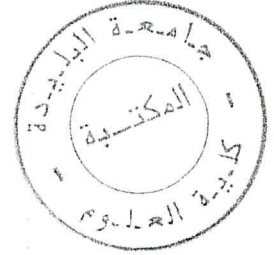


الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique  
Université de Saad Dâhlab Blida

FACULTE DE SCIENCES EXACTES  
DEPARTEMENT D'INFORMATIQUE

Mémoire de projet de fin d'études  
Pour l'obtention du diplôme  
D'ingénieur d'état en informatique



Option : Intelligence Artificielle

Thème:

**MODELISATION ET REALISATION D'UN SYSTEME  
DE  
GESTION COMMERCIAL SOUS RESEAUX DE  
DONNEES  
D'UN OPERATEUR DE TELECOMMUNICATION**

Thème dirigé par :

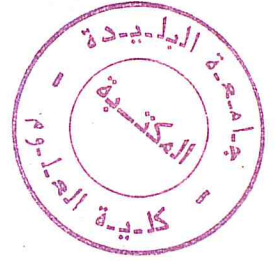
M<sup>r</sup> BENNOUAR Djamel

Présenter par :

M<sup>r</sup> MEZAOUROU Brahim  
M<sup>lle</sup> ZERF Nadjat

Promotion 2003-2004

MIG-004-32-1



بسم الله الرحمن الرحيم  
الحمد لله رب العالمين  
والصلاة والسلام على  
سيدنا محمد وآله الطيبين  
الطاهرين

أشكر الله على ما  
أنعم به عليّ وعلى  
والديّ وأهليّ وأمتي  
والمسلمين أجمعين

وأشكره على ما  
أنعم به عليّ وعلى  
والديّ وأهليّ وأمتي  
والمسلمين أجمعين

والحمد لله رب العالمين  
والصلاة والسلام على  
سيدنا محمد وآله الطيبين  
الطاهرين

# Dédicace

*Ce mémoire est dédié :*

*A mes très chers parents pour leur soutien durant  
toute ma carrière,*

*Pour leur bienveillance, leurs efforts constants dans mes études, et  
Pour leur encouragements.*

*A mes frères et mes sœurs*

*Aux petits princes : Yasmina, Chahira, Mohamed, Maleke*

*A ma grand-mère, mes oncles, mes tantes*

*A tous les employés de l'école de la formation en Informatique*

**MEZAOUROU ( EFIM)**

*A mon binôme Nadjat et à toute sa famille*

*A tous mes amis que j'aime et qui m'aiment.*

**M. IBrahim**

# Dédicace

*Ce mémoire est dédié :*

*A mes très chers parents pour leur soutien durant  
toute ma carrière,*

*Pour leur bienveillance, leurs efforts constants dans mes études, et  
Pour leur encouragements.*

*A mes frères Fouad et Mohamed,*

*A mes sœurs Nora et ses enfants, Samira, Ahlem*

*La petite princesse : Ibtissem*

*A ma grand-mère MIMI*

*A mon binôme IBrahim et à toute sa famille*

*A mes amis Nadjia, Salima, Nawel, Mahdia, Wassila, Darine,  
hannan, Hamza, B. Amine, Fateh et Kechih. H.*

*A tous mes amis que j'aime et qui m'aiment.*

*Z. Nadjjet*

# Remerciements

*Nous remercions avant tout le bon dieu qui nous a aidé à réaliser ce modeste travail.*

*Nous tenons à remercier M<sup>r</sup>. BENNOUAR Djamel et qui a bien voulu proposer le sujet pour son aide, sa patience, sa disponibilité et sa compréhensibilité.*

*Nous remercions M<sup>me</sup> le chef du département d'Informatique, tous les enseignants de la faculté des sciences de BLIDA et surtout ceux du département informatique.*

*Nous remercions les membres du jury pour nous avoir fait l'honneur de juger notre travail.*

*Les plus sincères remerciements sont destinés au laboratoire LRDSI pour leur impeccable accueil*

*Un remerciement tout spécial à M<sup>r</sup> MENACER, M<sup>me</sup> GHATASSE et à toute l'équipe de l'école logitel annexe de certification MicroSoft*

*Nous remercions, de tout coeur, tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.*

*Et enfin, Nous tenons à remercier tous nos amis, Boudjettou Hamza, Nadjia Merabet, Hassaine Salima, Mazari Farid, Fatma, Raouf pour l'aide qu'il nous a apportée pour l'édition de la thèse.*

## Table des matières :

### **Introduction**

1. Objectifs.....	4
2. Cahier de charge.....	5
3. Sommaire.....	5

### **Chapitre I La norme X.25**

<b>I.</b> La norme X.25.....	8
<b>II.</b> Caractéristiques de X.25.....	9
<b>III.</b> Les trois niveaux du protocole X.25.....	9
III.1- Le niveau physique X25-1.....	10
III.2- Le niveau liaison X25-2.....	10
III.3- Le niveau réseau X25-3 .....	11
<b>IV.</b> Connexion des terminaux asynchrones .....	16
IV.1- PAD .....	16
<b>V.</b> Les services facultatifs de la norme X25.....	18
<b>VI.</b> Gestion de X25.....	18

### **Chapitre II UML et la méthode de conception**

<b>I UML</b> .....	20
I.1 Les concepts.....	20
I.2. Les relations .....	21
I.3. Les diagrammes d'UML.....	22
I. 3.1 Diagrammes statiques.....	22
I.3.2 Diagrammes dynamiques.....	25
<b>II. Méthodologie de conception RUP</b> .....	27
II.1. La vue logique .....	28
II.2. La vue de processus.....	28
II.3. La vue des composants .....	29
II.4 La vue physique.....	29
II.5 La vue des cas d'utilisation.....	29

III. Conclusion.....	31
----------------------	----

### **Chapitre III L'architecture du système**

I. Introduction.....	33
II. Description globale du système .....	33
III. Indépendance entre les sous systèmes .....	35
IV. Description détaillée du système .....	36
I. Préparation de flux .....	36
I.1 Sous système d'Acquisition .....	37
I.2 sous système de Préparation.....	38
II. Flux du Facturation .....	43
II.1 sous système : Valorisation du coût unitaire .....	44
II.2 Sous système de Valorisation de produit.....	45
III.3 Sous système de Facturation.....	46
III. Interface homme machine.....	46
IV. Les données du système .....	47
IV.1 Données en entrée.....	47
IV.2 Données en sortie.....	47
V. Conclusion.....	48

### **Chapitre IV La conception et la réalisation**

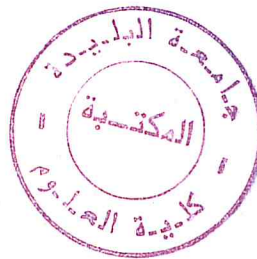
Introduction .....	50
I. La vue cas d'utilisation.....	50
I. Diagramme de cas d'utilisation.....	51
I.1 Définition des acteurs.....	51
II. Le cas d'utilisation de haut niveau .....	52
III. Les interactions entre le système et les utilisateurs .....	53
III.1 Les modes.....	53
III.2 Fonctionnalités génériques .....	54

IV. Gestion des utilisateurs et l'administration.....	55
IV.1 Paramétrage du système.....	55
IV.2 Gestion des utilisateurs.....	56
V. Gestion des clients.....	58
V.1 Création d'un client .....	58
V.2 Consultation et Modification des données d'un client.....	59
VI. Gestion des abonnements .....	61
VII. Gestion des types de jour et tranches horaires.....	62
VII.1 les types des jours.....	62
VII.2 Les tranches horaires .....	63
VIII. Les tarifications.....	65
VIII.1 Traitement des tarifications.....	65
VIII.2 Réductions et tarifs spécifiques .....	66
IX. Facturation et suivi de paiement.....	66
<b>II. La vue logique.....</b>	<b>69</b>
I. Diagramme de classe .....	69
II. Paquetage de gestion des clients.....	71
III. Paquetage de l'administration .....	73
IV. Le paquetage de tranches horaires .....	73
V. Paquetage de communication .....	75
VI. Le paquetage de gestion des tarifications .....	76
VII. Paquetage de la facturation .....	78
<b>III. La Vue Dynamique.....</b>	<b>79</b>
I. Diagramme d'activités.....	79
I.1 Diagramme d'activité du sous système d'acquisition .....	79
I.2 Diagramme d'activités du sous système préparation.....	81
I.3 Diagramme d'Activités sous système valorisation du coût unitaire.....	83
I.4 Diagramme d'activités du sous système Calcul des	



---

***INTRODUCTION***  
***GENERALE***



L'étude abordée dans ce projet se rapporte à la modélisation et la réalisation d'un système permettant la gestion commerciale de produit réseaux de données d'un opérateur de télécommunication.

Un opérateur de télécommunication dispose de plusieurs réseaux de données, offrant chacun divers services et possibilités. Dans le contexte de chaque réseau, l'opérateur commercialise divers produit. L'exemple le plus simple d'un produit de télécommunication est l'établissement d'une connexion entre deux clients pour une certaine durée nécessité par l'échange d'une masse d'informations bien précise.

Souvent pour permettre une exploitation efficace du réseau, les opérateurs commercialisent leurs produits avec des tarifs différents selon la période d'exploitation. (tarif de nuit, tarif de jour, d'été, de période spéciale etc.). Il est clair que les périodes ne sont pas figées et sont définies selon les exigences du marché des services de télécommunication

En plus des services propres aux réseaux, l'opérateur peut commercialiser d'autres produits dans le contexte de l'abonnement. Ainsi, il y'a les prix fixe du à l'abonnement, un prix fixe du au type de communication (locale, régionale, nationale, internationale etc.) Et d'autres prix fixes du à la location de certain équipement de télécommunication, notamment les modems et convertisseurs.

