire d'un réseau, sur deux machines différentes interconnectées. Le logiciel client/serveur masque aux clients la localisation du serveur en redirigeant les demandes de service si nécessaire. Un programme peut être client, serveur ou les deux.
- **Echange de messages** : client et serveur sont des systèmes à liaison épisodique qui interagissent au moyen de messages. Le message est le mécanisme d'émission des demandes de service et des réponses à celles-ci.
- **Asymétrie des protocoles** : la relation entre clients et serveur est de type plusieurs vers un. C'est toujours le client qui déclenche le dialogue en demandant un service. Les serveurs attendent passivement les requêtes des clients. Notez que, dans certains cas, un client peut transmettre une référence à un objet de type callback lorsqu'il invoque un service. Cela permet au serveur de rappeler le client, lequel devient alors un serveur.

I.2.8 Types de clients : [GEO 00]:

- **Client léger** :

Le poste client accède à une application située sur un ordinateur dit « serveur » via une interface et un navigateur Web. L'application fonctionne entièrement sur le serveur, le poste client reçoit la réponse « toute faite » à la demande (requête) qu'il a formulée.

- **Client lourd :**

Le poste client doit comporter un système d'exploitation capable d'exécuter en local une partie des traitements. Le traitement de la réponse à la requête du client utilisateur va mettre en œuvre un travail combiné entre l'ordinateur serveur et le poste client.

- **Client riche :**

Une interface graphique plus évoluée permet de mettre en œuvre des fonctionnalités comparables à celles d'un client "lourd". Les traitements sont effectués majoritairement sur le serveur, la réponse "semi-finie" étant envoyée au poste client, où le client "riche" est capable de la finaliser et de la présenter

I.2.9 Les types des serveurs : [BNOK]

- **Serveur de fichiers :**

Dans le cas de serveur de fichiers, le client requiert des enregistrements de fichiers en émettant des requêtes au serveur de fichiers. Les serveurs des fichiers sont utiles pour partager des fichiers sur un réseau et ils sont indispensables pour créer des banques de documents, d'images,...etc. Mais l'obtention de l'information nécessite de nombreux échanges de message sur le réseau.

- **Serveur de base de données :**

Dans le cas de serveur de base de données, le client émet des requêtes SQL sous forme de message en direction du serveur, le résultat est renvoyé au client. Le serveur utilise sa capacité de traitement pour rechercher les données demandées au lieu de transmettre tous les articles au client et de le laisser en faire la sélection, donc la puissance est répartie et utilisée de façon beaucoup plus efficace.

- **Serveur groupware :**

Le groupware s'intéresse à la gestion d'information semi-structurée telle que le texte, l'image, courrier, messagerie et ordonnancement de tâches.

- **Serveur d'application objet :**

Dans le cas d'un serveur objet, l'application client/serveur est écrite sous forme d'un jeu d'objets communications. Les objets client communication avec les objets serveur au moyen d'un courtier d'objets ou ORB (Object Request Broker). Le client invoque une méthode sur un objet distant, l'ORB localise une instance de la classe, appelle la méthode demandée et renvoie les résultats à l'objet client.

- **Serveur d'application web :**

Le world wide web est la première application client/serveur, ce modèle consiste en des clients légers, portables qui communiquent avec de très gros serveurs, et un serveur web qui renvoie des documents lorsque le client les demande.

I.2.10 Architecture client/serveur : [WIKIPÉDIA, ORD 99]

- **Architecture à 2 niveaux :**

L'architecture à deux niveaux (aussi appelée *architecture 2-tier*, *tier* signifiant *rangée* en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service.

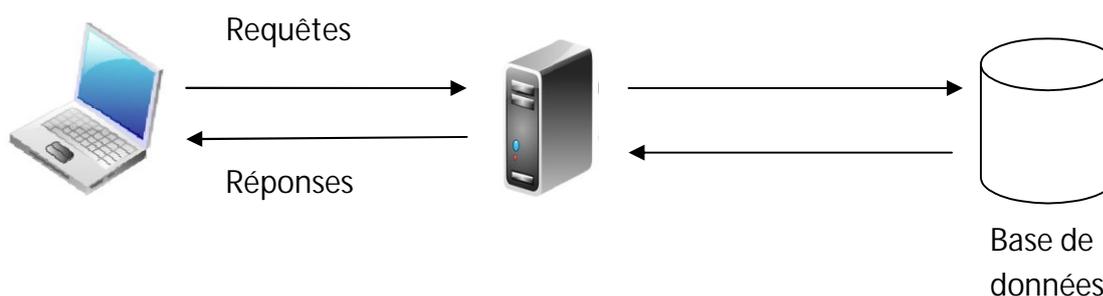


Figure I.9 : Le client/serveur à deux niveaux

Dans cette architecture la technologie applicative est enfouie soit dans le serveur, soit dans le client et soit dans les deux.

- **Architecture client/serveur à 3 niveaux :**

Cette architecture est généralement constituée par :

- **Un client** : qui demande la ressource.
- **Un serveur d'application** : (middleware) c'est le serveur qui charge de fournir la ressource mais en faisant appel à un autre serveur.
- **Un serveur secondaire** : c'est celui qui fournit le service au premier serveur (souvent un serveur de base de données). Dans cette architecture la logique applicative réside

dans le niveau intermédiaire et est séparée des données et de l'interface utilisateur. Les serveurs d'application web sont un bon exemple de système client/serveur à trois niveaux, L'architecture internet se décompose en trois niveaux logiques comme suit :

- l'interface utilisateur gérée par le langage html (hyper texte markup langage) et éventuellement enrichie par l'emploi de html dynamique et de composants java ou ActiveX.
- serveur web ou http(hyper texte transfert protocole).
- le serveur de données .

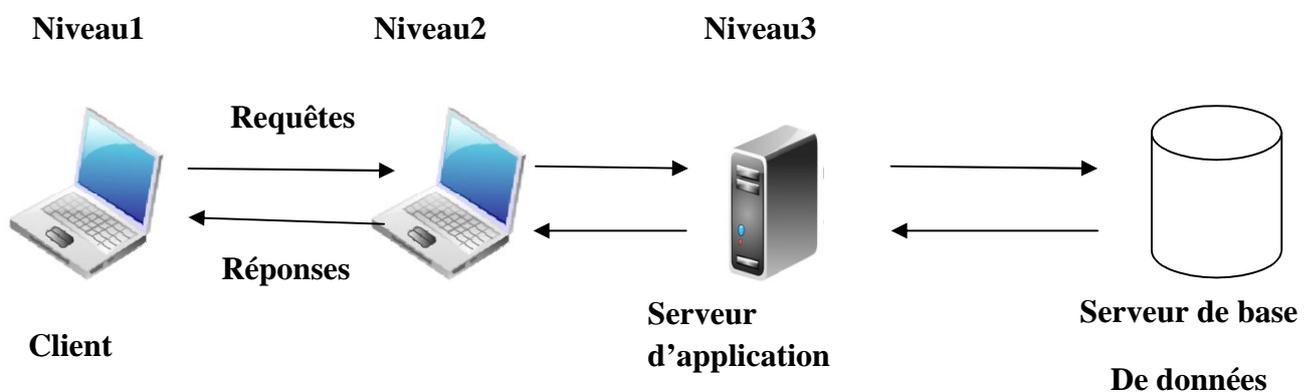


Figure I.10 : Le client/serveur à trois niveaux

- **Architecture à multi niveaux :**

Dans l'architecture à 3 niveaux, chaque serveur effectue une tâche (un service) spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à N niveaux

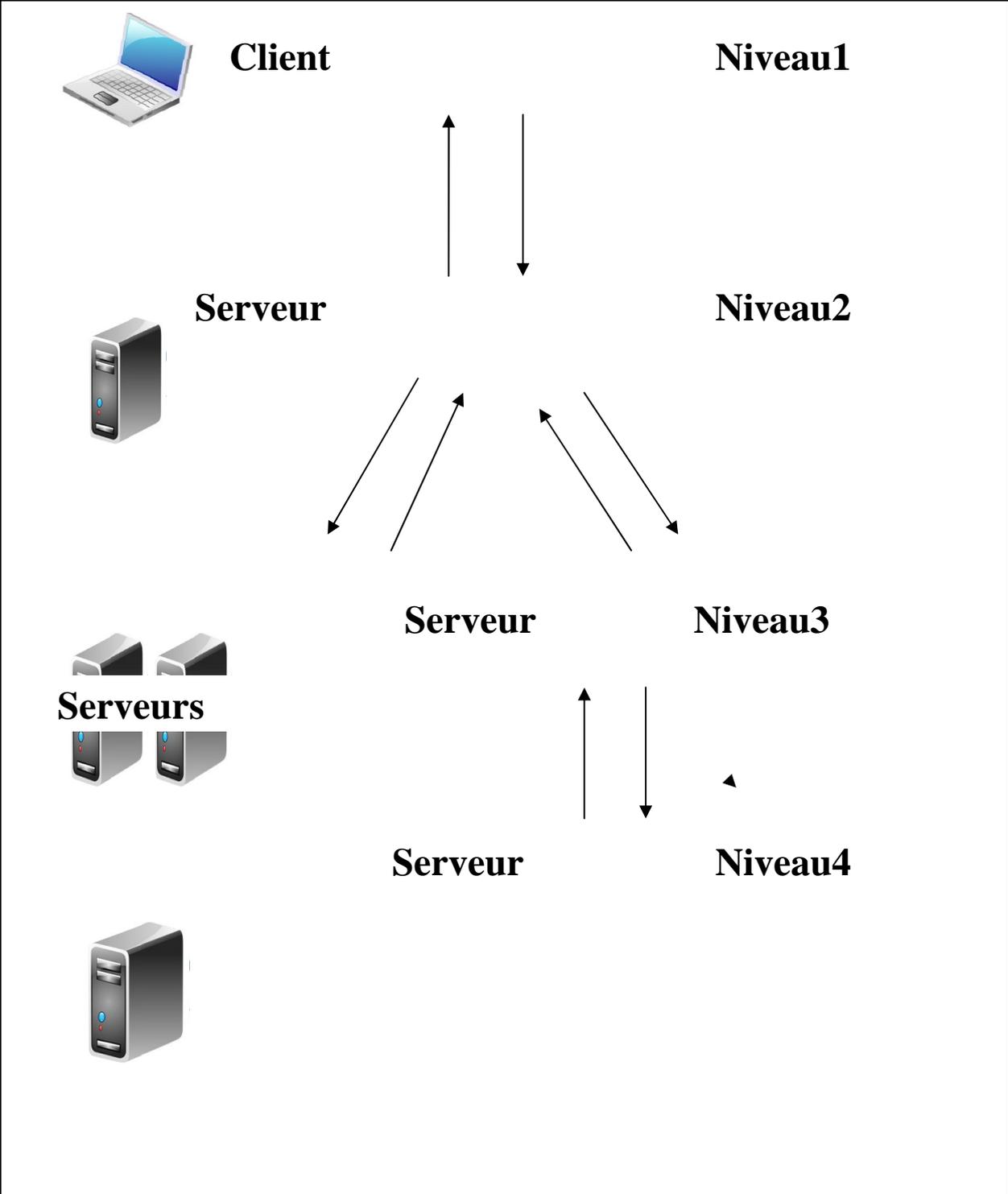


Figure I.11 : Le client/serveur à multi niveaux

I.2.11 Avantages de l'architecture client/serveur : [GEO 00]:

- Grande vélocité sur des grands volumes de données et de traitements.
- Toutes les données sont centralisées sur un seul serveur, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels.
- Les technologies supportant l'architecture client-serveur sont plus matures que les autres.
- La complexité du traitement et la puissance de calculs sont à la charge du ou des serveurs, les utilisateurs utilisant simplement un client léger sur un ordinateur terminal qui peut être simplifié au maximum.
- Maintenance matériel minime.

I.2.12 Inconvénients de l'architecture client/serveur :

- Utilisation de langages de programmation anciens.
- Calcul scientifique complexe impossible.
- Si le serveur n'est plus disponible, plus aucun des clients ne fonctionne (le réseau pair-à-pair continue à fonctionner, même si plusieurs participants quittent le réseau).
- Les coûts de mise en place et de maintenance peuvent être élevés.

Conclusion :

Dans ce chapitre, on a essayé de présenter des généralités sur les réseaux : leurs différents types, modèles de réseau ...etc. et on a présenté un petit aperçu sur les protocoles utilisés et nous avons présenté le paradigme client /serveur, son intérêt, sa mise en œuvre et son fonctionnement. Le chapitre suivant est consacré à la présentation de la technologie des services web.

II Introduction

Depuis toujours, les systèmes d'information ont cherché à offrir un accès rapide, généralisé à l'information pertinente. Traduit ces dernières années par un couplage en interne aux technologies de l'intranet et en externe, à celles de l'internet, particulièrement dans le cadre du Web. En quelques mois, les web services (services web en français) sont devenus le nouveau point de convergence technologique de l'ensemble des acteurs du marché de l'informatique et de l'entreprise. L'objectif des Web Services est de consentir à la réalisation rapide et efficace des systèmes d'information répartis sur internet et intranet, en incorporant des applications existantes et nouvelles afin de faciliter leurs accès entre entreprises et ainsi simplifier les échanges de données.

Les services web sont des applications auto descriptives, modulaires et faiblement couplées qui fournissent un modèle simple de programmation et de déploiement d'applications, basé sur des normes, et s'exécutant au travers de l'infrastructure web.

Pour comprendre le fonctionnement des Web services, ce chapitre commence par présenter le concept d'architecture Orientée services (SOA) respecté par les Web services. Ensuite il introduit les web service en mettant l'accent sur les standards sur lesquelles ils se basent et leurs architectures et leurs nombreuses utilisations par les entreprises.

II.1 SOA, Services Oriented Architecture [CR 185]

II.1.1 Introduction

Le système d'information de l'entreprise est généralement constitué d'applications et de données constituant son héritage. Devant s'adapter en permanence et être de plus en plus réactives aux variations des marchés notamment les fusions de groupe, commerciales, le changement technologiques, et autres, cet héritage a tendance à devenir hétérogène et à se spécialiser par métier (entité, service, ect), ce qui provoque un fonctionnement en silo, c'est-à-dire un cloisonnement des différents métiers empêchant certaines formes de transversalité et masquant au décideur une vision globale du système d'information de son entreprise.

L'Architecture orienté service est une solution à ce problème. En se basant sur l'intégration des applications de l'entreprise (EAI, Entreprise Application Intégration) qui consiste à développer des connecteurs spécifiques permettant de faire communiquer entre eux les différents

silos de l'entreprise et en fédérant plusieurs technologies avancées, l'architecture SOA promet une évolution majeure dans la conception du système d'information de l'entreprise.

II.1.2 Qu'est-ce que l'Architecture Orienté Services ?

Lancée par Gartner Group, la notion d'Architecture Orientée Services (calque de l'anglais Service Oriented Architecture, ou SOA) n'est ni une technologie, ni un progiciel, mais une composites en services interopérables et réutilisables.

L'idée sous-jacente est de cesser de construire la vie de l'entreprise autour d'applications pour faire en sorte de construire une architecture logicielle globale décomposées en services correspondant aux processus métiers de l'entreprise.

II.1.3 pourquoi le SOA ?

Aujourd'hui, la faiblesse des progiciels réside principalement dans la dynamique des entreprises. Si certaines applications restent pérennes, d'autre doivent être rapidement améliorées ou remplacées. Certaines autres ne constituent pas un frein à la bonne marche de l'ensemble.

Il fallait donc imaginer une architecture modulaire qui saurait incorporer l'existant tout en permettant le développement de nouveaux services performants et mutualiste.

II.1.4 Les objectifs de l'architecture orientée SOA

Les axes majeurs de la SOA sont :

- La réutilisation et la composition, permettent le partage de modules entre applications et les échanges inter-applicatifs ;
- La pérennité, qui implique notamment le support des technologies existantes et à venir ;
- L'évolutivité, car toute application est vivante peut se voir griffer de nouveau modules et doit pouvoir répondre aux nouveaux besoins fonctionnels ;
- L'ouverture et l'interopérabilité, pour partager des modules applicatifs entre plateformes et environnements ;
- La distribution, pour pouvoir utiliser ces modules à distance et les centraliser au sein de l'entreprise par exemple ;
- La performance, avec en priorité l'accent mis sur la montée en charge.

II.2 Les Web Services [ING.info.93]

Après avoir abordé les points cruciaux relatifs à l'architecture SOA, nous allons aborder en détail les concepts appelées collectivement Web Services grâce auxquelles l'architecture SOA est entrée depuis peu dans le domaine du réel.

II.2.1 Définition

Nonobstant les nombreux débats sur la définition du terme Web Services, nous utiliserons ces différentes définitions :

- Au sens large, Un web service est :
 - Une « unité logique applicative » accessible en utilisant les protocoles standards d'Internet ;
 - Une « librairie » fournissant des données et des services à d'autres applications ;
 - Un objet métier qui peut être déployé et combiné sur Internet avec une faible dépendance vis-à-vis des technologies et des protocoles ;
 - Une combinaison des meilleurs aspects du développement à base de composants et du Web ;
 - Un mécanisme de communication entre applications utilisant Internet et un protocole universel comme moyen de transport ;

- Plus spécifiquement :
 - Un service Web est une application logicielle identifiée par un URI dont les interfaces et les liaisons sont définies, décrites et découvertes et supporte une interaction directe avec les autres applications logicielles en utilisant des messages XML via un protocole internet.
 - Les Web services sont la nouvelle vague des applications web. Ce sont des applications modulaire, auto-contenues et auto-descriptives qui peuvent être publiées, localisée et invoquée depuis le web. Les web services effectuent des actions allant de simple requête à des processus métiers complexes. Une fois qu'un web service est déployé, d'autres applications (y compris des web services) peuvent les découvrir et l'invoquer.

II.2.2 Historique [CR 185]

Les services Web prennent leur origine dans l'informatique distribuée et dans l'avènement du Web. Le web en passant par trois grandes générations, dont la première se basait sur des sites

web fournissant des pages Web statiques (HTML) sans structure et impossible à fusionner et la seconde où un site web fournissait des applications réelles (Pages Web dynamiques, ASP, JSP ,PHP,.....) a enfin abouti à une troisième génération où un site web est un composant fournissant des services en XML (web services) qui sont devenus structurés et possible à fusionner. Le but de l'informatique distribuée quand à elle est de permettre à une application sur une machine d'accéder à une fonction d'une autre application sur une machine distante et ce, de la même manière que l'appel d'une fonction locale, indépendamment des plates-formes et des langages utilisés.

Le second point qui a favorisé l'essor des services web, c'est le développement d'Internet par la démocratisation du haut débit, la structuration des données via XML et la recherche d'interopérabilité. Les entreprises publiaient déjà de l'information via des sites web, utilisaient la messagerie et faisaient du commerce électronique, elle l'utilise maintenant pour leurs applications métier (ressources humaines, ventes, finances, ect...). Une grande course entre les principaux acteurs du marché ont progressivement, par soucis d'interopérabilité, adopté les principaux standards des services web que sont SOAP , WSDL et UDDI.

Et enfin le dernier point qui a contribué à la naissance des web services est une nouvelle façon de penser qui a commencée depuis 1990 avec le développement orienté objet qui a atteint très vite ses limites, suivi en 1998 du développement composant (ou super-objet comportant des services mais sans polymorphisme ni héritage) comme le EJB dans le modèle J2EE , DCOM dans l'architecture DNA et CORBA. Les services Web sont nés et cela en 2001, mixant de données, permettant aux sociétés d'être libres pour développer des offres complètes ou d'externaliser des services en toute transparence sans se soucier des technologies utilisées par les clients.

II.2.3 Services web=HTTP+SOAP+WSDL+Composants logiciels

Les services web assurent, à travers le réseau Internet, l'interaction entre les applications, les ordinateurs et les processus métier via les protocoles Internet et XML (eXtensible Markup Language) en permettant d'accéder de manière uniforme, à partir d'un seul site web à plusieurs services applicatifs distants.

Ce nouveau modèle de programmation et de déploiement d'applications assure l'interconnexion de services logiciels en se basant sur les standards suivants facilitant le transport, l'invocation, la description et la recherche des services web :

- **Le protocole HTTP (hypertext transfert protocol)** est un protocole de niveau application qui contrôle le transport de messages sur le web via le port 80 du protocole TCP/IP. La communication http est synchrone et se compose d'une série de messages échangés entre un client et un serveur.
- **Le protocole SOAP (simple object access Protocol)** se charge du transport des messages via HTTP. SOAP définit une enveloppe dans laquelle sont placées les données. C'est un protocole de communication inter applicatif comportant un ensemble de règles communes pour structurer les messages XML et invoquer un service web. Le message SOAP transitant sur internet est lui-même codé en XML. il transporte à la fois les données métiers échangées entre les deux objets distants et les données techniques nécessaires à la bonne exécution de l'appel : nom de la méthode appelée, types des données métier transportées, etc.
- **WSDL (Web Services Description Languages)**, langage basé sur XML, décrit les fonctionnalités constituant l'interface d'un service web. Les outils de développement doivent être capables de générer et de consommer automatiquement ces documents WSDL pour faciliter le développement et l'utilisation des services. Le fichier WSDL résultant expose les méthodes de l'objet COM, Java ou de toute autre application, de manière à ce que n'importe quel autre service web puisse comprendre cette interface.
- **UDDI (Universal Description, Discovery and Intégration)** est le standard qui définit un système d'annuaire permettant d'automatiser la découverte dynamique des services web existant. Ce protocole, également au format XML. Permet de prétorier de façon homogène l'ensemble de services dans un annuaire XML accessible sur internet.

Dès qu'un service web est déployé, les autres applications (et autre service web) Peuvent le découvrir et l'invoquer. De granularité variable, les services web peuvent être répartis sur différents serveurs, s'invoquer mutuellement à travers un réseau local ou étendu dans le cadre des échanges électroniques B2B (business To_ business) ou B2C (business To_consumer) en utilisant les standards XML. Un service web peut être implémenté comme une application autonome, un ensemble d'applications ou un ensemble de composants s'exécutant sur une infrastructure d'intégration ad hoc. Les infrastructures à base de composants (cobra, J2EE...) constituent actuellement les principales plates formes d'intégration des services web.

Tout comme les composants, les services web représentent des fonctions réutilisables. Contrairement aux principales technologie actuelles de composants, les services web reposent sur les standards XML. L'interopérabilité entre services est basée sur l'échange de message XML à l'aide de protocoles web (HTTP, SMTP, FTP).

II.2.3.1 SOAP ?le protocole

II.2.3.1.1 Définition

SOAP (acronyme de Simple Object Access Protocol) est un protocole de communication défini à l'origine par Microsoft, puis standardisé par le W3C (Word Wide Web Consortium), utilisant la notation XML permettant de définir les mécanismes d'échanges d'information entre des clients et des fournisseurs de services web. Le mécanisme de transport des messages SOAP peut être le protocole HTTP, SMTP, FTP,...

II.2.3.1.2 Structure des messages SOAP

Un message SOAP n'est en effet pas qu'un fichier XML classique, avec seulement un espace de nom prenant en charge les diverses extensions. Le document décrivant un message de ce type est en réalité une déclaration très structurés, appelée « enveloppe SOAP peut être décomposée en deux composants : l'entête (header) et le corps (Body) du message. Il peut y avoir plusieurs entêtes, ou un entête vide.