Les réseaux de transport de données sont gérés par des ordinateurs appelés les stations de supervision. Ces stations de supervision reportent toutes les informations concernant l'exploitation du réseau par les abonnés. Les informations sur une communication sont souvent reportées sous forme de ce qui a convenu d'appeler des tickets de taxation.

Un ticket est un ensemble de données dans un format bien précis (défini par le concepteur du réseau de données). A titre d'exemple un ticket contiendra la référence de l'abonné ayant initié la communication, la référence de l'abonné appelé, la date et l'heure de l'établissement de la communication, la durée de la communication, le nombre d'octets ou kilo octets échangés etc..

Un opérateur de télécommunication, pour le même type de réseaux, peut disposer d'infrastructure mise en place par divers fournisseurs. Chaque fournisseur d'une partie du réseau peut avoir ses propres stations de supervision et ses propres formats des tickets de taxation.

## 1. Objectifs :

La gestion commerciale des produits d'un réseau de données couvre toutes les activités liées à son exploitation tels que: la gestion des abonnés, des abonnements, des produits, des tarifs, des politiques tarifaires, l'ouverture effective d'un abonnement, la résiliation effective d'abonnements, la reprise d'un abonnement, l'ajout ou retrait de service, la facturation des produits et usages, le suivi des paiements. l'ouverture sur d'autres logiciels, notamment ceux de comptabilité, le suivi de l'exploitation par l'établissement de statistiques etc..

Le but du projet est de:

- Mettre en place un système informatique permettant de gérer la commercialisation des divers produits d'un réseau de données. Nous prendrons comme exemple pratique le réseau X25
- Utiliser les concepts de l'architecture DCOM sous la plate forme .Net et l'architecture des réseaux à 3 niveaux (appelées *architecture 3-tiers*), on a généralement une architecture partagée entre Le client (le demandeur de ressources), Le serveur d'application qui est chargé de fournir la ressource mais faisant appel à un autre serveur secondaire (généralement un serveur de base de données), fournissant un service au premier serveur.
- Utiliser UML comme un langage de modélisation et suivi une méthode de conception.

Dans son état final le système devra être capable de nous fournir entre autres, les informations suivantes :

- Facture globale ou détaillé de clients donnés pour des périodes données.
- Divers états (exemple: visualisation des tables de la base de données).
- Les messages d'erreurs (ou alarmes) nous renseignant sur le bon ou le mauvais fonctionnement de certaines parties du système.

## 2. Cahier de charge:

Le système à réaliser sera conçu selon une approche objet. Le langage de spécification des divers aspects du système sera UML. Pour arriver à ce but, les travaux suivants seraient nécessaires :

- 1- Etude du monde réel des réseaux X25, que ce soit du point de vue implantation du réseaux et les divers services qu'ils procurent et les diverses actions de gestion et contrôle direct sur ce réseau ou point de vue gestion commerciale des produits de ce réseau.
- 2- Etude poussée D'UML et apprentissage des outils (un IDE) basé sur UML.
- 3- Etude d'une ou deux méthodologies de développement de système en UML.
- 4- Définition informelle de plusieurs alternatives architecturales : Quelles sont les composantes du système, que fait chaque composante, selon quel style architectural ces composantes sont liées et comment elles sont déployées
- 5- Choix des cibles d'implantation pour chaque alternatives : Plate-forme, environnement d'exécution ; langage de programmation, SGBD.
- 6- Choix de l'architecture et le déploiement du système sur le matériel.
- 7- Modélisation UML de chaque alternative: Quelles sont les vues à utiliser pour définir les aspects nécessaires du système ?
- 8- Implémentation effective du système.

## 3. Sommaire :

Le premier chapitre expose brièvement les notions fondamentales des réseaux et la norme X25 (cette partie utile pour la réalisation de la base de données).

Le chapitre suivant, expose le langage de modélisation UML et ses concepts fondamentaux puis la méthode que nous allons suivre pour la modélisation et la réalisation du système et ses étapes.

Dans le chapitre III nous allons présenter une définition informelle de l'architecture proposer: Quelles sont les composantes du système, que fait chaque composante, selon quel style architectural ces composantes sont liées et comment elles sont déployées.

Les chapitres suivants seront consacrés au travail effectué (modélisation et réalisation d'un gestion commerciale des Services d'un réseau de données) ; tout d'abord nous allons présenter les problèmes et les expressions des besoins, puis en va montrer l'analyse et la conception et proposer une solution pour résoudre les problèmes citer.

Les détails du travail effectué qui seront décrits dans les chapitres de modélisation seront notés par le langage de modélisation UML.

Enfin dans la dernière partie consacrée aux tests : on montrera les différentes étapes de réalisation de notre système ainsi que les différentes évaluations des tests réalisés (selon les besoins actuels).

---

*CHAPITRE I*  
*LA NORME X25*

## I. La norme X.25 :

Cette norme a été établie en 1976 pour les réseaux à commutation de paquet sur proposition de quatre pays qui l'utilisent pour leurs réseaux publics de communication : Transpac pour la France, EPSS pour la Grande-Bretagne, Datapac pour le Canada et Telenet pour les USA [1].

La recommandation X.25 précise un protocole définissant l'interface entre un ETTD (Equipement Terminal de Transmission de Données) et un ETCD (Equipement Terminal de Circuit de Données) pour la transmission des paquets.

La définition de X.25 a été étendue pour prendre en compte des transmissions soit sur l'interface d'entrée du réseau, soit entre la machine de l'utilisateur (ETTD) et l'équipement d'accès de l'opérateur (ETCD), soit de la machine de l'utilisateur à la distante (ETTD à ETCD).

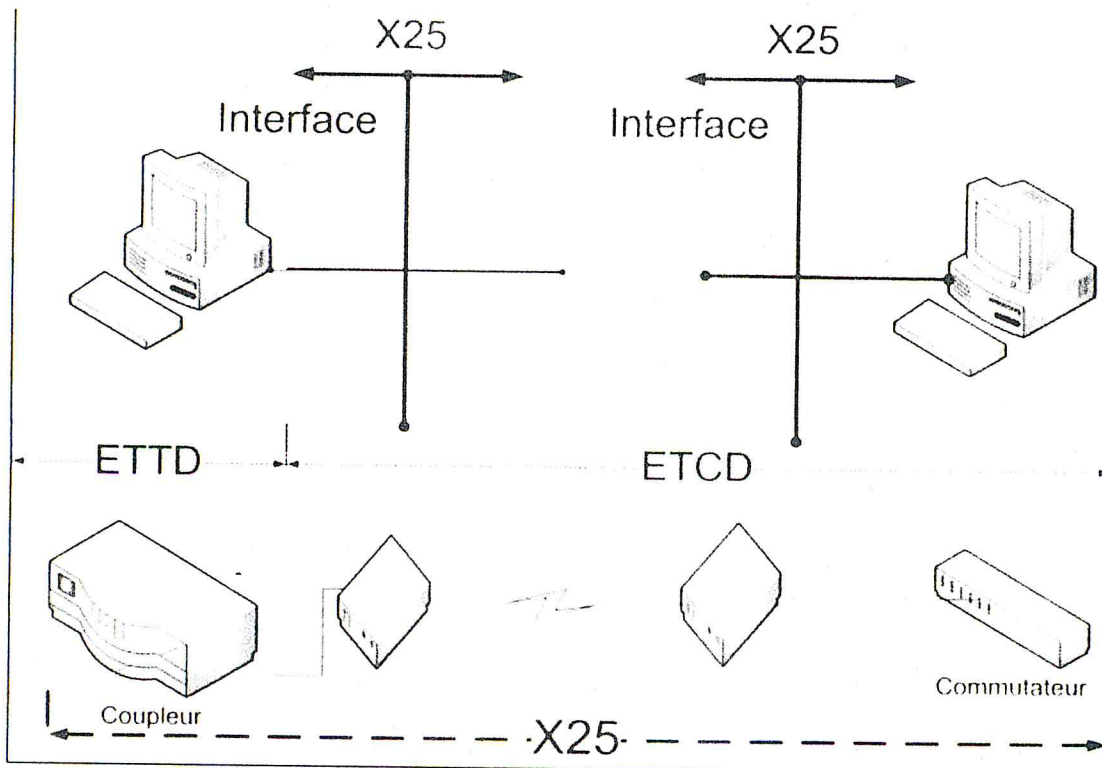


Figure I.1 Implantation du protocole X.25 [1]

## II. Caractéristiques de X.25 : [2]

- Le protocole X.25 utilise le mode avec connexion.
- La recommandation X.25 définit les types des paquets et leurs formats. Elle ne spécifie pas comment certaines informations de contrôle doivent être interprétées. En particulier, la fenêtre de contrôle de flux peut être interprétée au niveau local entre ETTD et ETCD, ou au niveau global entre l'ETTD émetteur et l'ETCD récepteur.
- Le niveau paquet de X.25 permet un maximum de 16 groupes de 256 voies logiques entre un ETTD et un ETCD. L'en-tête du paquet contient un champ de 4 bits qui identifie le groupe et un champ de 8 bits pour le numéro de la voie logique. Ainsi, 4095 voies logiques (la voie 0 joue un rôle particulier) sont utilisables sur une entrée.