En résumé, la structure d'un message SOAP peut être illustrée par la figure suivante :

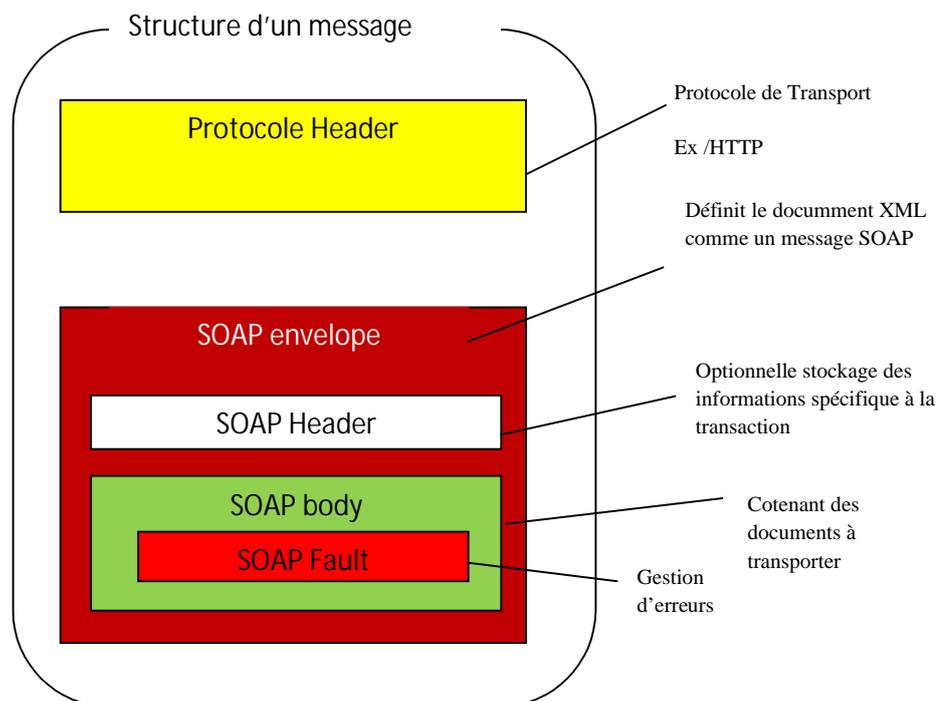


Figure II.1 Structure d'un message SOAP

- **SOAP envelope** (enveloppe) est l'élément de base du message SOAP. L'enveloppe contient la spécification des espaces de désignation (namespace) et du codage de données.
- **SOAP header** (entête) est une partie facultative qui permet d'ajouter des fonctionnalités à un message SOAP de manière décentralisée sans agrément entre les parties qui communiquent. C'est ici qu'il est indiqué si le message est mandataire ou optionnel. L'entête est utile surtout, quand le message doit être traité par plusieurs intermédiaires.
- **SOAP body** (corps) est un container pour les informations mandataires à l'intention du récepteur du message, il contient les méthodes et les paramètres qui seront exécutés par le destinataire final.
- **SOAP fault** (erreur) est un élément facultatif défini dans le corps SOAP et qui est utilisé pour reporter les erreurs.

II.2.3.2 UDDI, l'annuaire universel

II.2.3.2.1 Présentation

Universal Description Discovery and Integration, connu aussi l'acronyme UDDI, est un annuaire de service basé sur XML et plus particulièrement destiné aux Web service, il peut être vu comme les pages blanches (ou jaunes) des services-web. C'est un annuaire permettant à des fournisseurs de présenter leurs services à des 'clients'.

L'annuaire UDDI est consultable sous plusieurs facettes :

- **Les pages blanches** : recensent les entreprises : elles comportent des informations sur les entreprises et contiennent des informations telles que le nom de l'entreprise, ses coordonnées et des descriptions de l'entreprise consultable par l'utilisateur.
- **Les pages jaunes** : comprennent la description des services Web déployés par les entreprises. Elles répertorient les services Web par catégorie.
- **Les pages vertes** : fournissent des informations techniques (souvent en format WSDL) détaillées sur les services fournis.

II.2.3.2.2 Architecture et Fonctions [CR 185]

L'annuaire des services accessible en SOAP comme un service. Il s'agit donc d'un méta service offrant des fonctions d'enregistrement et de recherche. Il est ainsi possible :

- D'enregistrer votre société en fournissant son identité, une description, une liste de catégories et les types de services offerts ;
- D'enregistrer des services pour votre société en déclinant leurs types, noms et catégories ;
- D'enregistrer des opérations pour chacun des services définis avec les données techniques correspondantes (description en WSDL).

Tout ceci peut être effectué par appel d'opération via SOAP avec des documents XML en paramètres. De plus, l'annuaire UDDI peut être répliqué en plusieurs points. Une gestion de copies primaires et secondaires avec synchronisation implicite est spécifiée.

La découverte de services s'effectue par une requête FIND émise à un gérant de requêtes. Celui-ci invoque les services de l'annuaire et retourne un document XML décrivant le service. La figure illustre la découverte des fonctions d'un service.

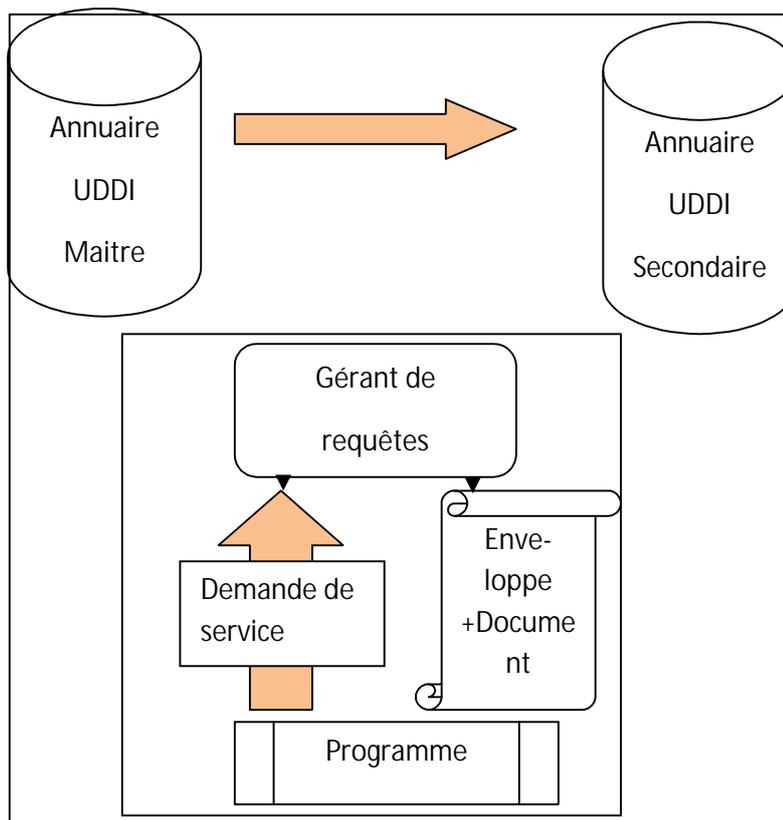


Figure II.2 : Découverte des fonctions d'un service

II.2.3.2.3 Contenu de l'annuaire

Le contenu de l'annuaire est représenté par figure sous forme d'entités liées. Il s'agit de pages blanches définissant les sociétés participantes (*Business Entity*), de pages jaunes décrivant les services (*Business Service*) et de pages vertes (*Binding Templates*) donnant les infor-

mations techniques. Plusieurs modèles techniques sont supportés (*tModel*), en particulier la description en WSDL du service. Des relations entre parties peuvent aussi être publiées (*Publisher Assertion*).

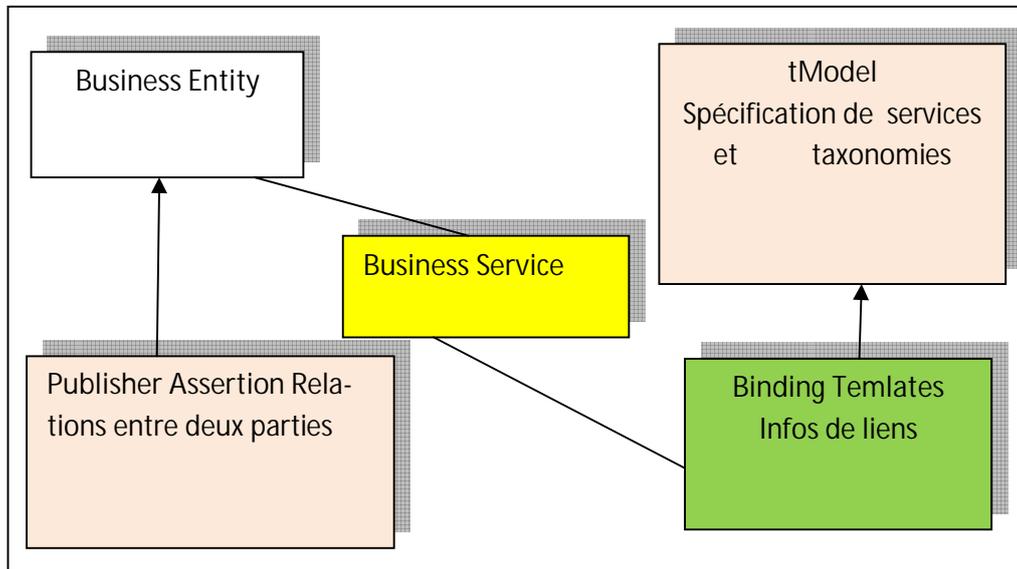


Figure II.3 : Entité composant un annuaire.

II.2.3.3 WSDL, méta-informations [ING .info.93]

II.2.3.3.1 Présentation

WSDL (Web Service Description Language) est un langage reposant sur XML dont on se sert pour décrire les services-web. Il est indispensable à UDDI pour permettre aux clients de trouver les méthodes leur permettant d'invoquer les services web.

Le WSDL décrit une interface publique d'accès à un service Web, notamment dans le cadre d'architectures de type SOA (*Service Oriented Architecture*).

II.2.3.3.2. Éléments d'une définition WSDL

Au-delà de la notion de service et de définition de types de données classiques avec les schémas, WSDL introduit les notions de **port**, **liaison** et **message** pour décrire les services.

- **Élément <type>** : Contient les définitions de types utilisant un système de typage (comme XSD).
- **Élément <message>** : Décrit les noms et types d'un ensemble de champs à transmettre (paramètres d'une invocation, valeur du retour,.....).
- **Élément <porttype>** : Décrit un ensemble d'opération. Chaque opération a zéro ou un message en entrée, zéro ou plusieurs messages de sortie ou de fautes.
- **Élément<binding>** : Spécifie une liaison d'un <porttype> à un protocole concret (SOAP1.1 , HTTP1.1,.....)Un <porttype> peut avoir plusieurs liaison !
- **Élément <port>** : Spécifie un point d'entrée (endpoint) comme la combinaison d'un <binding> et d'une adresse réseau.
- **Élément <service>** : Une collection de points d'entrée (endpoint) relatifs.

D'où la structure d'un document WSDL suivante :

```

<definitions>

    <type>définition des types de données </type>

    <message>définition des messages</message>

    <portType>définition des interfaces</portType>

    <binding>définition des bindings</binding>

    <service>définition de endpoint </service>

</definitions>

```

L'élément **<définitions>** est la racine du document WSDL. Nous définissons l'espace de noms WSDL comme le nom d'espace par défaut, si bien tout élément sans nom d'espace sera une commande WSDL. D'autres espaces de noms gagnent à être spécifiés : l'espace de travail wns ou l'on trouve le WSDL le nom d'espace cible, l'espace des types des schémas xsd, l'espace de définition de SOAP.

II.2.4 Pourquoi les services web ?

Pour l'entreprise, disposer des services web c'est avant tout ouvrir son système d'information à d'autres usages, d'autres besoins et d'autres clients extérieurs à l'entreprise. La mise en œuvre des services web dans l'entreprise accélère ce processus :

- Les services web permettent d'automatiser facilement les processus métier :

Les services web permettent d'intégrer, gérer et automatiser rapidement les processus métier intra et interentreprises en échangeant des informations au format XML. On peut ainsi donc, par ce biais, à son rythme, intégrer dans le système d'information, les différentes activités constituant les chaînes de valeur de l'entreprise.

L'assemblage des différents services peut être orchestré par un moteur de workflow qui contrôle à la volée d'exécution de l'application en fonction du contexte d'exécution et des règles de déroulement du processus métier. Un processus métier

Peut être modélisé avec des langages de types XML tels que BPML, XLANG ou WSFL. Ces derniers permettent d'indiquer l'ordre d'exécution des activités du processus et les actions à effectuer au cours de son déroulement.

Les services web permettent d'intégrer plus facilement des applications et des entreprises dans des processus métier opérationnels. On peut disposer dans ce cadre d'une riche palette d'outils graphiques pour concevoir des schémas XML, transformer des schémas, établir des relations transactionnelles sur Internet et assurer le suivi que l'analyse des données et document qui sont échangés.

- Les services web facilitent l'interopérabilité entre systèmes et plate formes hétérogènes :

La mise en place des services web facilite le dialogue entre environnement hétérogène. Comme les services web peuvent être implémentés sur différents plates formes et avec des langages variés, ils deviennent un moyen technique intéressant pour interconnecter des modules s'exécutant sur des plates formes hétérogènes.

Grace au support étendu des standards publics et des spécifications telles que le XML et HTTP , ainsi qu'aux normes de sécurité telles que le cryptage par clé publique , les signatures numériques, les plates –formes des services web assurent le plus haut niveau d'interfonctionnement et de sécurité entre les applications de l'entreprise et celles de partenaires.

- Les services web facilitent l'intégration d'applications et de services :

L'intégration d'applications et de services a pour objectif de décloisonner les différentes applications informatiques du système d'information de manière à :

1. fluidifier les flux entre les applications ;
2. réduire le temps de latence sur ces flux ;
3. faciliter l'évolutivité de ces flux ;

Face à une problématique croissante d'intégration, tant au niveau des partenaires qu'en interne, le système d'information va s'orienter de plus en plus vers un modèle de composants assemblés selon des processus métier de l'entreprise.

Il existe deux manières concrètes de mise en œuvre des services web pour y parvenir. La première consiste à utiliser les technologies SOAP, WSDL dans une logique d'EAI (Entreprise Application Intégration). Cette approche se révèle rapide, peu coûteuse et largement suffisante dans bien des cas. La seconde application est aussi liée à l'intégration, mais cette fois en dehors de l'entreprise. Dans ce dernier cas, on peut employer le terme de collaboration qui dépeint beaucoup mieux le potentiel des services web dans une logique d'intégration de services métiers.

II.2.5. Modèle d'interaction des services web

La collaboration entre services web s'appuie sur un modèle d'interaction dont les composants assurent trois rôles : le fournisseur de services, l'annuaire de service et le demandeur de service.

II.2.5.1 Scénario général de fonctionnement des services web

Dans le scénario de fonctionnement normal, un fournisseur de services héberge un module logiciel implémentant un ou plusieurs services web accessibles via le réseau. Il définit une description du service et le publie en le faisant enregistrer dans un annuaire de services. Le demandeur de services invoque une opération de recherche pour trouver la description du service pour établir une connexion avec le fournisseur de services et invoquer ou interagir avec l'implémentation du service web. Ce scénario peut être récursif dans le cadre des invocations de services récursives.

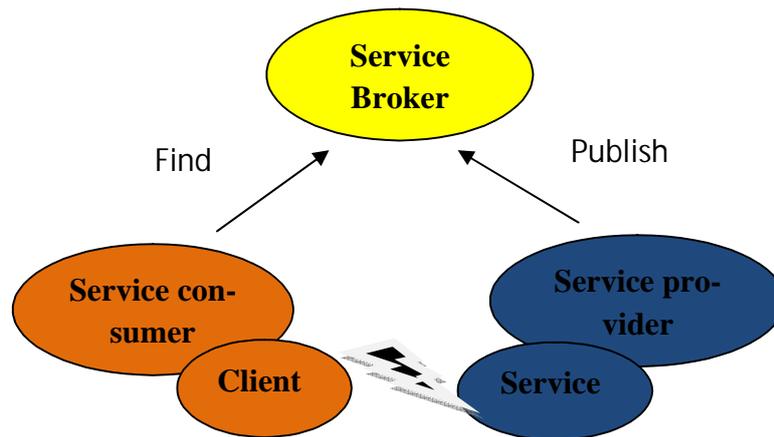


Figure II.4 : Modèle d'interactions des services

Il existe trois types d'opérations pour tirer pleinement parti de ce modèle :

- La publication de description de services ;
- La recherche et la découverte de la bonne description du service ;
- L'association ou l'invocation des services basés sur la description.

Afin d'assurer la collaboration, chacun des composants de ce modèle d'interaction présente une facette métier et une facette d'architecture technique :

- En ce qui concerne le fournisseur de services : d'un point de vue métier, il s'agit du propriétaire du service et d'un point de vue architecture technique, il s'agit de plateforme qui héberge l'accès au service.
- En ce qui concerne le demandeur de services : d'un point de vue métier, cela consiste à demander certaines fonctionnalités à satisfaire et d'un point de vue technique, il s'agit d'une application qui recherche et qui invoque ou initialise une interaction avec un service. Le rôle du demandeur du service peut être assuré par un browser piloté par une personne ou un programme sans interface utilisateur, par exemple un autre service web.
- En ce qui concerne l'annuaire de services : c'est un annuaire de recherche de description de services où les fournisseurs de services publient leurs descriptions de services. Les demandeurs de services trouvent des services et obtiennent les informations correspondantes à leur demande pendant le développement de manière statique ou en cours d'exécution, dynamiquement.

II.2.5.2 Etapes d'exécution des services web [CR 185]

D'une manière simplifiée, les principales étapes d'exécution d'un service web sont les suivantes :

- Découverte du service .le demandeur de service lance la recherche d'un service correspondant à ses besoins sur un annuaire UDDI qui peut être public ou privé.
- Récupération des informations de description du service. Le demandeur de service récupère de l'annuaire UDDI la description de ce service au format WSLD.
- Connexion au service web. la communication entre le composant demandeur du service et celui fournisseur du service est assurée en phase d'exploitation à travers des wrappers (listener) SOAP qui servent d'interfaces entre ces composants et les protocoles de communication de l'infrastructure de déploiement. Le proxy du composant demandeur du service émet une requête SOAP au composant fournisseur du service. Le protocole HTTP véhicule le message SOAP jusqu'au listener du fournisseur du service.
- Le service web renvoie sa réponse. le service web du fournisseur renvoie sa réponse au demandeur sous la forme d'un document XML via SOAP et HTTP.

Dans la pratique, la synchronisation de traitements impliquant plusieurs services web reste difficile. L'orchestration de processus étendus reste difficile à réaliser faute de fonctions transactionnelles au niveau de la couche de transport SOAP et d'outils adaptés pour effectuer le déploiement.

II.3. Apport de xml

XML, le langage fondateur des services web, est le résultat de la coopération d'un grand nombre d'entreprises et de chercheurs partenaires du World Wide Web Consortium (w3c). C'est un langage de description et d'échange de documents structurés. Il permet, comme SGML (Standard Generalized Markup language) dont il est un sous-ensemble, de décrire la structure logique de documents, à l'aide de balises permettant de marquer les éléments qui composent la structure et les relations entre ces éléments.

XML pallie aux limites de HTML en permettant de séparer le contenu d'un document de sa structure et de sa représentation. Le langage XML permet de disposer d'un format universel pour l'échange d'informations entre ordinateurs. Cette séparation structurelle des documents

et de leur réalisation physique offre beaucoup d'avantages en termes de facilité d'échange et de production coopérative de documents.

L'enjeu stratégique de ce langage se situe sur le partage et l'échange d'informations dans tous les domaines métier que ce soit dans le cas de la mise en œuvre du B2C avec, par exemple, l'automatisation du processus de vente sur le web ou du B2B permettant l'amélioration de la réactivité des échanges d'une entreprise avec ses partenaires.

II.3.1 Les technologies XML

Le langage XML est à l'origine d'une multitude de nouvelles spécifications technologiques. Les spécifications de base XML sont les suivantes :

- XML (eXtensible Markup language) : le noyau du langage définit un format d'échange universel.
- DTD (Document type definition) : permet de définir la structure d'un document XML.
- XSL (Extensible markup language) : permet de réorganiser et de représenter les documents XML.
- DOM (Document Object Model) : est une API permettant de manipuler un document XML.
- Namespaces : permet de cataloguer les vocabulaires XML.

Tout autour du langage XML s'est développé et se développe encore un ensemble de standards qui permettent de construire les services web et d'assurer l'interopérabilité entre composants applicatifs de manière à :

- Assurer le déploiement des services web : SOAP , UDDI ,.....
- Effectuer l'échange de données dans un contexte de commerce interentreprises : Biz-Talk ;....
- Assurer l'intégration » de processus collaboratifs sur le web (workflow) BPML ,
- Fournir des mécanismes de sécurité : XKMS , XML signature, XML Encryption.....

II.3.2 XML et services distribués

Cobra et COM sont les technologies majeures actuelles de systèmes distribués à base de composants. Ces deux technologies permettent à une application de faire appel à des services distants par l'intermédiaire d'objets qui exposent leurs fonctionnalités au travers d'interfaces. Techniquement, les services web proposent un modèle objet distribué proche de celui de co-

bra, mais sans nécessité de disposer d'une API commune. Ils reposent sur une architecture d'objets distribués XML/HTTP, étendent les modèles java, COM et Cobra à Internet et assurent l'extension de l'interopérabilité des composants existants.

L'avantage de XML, par rapport aux systèmes d'objets distribués, est sa simplicité. Il permet, en effet, de définir un format de description standard de données.

Ce langage peut être utilisé pour créer des messages auto descriptifs destinés aux échanges entre les applications. De tels messages sont indépendants des systèmes d'exploitation, des langages de programmation et des formats d'affichage.

II.3.3 XML et le remote procedure Call

Dans un système d'information distribué, les applications interagissent les unes avec les autres souvent à l'aide du RCP (Remote Procedure Call).

SOAP, une forme de RCP utilisée dans les services web, est un langage basé sur XML qui permet de décrire et de déclencher des procédures distantes au dessus du protocole HTTP. Pour cela le client transmet un message XML au serveur qui contient le nom de la procédure à exécuter et la valeur des paramètres de cette procédure. La figure II.3 positionne le protocole SOAP parmi les autres composants de communication de l'environnement d'exécution des services web.

SOAP permet donc, à l'aide de XML, la réalisation d'un appel RCP dans un format non propriétaire. En réponse à l'appel SOAP effectué par le client, le serveur exécute le traitement et retourne la valeur du résultat sous forme d'un document XML indiquant si l'exécution a réussi et en précisant la (ou les) valeur(s) de retour. Dans ce cas XML n'est pas utilisé pour transporter des documents mais pour véhiculer des appels de procédure et leurs résultats.

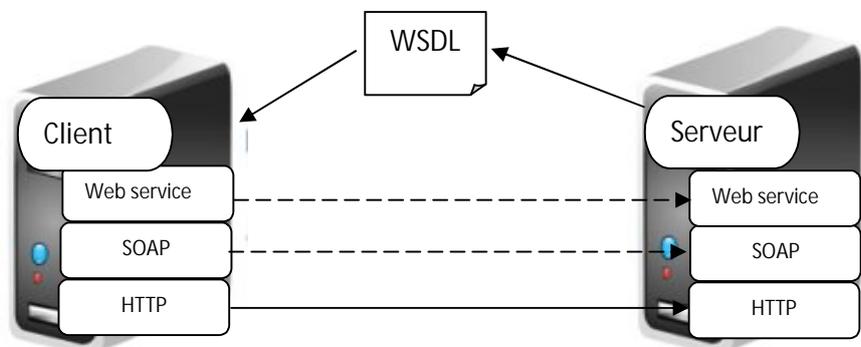


Figure II.5: Place de SOAP dans la pile de communication entre services web

II.3.4 XML et l'architecture des systèmes d'information

A l'issue d'une démarche d'urbanisation d'un système d'information, chaque application devrait apparaître comme une composante indépendante à même de produire et de consommer des données XML tout en assurant le découplage et la minimisation des flux entre les applications.

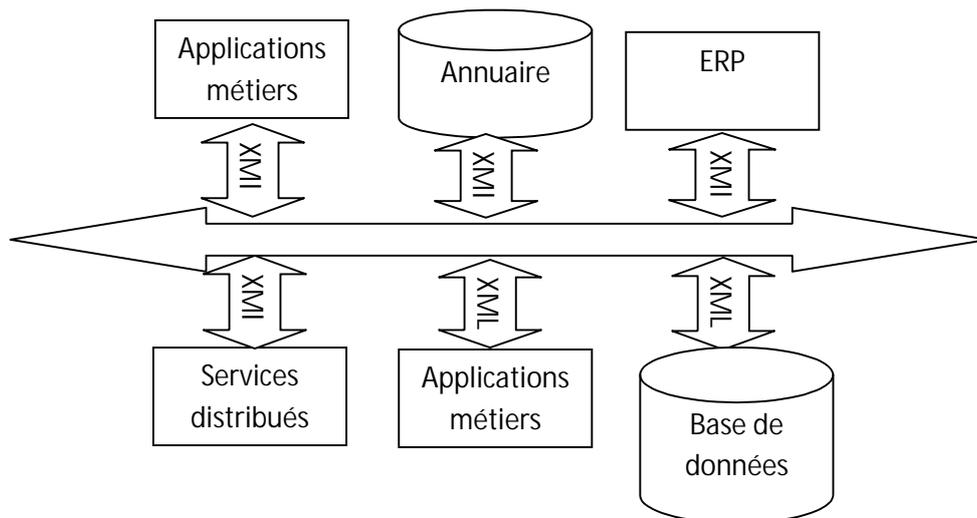


Figure II.6 : XML comme langage commun de communication entre composants du S.I

Comme illustré sur la figure II.8 XML semble être un moyen efficace pour atteindre l'objectif conduisant à la mise en œuvre de ce type d'architecture. Chaque composante du système d'information : les bases de données, les annuaires, les services distribués mais aussi les applications et les ERP (Entreprise Ressource planning) doivent alors fonctionner comme des consommateurs ou des procédures de données XML. XML est ainsi le langage fédérateur pour mettre en place le bus applicatif du système d'information. Le système d'information devient, par ce biais, un simple assemblage de services XML qui communiquent entre eux à travers le bus applicatif.

D'une manière évidente, XML est utilisé dans cette architecture comme format d'importation et d'exportation d'informations. Cette solution simple permet cependant de bénéficier de tous les avantages de XML :

- XML étant autodéscriptif, la description des données est jointe à leurs valeurs. Ceci permet d'envisager un traitement des données sans en connaître le format.

- Le fichier XML obtenu peut être validé si on lui adjoint une DTD. Ceci permet, par exemple, de réaliser un précontrôle simple des fichiers importés.
- Il est envisageable de réaliser des traducteurs permettant d'intégrer les données dans d'autres applications supportant XML en utilisant XSL ou des outils permettant de générer de XSL.

II.3.5 XML et l'intégration d'applications

L'intégration des applications d'entreprises (EAI, Entreprise Application Intégration) est un domaine privilégié pour exploiter l'universalité et la flexibilité apportées par les différentes normes XML :

- Au niveau des données, par la définition des formats de documents XML mais aussi dans la validation des messages avec les schémas XML ou dans leurs transformations de formats avec XSL ;
- Au niveau de l'architecture technique, XML rend possible le découplage nécessaire à la communication entre application hétérogènes.

En effet, l'EAI représente actuellement une combinaison de technologies mises en œuvre dans différents types de produits. une solution EAI complète utilise les services de connectivité pouvant également être offerts par les produits de type middleware, les services de transformation de données pouvant également être offerts par les produits ETL (Extracting, Transforming and Loading data Tools) et les services de gestion de processus pouvant également être offerts par les produits de gestion de workflows.

Intégration de processus Modélisation du workflow ou de processus
Intégration d'application Translation et transformation de données Routage basée sur les règles.
Intégration de composants Serveurs d'application
Intégration de données Outils d'extraction, transformation et changement de données. Management de métadonnées
Intégration de plates- formes Messagerie,ORB et RPC

Figure II.7 : Niveaux d'intégration EAI dans le S.I

Comme illustré sur la figure II.9 nous pouvons identifier cinq niveaux d'intégration de composants d'un système d'information en utilisant les outils EAI :

- **Intégration au niveau de la plate-forme :** Ce niveau d'intégration assure la connectivité entre matériels, operating systems et plates formes applicatives hétérogènes. Les technologies qui offrent une intégration au niveau de la plate forme concernent la messagerie, l'ORB (Objet Request Broker) et le RPC (Remote Procedure Call). La messagerie fournit la connectivité asynchrone. le RPC offre la connectivité synchrone et les ORB couplées à la messagerie offrent les deux modes de connectivité.
- **Intégration de données :** Deux catégories de produits permettent actuellement d'assurer l'intégration de données de l'entreprise : les passerelles vers les bases de données et les outils d'extraction, transformation, déplacement et chargement de données couramment appelés en anglais ETL tools (Extracting, Transforming, Moving and loading data). Les passerelles vers les bases de données assurent l'accès SQL aux sources de données hétérogènes. Ce sont des produits d'accès synchrones aux données. Les outils ETL généralement batchs sont mieux adaptés aux chargements initiaux des entrepôts de données de l'entreprise ou aux transferts de masse batchs. Dans une architecture multiniveaux, les outils de cette catégorie extraient et chargent les données directement sans transiter par la logique de l'application. La plupart de ces outils étaient originellement développés pour la construction des entrepôts de données (datawarehouse).
- **Intégration au niveau des composants :** Ce niveau d'intégration facilite l'introduction de nouvelles fonctionnalités aux packages ERP (Entreprise Resource planning), aux applications clients /serveurs et aux applications existantes. Les serveurs d'applications facilitent l'introduction de nouvelles fonctionnalités aux composants et applications existants. les serveurs d'applications, initialement focalisés sur la gestion de transactions , sont des évolutions naturelles de moniteurs TP (Transactional Processing) et offrent actuellement un ensemble de services en complément à la gestion de transactions comportant le load balancing, la tolérance aux pannes, le pooling de connexions, la gestion des états et de sessions, la sécurité et l'accès aux sources de données relationnelles et non relationnelles.

- **Intégration au niveau des applications :** Ce niveau fournit un cadre général de mise en œuvre d'un ensemble de technologie assurant l'intégration d'applications. La logique de connexion à une application est définie à travers un adaptateur d'application dédié. À ce niveau, le message Broker gère la transformation et la translation de données, le routage basé sur les règles et la connectivité aux applications via les adaptateurs d'applications dédiés (les adaptateurs SAP par exemple).
- **Intégration au niveau processus :** L'intégration au niveau processus est le plus haut niveau d'abstraction et d'adaptabilité pour une solution EAI. Pour s'adapter aux processus complexes interentreprises, les offres d'intégration B2B combinent les outils d'intégration EAI classiques-adaptateurs et courtiers de messages à un moteur de workflow permettant de gérer les processus. Le moteur de workflow pilote le courtier de messages selon des scénarios établis par les architectes métiers. Cela est facilité par l'utilisation des outils à interfaces graphiques de modélisation des processus métiers. Un workflow est capable de piloter à la fois des applications et des hommes, des procédures, des protocoles de transport inclus s'exécutant sur une infrastructure sécurisée. Le moteur de workflow assurant le BPM (business Process Management) est un composant de l'infrastructure d'exécution chargé de piloter le courtier de messages.

Doté de sa fonction BPM, l'EAI nous semble être une bonne technologie pour :

- Supporter la vision processus dans la conception et l'urbanisation des SI ;
- Répondre aux besoins de la flexibilité et de la capacité d'adaptation ;
- Fournir une solution de robustesse et transactionnelle minimale face à l'impossibilité actuelle de disposer des transactions réparties ACID.

La figure suivante présente une architecture d'intégration d'applications d'un système d'information à l'aide des technologies EAI services web.

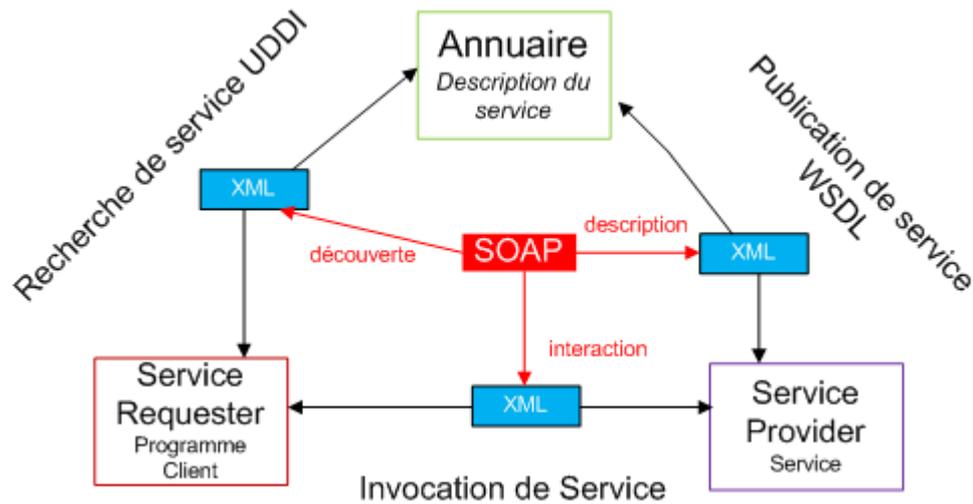


Figure II.8 : Intégration d'un SI à l'aide des technologies EAI

II.4. Architecture des services web

L'architecture des services web est implémentée à l'aide des diverses technologies organisées en quatre couches comme illustré sur la figure suivante :

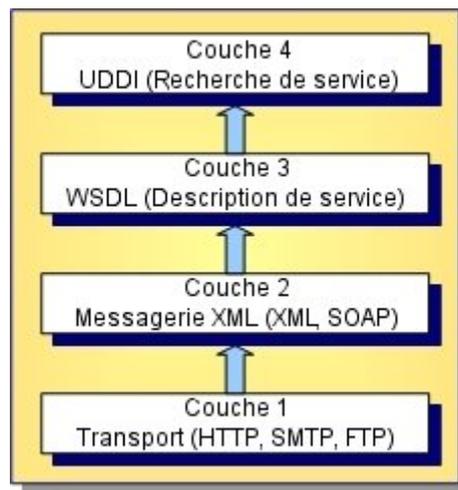


Figure II.09 : Les couches des services web

Chaque couche de la pile des services web répond à des préoccupations fonctionnelles différentes telles que la sécurité, la messagerie fiable, les transactions, le routage, le workflow etc. Comme représenté sur la figure II.12. Dans un souci d'interopérabilité, les différentes couches de la pile des services web s'interfaçent avec les standards.

Les fonctions des couches supérieures reposent sur celles de couches inférieures. La partie gauche représente les standards appliqués à chaque couche.

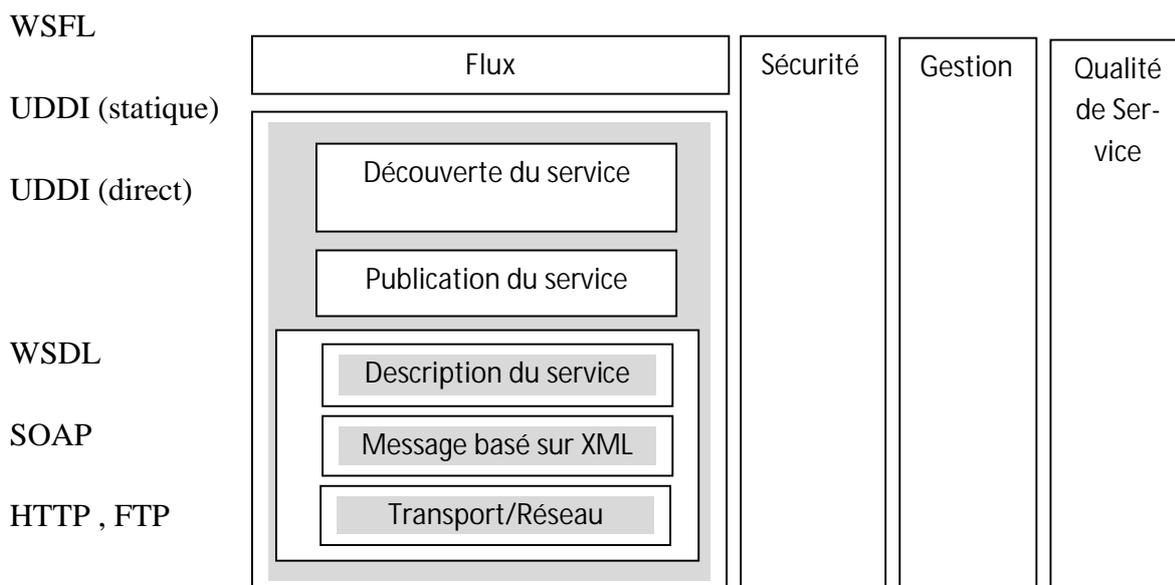


Figure II.10 : Cadre architecturale des services web

Au niveau transport pour assurer la connectivité physique, plusieurs protocoles peuvent supporter les services web : HTTP, SMTP, FTP. Des composants d'infrastructures de type Cobra, RMI.....peuvent également être utilisées à ce niveau.

Le niveau suivant est le niveau message basé sur le standard SOAP qui assure le transport de messages encapsulés au format XML.

Le niveau description de service est actuellement dédié à la description de document en WSDL. WSDL définit l'interface et les mécanismes d'interaction de services. En outre, il est nécessaire de spécifier le contexte métier, la qualité de service et les relations de service à service. Le document WSDL peut être complété par d'autres documents de description de services pour décrire ces aspects de haut niveau.

Actuellement, les standards des services web s'avèrent encore insuffisants pour la construction des systèmes complexes. La majorité des développements en cours dans ce domaine restent circonstrits à l'entreprise et les dictionnaire UDDI demeurent encore privés. Parmi les standards, il manque actuellement, entre autres, les standards relatifs à la sécurité, l'intégrité transactionnelle, la gestion de processus métier ou de gestion contractuelle et les standards verticaux sectoriels.

Les principaux standards ou travaux de standardisation plus au moins finalisés pour couvrir l'ensemble des fonctions de chacune des couches du modèle d'architecture des services web sont présentés dans le tableau 4.1.

Tableau II.1 : Principaux travaux de standardisation

Transport	
DIME (Direct Internet Message Encapsulation)	Un format du message binaire léger permettant d'encapsuler un ou plusieurs contenus utilisateurs permettant la construction de messages simples.
HTTPR (reliable HTTP)	Un protocole qui assure une livraison fiable des paquets HTTP entre le serveur et le client.

Packaging et Extensions	
SOAP (simple object access protocol)	Un protocole léger basé sur XML permettant d'assurer l'échange d'informations dans un environnement distribué et décentralisé.
SOAP DSIG (SOAP Security Extensions : Digital signature)	Une extension SOAP permettant d'utiliser la syntaxe de la signature digitale XML dans les messages SOAP.
SWA (SOAP Messages with Attachments')	Une liaison (binding) permettant au message SOAP d'être véhiculé dans un message MIME multipartis.
WS-License (web services License language)	Une extension SOAP permettant de décrire le codage communément utilisé dans le domaine de la sécurité pour gérer les types de licences.
WS- Referral (web services referral protocol)	Un protocole basé sur SOAP qui implémente les stratégies de routage utilisées par les nœuds SOAP dans un chemin de message afin de pouvoir être dynamiquement configuré.
WS-Routing (web services routing protocol)	Une extension SOAP permettant de router les messages SOAP de manière asynchrone.

WS-Security (Web Services Security Language)	Une extension SOAP qui assure les interactions sécurisées entre les services web fournissant, pour des besoins de sécurité, l'échange de crédit, le maintien de l'intégrité de messages et la confidentialité de messages.
---	--

Description	
WSCL (web services conversation language)	Un format XML de description de séquences des échanges de messages légaux (interactions) que les services web peuvent échanger.
WSCM (web services Component Model)	Un modèle de composant qui définit le mécanisme de packaging de composants d'affichage d'un service web.
WSDL (Web Services Description Language)	Un format XML pour la description des services réseau comme un ensemble des points finaux agissant sur le contenu de messages ou les informations liées aux documents ou aux procédures.
WSEL (web services Endpoint Language)	Un format XML de description de caractéristiques non fonctionnelles des points finaux de service tels que la qualité de service, le coût ou les propriétés de sécurité.
WSFL (Web Services flow language)	Un format XML de description des compositions des services web qui spécifie les patterns d'interactions et d'usage.
WSML (Web Services Meta Language)	Un format XML de description d'informations qui mappent les opérations d'un service avec les méthodes de l'objet COM.
WSXL (Web Services Experience Language)	Un modèle de composants centré services web pour les applications web interactives.
WSUI (Web Services User Interface)	Un modèle de composants permettant d'intégrer la présentation et l'interaction multi étapes aux services réseau XML basés sur le protocole SOAP.
XLANG	Un format XML pour la description du comportement des échanges des messages entre les services web copérants.

En terme d'architecture, idéalement, grace aux services web, il est permis d'envisager à moyen terme la conception de systèmes complètement dynamiques qui auraient la capacité de s'auto-adapter en découvrant

II.4.1 Principes de développement des services web

Le développement d'un service web consiste essentiellement- pour l'instant à étendre le champ d'action d'un objet COM ou java existant par la génération d'une API (Application Programming Interface) ; cette API standard reposant sur un schéma XML normalisé. Dans ce contexte, les plates formes de développement et d'exploitation des services web assurent donc essentiellement deux objectifs ;

- Le premier consiste à créer le fichier de description d'un composant existant au format WSDL. Il s'agit en fait de décrire les différentes méthodes de l'objet en XML, ainsi que l'ensemble de ses propriétés. L'outil de développement doit être, par introspection du composant, capable de lire un fichier WSDL en sens inverse afin d'exposer l'interface des objets distants au développeur.
- Le deuxième objectif concerne la génération des listeners SOAP. Ces modules traduisent les requêtes des plates-formes d'exécution (COM, Java ou autre) au format SOAP et vice versa. Ils jouent le rôle de traducteur et de passerelle entre le monde Internet et le système d'information de l'entreprise. la génération de ces composants (fichier WSDL) peut être statique ou dynamique.

Comme tout développement maîtrisé, il est indispensable de s'appuyer sur un processus de développement approprié au développement des services web. Le processus de développement des services web que nous préconisons repose sur la modélisation des services web avec la notation UML dans le cadre du Unified Process (UP).

Le standard d'interopérabilité de processus du WFMC (Workflow Management Coalition) sert de cadre de description de collaborations interprocessus.

Les services web permettent de concevoir et de déployer rapidement des processus métier intra ou interentreprises. La figure II.13 exemple simple d'architecture d'implémentation des services web.

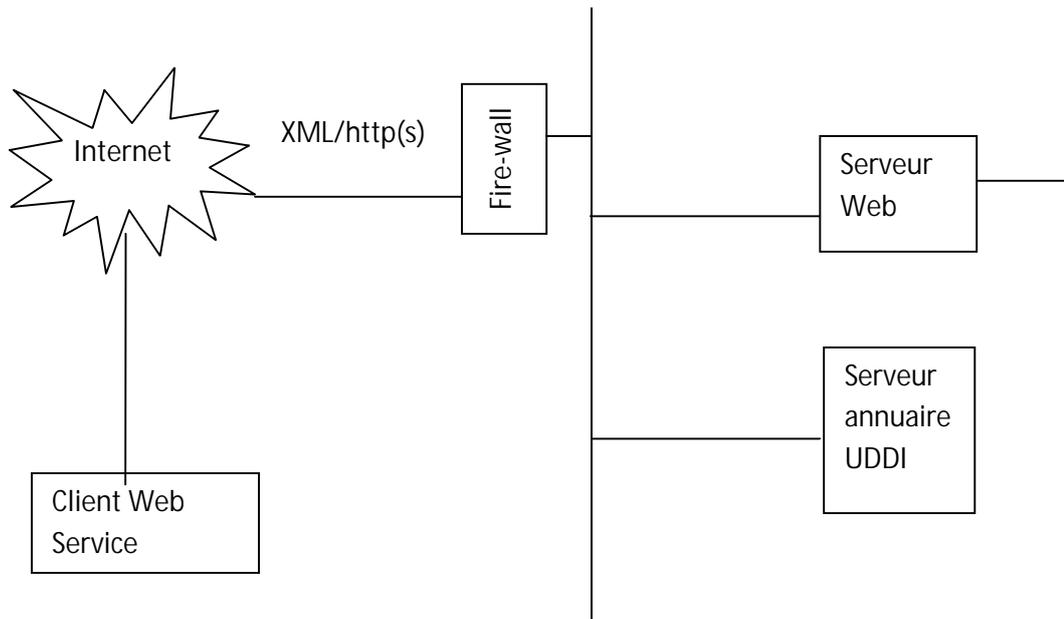


Figure II.11 : Architecture d'implémentation des services web

Conclusion

Les services web sont des applications accessibles sur Internet réalisant chacune une tâche spécifique. Pour fournir une solution à une tâche complexe, on peut regrouper des services web pour n'en former qu'un seul ; on parle alors de composition de services web.

Dans la suite de notre travail nous allons faire l'étude de l'existant.

III Introduction

Dans ce troisième chapitre nous allons d'abord présenter l'entreprise NAFTAL où nous avons effectué notre stage, à savoir, son historique, ses missions qu'elle effectue auprès de ses différentes stations. Nous allons aussi présenter l'organisation de l'entreprise à travers son organigramme générale, ensuite nous nous intéresserons plus particulièrement à la branche commercialisation où nous étudierons son organisation à travers son district puis nous présenterons le diagramme de flux pour les tâches qu'elle effectue et nous citerons les acteurs qui rentrent en jeux.

Nous allons aussi étudier la situation informatique de l'entreprise en définissant l'aspect humain, logiciel ainsi que matériel.

Finalement, nous ferons un diagnostic de la situation existante et nous proposerons des solutions pour y remédier.

III.1 Présentation de NAFTAL

III.1.1 Historique de NAFTAL

Avec un personnel de plus de trente-mille (30000) agents, NAFTAL est le premier distributeur de produits pétroliers en Algérie.

NAFTAL est issue de deux restructurations de l'entreprise SONATRACH. Depuis 18/04/1998 NAFTAL est devenue une société par action (SPA), son capital social est fixé à 6 650 000 000 de dinars algériens (DA) est détenue entièrement par l'entreprise mère SONATRACH à 12 650 000 000.00 DA en aout 2002.

NAFTAL fonctionne sur le plan régional en unité de distribution (UND) jusqu'au 21 décembre 2001 un autre plan de restructuration s'est traduit par la création de trois divisions :

CLPB : Carburant, Lubrifiant, Pneumatique et Bitumes ;

GPL : Gaz de Pétrole Liquéfié;

AVM : Aviation Maritime.

La direction générale a opérée des réajustements suivants à dater du 01 janvier 2004 (décision N 760 du 12/11/2003) :

a. Au niveau central

- Création d'une branche chargée des activités internationales ;
- Création d'une branche direction communication et relation publiques ;

- Séparation de la branche CLPB (carburant, lubrifiant, pneumatique et bitumes) en deux branches ;
- Maintien de la branche «GPL ».

b. Au niveau opérationnel

- Création du district commercialisation orienté client chargé de préserver l'image de marque de la société, de maintenir et améliorer les parts de marché.
- Création de district carburant à vocation régionale chargé de gérer les différents dépôts et les approvisionner en produits à partir des ressources.

III.1.2 Les missions de NAFTAL

NAFTAL est chargée de la commercialisation et de la distribution des produits pétroliers y compris ceux destinés à l'aviation, à la marine, le GPL, les combustibles, les lubrifiants, les bitumes et les pneumatiques. Son activité principale consiste à commercialiser les produits pétroliers, à travers ses différentes organisations et la gestion d'un réseau de distribution sur l'ensemble du territoire national. Ces missions sont :

- Organiser et développer l'activité de commercialisation et de distribution des produits pétroliers et dérivés.
- Stocker et transporter les produits pétroliers commercialisés sur le territoire national.
- Veiller à l'application et au respect des mesures relatives à la sécurité industrielle et à la sauvegarde et la protection de l'environnement.
- Procéder aux études de marché des produits pétroliers.
- Définir et développer une politique en matière d'Audit (concevoir et mettre en œuvre du système intégré d'information).
- Développer et mettre en œuvre des actions visant à une utilisation optimale et rationnelle de ses infrastructures et de ses moyens.
- Veiller à l'application et au respect des mesures liées à la sûreté interne de l'entreprise, conformément à la réglementation.

III.1.3 Organisation de NAFTAL

La nouvelle organisation de NAFTAL est datée du 1 janvier 2006, dans un contexte de marché marqué par de profonds bouleversements. NAFTAL, pour assurer sa pérennité, à l'ambition de se hisser au niveau des meilleurs de l'aval pétrolier.