## III. Les trois niveaux du protocole X.25 :

Comme illustré dans la figure I.2 le protocole X.25 contient les trois premières couches des modèles de référence sur l'architecture des réseaux informatiques. [1]

- Le niveau physique provient principalement de la norme X.21.
- La couche liaison est constituée par un sous-ensemble de la norme HDLC : le protocole LAP-B,
- le niveau 3 de la norme X.25 qui est le niveau réseau.

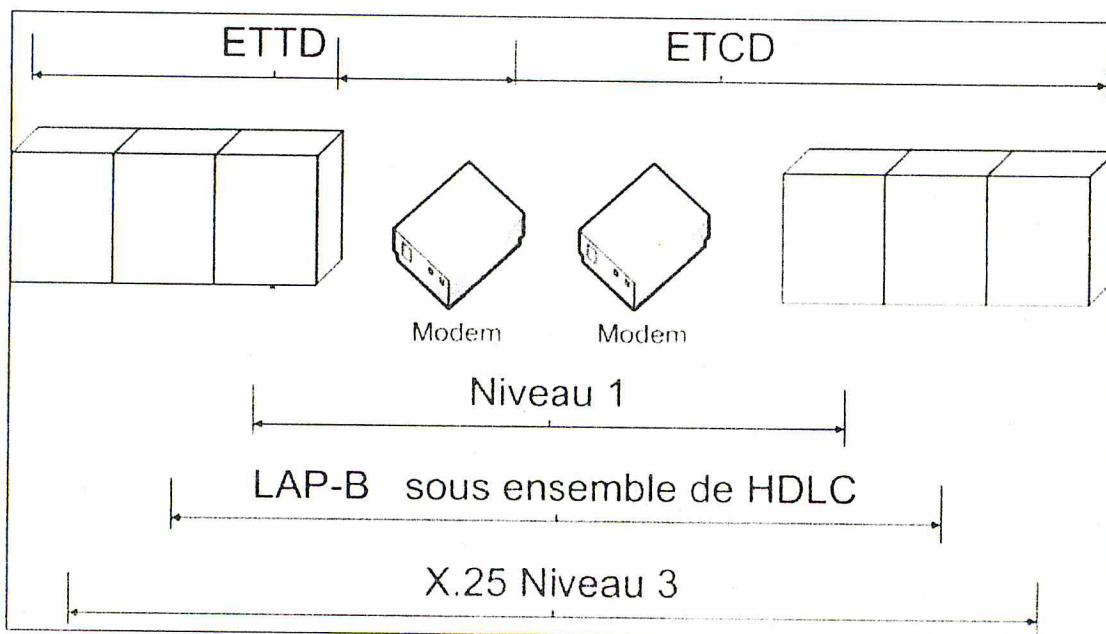


Figure I.2 Les niveaux du protocoles X.25



#### IV.1- Le niveau physique X25-1:

Le protocole X.25 au niveau 1 reprend l'avis X21 (interface numérique). La plus part des réseaux reprennent l'avis X21 (interface analogique), définissent une interface compatible avec la jonction V24, V28 et V35.

Le protocole X.25 niveau 1 spécifie :

- Les caractéristiques physiques de l'interface;
- Les caractéristiques électriques (X26 et X27);
- La transmission par bits de synchronisation;
- La procédure point à point en full duplex<sup>1</sup>.
- La procédure pour établir la liaison physique.

#### IV.2- Le niveau liaison X25-2 :

##### IV.2.1- Le protocol HDLC: (High Level Data Link Control):

C'est le premier protocole normalisé de niveau liaison : c'est un ensemble de classes de procédures et de fonctionnalités optionnelles (normalisé par l'ISO en 1976). La génération HDLC procède par anticipation : l'attente de l'acquittement n'empêche pas la transmission des trames suivantes.

Chaque liaison de données choisit sa procédure en fonction de ses besoins (coût, ressources...).

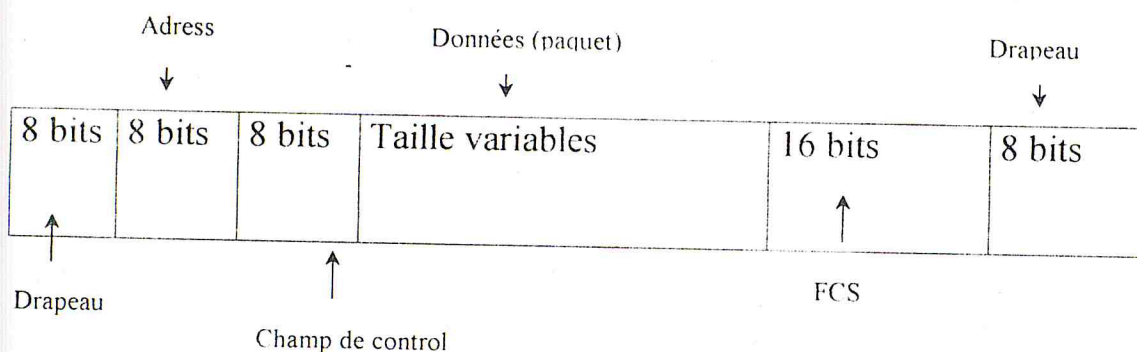


Figure I.3 Format standard d'une trame LAP-B

<sup>1</sup> Full duplex: Communication bidirectionnelle en même temps

- **Champ drapeau :**

La longueur du champ est d'un octet. Il est utilisé pour délimiter les trames, il apparaît au début et fin de chaque trame.

- **Champ adresse :**

Il sert à séparer les commandes des réponses et peut seulement prendre les valeurs 0x01 et 0x03. 01 désigne une commande de l'ETTD à l'ETCD et 03 contient une réponse de l'ETCD à l'ETTD.

- **Champ de contrôle :** Identifie le type de trame ; en plus, inclut la séquence de nombre, les fonctions de contrôles des erreurs en fonction du trame.
- **Champ séquence de contrôle de trame :** FCS (Frame Check Sequence) c'est un nombre codé sur 16 bits calculé à la fois par l'émetteur et le destinataire de la trame, il est utilisé pour s'assurer que l'intégrité de la trame n'est pas perturbée.
- **Champ de données :** Il est utilisé pour la transmission de données.

#### IV.2.2- Types de trames :

Il existe trois types de trames qui permettent de faire fonctionner le protocole :

- **La trame I (information) :** permet d'effectuer le transfert de l'information, chaque trame I contenant un numéro d'ordre N(S), un numéro d'ordre N(R), qui peut ou non accuser d'autres trames I à la station réceptrice.
- **La trame S (Supervision) :** Elle sert à réaliser les fonctions de commande de supervision de la liaison (la demande de retransmission ou suspension temporaire de transmission, l'accusé de réception...), chaque trame contenant un numéro d'ordre N(R), qui peut ou non accuser réception d'autres trames
- **La trame U (Unnumbered, non numérotées, ou encore trames de gestion) :** Elle est responsable de l'établissement et de la fermeture des connexions du niveau liaison. Elle permet aussi l'émission et la réception d'information en dehors du flux normal [1].

#### IV.3- Le niveau réseau X25-3 :

L'ETTD et l'ETCD partagent le même numéro de voie logique (plus tard, nous allons décrire la notion de voie logique) lors de la formation de la connexion. On profite de la mise en

place de la connexion pour réaliser un circuit virtuel, qui s'établit lors du routage du paquet d'appel. Ce circuit est emprunté par l'ensemble des paquets d'un même message. Dans ce cas, le numéro de voie logique que joue également le rôle d'une référence.

L'ouverture du circuit virtuel peut s'accompagner d'allocations des ressources pour assurer le contrôle de flux et garantir le séquençement des paquets dans un réseau.

#### IV.3.1- Notion de voie logique :

L'utilisateur appelant choisit dans son paquet d'appel un numéro de circuit virtuel parmi les numéros disponibles sur son tronçon, par contre, c'est le réseau qui choisit le numéro sur le tronçon de l'appel.

#### IV.3.2- Les circuits virtuels :

##### IV.3.2.1- Notion du circuit virtuel :

Les unités de donnée échangées à ce niveau sont les paquets. Chaque paquet est émis par un ETTD vers ETCD, l'élément du réseau est destiné à être acheminé à travers le réseau vers un autre ETTD grâce à un circuit qui s'appelle le « circuit virtuel ».

##### IV.3.2.2- Types de circuit virtuel :

On distingue deux types de circuit virtuel :

- **Les circuits virtuels commutés (CVC) :** qui sont des circuits établis à chaque communication sur demande d'un usager, Un ETTD peut, dans les limites de la capacité de sa liaison physique, établir plusieurs circuits virtuels simultanément.
- **Les circuits virtuels permanents (CVP) :** qui sont établis et maintenus sur le long terme (plusieurs mois), même si ceux-ci ne communiquent pas.