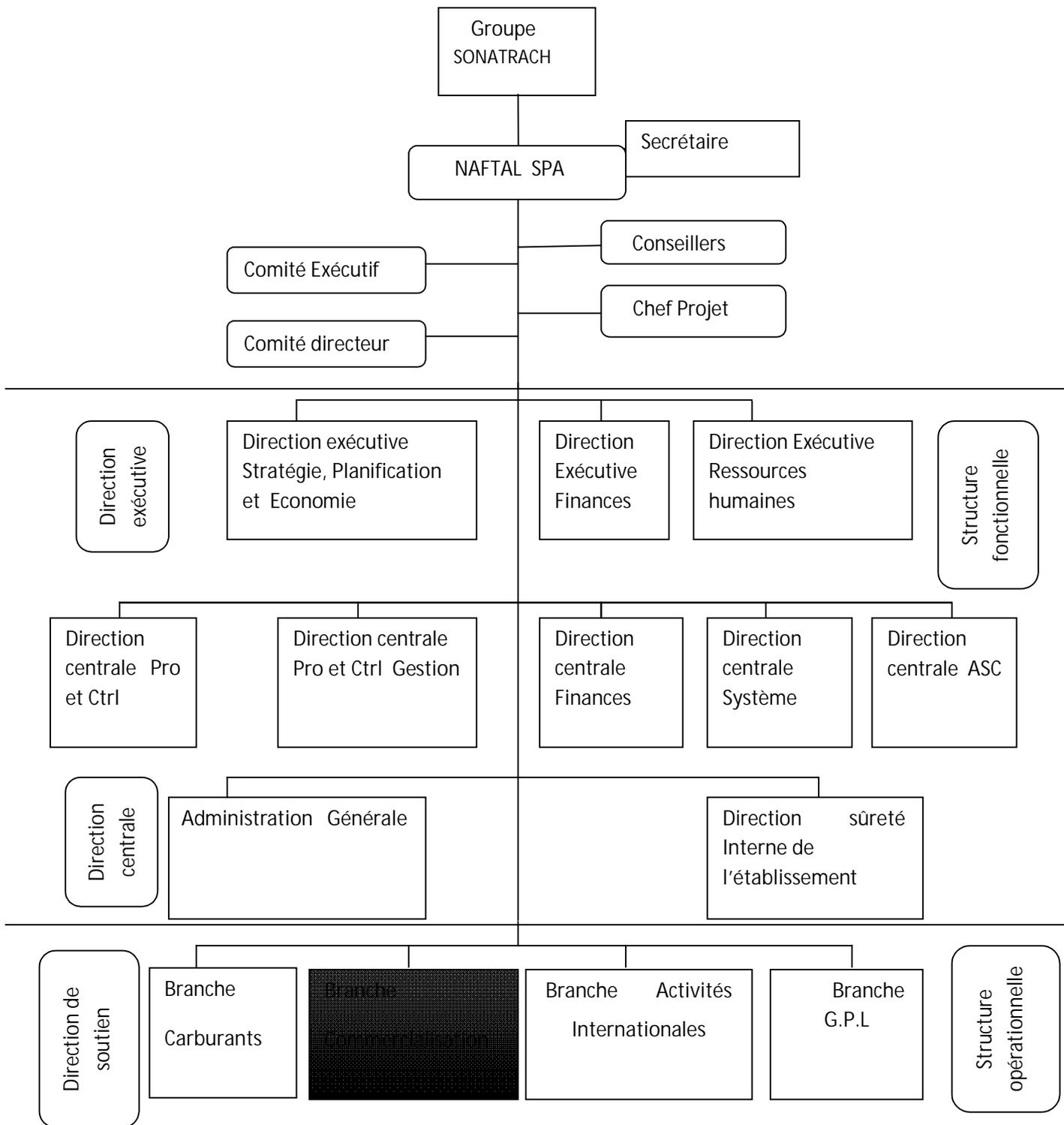
La dernière organisation de la société NAFTAL, résultat de la séparation des activités commercialisation et carburant est structurée comme suit :

Une direction générale qui comporte deux types de structures :

- **Des structures fonctionnelles**, elles sont subdivisées en trois (03) types de directions qui sont :
 - Direction exécutive.
 - Direction centrale.
 - Direction de soutien.
- **Des structures opérationnelles**, contenant quatre (04) branches :
 - Branche carburants.
 - Branche commercialisation.
 - Branche d'activités internationales.
 - Branches GPL.

Cette structuration est schématisée dans l'organigramme figure (**Figure III.1.**), ce processus de séparation nécessite la réorganisation des structures, des districts et des centres de stockage carburants.

La figure suivante illustre l'organigramme l'entreprise NAFTAL,



 Champs d'étude:

Figure III- 1 : Organigramme de l'entreprise NAFTAL .

III.1.4 Branche commercialisation NAFTAL

La branche commercialisation se répartit en douze (12) Districts commerciaux et parmi eux le District commercial de Tizi-Ouzou.

La branche commercialisation est chargée de :

- Commercialiser les produits pétroliers carburants, lubrifiants, pneumatiques et bitumes et autres produits destinés à l'automobile ou à l'automobilisme.
- Proposer toutes mesures en vue d'adapter l'organisation de la branche à la vision stratégique entreprise.
- Mettre en place des systèmes de gestion performants, assainir et actualiser la nomenclature des postes de travail.
- Assurer les prestations de service sur l'ensemble de points de vente.
- Gérer, suivre et contrôler les relations avec la clientèle carburants, pneumatique et bitumes (réseau de station-service, gros consommateurs, autre clients).
- Élaborer et mettre en œuvre la stratégie de marketing du réseau (GD, GL, PVA et SERVICE) ainsi qu'une stratégie de développement (modernisation, réhabilitation et mise en conformité).
- Gérer, contrôler et suivre le réseau commercial en gestion directe.
- Procéder à des inspections régulières pour s'assurer de la qualité des services, du respect des normes de sécurité et de gestion.
- Évaluer et procéder à l'achat des équipements nécessaires à la maintenance et à la rénovation du réseau.
- Initier et coordonner les actions de communication destinées à la promotion et la vulgarisation des produits et veiller au développement du GPL/carburants.
- Initier, diriger et superviser les études du marché.

III.2 Présentation du district commercial de Tizi-Ouzou

Le district commercial de Tizi-Ouzou est situé dans la zone industrielle AISSAT IDIR à 15 km de Tizi-Ouzou station.

Il gère un centre de distribution dépôt OUED AISSI , un centre LUB/PNM et un réseau de dix sept (17)stations en gestion directe (GD).Il se charge aussi du contrôle de six bavardes (06) gestion libre (GL), Dix (10) réseaux ordinaires (RO), quarante deux (42) points de vente agréés (PVA), cent cinq (105) stations lavage graissage (SLG),quarante six (46) revendeurs pneumatiques et quatre (04) revendeurs pièces de rechange .

En général le district analyse sa part dans le marché, étudie les propositions et met en œuvre toute action susceptible de renforcer sa position par rapport à ses concurrents.

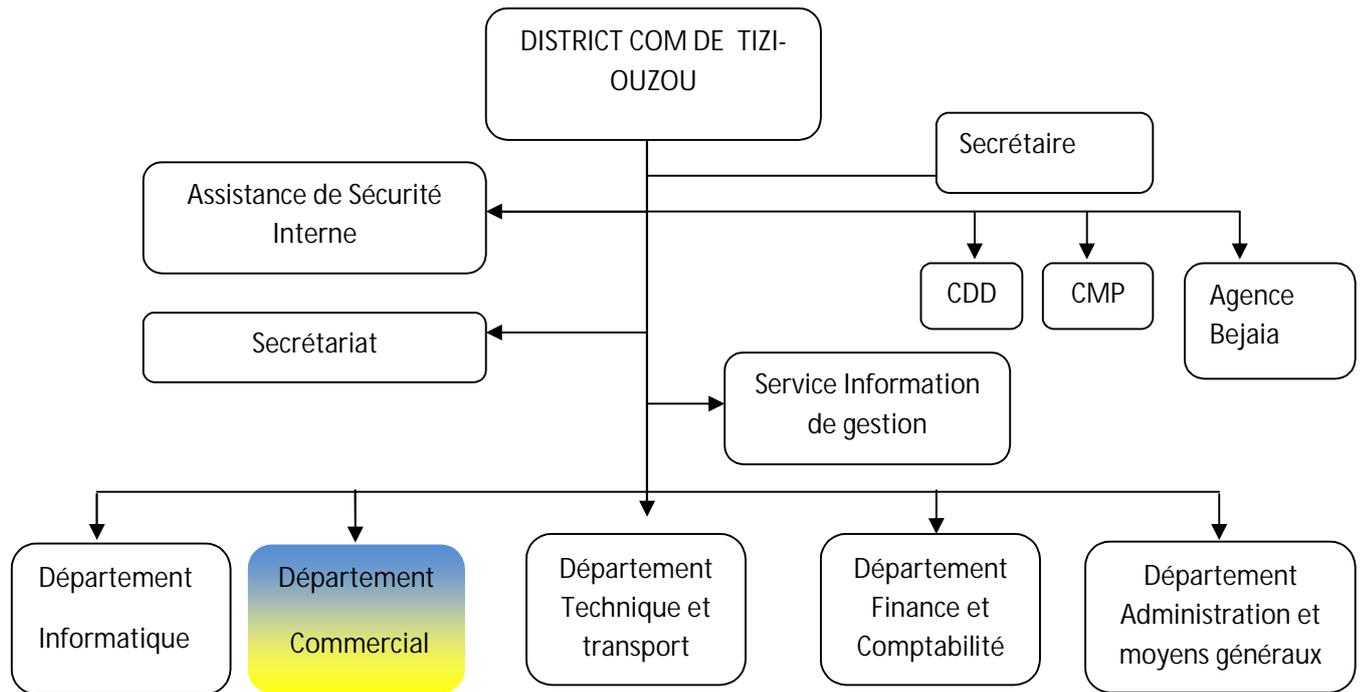
III.2.1 Missions et attributions du district commercial de Tizi-Ouzou

Le district commercial de Tizi-Ouzou est chargé des activités liées au transport, stockage, distribution, commercialisation, promotion et développement des produits pétroliers de NAFTAL. Ses principales missions sont :

- Gérer, organiser, promouvoir et développer l'activité de distribution des carburants lubrifiants, pneumatiques et autres produits spéciaux.
- Tenir la comptabilité générale du district.
- Élaborer le bilan consolidé au district.
- Veiller à l'application des politiques : règles et procédures du district dans les domaines de la gestion, de la maintenance du développement et de la sécurité des ressources humaines et matérielles.
- Veiller à la tenue rigoureuse de la comptabilité des flux physiques et financiers et aux suivis et recouvrement des créances détenues sur les tiers.

III.2.2 Organisation du district commercial de Tizi-Ouzou

La figure suivante illustre l'organigramme du district commercial de Tizi-Ouzou,



Légende :



: Champs d'étude.

Figure III- 2. Organigramme du district commercial de Tizi-Ouzou .

La gestion du district commercial de Tizi-Ouzou demande une organisation, pour cela le district s'est structuré et a réparti ses tâches en départements.

Mais avant d'entamer la présentation de ces départements il y a eu lieu de citer les cellules liées directement à la direction, d'une part :

- a. service information de gestion ;**
- b. Secrétariat ;**
- c. Assistant sécurité interne ;**
- d. Agence commerciale de Bejaia**

L'agence commerciale de Bejaia est un système miniaturisé du district Commercial Tizi-Ouzou, elle dirige coordonne et contrôle des activités, moyens humains et matériels de l'agence commerciale de Bejaia. Elle est composée d'un centre de distribution dépôt 06 et 07, d'un magasin de vente LUB/PNM et d'un réseau de huit (08) stations-service gestion directe (GD) situées à Bejaïa.

e. Centre Multi-Produit (CMP)

Le centre multi-produit est un lieu de stockage et de vente d'une gamme variée de produits à savoir : les lubrifiants, les pneus, Les acides et l'eau distillée. Il se situe sur la route nationale N° 12 à 500 m du centre ville de Tizi-Ouzou, pour l'alimentation de son stock, il s'approvisionne de son fournisseur principal qui est la raffinerie d'Arzew.

f. Centre de stockage et de distribution (CDD) de Tizi-Ouzou

Alger est considérée comme le centre partenaire d'où NAFTAL de Tizi-Ouzou transporte les produits pétroliers au centre secondaire (CDD) (Centre De stockage et de Distribution) à Oued-Aissi. Puis ce dernier effectue l'activité de distribution vers les clients réseaux et autres. Le district de Tizi-Ouzou utilise les moyens de transport classique, il met en œuvre des livraisons par camions (transport régulier) qui représentent le seul moyen disponible actuellement.

g. Département transport et technique

Ce département assure la gestion administrative du matériel roulant, des pièces de rechange et du personnel de conduite. Mais aussi, il élabore le plan de maintenance préventive et curative des équipements.

h. Département finance et comptabilité

Il se consacre principalement à l'interprétation des flux financiers en écriture comptable, puis il les traduit en bilan en fin de chaque période comptable. Il coordonne toutes les activités de comptabilité, trésorerie, budget et patrimoine.

i. Département administration et moyens généraux

Ce département assure la gestion des moyens généraux, l'administration et la gestion des ressources humaines du district.

j. Département informatique

Le département informatique se répartit en deux (02) services :

- Service ING : Information de gestion qui assure le suivi des activités de la station, il veille à la récupération, consolidation et traitement de données.
- Service systèmes et réseaux : Il veille au bon fonctionnement du réseau intranet, s'occupe de différentes installations et mises à jour des systèmes et des logiciels et corrige les pannes matérielles.

k. Département commercial

Le département commercial veille à l'exécution et l'application de la politique commerciale adoptée par les directions produites. Mais aussi, elle élabore le plan budget prévisionnel des ventes des produits, puis analyse des écarts et entreprend des actions de redressement de l'entreprise. Le département commercial se compose de :

- **Cellule recouvrement**

C'est une cellule rattachée directement au département. Elle a pour fonction principale :

- Suivi et contrôle du règlement des clients à terme.
- Recouvrement des créances sur clients.
- Etablissement des reçus d'encaissement et saisie éventuelle des clients retardataires.

Elle est donc en contact permanent avec les autres services.

- **Service carburants** : chargé de,

- La commercialisation des carburants.
- Traiter et analyser les données du marché potentiel pour pouvoir établir les plans prévisionnels.
- Gérer et contrôler le mouvement des stocks puis s'assurer de la qualité et de la conformité du produit.
- Enregistrer, analyser et suggérer les solutions remédiant aux doléances de la clientèle.

- **Service lubrifiants et produits spéciaux** : qui s'occupe principalement de :

- La présentation du conseil de la clientèle en matière d'utilisation des lubrifiants de NAFTAL.

- Suivi et analyse des ventes des produits commercialisés par le centre multi-produits.
- Elaboration des plans d'approvisionnement, de distribution et répond aux doléances des clients.
- Elaboration des plans de ventes.
- Etablissement des programmes et planning d'approvisionnement.
- Suivi et analyse des ventes pneumatiques.

Service réseau : se compose de :

- Un chef de service ;
- Deux cadres d'étude ;
- Trois chefs de secteur ;
- 16 Chefs de stations ;
- 30 chefs d'équipes ;
- 136 pompistes et des laveurs graisseurs.

Il est chargé de :

- Gérer les réseaux des stations-service en l'occurrence les gestions directes (GD), les gestions libres (GL), les points de vente agréés (PVA)...
- Analyser la performance de ses stations en gestion direct.
- Veiller aux respects des normes de gestion.
- Veiller à la présentation de l'image de marque de NAFTAL.

III.2.3 Organisation du département commercial

La figure ci-dessous présente l'organigramme du département commercial,

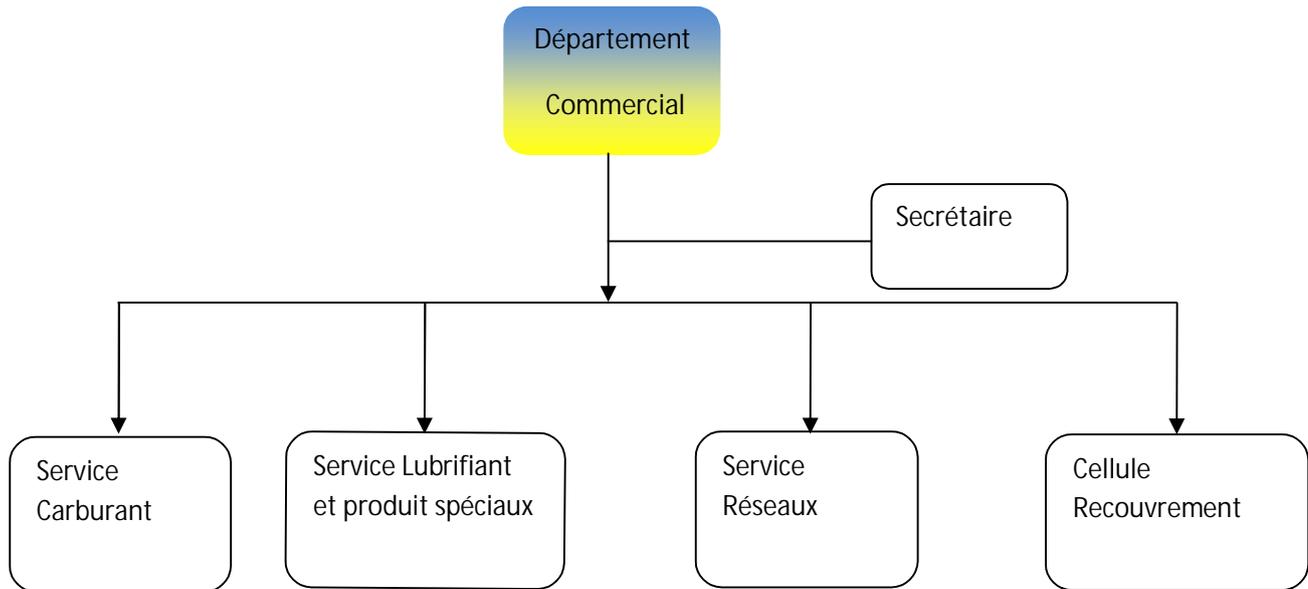


Figure III- 3 : Organigramme du Département commercial .

Conclusion

Dans ce chapitre, nous avons présenté le fonctionnement de l'entreprise et en étudiant les tâches du département commercial, le chapitre suivant est consacré à la présentation de l'analyse et la conception de la solution dégagée dans ce présent chapitre.

IV Introduction :

L'étude de l'existant nous a permis de collecter les informations qui concerne le fonctionnement du système existant, et ainsi de connaître les besoins et les orientations des utilisateurs.

Pour le développement de notre application de gestion des écarts des stations de service cas: NAFTAL nous avons opté pour une démarche de conception orientée objet (RUP, Rational Unified Process), en se basant sur la modélisation UML. Pour cela, nous suivons les phases suivantes :

- **Initialisation** : vision approximative de la finalité du projet, cas d'entreprise, périmètre, estimation globales.
- **Elaboration** : vision plus élaborée, implémentation itérative de l'architecture noyau, résolution des risques élevés, identification de la plupart des besoins et du périmètre réel, estimation plus réalistes.
- **Construction** : implémentation itérative des éléments qui présentent des risques et une complexité moindres et préparation du déploiement.
- **Transition** : déploiement.

IV.1 Phase d'Initialisation : une analyse approximative de notre projet permet de desseller les points suivants :

- Notre application touche à la gestion des écarts des stations de service cas: NAFTAL qui est un domaine connu ;
- Notre application est de type Client/serveur ;
- Notre application est isolée à internet, le risque de sécurité ne présente pas un énorme problème.

Vu les points précédent de notre analyse de surface, nous pouvons dire que ce projet est abordable.

IV.2 Phase d'Elaboration : Notre projet porte sur la mise en place d'une application client/serveur 2-tiers, pour l'automatisation des activités de la gestion des écarts des stations de service cas: NAFTAL

Dans le but de maîtriser le suivi des mouvements des documents et dossiers entre les différents services du champ d'étude, l'application doit offrir un environnement interactif et

convivial aux différents acteurs de tous les services de la direction district NAFTAL et les différentes stations de service leur permettant de travailler et de communiquer mutuellement, dans l'objectif de la gestion, et ce via internet.

IV.2.1 Identification des acteurs : dans le stage que nous avons fait nous a permis de procéder à l'identification des principaux acteurs qui seront les futurs utilisateurs de notre application.

- **Administrateur (ADMIN):** c'est le chef de service réseaux, il peut modifier, ajouter et supprimer un produit ; consulter les ventes et les écarts des carburants des stations ainsi que les commentaires laissés par les différents chefs de station. Confirme ou infirme des préinscriptions, bloque des inscriptions.
- **Chef de station(CDS) :** son rôle est d'ajouter des ventes et les écarts des carburants ainsi que laisser des commentaires sur ces derniers.
- **Chef de département commercial(CDC) :** visite le site à fin de consulter les valeurs des écarts des carburants, les commentaires laissés par les chefs de station et celui du chef de service réseaux, ainsi que les ventes.

IV.2.2 Diagramme de contexte : le diagramme de contexte est un modèle conceptuel de flux qui offre une vision globale des interactions entre le système et l'environnement extérieur. Il permet aussi de bien délimiter le champ de l'étude. Pour notre cas, le diagramme de contexte est donné par la figure suivante :

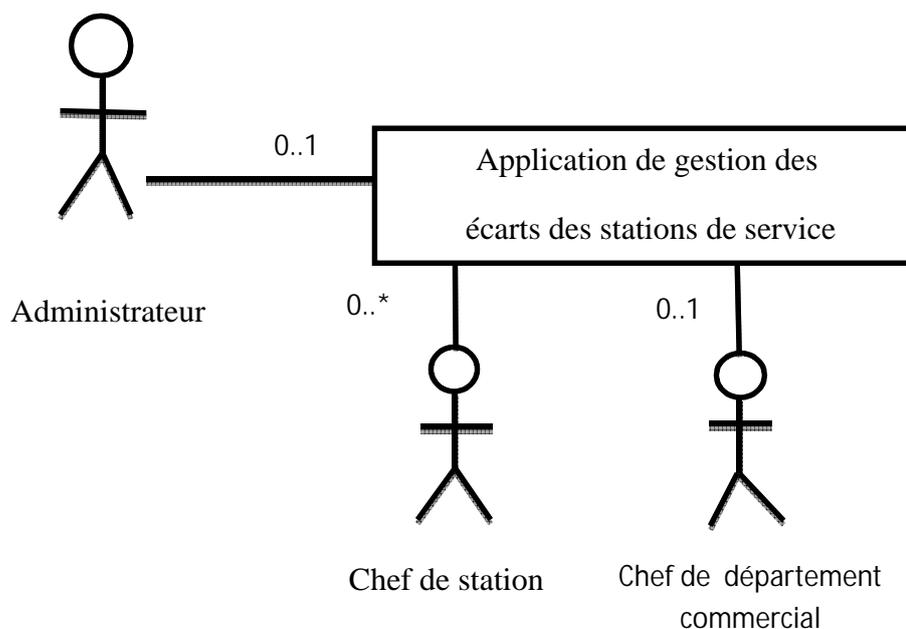


Figure IV.1: diagramme de contexte

IV.2.3 Représentation des diagrammes de cas d'utilisation : les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation (use cases), les comportements sont modélisés sous la forme de cas d'utilisation via un diagramme de cas d'utilisation.

• Représentation les cas d'utilisation globale :

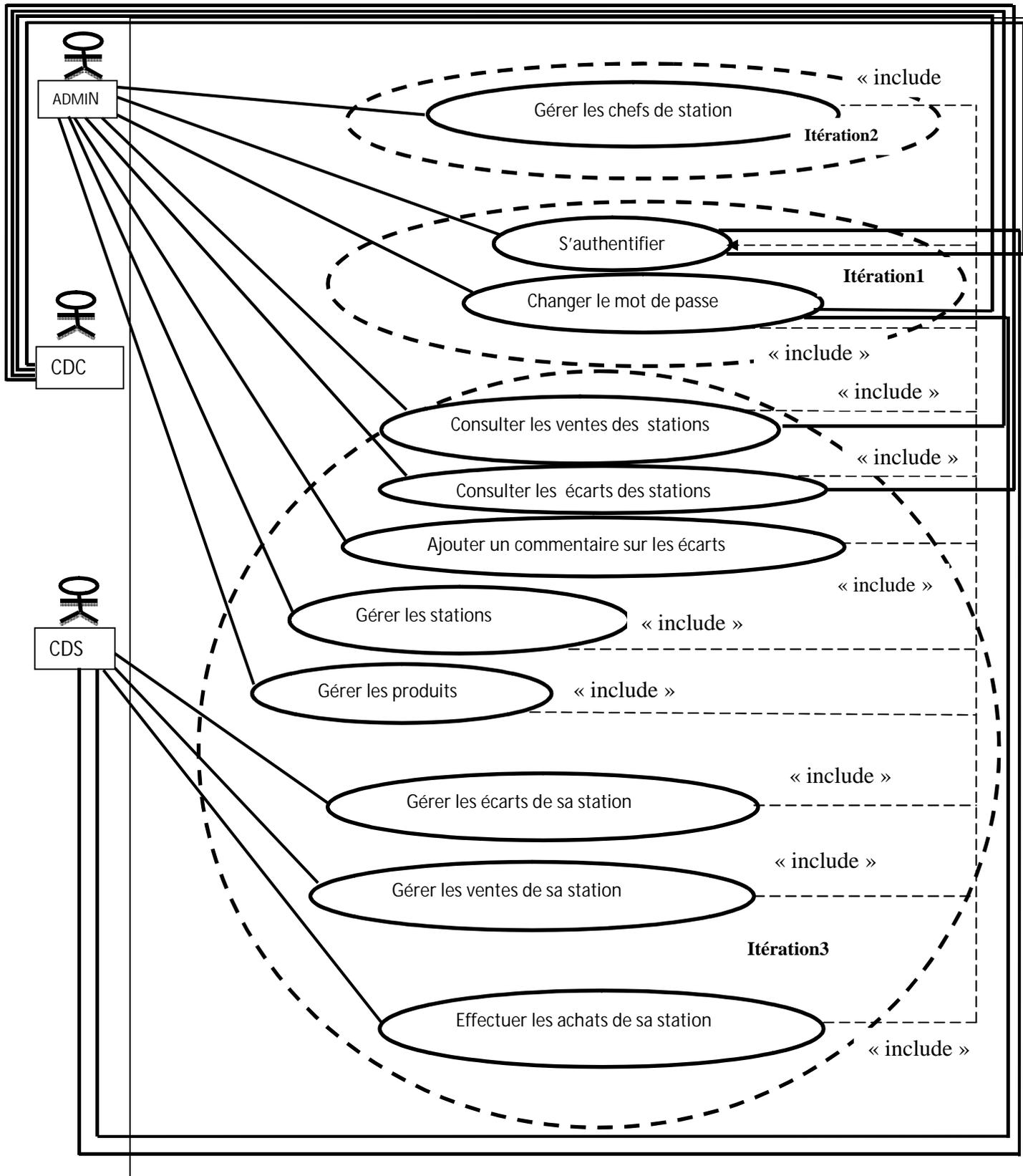


Figure IV.2 : Diagramme de cas d'utilisation

IV.2.4 Définition des itérations : le critère de définition des itérations se base sur les deux points suivants :

1. Ressemblance entre cas d'utilisation.
2. Dépendance entre cas d'utilisation.

IV.3 Construction : pour chaque itération réalisée précédemment, nous allons suivre un cycle de vie en cascade pour aboutir à un diagramme de classe en passant par un diagramme de séquence ou de collaboration.

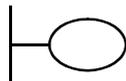
IV.3.1 Réalisation de l'itération 1

IV.3.1.1 Diagrammes de séquence pour itération 1: les diagrammes de séquence présentent la coopération entre différents objets. Les objets sont définis et leur coopération est représentée par une séquence de messages entre eux.

Après avoir déterminé nos besoins et réaliser les diagrammes des cas d'utilisations nous allons élaborer quelques diagrammes de séquences de l'itération 1. Les classes d'objets utilisées dans la représentation du diagramme de séquence d'analyse peuvent être réparties dans les trois catégories selon les stéréotypes suivant:

- **Les objets de type interface :** l'interface entre l'acteur et le système, par exemples application web, des pages web complètes.

L'icône utilisée :



- **Les objets de type entité :** sont des objets décrits dans un cas d'utilisation, qui peuvent apparaître dans de nombreux cas d'utilisation, et généralement sa durée de vie dépasse celle de toute Interaction où il participe.

L'icône utilisée :



- **Les objets de type contrôle :** représentent les processus, c'est-à-dire les activités système, ces objets dirigent les activités des objets entité et d'interface.

L'icône utilisée:



• Diagramme de séquence du cas d'utilisation : « Authentification »

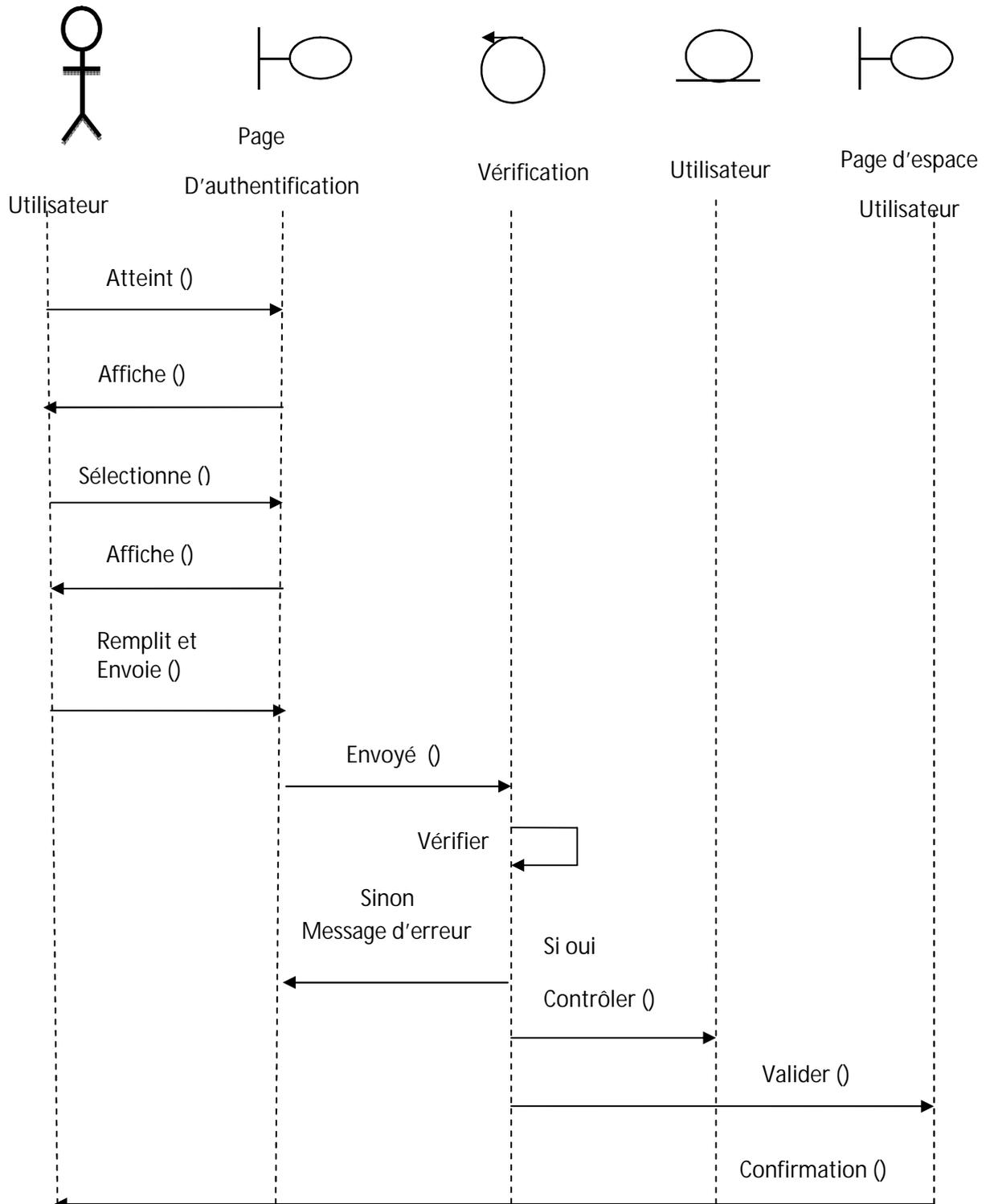


Figure IV.3. Diagramme de séquence du cas d'utilisation : « Authentification »

- **Description du cas d'utilisation avec des scénarios:**

Cas utilisation : Authentification

Acteur : UTILISATEUR

Résumé : Ce cas d'utilisation permet à l'utilisateur (UTILISATEUR) de s'authentifier et d'avoir un espace personnel.

Scénario :

- 1- L'UTILISATEUR atteint son espace.
- 2- Le système affiche « espace UTILISATEUR ».
- 3-L'UTILISATEUR sélectionne la rubrique «authentification».
- 4- Le système retourne le formulaire «authentification ».
- 5- L'UTILISATEUR saisit et soumet les données (nom, prénom, mot de passe, État utilisateur).
- 6- Le système retourne une page de confirmation .

- Diagramme de séquence du cas d'utilisation : « Modification du mot de passe »

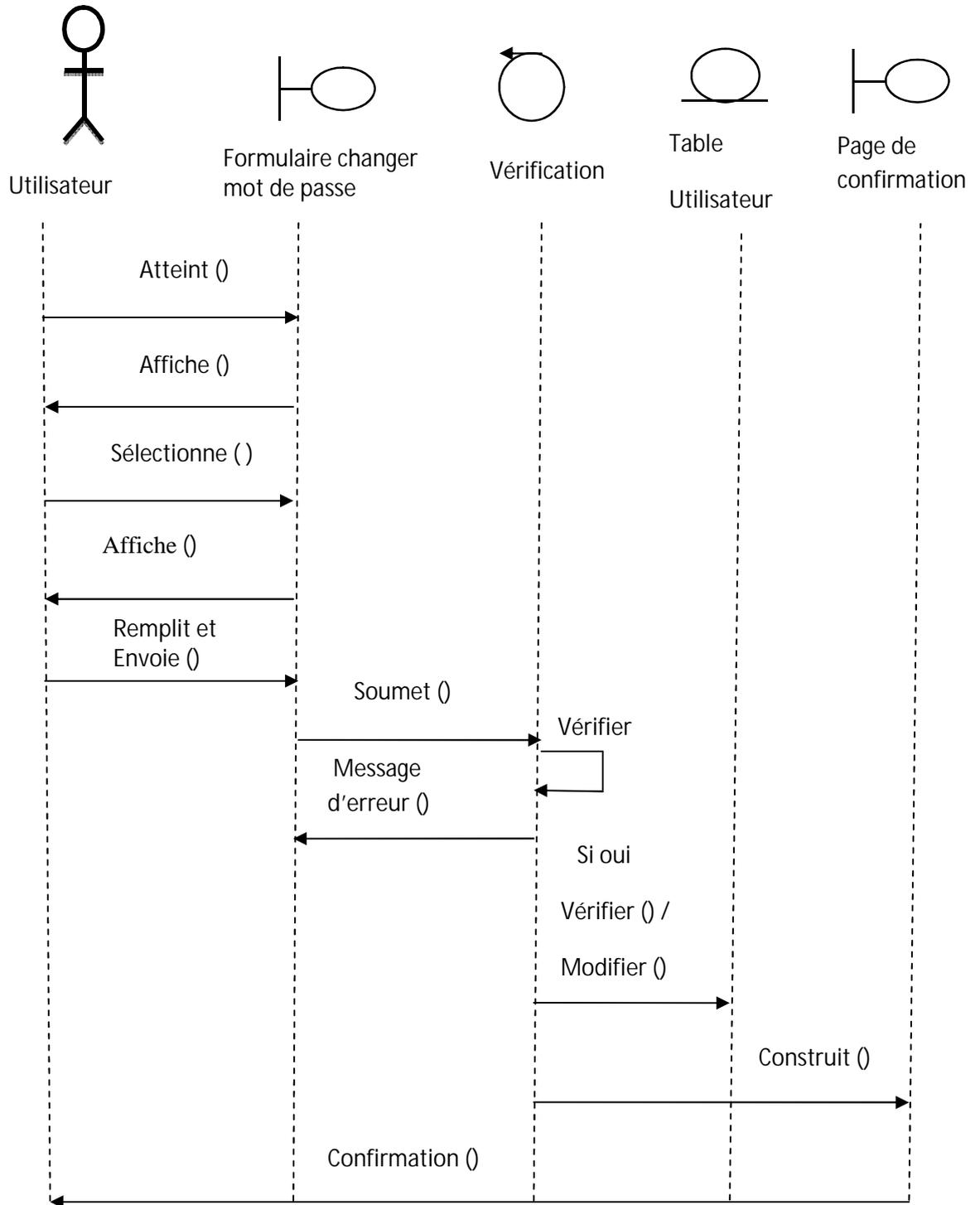


Figure IV.4. Diagramme de séquence du cas d'utilisation : «Modification du mot de passe»

- **Description du cas d'utilisation avec des scénarios:**

Cas utilisation : Authentification

Acteur : UTILISATEUR

Résumé : Ce cas d'utilisation permet à l'utilisateur (UTILISATEUR) modifier son mot de passe

Scénario :

1-L'UTILISATEUR atteint son espace.

2-Le système affiche « espace UTILISATEUR ».

3- L'UTILISATEUR sélectionne la rubrique « changer le mot de passe».

4-Le système retourne le formulaire «changer le mot de passe».

5- L'UTILISATEUR saisit et soumet les données.

6-Le système retourne une page de confirmation

IV.3.1.2 Diagrammes de classe pour l'itération 1

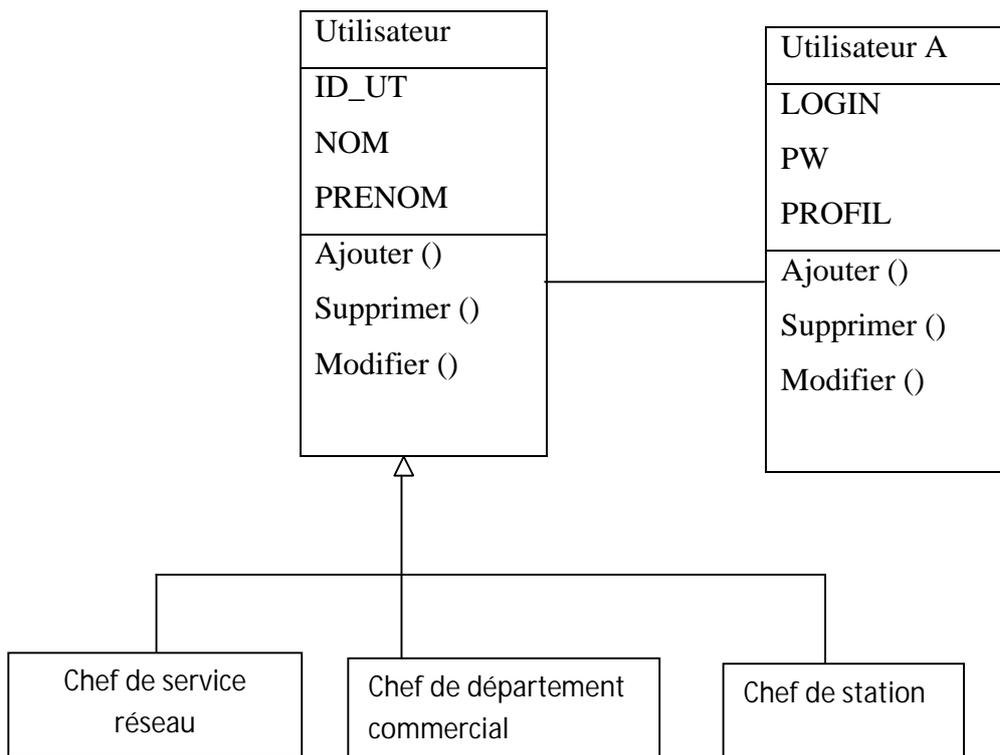


Figure IV.5 : Diagramme de classes pour l'itération 1

IV.3.2 Réalisation de l'itération 2

IV.3.2.1 Diagrammes de séquence pour itération 2: Un diagramme de séquence de l'itération 2.

• Diagramme de séquence du cas d'utilisation : «Ajouter un chef de station»

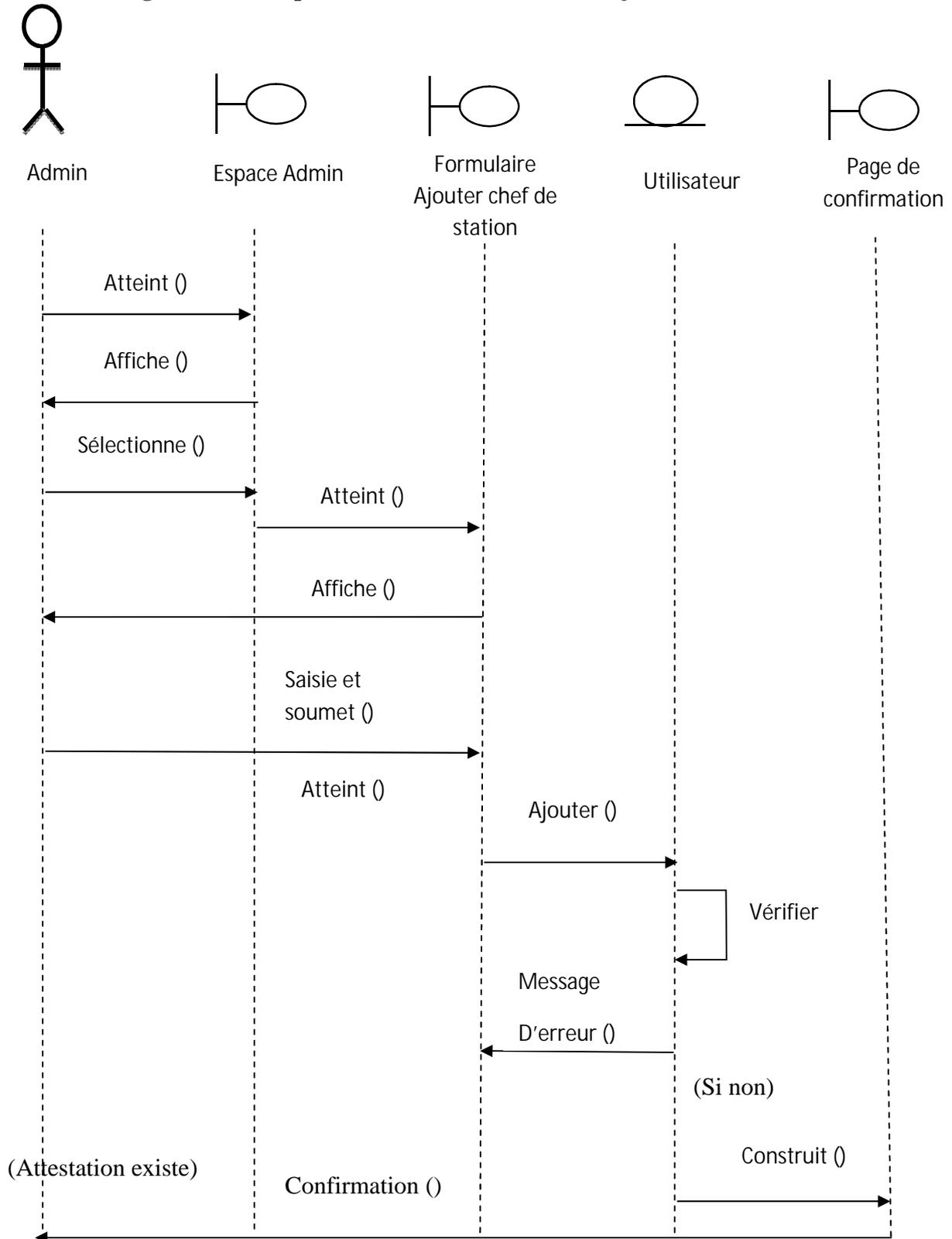


Figure IV.6 :Diagramme de séquence du cas d'utilisation : «Ajouter un chef de station »

Cas utilisation : Ajouter un chef de station

Acteur : ADMIN

Résumé : Ce cas d'utilisation permet à l'administrateur(ADMIN) D'ajouter un employeur

Scénario :

- 1- L' ADMIN atteint son espace.
- 2- Le système affiche « ADMIN ».
- 3- L'utilisateur sélectionne la rubrique « Ajouter chef de station»
- 4- Le système retourne le formulaire « Ajouter chef de station».
- 5- L' ADMIN remplit le formulaire et clique sur le bouton « valider ».
- 6- Le système ajoute un nouvel employeur après avoir vérifié la validité Des données saisies.

IV.3.2.2 Diagrammes de classe pour l'itération2

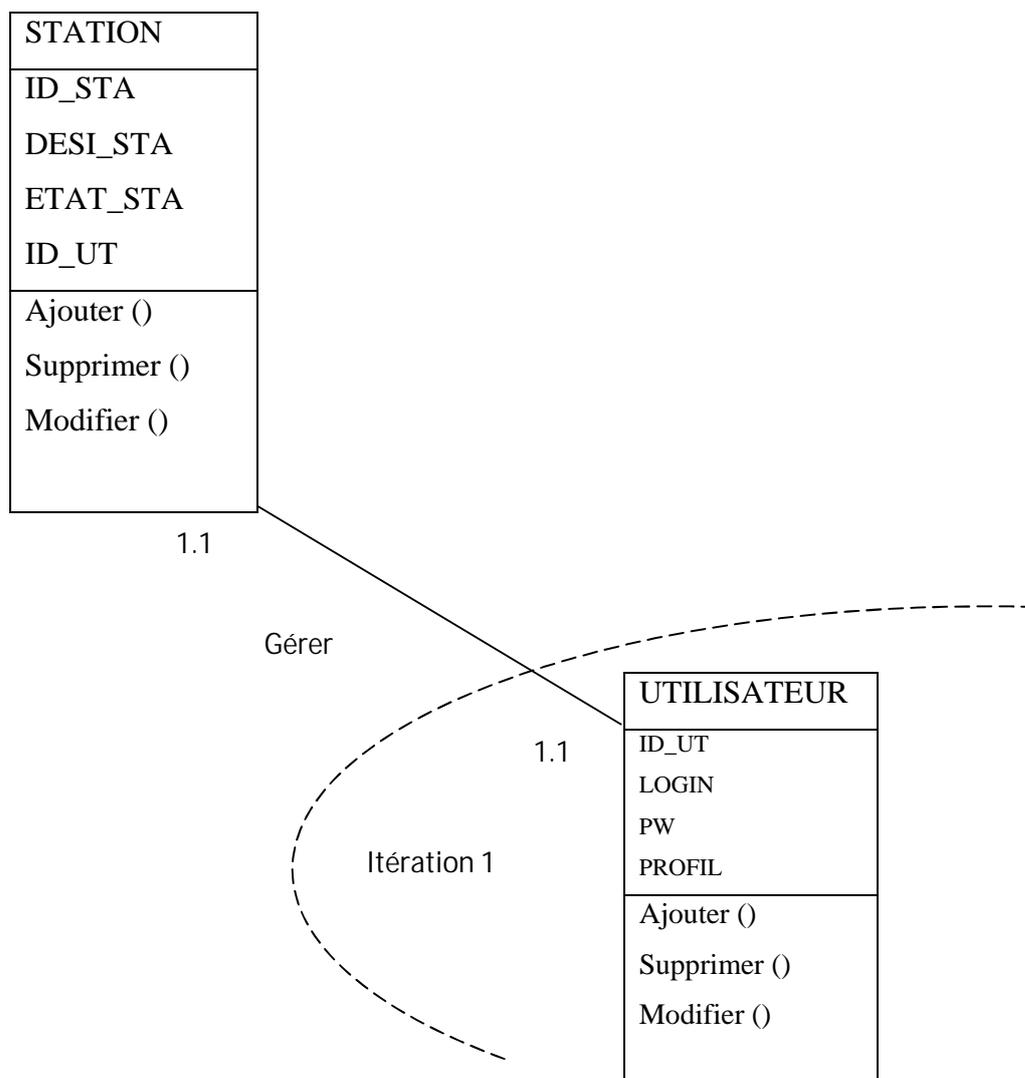


Figure IV.7 : Diagramme de classes pour l'itération2

IV.3.3 Réalisation de l'itération 3 :

IV.3.3.1 Diagrammes de séquence pour itération 3: des diagrammes de séquence de l'itération.

- Diagramme de séquence du cas d'utilisation : «Saisie des valeurs des écarts»

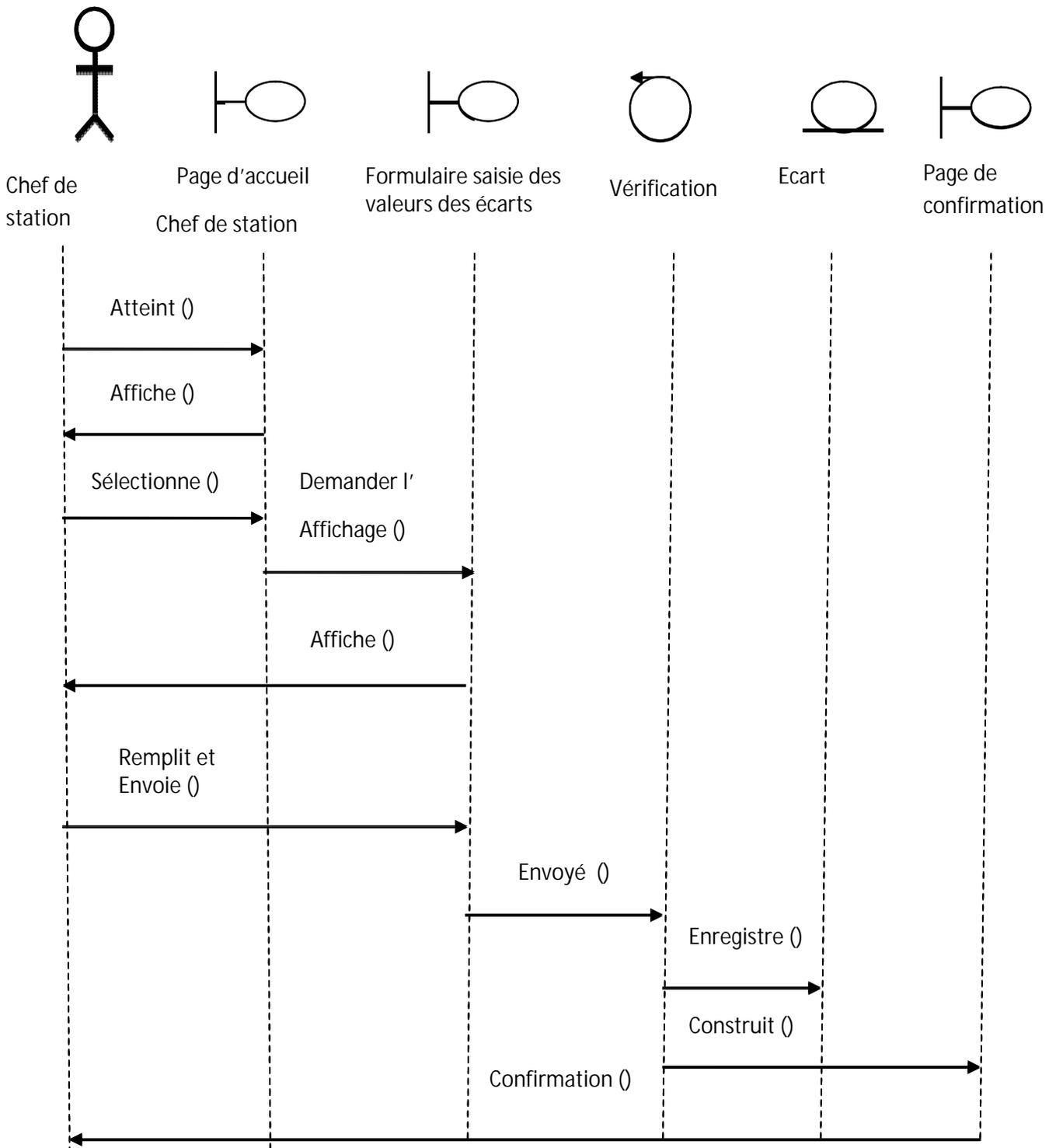


Figure IV.8 :Diagramme de séquence du cas d'utilisation : «Saisie des valeurs des écarts »

- **Description du cas d'utilisation avec des scénarios:**

Cas utilisation : saisie des valeurs des écarts

Acteur : CHEF DE STATION

Résumé : Ce cas d'utilisation permet au chef de station de saisir des valeurs des écarts

Scénario :

- 1- Le CHEF DE STATION atteint son espace.
- 2- Le système affiche « espace CHEF DE STATION ».
- 3- Le CHEF DE STATION sélectionne la rubrique «saisie des valeurs des écarts ».
- 4- Le système retourne le formulaire « saisie des valeurs des écarts».
- 5- Le CHEF DE STATION saisit et soumit les données.
- 6- Le système retourne une page de confirmation.