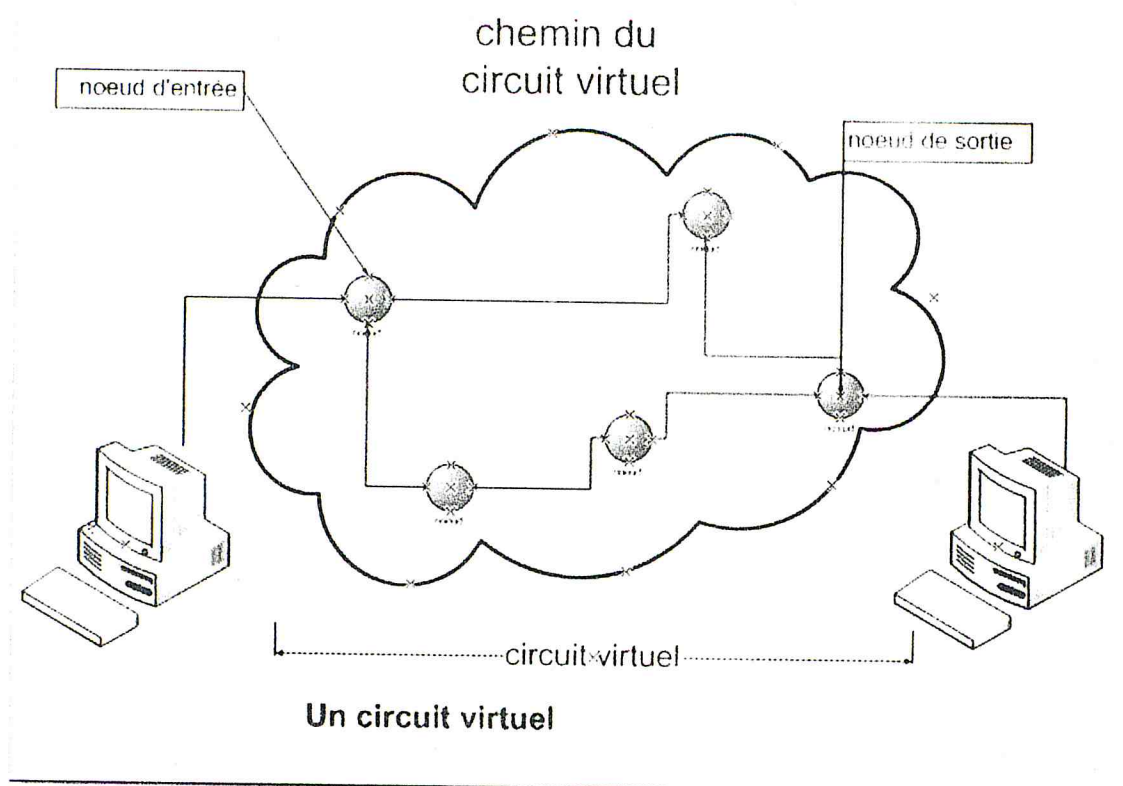


Figure I.4 Un circuit virtuel

#### IV.3.2.3- Paramètres du circuit virtuel :

Pour chaque sens de transmission, un circuit virtuel est caractérisé par :

- La classe de débit (bit/s) : elle peut être interprétée le nombre de paquets de données par seconde, ce paramètre ne peut pas dépasser la classe de débit maximale assurée par le réseau.
- Les paramètres de contrôle de flux : ils sont constitués par la taille et la longueur du paquet de données.

#### IV.3.3- La vie d'un circuit virtuel :

L'établissement d'un circuit virtuel est définie par l'ouverture et la fermeture de la connexion, qui se fait de par :

- l'envoi d'un paquet d'appel (call request) sur une voie logique (qui contient le numéro de la voie logique, obtenu par l'utilisateur, l'adresse réseau des abonnés ETDD des services complémentaires). A un instant donné, plusieurs CVC et CVP peuvent coexister,

chaque circuit virtuel utilise une voie logique repérée par un numéro de groupe logique et un numéro de voie.

Pour un CVC, les deux numéros sont attribués pendant la phase d'établissement de la communication, pour un CVP ils le sont lors de l'abonnement.

- Lorsque le paquet d'appel arrive à l'ETCD destinataire, celui-ci peut refuser la connexion en envoyant une demande de libération (clear request) ou accepter en envoyant un paquet de communication acceptée (call accepted).
- S'il ne répond pas après un certain délai (T1 de X25), le réseau entame une procédure de libération en émettant vers ETDD appelant un paquet d'indication de libération.
- Quand l'ETCD accepte, à ce moment le circuit virtuel est établi et les données peuvent être échangées, celles-ci empruntent alors toutes le même chemin marqué à travers le réseau par le paquet d'appel. Pour leur part la demande et la confirmation de libération sont traitées localement.
- L'émetteur ou le récepteur peut mettre fin au circuit virtuel en envoyant une demande de fermeture.

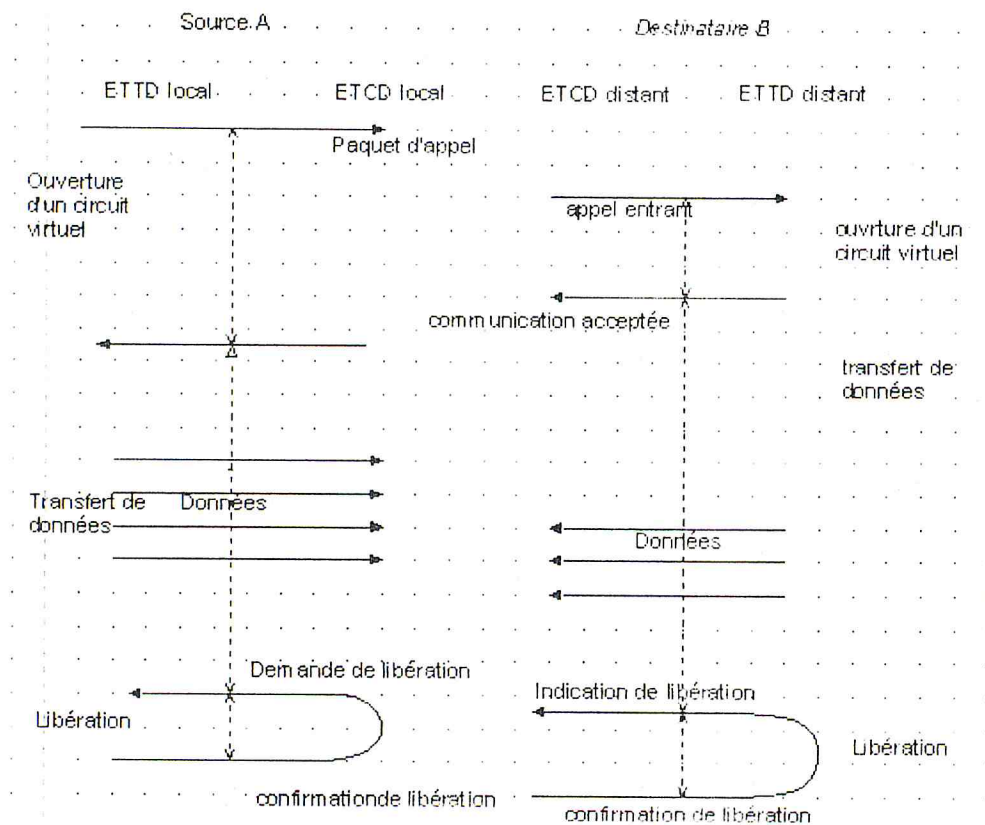


Figure 1.5 La vie d'un circuit virtuel (ouverture-libération)

- Les paquets de reprise ;
- Les paquets de diagnostic ;
- Les paquets d'enregistrement... etc.

Pour plus d'information consulter l'annexe 1, vous trouverez en détail les formats de chaque paquet.

#### IV.3.4.2- Fractionnement des paquets :

- Le protocole HDLC de la couche liaison fractionne les paquets en paquets de plus en plus petite taille.
- L'ETTD récepteur reconstitue le message d'origine en le rassemblant.

#### IV.3.4.3- Transfert des données :

La recommandation X25 exige que :

- Toutes les données d'usager soient transmises sans aucune altération par le réseau (Changement dans l'état).
- L'ordre des éléments binaires dans les paquets de données et l'interruption soit conservé.
- Pendant le transfert de données, un ETTD peut recevoir des paquets :
  - De données transportant des informations ;
  - D'interruption transportant également des informations ;
  - De contrôle de flux limitant la vitesse de transfert des paquets de données ;
  - De réinitialisation utilisés pour régler des différents cas d'erreurs ou d'incidents qui se produisent au niveau du réseau ou des équipements.

### V. Connexion des terminaux asynchrones : le triple X(x.3, x.28, x.29) :

#### V.1- PAD (Packet Assembler Desassembler):

La norme X25 définit un format de paquet ainsi que les contrôles d'entrée dans la couche réseau comme nous avons dit précédemment. Les réseaux publics acceptent les paquets suivant l'avis X25. Seuls les terminaux haut de gamme, synchrones, peuvent se permettre de supporter le protocole x25.

supporter le protocole X25, mais les terminaux asynchrones (sont incapables de fabriquer des paquets ils fonctionnent en mode caractère) ne peuvent pas connecter à un réseau public.

Pour remédier à ce problème il faut ajouter une boîte intermédiaire capable de les connecter qui s'appelle PAD (Packet Assembler Desassembler).