• Diagramme de séquence du cas d'utilisation : «Supprimer une vente»

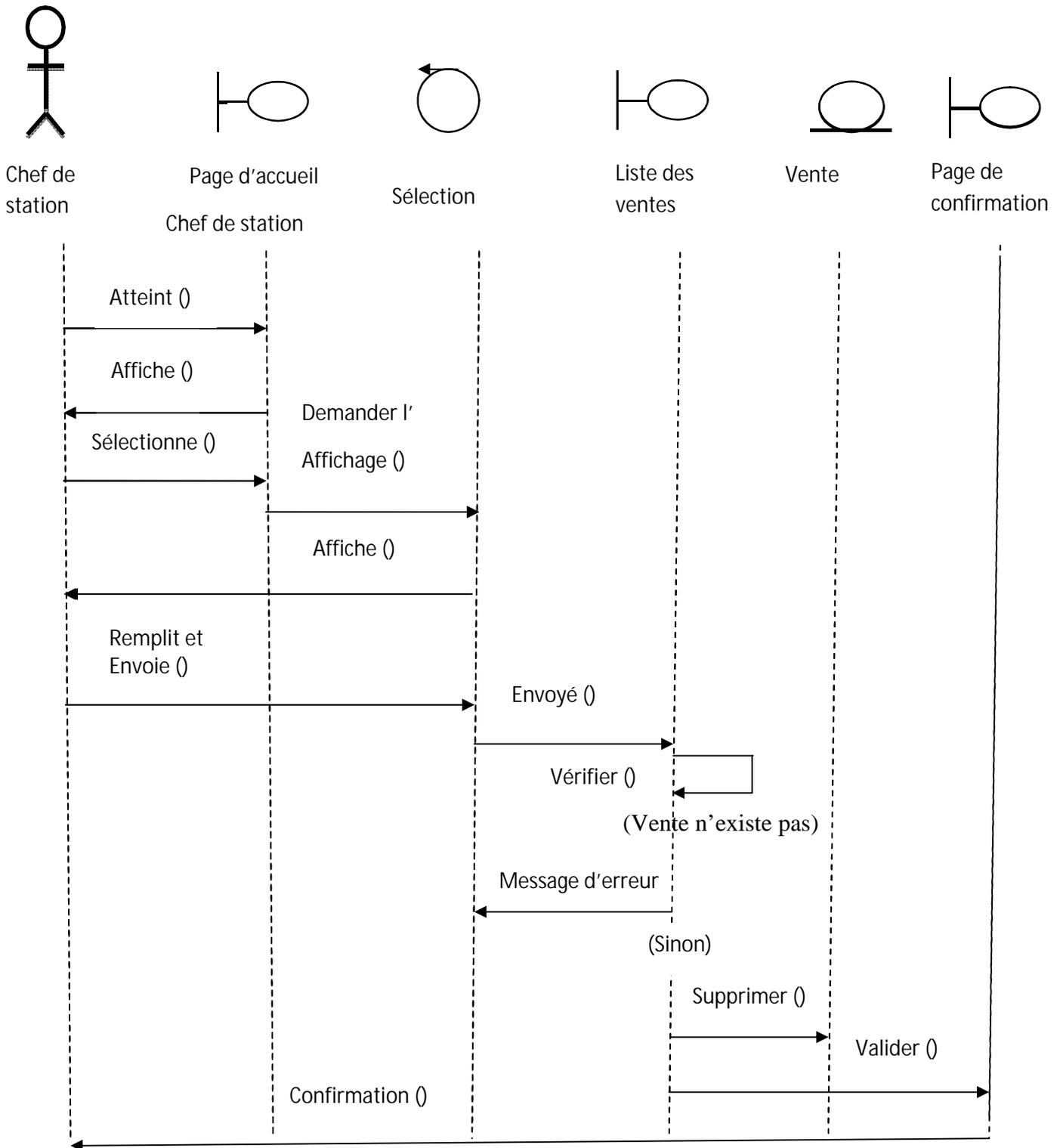


Figure IV.9 :Diagramme de séquence du cas d'utilisation : «Supprimer une vente »

- **Description du cas d'utilisation avec des scénarios:**

Cas utilisation: Supprimer une vente

Acteur : CHEF DE STATION.

Résumé : Ce cas d'utilisation permet au CHEF DE STATION de Supprimer une vente

Scénario :

- 1- Le CHEF DE STATION atteint son espace.
- 2- Le système affiche « espace CHEF DE STATION.».
- 3- L'utilisateur sélectionne la rubrique «Supprimer une vente».
- 4- Le système retourne le formulaire «Supprimer une vente».
- 5- Le CHEF DE STATION remplit le formulaire et clique sur le bouton « valider ».
- 6- Le système retourne une page de confirmation.

IV.3.3.2 Diagrammes de classe pour l'itération 3

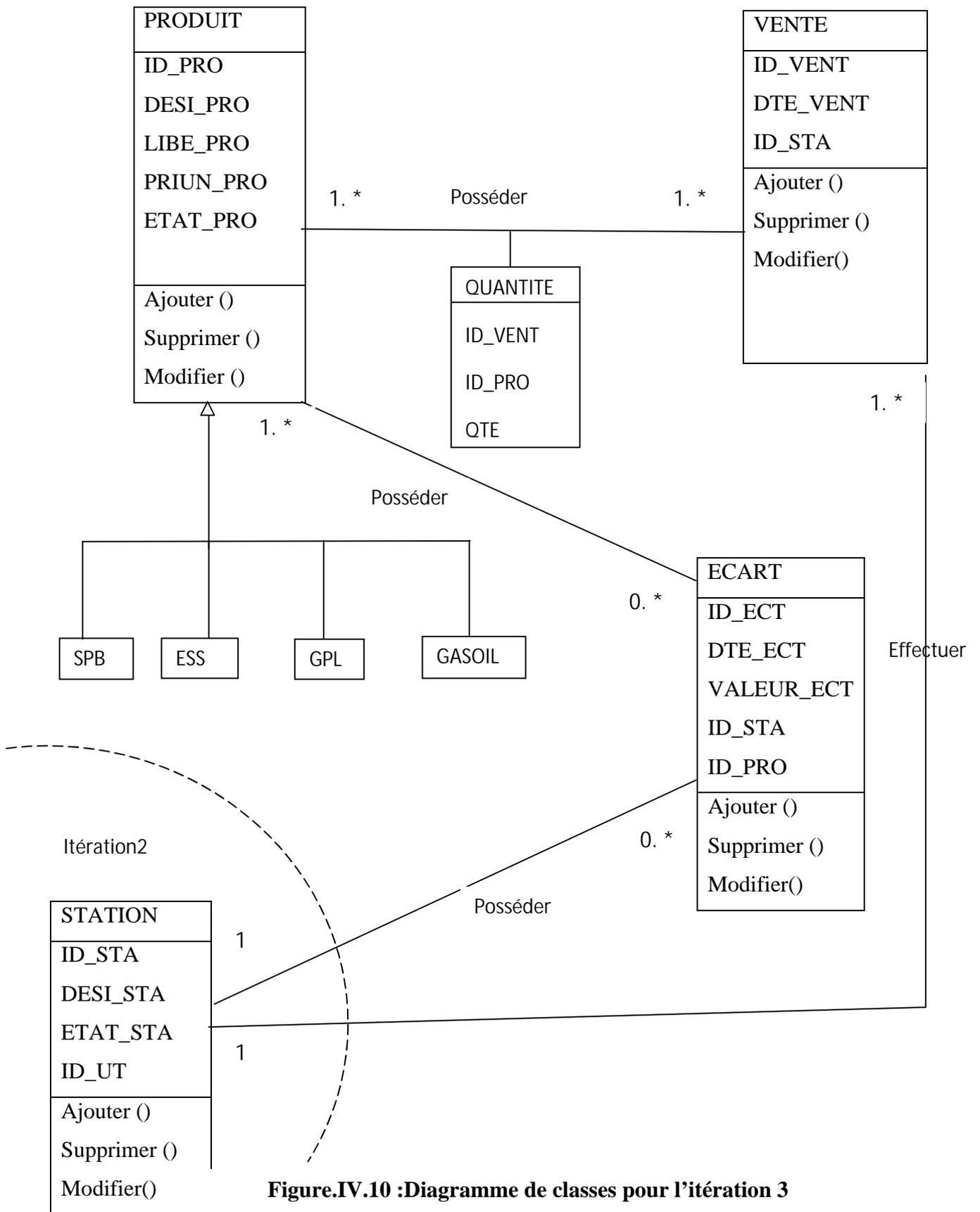


Figure.IV.10 :Diagramme de classes pour l'itération 3

-La classes STATION relié avec la classe UTILISATEUR de l'iteration1 par les cardinalité (1.....1).

-La classe STATION relié avec la classe ECART de l'iteration3 par les cardinalité (1..... 0.*).

-La classe STATION relié avec la classe VENTE de l'iteration3 par les cardinalité (1.....0.*).

Conclusion :

Dans ce chapitre nous avons utilisé le processus unifié RUP pour l'analyse et la conception de notre application en utilisant le langage de modélisation UML. Dans le chapitre suivant, qui touche à la réalisation, nous essayerons de construire notre application par itération.

V Introduction

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, nous allons présenter dans ce dernier chapitre l'environnement de développement, les outils qui ont servi à la réalisation de notre application, pour illustrer quelques fonctionnalités de l'application.

V.1 Les outils de développement :

❖ **Hibernate Framework :**

C'est un logiciel, écrit en java, qui permet de faire le mapping entre Objets Java et objets stockés en base relationnelle en assurant la persistance, et aussi s'occupe du transfert des classes Java dans les tables de la BDD et des types de données dans les types de données SQL.

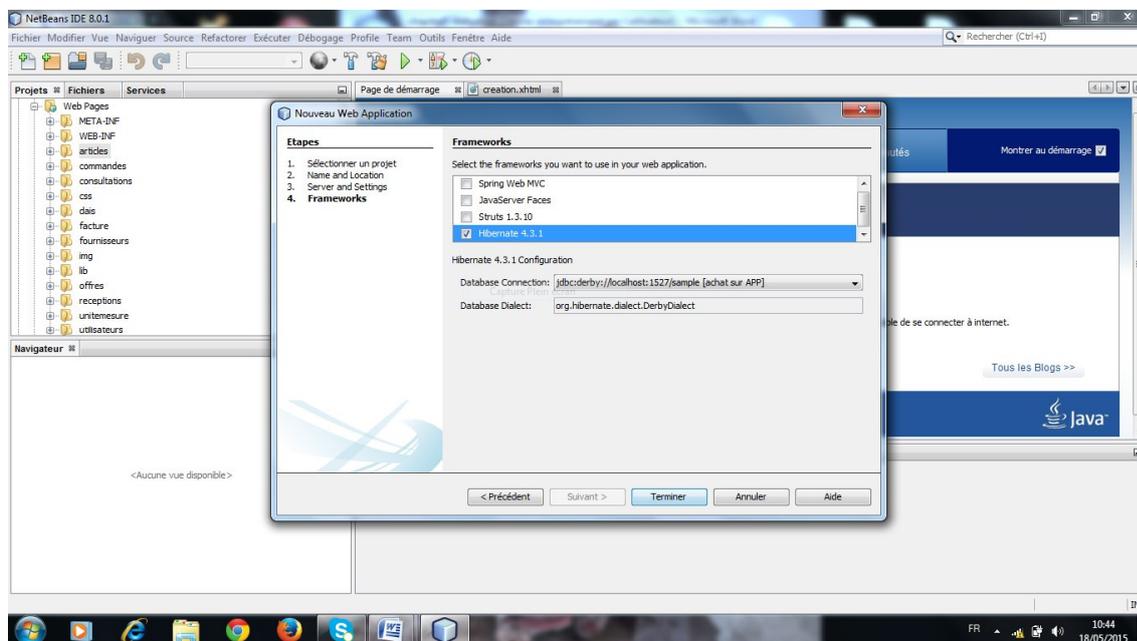


Figure V.1 : L'interface de Hibernate.

❖ **EasyPHP :**

EasyPHP installe et configure automatiquement un environnement de travail complet qui permet de mettre en œuvre toute la puissance et la souplesse qu'offrent le langage dynamique PHP et son support efficace des bases de données MySQL. EasyPHP regroupe un serveur Apache, un serveur de base de données MySQL, Le langage PHP ainsi que des outils facilitant le développement de sites dynamiques ou d'applications.

La figure suivante montre une copie d'écran de la page d'accueil d'EasyPHP

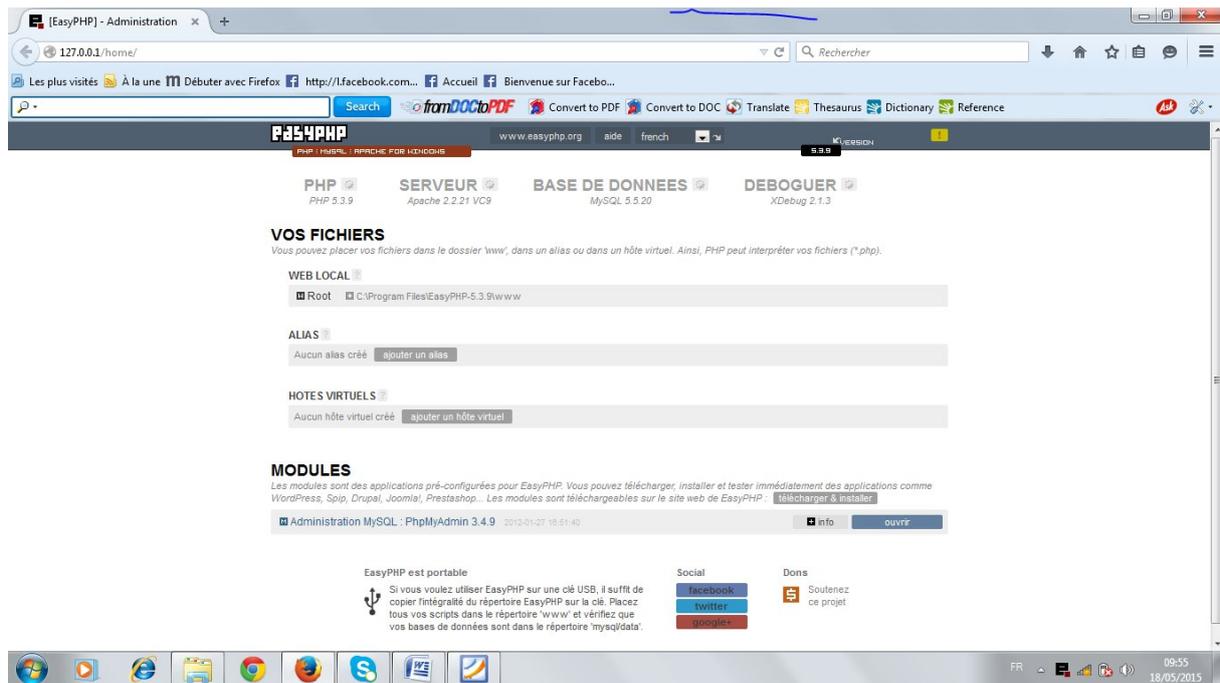


Figure V.2 : L'interface EasyPHP .

❖ PhpMyAdmin :

PhpMyAdmin est une application web permettant de gérer un serveur de bases de données MySQL. Cette interface est écrite en PHP et s'appuie sur le serveur Apache.

La figure suivante montre une copie d'écran de la page d'accueil de PhpMyAdmin.

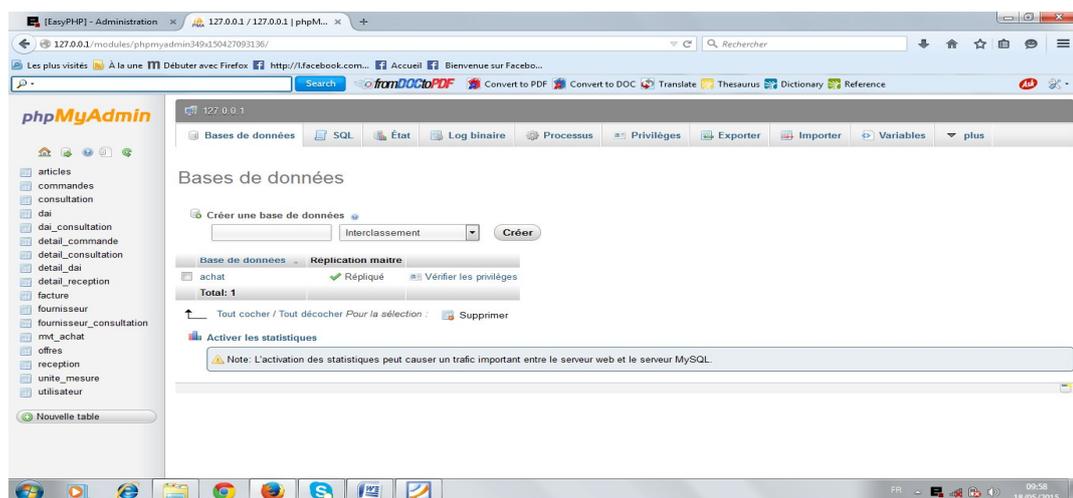


Figure V.3: L'interface PhpMyAdmin.

❖ Serveur Apache Tomcat:

Apache Tomcat est une implémentation open source d'un conteneur web qui permet donc d'exécuter des applications web reposant sur les technologies servlets et JSP.

Tomcat est diffusé en open source sous une licence Apache. C'est aussi l'implémentation de référence des spécifications servlets jusqu'à la version 2.4 et JSP jusqu'à la version 2.0 implémentées dans les différentes versions de Tomcat.

❖ Netbeans

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme.

L'IDE Netbeans s'enrichit à l'aide de plugins.

Il comprend toutes les caractéristiques d'un IDE moderne (coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web, etc).

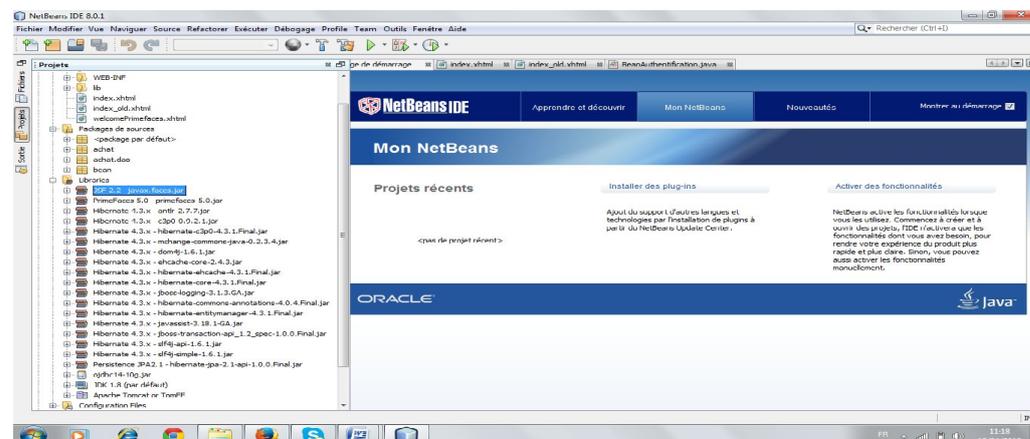


Figure V.4: L'interface de Netbeans .

V.2 Les langages de programmation :

❖ Java EE

Java Enterprise Edition, ou **Java EE** (anciennement **J2EE**), est une spécification pour la technique Java de Sun plus particulièrement destinée aux applications d'entreprise. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au *Framework* Java standard (JSE, Java Standard Edition) afin de faciliter la création d'applications réparties.

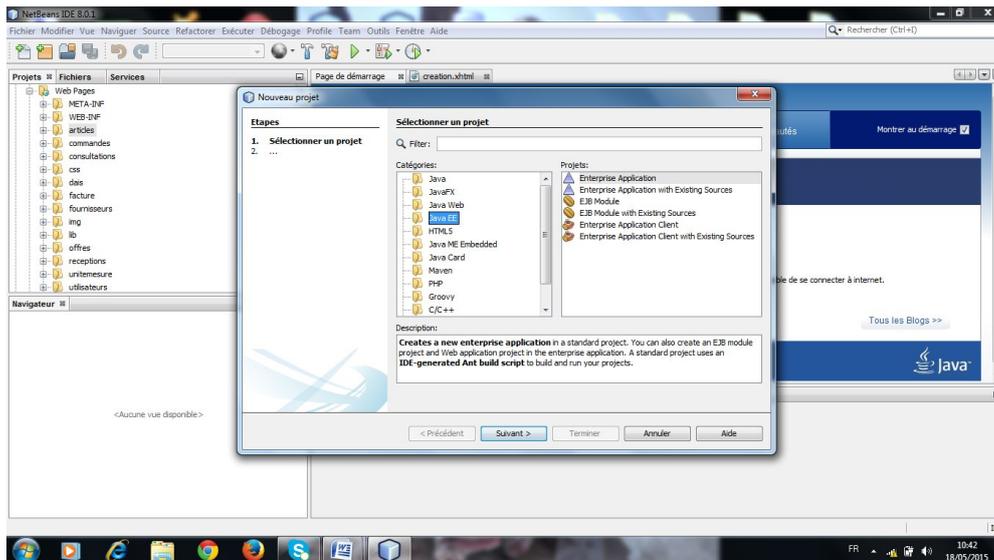


Figure V.5: L'interface de JEE.

❖ XHTML :(Extensible Hyper text Markup Language)

XHTML est un langage balisé servant à l'écriture de pages du World Wide Web. XHTML est le successeur de HTML (de l'anglais HyperText Markup Language), XHTML respectant la syntaxe définie par XML, plus récente et plus simple que la syntaxe définie par SGML respectée par HTML.

V.3 les Itérations :

V.3.1 Itération I :

- **Son modèle relationnel :**

Utilisateur (ID_UT, LOGIN, PW, PROFIL)

- **Table Utilisateur**

#	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/>	1 <u>ID_UT</u>	decimal(19,0)			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	2 <u>LOGIN</u>	varchar(8)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	3 <u>PW</u>	varchar(8)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	4 <u>PROFIL</u>	varchar(45)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼

V.3.2 Itération II :

- **Son modèle relationnel**

Station (ID_STA, DESI_STA, ETAT_STA, ID_UT*).

Utilisateur (ID_UT, LOGIN, PW, PROFIL).

- **Table Station**

#	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/>	1 <u>ID_STA</u>	varchar(10)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	2 <u>DESI_STA</u>	varchar(25)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	3 <u>ETAT_STA</u>	varchar(25)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	4 <u>ID_UT</u>	decimal(19,0)			Non	Aucune		Modifier Supprimer plus ▼

V.3.3 Itération III:

- **Son modèle relationnel :**

Produit :(ID_PRO, DESI_PRO, PRI_PRO, ETAT_PRO)

Vente:(ID_VENT, DTE, ID_PRO*, ID_STA*, QUNTITE)

Ecart:(ID_ECT, DTE_ECT, VALEUR_ECT, ID_PRO*, ID_STA*)

Quantite :(ID_VENT* , ID_PRO* ,QTE)

Quelque table pour l'itération3

- **Table Produit :**

#	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/>	1 <u>ID_PRO</u>	int(5)			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	2 DESI_PRO	char(25)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	3 PRI_PRO	float			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	4 ETAT_PRO	varchar(5)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼

- **Table Vente :**

#	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/>	1 <u>ID_VENT</u>	varchar(20)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	2 DTE_VENT	date			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	3 ID_STA	varchar(10)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	4 ID_PRO	int(5)			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	5 QUANTITE	double			Non	Aucune		Modifier Supprimer plus ▼

- **Table Ecart:**

#	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/>	1 <u>ID_ECT</u>	varchar(8)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	2 DTE_ECT	date			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	3 ESS_ECT	float			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	4 PRI_ESS	float			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	5 GASOIL_ECT	float			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	6 PRI_GASOIL	float			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	7 SPB_ECT	float			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	8 PRI_SPB	float			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	9 GPL_ECT	float			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	10 PRI_GPL	float			Non	Aucune		Modifier Supprimer plus ▼
<input type="checkbox"/>	11 ID_STA	varchar(10)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼

V.4 Présentation de quelques interfaces :

Nous n'allons présenter que quelques interfaces :

- **Page d'accueil :** c'est la première page qui apparaît au lancement de l'application.

V.4.1 Présentation d'interface page d'accueil :



Figure V.6 :Interface page d'accueil

V.4.2 Présentation d'interface authentification :



Figure V.7 : Interface authentification

V.3.3 Présentation d'interface d'un produit:



Figure V.8 : Interface d'un produit

V.3.4 Présentation d'interface d'ajouter un produit:



Figure V.5 :Interface ajouter un produit

Conclusion :

Dans ce chapitre nous avons présenté les outils et l'environnement développement de L'application.

Nous avons en outre explicité les composants de la base de données, puis pour terminer nous avons présenté quelques interfaces de l'application.

Conclusion générale

La présente étude que nous avons réalisée au sein de l'entreprise NAFTAL avait pour mission la conception et la réalisation d'un service web pour la gestion des écarts des stations de services. Pour mener à terme notre projet, nous avons d'abord analysé le fonctionnement de l'entreprise, ainsi nous avons détecté les anomalies présentées et cernées les tâches à informatiser pour apporter des améliorations. Ensuite, nous avons mis en œuvre une conception appropriée représentant l'ensemble des solutions envisagées. Enfin, nous avons présenté la réalisation de notre application, qui contient désormais l'ensemble des fonctionnalités répondant aux besoins de l'entreprise. Le projet effectué nous a permis d'affronter des problèmes concrets, en mettant en pratique nos connaissances acquises lors de notre cursus de licence en informatique, et nous a permis en outre, de bénéficier d'une première expérience dans le monde professionnel. En guise de perspectives, nous envisageons d'enrichir notre application par d'autres fonctionnalités pour assurer une gestion plus complète, notamment de permettre aux chefs de station d'effectuer des demandes d'approvisionnement en stock des différents produits au sein du district commercial de Tizi-Ouzou.

[PR185]:Kadima Hubert, Les services web

[ING.INF.93]: Conception implimentation d'un web services pour un environnement SDDS,
2007 /2008

[ARS 09]: Mr ARHAB Samir & Mr SADOU Samir

« Conception et réalisation d'une application sous oracle, cas : Gestion du budget d'investissement chez SONATRACH, HASSI R'MELL» Mémoire d'ingénieur UMMTO, 2008/2009.

[GUP 05]: Guy Pujolle «Les réseaux», Edition 2005.

[DJADJN 2010]: Mr DJEBRANI Aziz & Mr DJEMA Nacer

« Conception et réalisation d'une application client/serveur sous ORACLE
Cas : Gestion de stock et approvisionnement Du groupe ETRHB HADDAD

[ASG 08] : Mr ASKEUR Akli & GUIDDIR Mohand Akli ,

«Conception et réalisation d'une base de données distribuée, multimédia sous Oracle
9» Mémoire d'ingénieur UMMTO, 2007/2008.

[GEO 00]: George et Olivier Gardain, «client / serveur», Edition 2000.

[ORD 99] : Orfali Robert,Dan HARKEY et jery EDWARDS ,

Traduction en français LEROY et jean-piere GOUT ,
« client/serveur guide de service »

[GEO 00]: George et Olivier Gardain; «client / serveur»;

[BNOK] MR BELKEBIR NAZIM &OUKOULOU KHELIFA

« Conception et réalisation d'une application client/serveur sous oracle »

Cas : Gestion des approvisionnements de l'électro-industrie

Mémoire d'ingénieur UMMTO 2008/2009

[GUY PUJOLLE] :« Les réseaux informatique », Edition 2005

[CHCK] Chikhi hassene &Chaba kamel

« Conception et réalisation d'une application client/serveur sous oracle »

Et visuel stdio 2008 pour le calcule de la paie Cas : ENIEM

[MAG 98] : MaranGraphics ; «Apprendre les réseaux» , Edition 1998.

[CCM 09]: <http://www.commentcamarche.net> , année 2009.

[USER] : www.user.skynet.be/ybet

[WIKIPÉDIA] : <http://www.wikipedia.com>

[CCM] : <http://www.commentcamarche.net>

1. Introduction :

Dans ce chapitre, nous allons entamer le processus de développement par une analyse qui mettra en évidence les différents acteurs intervenant dans le système cible ainsi que leurs besoins. La phase conception, s'appuyant sur les résultats de la phase analyse donnera la modélisation des objectifs à atteindre. Pour ce faire, notre démarche va s'appuyer sur le langage UML, qui permet la représentation de la dynamique et la statique du système à travers les différents diagrammes qu'il offre.

2. Définition UML :

UML est le facteur de standardisation, car il est impossible de prétendre imposer une méthode, une manière de faire, à toute l'industrie. En revanche on peut prétendre. Définir un langage de modélisation qui, s'il est suffisamment général, sera adopté comme moyen de communication. UML est un langage qui permet de représenter des modèles, mais il ne définit pas le processus d'élaboration d'un modèle. Cependant, dans le cadre de la modélisation d'une application informatique, les acteurs d'UML préconise d'utiliser une démarche :

- Itérative et incrémentale.
- Guidée par les besoins des utilisateurs de système.
- Centrée sur l'architecture logicielle.

3. La modélisation UML :

UML fournit une panoplie d'outils permettant de présenter l'ensemble des éléments du monde objet (classe ,objet...)ainsi que les liens qui les relie. Toutefois, étant donné qu'une seule représentation est trop subjective ,UML fournit un moyen astucieux permettant de représenter diverse projection d'une même représentation grâce aux vues .une vue est constituée d'un ou plusieurs diagrammes.

On distingue trois types de vues :

✚ Les vues statique : représentant le système physiquement .

- Diagramme de classe
- Diagramme d'objets
- Diagramme de composants
- Diagramme de déploiement

✚ Les vue Dynamique :

- Diagramme d'états
- Diagramme d'activités

- Diagramme de séquence
- Diagramme de collaboration
- ✚ Le vue fonctionnelle :
 - Diagramme de cas d'utilisation

UML suit une démarche en trois étapes :

- Analyse
- Conception
- Implémentation

Les briques d'UML : composées de :

- Les éléments
- Les relations
- Les diagrammes

4. Eléments d'UML :

Il existe quatre types d'éléments dans UML :

- les éléments structurels.
- les éléments comportementaux.
- les éléments de regroupement.
- les éléments d'annotation.

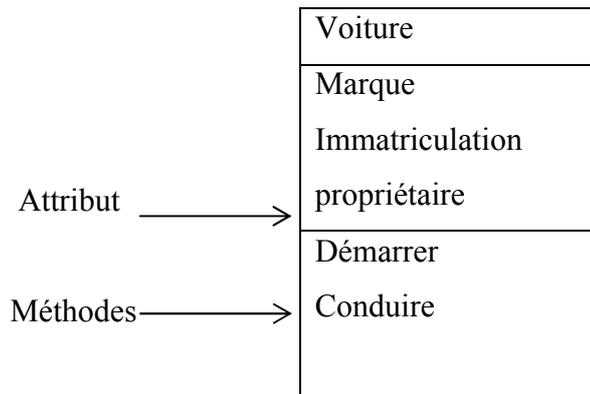
4.1. Les éléments structurels : Ils représentent les parties statiques du modèle.

Ceux sont des représentations conceptuelles ou physiques d'un système

Il existe 7 types d'éléments structurels :

- ✚ Classe
- ✚ Interface
- ✚ Collaboration
- ✚ Cas d'utilisation
- ✚ Classe d'activité
- ✚ Composant
- ✚ Nœud

4.1.1. Classe : c'est un ensemble d'éléments ayant les mêmes attributs, les mêmes opérations, les mêmes relations et la même sémantique.

Exemple :

4.1.2. Interface : c'est un ensemble d'opérations définissant la fonction d'un élément ou d'un composant. Elle définit seulement la signature des opérations.

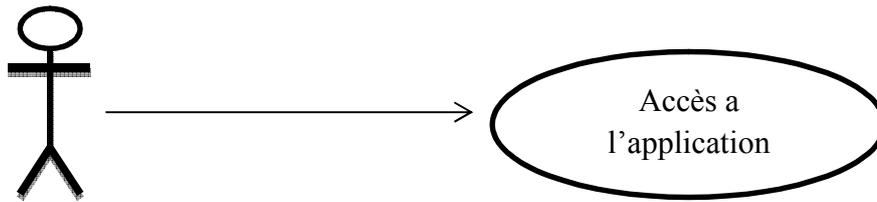
Exemple :

4.1.3 Collaboration : la collaboration Elle définit une interaction entre plusieurs éléments .Elle a un rôle structurel et comportemental.

4.1.4 Cas d'utilisation :

- ❖ Il s'agit de la solution UML pour représenter le modèle conceptuel.
- ❖ Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.
- ❖ Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés.
- ❖ Ils se limitent aux préoccupations "réelles" des utilisateurs ; ils ne présentent pas de solutions d'implémentation et ne forment pas un inventaire fonctionnel du système.
- ❖ Ils identifient les utilisateurs du système (acteurs) et leur interaction avec le système.
- ❖ Ils permettent de classer les acteurs et structurer les objectifs du système.

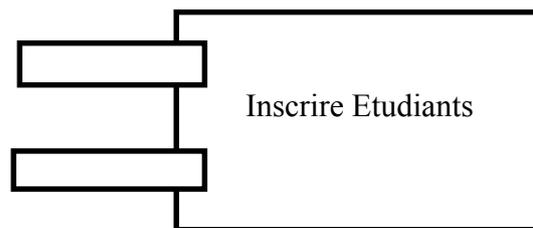
Exemple :



4.1.5 Classe d'activité : Classe particulière dont les objets possèdent un ou plusieurs processus pouvant lancer une activité de commande.

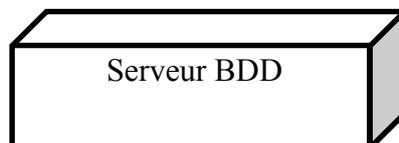
4.1.6 Composant : Le composant est une partie physique d'un système qui conforme a une spécification et fournit la réalisation d'un ensemble d'interface.il peut être un module.exe, com, javaBeans,etc ;il peut aussi être constitué de plusieurs sous composantes.

Exemple :



4.1.7 Nœud : C'est un élément physique. Lors d'une exécution, il représente une ressource ayant une capacité de calcul. En règle général ,il a moins de la mémoire et souvent de capacité de traitement .Un nœud est représenté par un cube.

Exemple :



4.2. Les éléments comportementaux

4.2.1 Les interactions :

Les interactions est l'ensemble des messages échanger entre les éléments du système(résultat de collaboration d'un groupe d'instances), elles sont des messages, des séquences d'actions ou bien des liens (relation entre des éléments). Une interaction peut être visualisée selon le point de vue du temps (diagramme de séquences) ou de celui d'espace (diagramme de collaboration). Un message par exemple est une interaction représenter par une ligne fléchée, indiquant le nom de son opération.

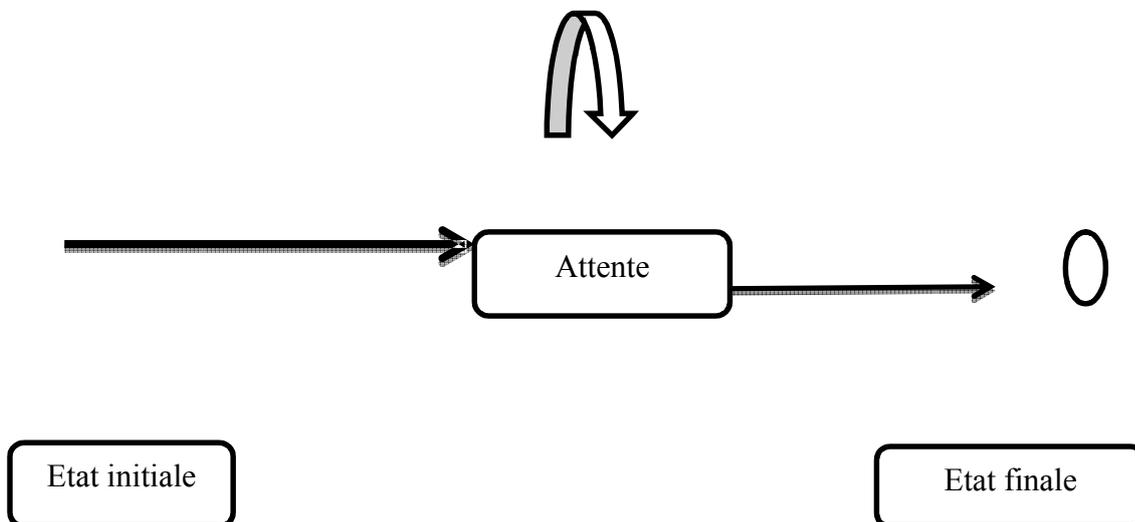
Exemple :



4.2.2 Etat :

La machine à état spécifie le cycle de vie d'un objet quand cet objet a un comportement dynamique. On représente graphiquement un état par un rectangle au coin arrondis (au centre), et en distingue les états de départ (à gauche) et terminaux (à droite). Comme décrit dans figure suivante.

Exemple :



4.3. Les éléments de regroupement :

Les éléments de regroupements représentent les parties organisationnelles des modèles

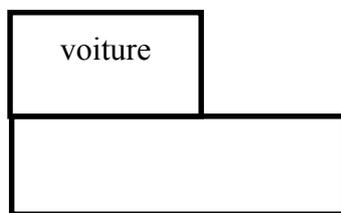
UML. Ce sont des boîtes dans lesquelles un modèle peut être décomposé. Il existe un seul type fondamental d'éléments de regroupement : le « paquetage ».

4.3.1 Package :

Un package en UML (ou paquetage en français) est un groupe d'éléments, dans le but de les grouper dans des ensembles cohérents. Un package peut contenir la plupart des éléments UML : classes, objets, cas d'utilisations, composantes, etc. Il peut également contenir des packages, créant une hiérarchie complète.

L'avantage des packages est qu'ils permettent de structurer les diagrammes et donnent une vision globale plus claire.

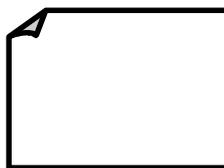
Exemple :



4.3.2 les éléments d'annotation :

Ils représentent les parties explicatives modèles. Ceux des commentaires. Il existe 1 type d'élément d'annotations : Ils font partie des décorations.

Exemple :

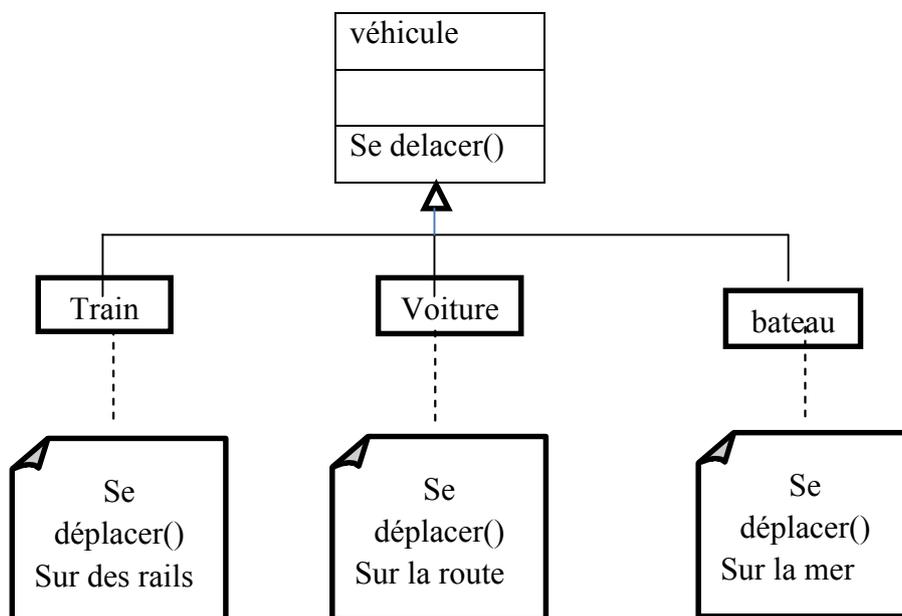


5. Les Relation : sont les liens entre les éléments.

Il existe Cinq types de relation interclasse sont couramment utilisés (il en existe d'autres) :

- héritage
- dépendance
- agrégation
- composition
- Association
- **Héritage :**
 - ✓ L'héritage est un mécanisme de transmission des propriétés d'une classe (ses attributs et méthodes) vers une sous-classe.
 - ✓ Une classe peut être spécialisée en d'autres classes, afin d'y ajouter des caractéristiques spécifiques ou d'en adapter certaines.
 - ✓ Plusieurs classes peuvent être généralisées en une classe qui les factorise, afin de regrouper les caractéristiques communes d'un ensemble de classes

Exemple :



- **Dépendance :**

C'est un lien sémantique entre 2 éléments .Un changement de l'un peut affecter la sémantique de l'autre.

Exemple :



- **Agrégation :**

L'agrégation est un cas particulier de relation d'association qui indique qu'une classe est une partie d'une autre classe. Cette relation comporte également les ordre de multiplicité On la représente graphiquement en décorant la relation d'association par un losange.

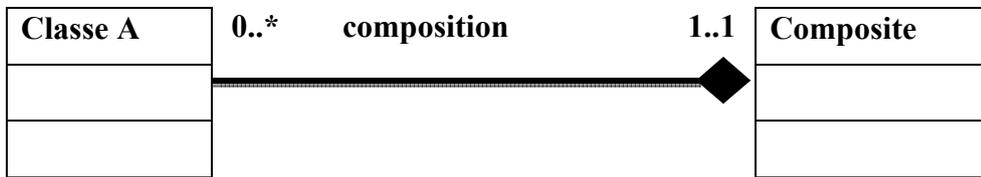
Exemple :



- **Composition :**

Composition Aussi appelée "agrégation forte" ou "agrégation par valeur", il s'agit en fait d'une agrégation à laquelle on impose des contraintes internes : un seul objet peut faire partie d'un composite (l'agrégat de la composition), et celui-ci doit gérer toutes ses parties. En clair, les composants sont totalement dépendants du composite.

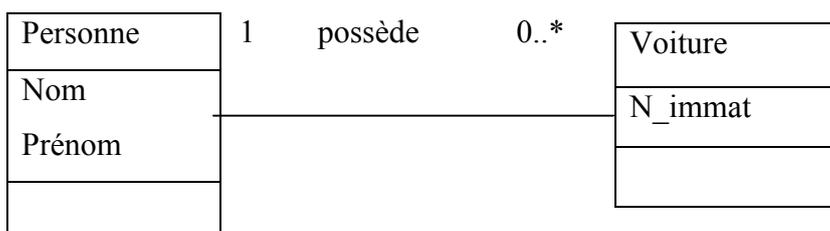
En UML, la composition est représentée de la même manière que l'agrégation, mais le diamant est plein.

Exemple :

- **Association :**

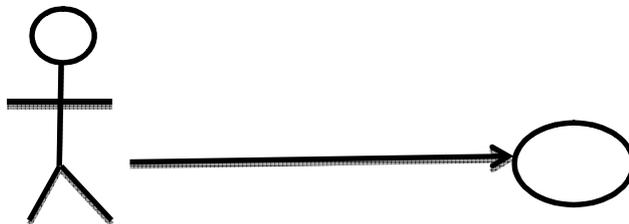
C'est la relation la plus simple entre deux classes. Elle existe à partir du moment où l'une des deux classes sert de type à un attribut de l'autre, et que cet autre envoie des messages à la première (condition nécessaire pour une association). Simplement, une association indique que deux classes communiquent entre elles (dans un sens ou dans les deux sens).

En UML, une association est représentée par une ligne entre deux classes, possiblement accompagnée d'une flèche si l'association n'est pas bidirectionnelle.

Exemple :**6. Les différents types de diagrammes UML :****6.1 diagramme cas d'utilisation :**

- Il s'agit de la solution UML pour représenter le modèle conceptuel. (est de comprendre et structurer les besoins du client.)
- Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.

- Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés.
- Ils se limitent aux préoccupations "réelles" des utilisateurs ; ils ne présentent pas de solutions d'implémentation et ne forment pas un inventaire fonctionnel du système.
- Ils identifient les utilisateurs du système (acteurs) et leur interaction avec le système.
- Ils permettent de classer les acteurs et structurer les objectifs du système.
- Ils servent de base à la traçabilité des exigences d'un système dans un processus de développement intégrant UML.

Exemple :

Acteur A

Cas d'utilisateur x

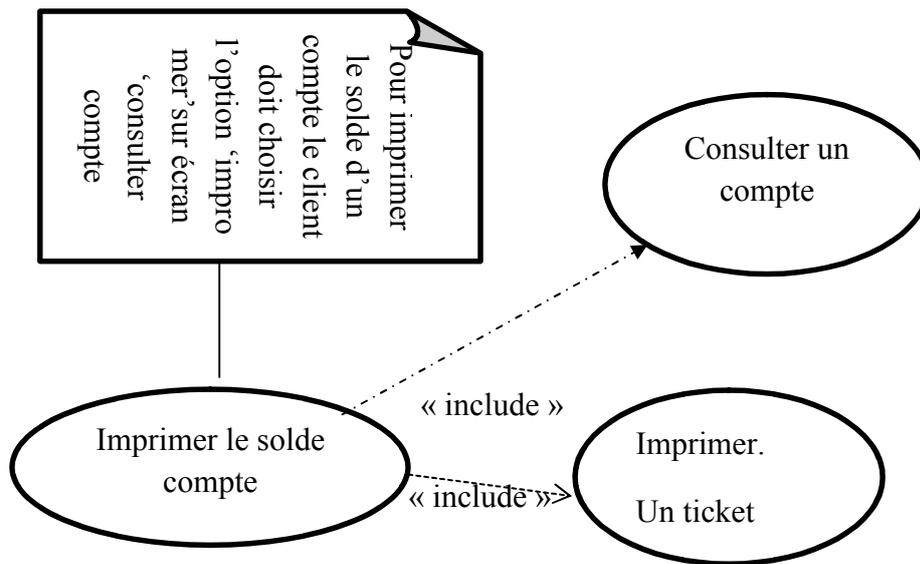
✓ Intérêt des cas d'utilisation :

- Le but de la conceptualisation est de comprendre et structurer les besoins du client.
- Il ne faut pas chercher l'exhaustivité, mais clarifier, filtrer et organiser les besoins
- Une fois identifiés et structurés, ces besoins :
 - définissent le contour du système à modéliser (ils précisent le but à atteindre),
 - permettent d'identifier les fonctionnalités principales (critiques) du système.
- Le modèle conceptuel doit permettre une meilleure compréhension du système.
- Le modèle conceptuel doit servir d'interface entre tous les acteurs du projet.
- Les besoins des clients sont des éléments de traçabilité dans un processus intégrant UML.