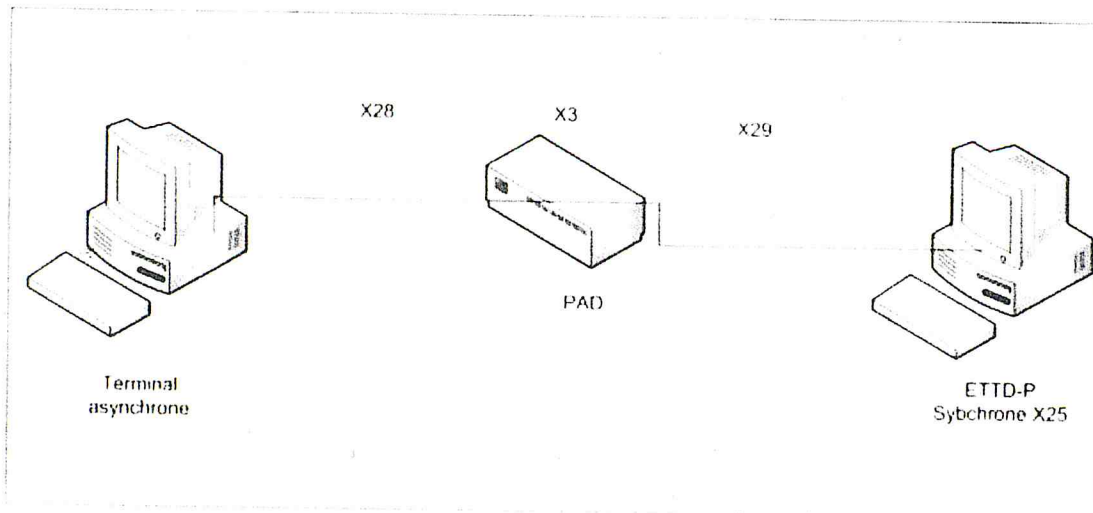


Figure I.6 fonctionnement du PAD [1]

Il faut que le réseau fournisse les fonctions qui leurs manquent, sous forme de PAD ou de PAVI<sup>2</sup> sont des points de concentration de données associés aux nœuds du réseau X25. Les terminaux se connectent à ces dispositifs soit par une liaison spécialisée soit par le RTC<sup>3</sup>.

Quand un appel est effectué via le RTC, le modem répond automatiquement du PAD qui assure la connexion et effectuent une détection automatique de vitesse du terminal appelant, puis la liaison passe à l'état de « ligne active ».

La fonction du PAD est d'assembler des caractères en paquets ou de désassembler des paquets en caractères:

- X.3 : précise les fonctions du PAD (spécifications de l'assemblage et du désassemblage de paquet).
- X.28 : langage de commande d'une liaison asynchrone (PAD-terminal).
- X.29 : décrit le protocole utilisé entre un PAD et un terminal mode paquet (l'ETTD, X25).

<sup>2</sup> PAVI : Point d'Accès Videotex

<sup>3</sup> RTC : Réseau téléphonique Commuté

## VI. Les services facultatifs de la norme X25

De nombreux services facultatifs sont spécifiés par la norme X25 :

- ❖ **Groupes fermés d'abonnés** : des utilisateurs peuvent former un groupe (sous-réseau privé). Ils peuvent communiquer entre eux, mais non à l'extérieur. Les abonnés n'appartenant pas au groupe fermé ne peuvent pas communiquer avec ceux qui y appartiennent.
- ❖ **Sélection des paramètres de contrôle de flux** : on peut sélectionner la longueur maximale des paquets, ainsi que les classes de débit.
- ❖ **Voie logique unidirectionnelle** : elle est appliquée pour éviter la collision de paquets d'appel.

### **Gestion de X 25 :**

Un opérateur de télécommunication, pour le même type de réseaux, peut disposer d'infrastructure mise en place par divers fournisseurs. Chaque fournisseur d'une partie du réseau peut avoir ses propres stations de supervision et ses propres formats des tickets de taxation. Un ticket de taxation émis par une station d'un fournisseur peut être différent du format d'un ticket émis par une station d'un autre fournisseur en terme de formats et de richesse d'informations.

Les informations sur une communication sont souvent reportées sous forme de ce qui a convenu d'appeler des tickets de taxation. Un ticket est un ensemble de données dans un format bien précis (défini par le concepteur du réseau de données). A titre d'exemple un ticket contiendra la référence de l'abonné ayant initié la communication, la référence de l'abonné appelé, la date et l'heure de l'établissement de la communication, la durée de la communication, le nombre d'octets ou kilo octets échangés etc..



---

*CHAPITRE II*  
*UML ET LA METHODE DE*  
*CONCEPTION*

## I UML (L'Unified Modeling Language):

L'Unified Modeling Language (UML) est un formalisme de modélisation orienté objet. Comme son nom l'indique, Il est issu de la fusion des trois méthodes de modélisation, tirant donc profit des avantages de chacune :

- méthode **Booch** de Grady Booch,
- méthode **OOSE** de Ivar Jacobson,
- méthode **OMT** de James Rumbaugh. [10].

UML est un langage graphique de modélisation objet permettant de spécifier, de construire, de visualiser et de décrire les détails d'un système logiciel.

Il comprend un vocabulaire et un ensemble de règles centré sur la représentation conceptuelle et physique du système.

UML supporte les concepts suivants:

- Des concepts (structurels, comportementaux, annotationnels, groupements).
- Des relations (association, généralisation, les agrégations, les compositions).
- Des diagrammes (statiques et dynamiques).

La section suivante a pour but de présenter le langage de modélisation de données UML:

### I.1 Les concepts:

UML supportent quatre (04) types de concepts : [11]

**Les concepts structurels** : représentés par les classes, les interfaces, les collaborations...

**Les concepts comportementaux** : représentés par les interactions et les états des objets.

**Les concepts annotationnels** : représentés par les notes.

**Les concepts de groupement**: Représentés par les sous-systèmes, les paquetages.

**Paquetages (Packages)** : Ils offrent un mécanisme générale pour la partition des modèles et le regroupement des éléments de modélisation. Il permet de définir des sous-systèmes formés d'éléments. [12]

Chaque package est représenté par un dossier, et doit avoir un nom différent de celui des autres packages, et peut être composé d'autres éléments, y compris d'autres packages.

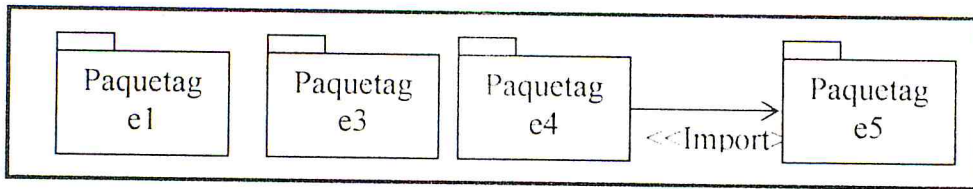


Figure II.1 Représentation graphique d'un package

**I.2. Les relations :** Elles permettent de relier les concepts entre eux.

Exemple : les associations, les généralisations, les agrégations, ... etc. [13].

**I.2.1. L'association:** Elle représente une relation précisant que les objets d'un élément sont reliés aux objets d'un autre élément.

**I.2.2. La généralisation:** Elle représente une relation entre un élément général et un élément dérivé de celui-ci, mais plus spécifique (désigné par sous-élément ou élément fils).

Le plus souvent, la relation de généralisation est utilisée pour représenter une relation d'héritage.

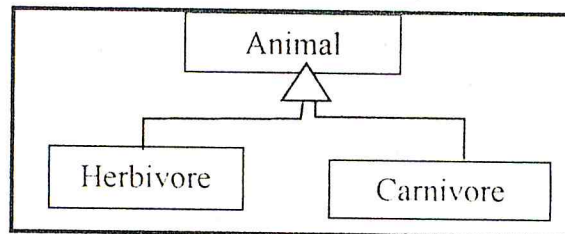


Figure II.2 Représentation graphique de la généralisation

**I.2.3. L'agrégation :** Elle représente une association non symétrique dans laquelle une extrémité joue un rôle prédominant par rapport à l'autre extrémité [6].

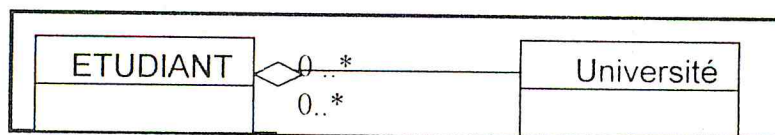


Figure II.3: Représentation graphique d'une agrégation

### I.3. Les diagrammes d'UML :

UML définit neuf types de diagrammes qui sont présentés dans l'extrait du méta modèle suivant :

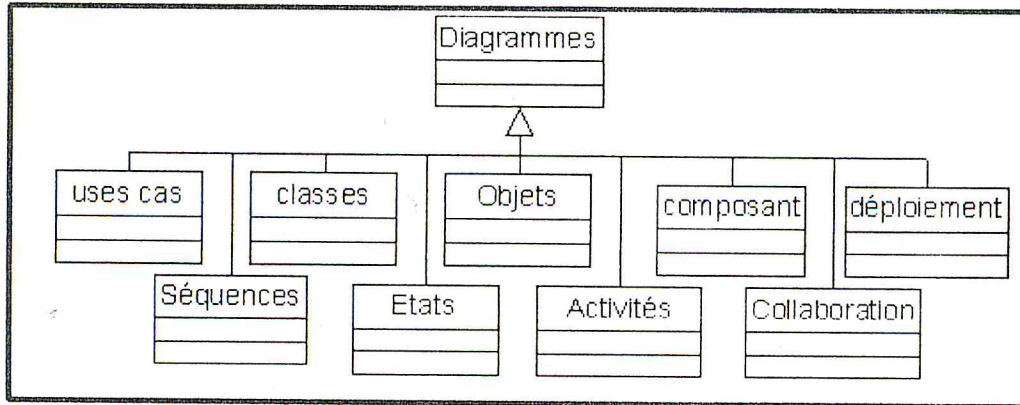


Figure II.4 : Les diagrammes d'UML [10]

#### I. 3.1 Diagrammes statiques:

##### I.3.1.1 diagramme de cas d'utilisation :

Un cas d'utilisation est une interaction effectuées par un acteur sur un système.

Un acteur peut représenter un rôle joué par une personne ou une logiciel qui interagit avec le système [12].

Un diagramme de cas d'utilisation permet de décrire les interactions entre les acteurs de l'organisation et le système dans chacun des cas d'utilisation envisagés. Il décrit le comportement d'un système au point de vue de l'utilisateur et fixe les limites du système et les relations entre le système et l'environnement [12].

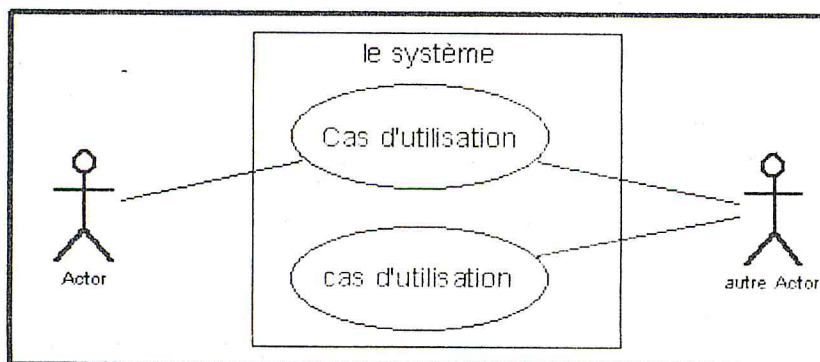


Figure II.5 : Représentation graphique de cas d'utilisation

### I.3.1.2 diagramme de classes :

Il montre la structure du système en terme des classes, des relations d'héritages entre classes, des associations, dont les agrégations, des attributs, des opérations et la spécification d'opérations et contraintes au niveau des entités [13].

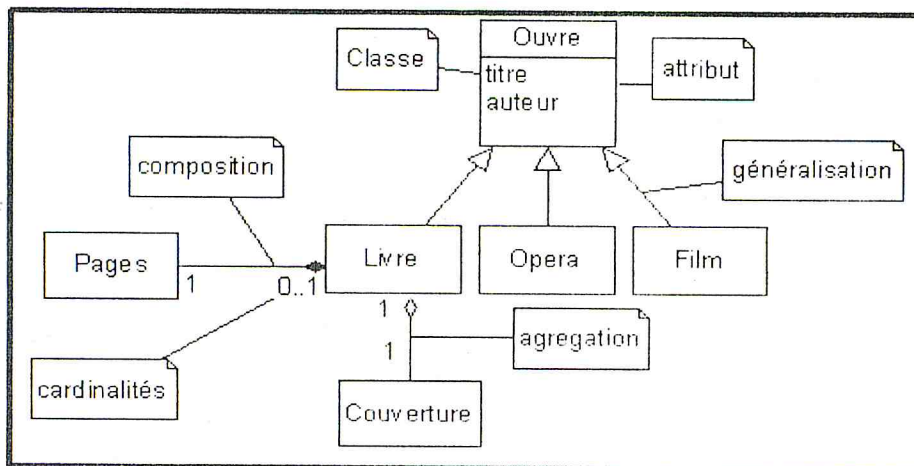


Figure II.6 Exemple de diagramme de classe

### I.3.1.3 Diagramme d'objets :

Ce type de diagramme UML montre des objets (instances de classes dans un état particulier) et des liens (relations sémantiques) entre ces objets [10].

Les diagrammes d'objets s'utilisent principalement pour montrer un contexte, par exemple avant ou après une interaction, mais également pour faciliter la compréhension des structures de données complexes [12].

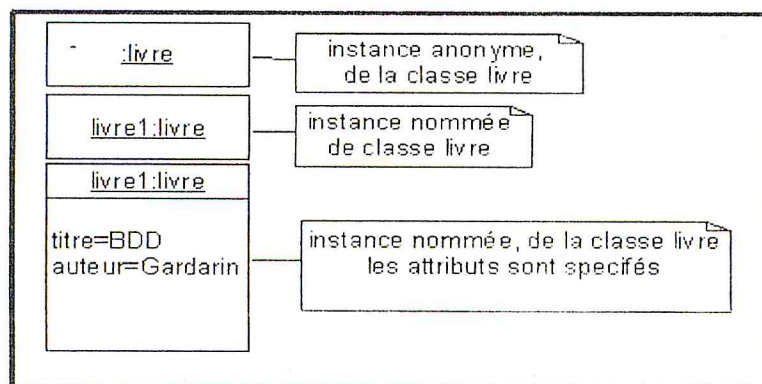


Figure II.7 Représentation graphique d'un objet

### I.3.1.4 Diagramme de composants :

Il permet de décrire l'architecture physique et statique d'une application en terme de modules : fichiers sources, librairie, exécutables etc. Ils décrivent les éléments physiques et leurs relations dans l'environnement de réalisation, ils montrent les choix de réalisation [12].

Ils montrent la mise en oeuvre physique des modèles de la vue logique avec l'environnement de développement.

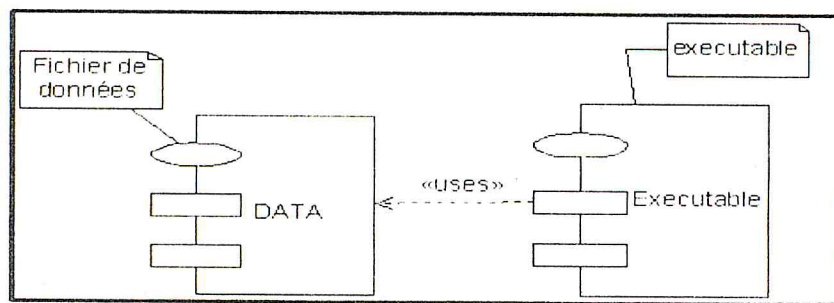


Figure II.8 Représentation d'un composant (fichier)

### I.3.1.5 Diagramme de déploiement :

Il montre la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels. Ils peuvent montrer des classes de nœuds ou des instances de nœuds [14].

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.

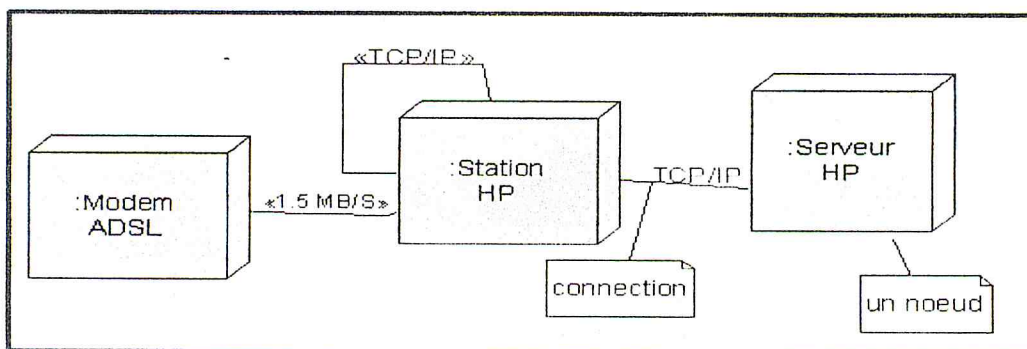


Figure II.9 Représentation de relations entre nœuds

### 1.3.2 Diagrammes dynamiques :

- **Interactions:** modélisent un comportement dynamique entre objets [6]. Elles se traduisent par l'envoi de messages entre objets.
- **Les messages :** Les messages échangés sont représentés au moyen de flèches horizontales partant de l'émetteur vers le récepteur. L'ordre de l'envoi est donné par la position sur l'axe vertical [15].

#### 1.3.2.1. Diagramme de séquences :

Un diagramme de séquence représente une interaction entre objets, en insistant sur la chronologie des envois de messages [13].

Ils peuvent être utilisés pour représenter les scénarios d'un cas d'utilisation donnée [16].

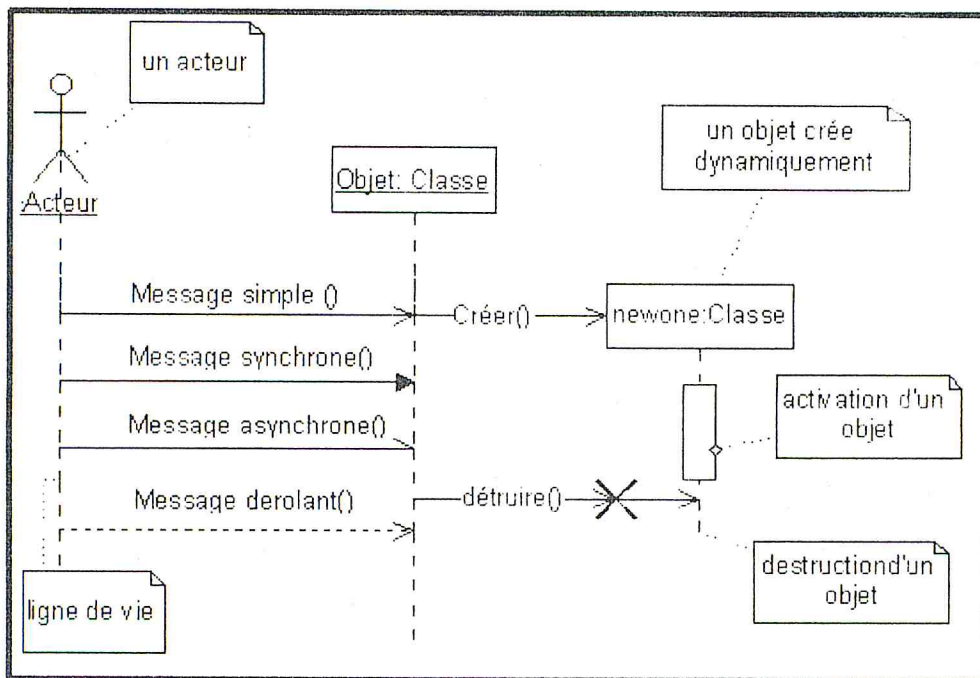


Figure II.10 Agencement de messages

**Période d'activation :** Correspond au temps pendant lequel un objet effectue une action, soit directement, soit par l'intermédiaire d'un autre objet.