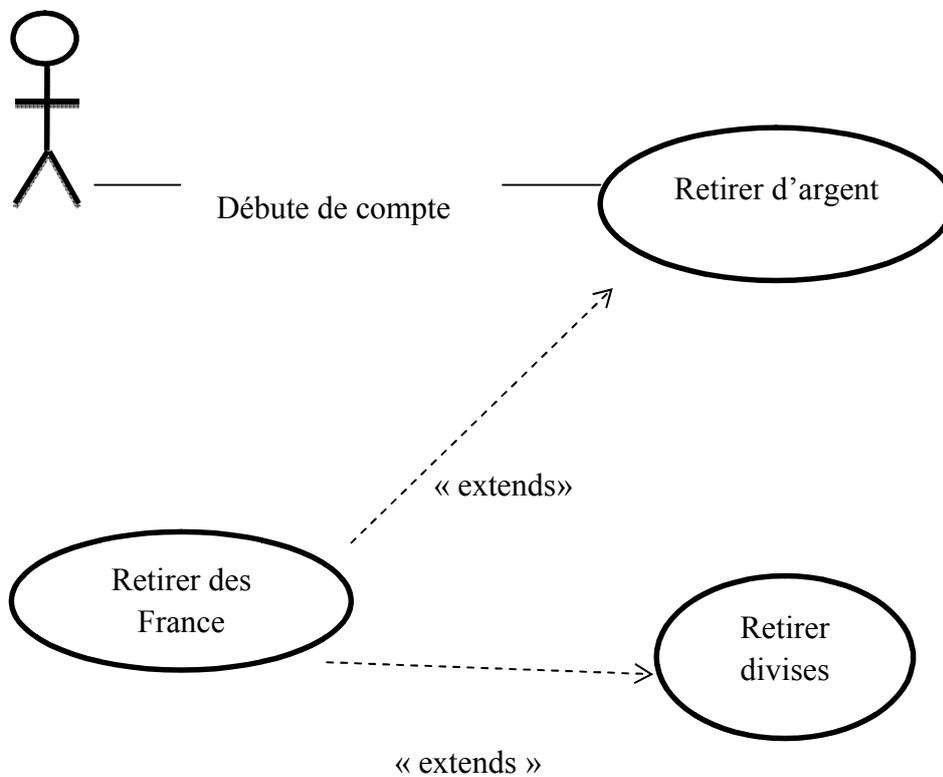
✓ La relation entre cas utilisation :

Il ya deux types de relation possibles entre cas :

- L'inclusion : quand les cas sources comprend le cas destination
- L'extension : quand la source ajoute son comportement cas.

Exemple :

- ✓ **Relation d'utilisation** : indiquer que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination

Exemple :

Le relation extension indiquer que le cas d'utilisation source étend (précise) les objectifs (le comportement) du cas d'utilisation destination

6.2 diagramme classes :

✓ Diagramme de classes : sémantique

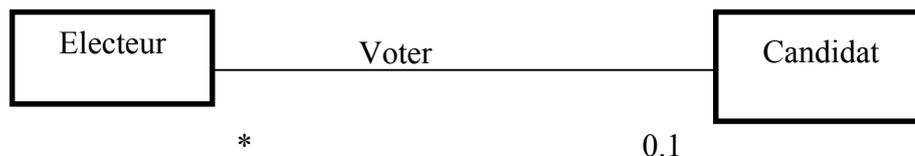
- Un diagramme de classes est une collection d'éléments de modélisation statiques (classes, paquetages...), qui montre la structure d'un modèle.
- Un diagramme de classes fait abstraction des aspects dynamiques et temporels.
- Pour un modèle complexe, plusieurs diagrammes de classes complémentaires doivent être Construits.

On peut par exemple se focaliser sur :

- les classes qui participent à un cas d'utilisation (cf. collaboration),
- les classes associées dans la réalisation d'un scénario précis,
- les classes qui composent un paquetage,
- la structure hiérarchique d'un ensemble de classes.

Les diagramme de classes permettent de représenter l'ensemble des informations formalisés, qui sont regrouper des classes.

Exemple :

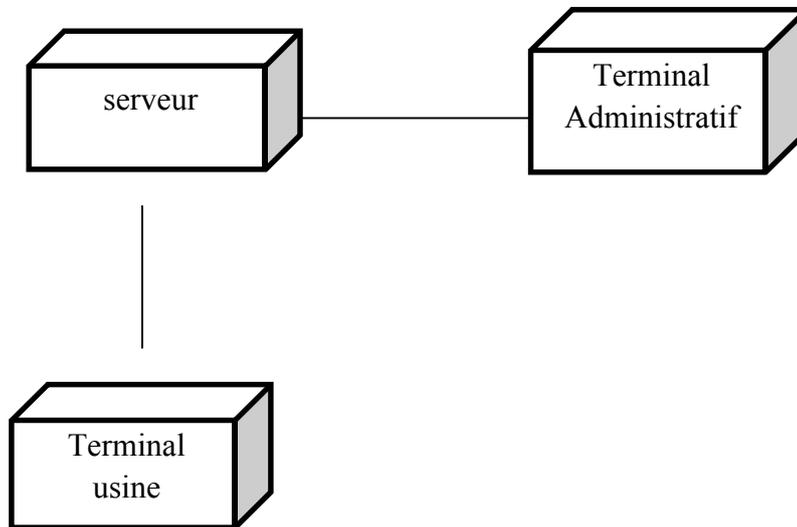


6.3. Diagramme de déploiement :

- Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.
- Les ressources matérielles sont représentées sous forme de nœuds.
- Les nœuds sont connectés entre eux, à l'aide d'un support de communication. La nature des lignes de communication et leurs caractéristiques peuvent être précisées.
- Les diagrammes de déploiement peuvent montrer des instances de nœuds (un matériel précis), ou des classes de nœuds.

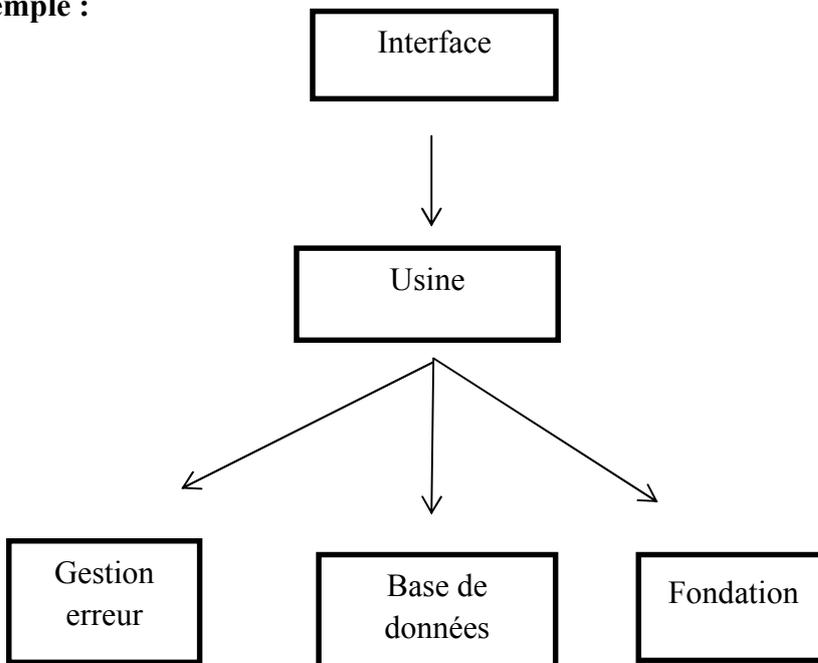
- Les diagrammes de déploiement correspondent à la vue de déploiement d'une architecture logicielle .

Exemple :



6.4 diagramme de composants : Il représente l'organisation et les dépendances des composants. Un composant est soit une (ou plusieurs) classe(s), interface(s) ou collaboration(s).

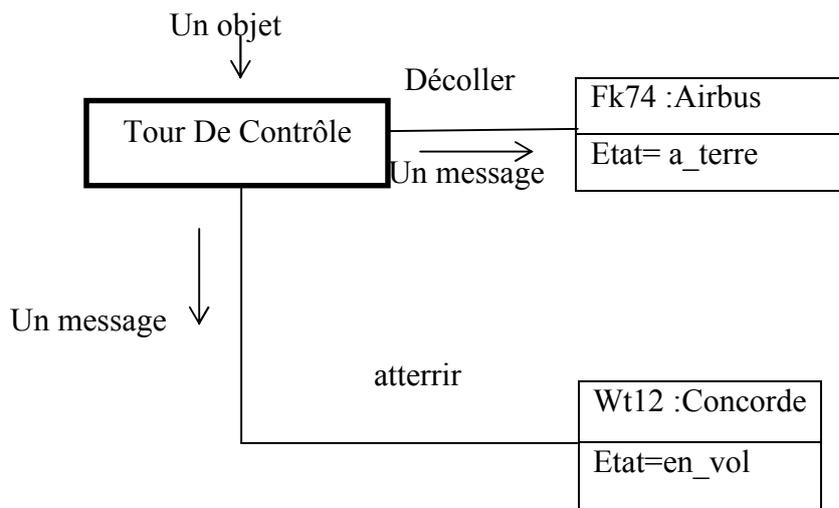
Exemple :



6.5 Diagramme de collaboration

- Les diagrammes de collaboration montrent des interactions entre objets (instances de classes et acteurs).
- Ils permettent de représenter le contexte d'une interaction, car on peut y préciser les états des objets qui interagissent.

Exemple :



6.6 Diagramme de séquence : sémantique

C'est un diagramme d'interaction. Il représente un ensemble d'objets et leurs relations, avec les messages qu'ils échangent. (aspect chronologique des messages).

6.6.1 Les différents types de messages :

- Message simple
- Message minuté
- Message synchrone
- Message asynchrone
- Message déroband

🚦 message simple

Message dont on ne spécifie aucune caractéristique d'envoi ou de réception particulière.

🚦 message minuté (timeout)

Bloque l'expéditeur pendant un temps donné (qui peut être spécifié dans une contrainte), en attendant la prise en compte du message par le récepteur. L'expéditeur est libéré si la prise en compte n'a pas eu lieu pendant le délai spécifié.

 **message synchrone**

Bloque l'expéditeur jusqu'à prise en compte du message par le destinataire. Le flot de contrôle passe de l'émetteur au récepteur (l'émetteur devient passif et le récepteur actif) à la prise en compte du message.

 **message asynchrone**

N'interrompt pas l'exécution de l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré (jamais traité).

 **message déroband**

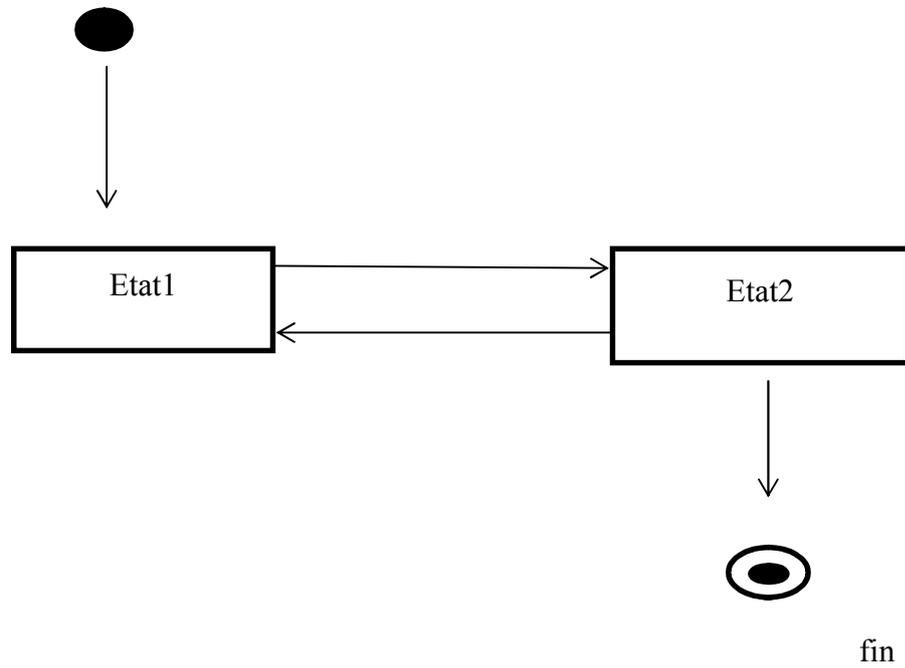
N'interrompt pas l'exécution de l'expéditeur et ne déclenche une opération chez le récepteur que s'il s'est préalablement mis en attente de ce message.

6.7 Diagramme d'états-transitions : sémantique

- Ce diagramme sert à représenter des automates d'états finis, sous forme de graphes d'états, reliés par des arcs orientés qui décrivent les transitions.
- Les diagrammes d'états-transitions permettent de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.
- Un état se caractérise par sa durée et sa stabilité, il représente une conjonction instantanée des valeurs des attributs d'un objet.
- Une transition représente le passage instantané d'un état vers un autre.
- Une transition est déclenchée par un événement. En d'autres termes : c'est l'arrivée d'un événement qui conditionne la transition.
- Les transitions peuvent aussi être automatiques, lorsqu'on ne spécifie pas l'événement qui la déclenche.

Exemple :

Début

**Conclusion :**

L'UML, comme l'on a vu à travers ce chapitre, ne propose pas une démarche objet mais une notation adapté au monde de développement orienté objet. Il nous a donc permis de s'initier aux techniques de modélisation objet. La notation UML peut s'adapter a tous les projets informatiques.

1. Introduction :

Processus unifié (up) Le processus unifié est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. Caractéristiques essentielles du processus unifié :

- ✓ Le processus unifié est à base de composants,
- ✓ Le processus unifié utilise le langage UML (ensemble d'outils et de diagramme),
- ✓ Le processus unifié est piloté par les cas d'utilisation,
- ✓ Centré sur l'architecture,
- ✓ Itératif et incrémental

2. RUP (Rational Unified Process) :

- Dérivée de UP (UP a été créée en 1996)
- Commercialisée par IBM en 1998.

2.1. Méthode de développement logiciel :

- itérative,
- Incrémentale
- pilotée par les cas d'utilisation.
- centrée sur l'architecture et la réduction des risques
- Produit de qualité

2.2. Phases de RUP :

2.2.1. Phase d'initialisation :

- Définir la portée du projet
- Spécification
- Affectation des tâches
- Evaluation des risques
- Cycle de vie

2.2.2. Phase d'élaboration :

- Planifier le projet,
- spécifier les fonctionnalités
- construire l'architecture
- Spécification du produit
- Conception de l'architecture
- Planification activités-ressources

2.2.3. Phase de construction :

- Construire le produit
- Implémentation du produit
- Tests
- produit opérationnel

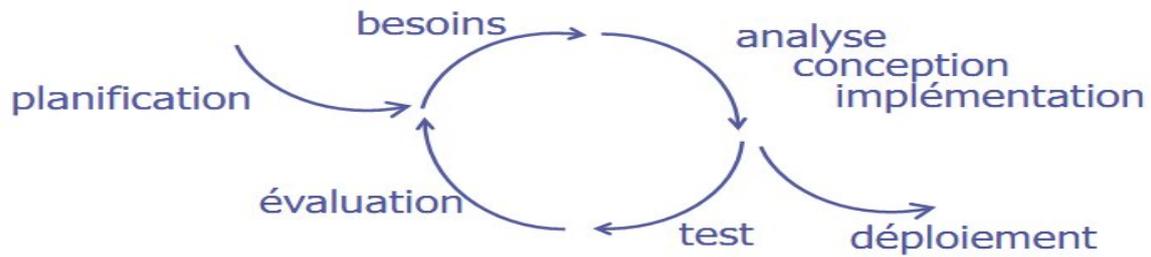
2.2.4. Phase de transition :

- Transition du produit vers les utilisateurs
- Livraison
- Formation
- Qualité

2.3. Cycle de vie de RUP :

- **Exigence** détermination des besoins :
 - fonctionnels (ce que l'on attend du système)
 - non fonctionnels (fiabilité, temps de réponse, environnement distribué, etc.)
- **Analyse et conception** : évoluer depuis la spécification des besoins jusqu'à une solution informatique
 - analyse besoins fonctionnels
 - conception intègre aussi les besoins non fonctionnels
- **Implémentation** :
 - Transcription dans un langage de programmation ou de base de données .
 - Utilisation de composants existants
- **Test** :
 - Estimer
 - Si les besoins sont satisfaits. S'il y a des erreurs/défauts à corriger Renforcer et stabiliser l'architecture
- **Déploiement** :
 - Distribuer le logiciel dans son environnement opérationnel, Installation, test Formation des utilisateurs, Migration des données

Cycle de base



Cycle de vie de RUP

2.4. Différence entre incrémental et itératif

✓ RUP est incrémental

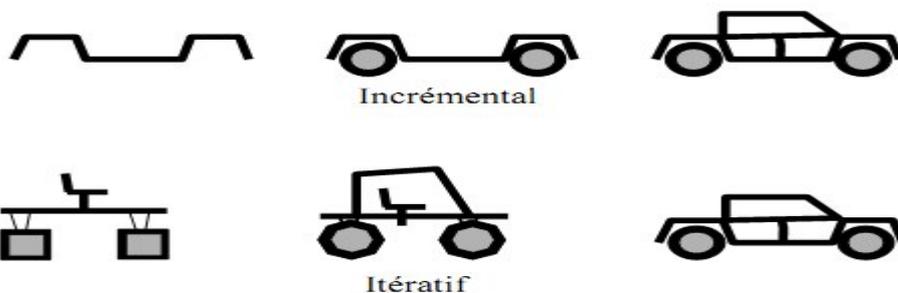
- Le produit final est livré en incréments
- Chaque incrément livre une fonctionnalité ayant une valeur pour l'utilisateur final
- Chaque incrément consiste en un ensemble de cas d'utilisation
- Les incréments doivent être de courte durée (quelques semaines à quelques mois)
- Les incréments sont des mini-projets dans un projet

✓ RUP est itératif

Une itération est une séquence d'activités qui répond à un plan et à des critères d'évaluation et qui produit une version exécutable

Exemple

Différence entre incrémental et itératif



❖ Forces de la méthode RUP

Traçabilité à partir des Uses Cases jusqu'au déploiement

Approche basée sur l'architecture

Gestion des risques dans les projets

❖ Faiblesses de la méthode RUP

Coût de personnalisation souvent élevés

Très axé processus

Conclusion :

Cette partie nous a permis de prendre connaissance le processus rup (rationnel unified process) est une méthode de développement pour logiciel orienté objet, c'est une méthode générique itérative et incrémentale.

1. Introduction :

Dans les productions informatiques d'aujourd'hui, il est inconcevable de perdre des données. Pourtant, les problèmes matériels et/ou erreurs utilisateurs peuvent toujours exister. Et là... c'est le drame !

Nous savons que nous devons faire des sauvegardes régulières afin de pouvoir retrouver nos données mais s'il faut restaurer les données, cela se traduit par une indisponibilité du service (au moins partielle) tant que la restauration n'est pas terminée.

Il existe plusieurs solutions pour permettre de récupérer rapidement une information et nous allons nous intéresser aux solutions d'infrastructure de résistance aux pannes proposées par Oracle.

2. Définition SGBD :

Un SGBD est un système complexe permettant de gérer de manière efficace, un volume important de données structurées, accessible par des utilisateurs simultanés locaux ou non. On plus SGBD est un ensemble d'outils logiciels permettant la création et utilisation de bases de données

3. Caractéristique d'un SGBD (système de gestion des base de données) :

- **gestion de gros volumes de données** (en consultation et en mise à jour),
- **Sécurité des données, qui se décline en :**

- ✓ **disponibilité**

Un SGBDR (système de gestion des base de données relationnelles) se doit d'offrir une bonne disponibilité des données. Une disponibilité totale des données est possible (temps de reprise nul) il suffit de s'en donner les moyens logiciels et matériels..

- ✓ **fiabilité**

Des mécanismes de sauvegarde variés (physique, logique, off-line, on-line, totale, partielle, incrémentale), ainsi que des mécanismes de journalisation, et de reprise permettent de restaurer une information sans pratiquement aucune perte, dans tous les cas de problème matériel ou logiciel.

- ✓ **confidentialité**

Tout n'est pas accessible à tout le monde! Se connecter à la base de données, donne un certain nombre de droits et de ressources en fonction d'un profil défini et maintenu par un administrateur. La granularité d'accès peut aller jusqu'à la vision unique d'un champ d'un enregistrement d'une table particulière. Encore une fois on ne manipule plus des fichiers..

✓ **cohérence**

Que les données soient réparties ou non –dans ce dernier cas les mécanismes mis en jeu seront plus complexes– elles doivent être cohérentes. Cela sous entend, d'une part que les accès concurrents d'utilisateurs, notamment lors de mises à jour, ne doivent pas compromettre l'intégrité des données et d'autre part que ces dernières satisfassent aux contraintes d'intégrité du modèle, mais aussi aux règles de gestion de l'entreprise.

- **concurrence d'accès en lecture et écriture** (avec une bonne granularité),
- **gestion (efficace) des transactions,**
- **portabilité sur différents OS, des données et du code**
- **Administrable:**
 - existence d'outils d'administration généraux : gestion des données, des utilisateurs, des fichiers physiques, des espaces logiques, des droits, des profils, des ressources systèmes,etc.
 - outils de surveillance en temps réel grâce à un moniteur, si possible graphique ou en temps différé grâce à des journaux ou à des traces paramétrables.
- **performances:**

offrir de bonnes performances et des outils permettant de les mesurer et de les contrôler via des paramètres de configuration. Des processus d'optimisation en temps réel des requêtes complexes sont également souvent présents. Les données peuvent être indexées, de manière souple, dynamique et complète (index simples, concaténés, multiples, tables de hashage, recherche textuelle, etc.). Un nombre important d'utilisateurs, ainsi qu'un volume conséquent de données peuvent être pris en compte.

4. Présentation de Easy PHP

4.1 Dans quels cas utiliser EasyPHP ?

- Traitement d'un gros volume de données (besoin d'optimiser l'utilisation de la mémoire, besoin important de performance sur traitements, ...)
- Entreprises ayant une problématique de sécurité (redondance et accès)
- Traitement des données particulières
- Nécessite souvent un Administrateur.

4.2 Pourquoi Easy PHP

Le langage PHP est un langage extrêmement puissant : il permet de créer des pages web, au travers desquelles l'utilisateur peut échanger des informations avec le serveur ; c'est ce qu'on appelle des pages web dynamiques. Programmer en PHP est assez simple. En revanche, PHP n'est pas un langage compilé, c'est un langage interprété par le serveur : le serveur lit le code PHP, le transforme et génère la page HTML. Pour fonctionner, il a donc besoin d'un serveur web. Donc si vous souhaitez utiliser des pages en PHP dans votre site web, pour les tester, il faudra les exécuter sur un serveur web. Donc deux solutions :

- soit vous les envoyez régulièrement grâce à un programme FTP sur votre serveur web pour les tester. C'est faisable, mais ça peut devenir fastidieux ;
- soit vous installez un serveur web en local, qui vous permettra de tester directement vos pages PHP. EasyPHP permet de réaliser simplement cette dernière solution.

Enfin, EasyPHP permet également d'installer MySQL, une base de donnée, le troisième et inséparable membre du trio Apache/PHP/MySQL. Une base de données est un programme permettant de gérer une grande quantité de données en les organisant sous forme de tables. Vous n'avez alors plus à vous occuper de la manière dont les données sont stockées sur le disque dur, de simples instructions permettent d'ajouter, de supprimer, de mettre à jour et surtout de rechercher des données dans une base de données. On peut de plus accéder très facilement à une base de données MySQL à partir de PHP, ce qui permet de développer des sites web très performants et interactifs (par exemple, le forum de Developpez.com). EasyPHP joint PHPMyAdmin à MySQL, un outil écrit en PHP permettant de gérer vos bases de données MySQL. En utilisant EasyPHP, vous pouvez installer un serveur web complet, qui vous permettra de faire tous vos tests de pages PHP en toute facilité.

4.3 Quelles fonctionnalités propose Easy PHP

Chose-la plus importante, EasyPHP propose le téléchargement en une fois et l'installation en un assistant des trois programmes précédemment cités, Apache, PHP et MySQL. Cela permet d'installer automatiquement ceux-ci, en se libérant des problèmes liés à la configuration manuelle qui est souvent nécessaire lorsqu'on les installe séparément.

Lorsqu'EasyPHP est lancé, les serveurs Apache et MySQL sont automatiquement lancés (il est même possible de le faire automatiquement au démarrage de Windows). Une petite icône s'installe dans la barre des tâches, à côté de l'horloge, permettant un accès rapide aux fonctions proposées par EasyPHP :

- arrêter et Redémarrer les serveurs Apache et MySQL ;
- accéder au « Web local », c'est-à-dire la racine des sites Web ;
- un panneau d'administration en PHP ;
- un outil de configuration d'EasyPHP ;
- l'accès aux logs ;
- l'aide

Conclusion :

EasyPHP est un excellent outil pour ceux qui ne sont pas à l'aise pour coder en PHP. Pour les codeurs plus expérimentés, il peut encore représenter un gain de temps séduisant. L'interface est assez intuitive et il ne faut pas longtemps pour prendre en main les fonctionnalités variées d'EasyPHP. - vous pouvez simplement importer vos bases de données via l'interface PHPMysqladmin. EasyPHP peut être installé sur un disque local, ou sur un média amovible comme un disque flash, donc vous pouvez l'emporter avec vous n'importe où. Et de citer quelques caractéristiques d'un SGBD.