### I.3.2.2. Diagrammes de collaboration :

Il montre des interactions entre les objets et les acteurs. Ils permettent de représenter le contexte d'une interaction, car on peut y préciser les états des objets qui interagissent et peuvent être utilisé pour identifier les principaux objets [16].

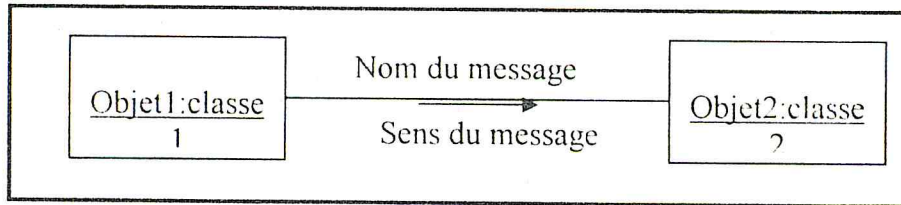
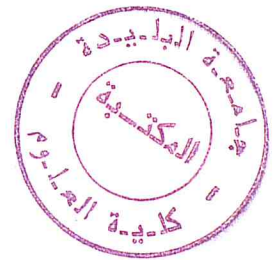


Figure II.11 Formalisme de base du diagramme de collaboration



### I.3.2.3 diagramme d'état-transition :

C'est un graphe constitué de nœuds représentant des états ainsi que des flèches représentant des transitions, portant des paramètres et des noms d'événements [10].

Les diagrammes d'états permettent de définir le comportement d'un objet particulier vis-à-vis des sollicitations internes ou externes auxquelles il peut être soumis [16]. Ils permettent aussi de décrire l'évolution dans le temps les états des objet d'une certaine classe, les événements auxquels ils réagissent et les transitions qu'ils effectuent.

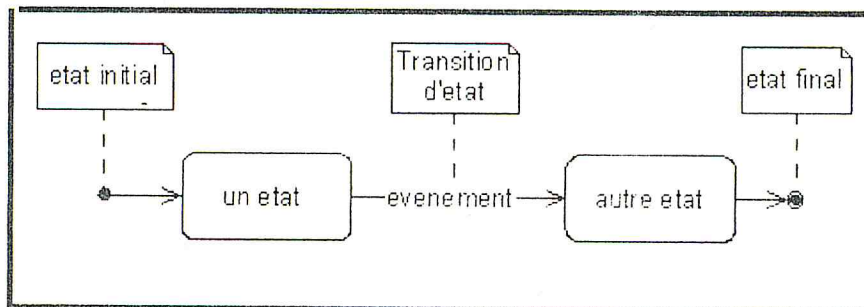


Figure II.12 Diagramme d'état-transition



#### I.3.2.4. Diagramme d'activité :

Il permet de décrire le déroulement d'un cas d'utilisation particulier. Il est possible de décrire les acteurs responsables de chaque activités par l'utilisation des «couloirs d'activités» qui permettent de répartir graphiquement les différentes activités entre les acteurs opérationnels [12]. Chaque activité est placée dans le «couloir» correspondant à l'acteur qui assume cette activité.

## II. Méthodologie de conception RUP :

Comme UML n'impose pas de méthode de travail particulière, il peut donc être intégré à n'importe quel processus de développement logiciel de manière transparente. UML est une sorte de boîte à outil, qui permet d'améliorer progressivement nos méthodes de travail. Intégrer UML par étapes dans un processus, de manière pratique, est tout a fait possible. Notre choix est basé sur un processus de modélisation qui s'appelle RUP (Rasional Unified Process). RUP est l'Unique implémentation commerciale d'UP (*Unified Process*), un processus théoriquement « standard » élaboré par Rationnel .

RUP est une évolution du processus *Objectory* initialement mis au point par la société Rationnel.

RUP est défini par ses créateurs comme un environnement d'exécution de processus génériques (*framework*) destinés à gérer l'ensemble des aspects d'un projet de développement de logiciel. RUP abrège des bonnes pratiques formalisées qui sont destinées aux différents acteurs d'un projet [5].

RUP est néanmoins décrit par ses auteurs comme un méta processus. C'est-à-dire qu'il est possible d'instancier RUP. En clair, cela signifie que RUP peut être adapté à tous les types de projets, pour tenir compte :

- De la taille du projet ;
- Du niveau de compétence initiale des acteurs (par exemple dans le cas où l'entreprise n'utiliserait pas de processus de développement).

### Les fondements de RUP:

RUP est, pour citer ses auteurs :

« Piloté par les cas d'utilisation, centré sur l'architecture, itératif et incrémental. » [5]

#### Piloté par les cas d'utilisation :

Les cas d'utilisation, dans RUP, ne servent pas qu'à améliorer la capture des exigences fonctionnelles des utilisateurs. Ils servent d'intrant aux différentes activités du processus : analyse, conception, implémentation, rédaction des tests et déploiement.

Le modèle de conception est développé à partir des cas d'utilisation ; le modèle d'implémentation reprend le modèle de conception et définit des composants qui regroupent les classes en unités d'implémentations indépendantes.. Chaque modèle est respectivement d'un niveau d'abstraction inférieur à celui dont il descend. C'est-à-dire que le modèle d'implémentation est plus proche de la structure du code source finalement produit que le modèle de conception. RUP contient les guides qui permettent aux développeurs de se familiariser avec ces concepts. Mais rien ne remplace la pratique et l'expérience accumulée en projet.

#### II.1. La vue logique :

Cette vue de haut niveau se concentre sur l'abstraction et l'encapsulation, elle modélise les éléments et mécanismes principaux du système ; Elle identifie les éléments du domaine, ainsi que les relations et interactions entre ces éléments. Le système est décomposé en ensemble d'abstractions principales, pris (la plupart du temps) du domaine de problème, sous forme *d'objets* ou *de classes d'objet* Ils exploitent les principes de l'abstraction.

Nous employons l'approche d'UML comme nous avons dit précédemment pour représenter l'architecture logique, au moyen de diagrammes de classe et diagrammes d'objets.

#### II.2. La vue de processus «la vue dynamique »:

Cette vue est très importante dans les environnements multitâches ; elle montre : [6], [7]

- La décomposition du système en terme de processus (tâches).
- Les interactions entre les processus (leur communication).
- La synchronisation et la communication des activités parallèles.

Cette vue est basé sur les diagrammes dynamiques d'UML.

### II.3. La vue des composants : « la Vue de Développement »

Elle se concentre sur l'organisation réelle de logiciel, cette vue est de bas niveau (aussi appelée "vue de réalisation"), montre : [6], [7]

- L'allocation des éléments de modélisation dans des modules (fichiers sources, bibliothèques dynamiques, bases de données, exécutables, etc...).
- L'organisation des composants, c'est-à-dire la distribution du code en gestion de configuration, les dépendances entre les composants...
- Les contraintes de développement (bibliothèques externes...).

En d'autres termes, cette vue identifie les modules qui réalisent (physiquement) les classes de la vue logique. Elle est représentée par des diagrammes de composants dans la notation d'UML.

### II.4 La vue physique : « la vue de déploiement »

Cette vue très importante dans les environnements distribués, décrit les ressources matérielles et la répartition du logiciel dans ces ressources : [6], [7]

- La disposition et nature physique des matériels, ainsi que leurs performances.
- L'implantation des modules principaux sur les nœuds du réseau.

### II.5 La vue des cas d'utilisation :

La description d'une architecture peut être organisée autour de ces quatre vues, et alors illustré par les cas d'utilisation ou *scénarios* qui deviennent une cinquième vue. [6], [7]  
Cette vue définit les besoins des clients du système et centre la définition de l'architecture du système sur la satisfaction (la réalisation) de ces besoins

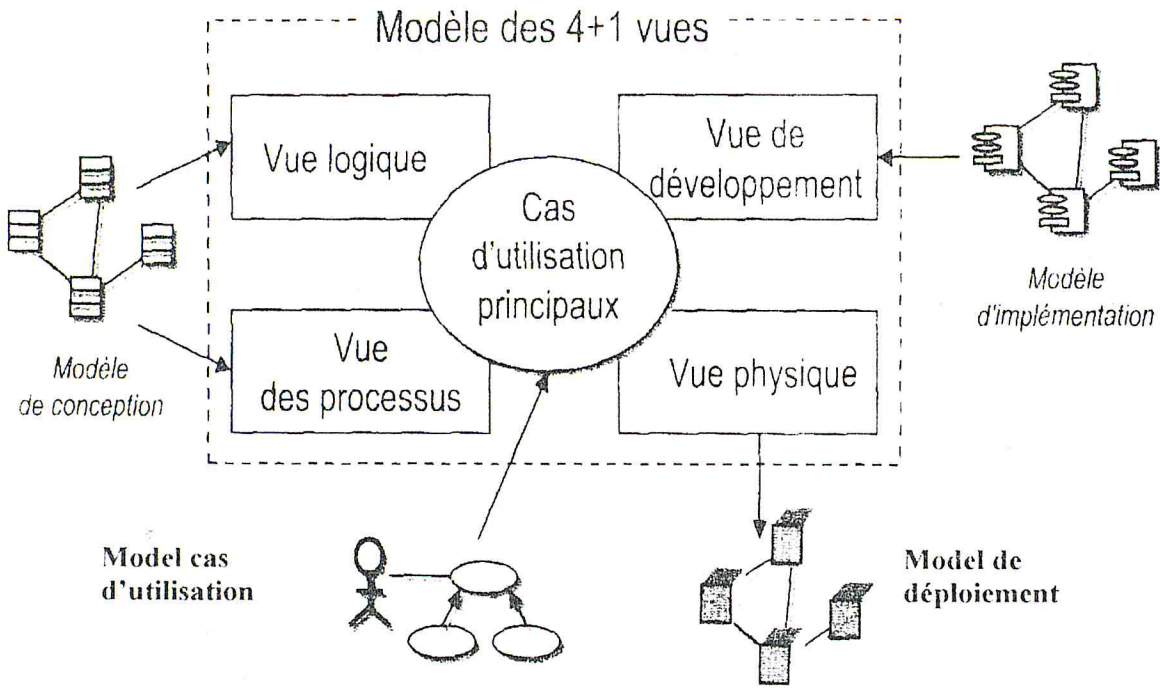


Figure II.13 Modèle 4+1. [7]

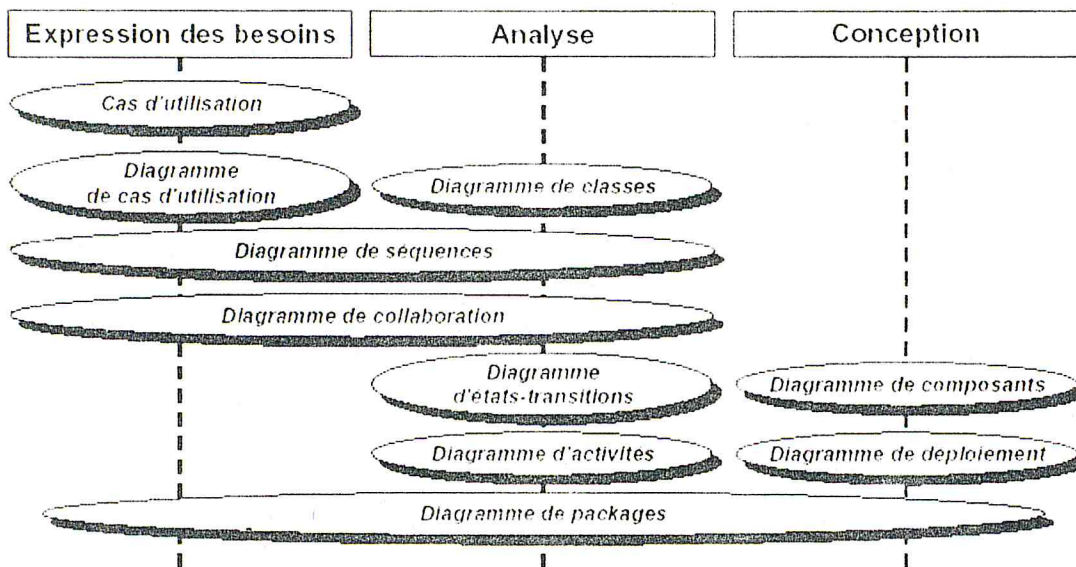


Figure II.15 : Diagrammes d'UML et architecture du logiciel [17]

### **III. Conclusion :**

Dans la pratique nous avons suivi la méthode de modélisation RUP<sup>1</sup> qui utilise le model 4+1 méthodes itératives et incrémentales. Pour présenter notre système on a choisi le langage UML. Pour modéliser et concevoir notre système on a utilisé cinq vues représentées dans le Chapitre suivant.

---

<sup>1</sup> RUP: Rational Unified Process

---

*CHAPITRE III*  
*ARCHITECTURE*  
*DU SYSTEME*

## **I Introduction:**

Dans ce chapitre nous allons présenter une vue d'ensemble des composants logiciels de notre système, nous avons utilisé une notation informelle pour permettre aux lecteurs de mieux comprendre notre architecture.

## **II Description globale du système :**

### **II.1 Préparation de flux :**

- 1- Acquisition : Les tickets émis par les réseaux X25 sont récupérés sur le centre de facturation.
- 2- Préparation : Ce sous-système a pour but d'uniformiser le format de tous les tickets et de ne garder que les champs utiles au centre de facturation.

### **II.2 Valorisation et Facturation :**

#### 1- Valorisation :

- 1-1 Valorisation le coût unitaire : Le but de ce sous système est de déterminer le coût unitaire des tickets de taxation
- 1-2 Valorisation de produit : Elle prend en compte les caractéristiques des accès du client pour le sous système suivant.

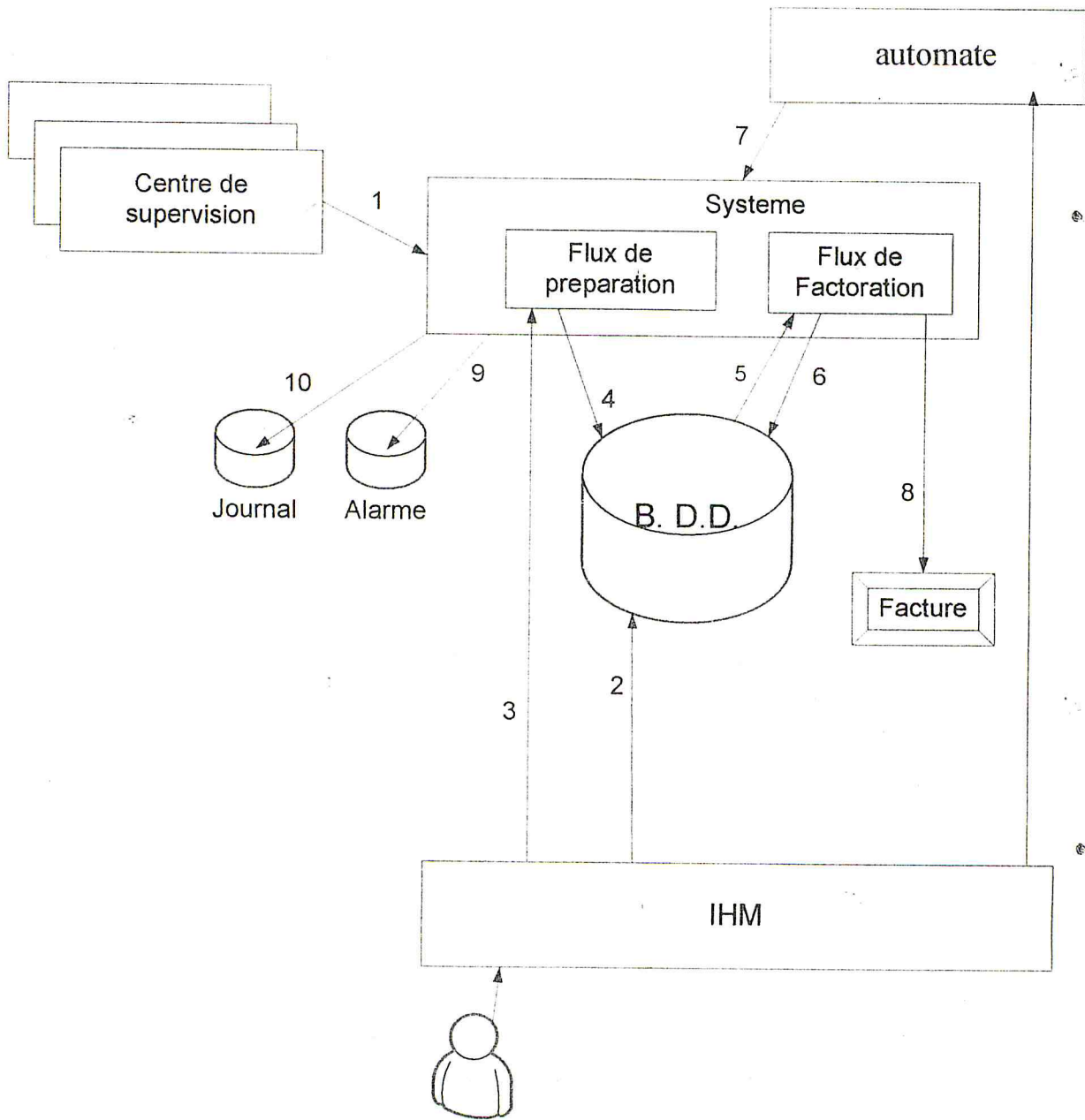
#### 2- Facturation : Obtenir dans le détail les coûts de toutes les communications nationales effectuées sur le réseau de l'opérateur.

- 2-1 Calcul de la facture,
- 2-2 Edition.

## **III Interface Homme Machine :**

C'est l'interaction entre l'utilisateur et le système pour l'échange des données.

**L'architecture globale du système :**



Architecture globale du système:

Figure III.1 Architecture globale du système



**Description :**

- 1 : Récupération des tickets.
- 2, 4,6 : Stockage des données.
- 3 : Intervention de l'utilisateur.
- 5 : Récupération des données.
- 7 : Contrôle automatique via le moniteur d'exploitation.
- 8 : Facturation.
- 9 : Les Alarme du système.
- 10 : Le journal du système.

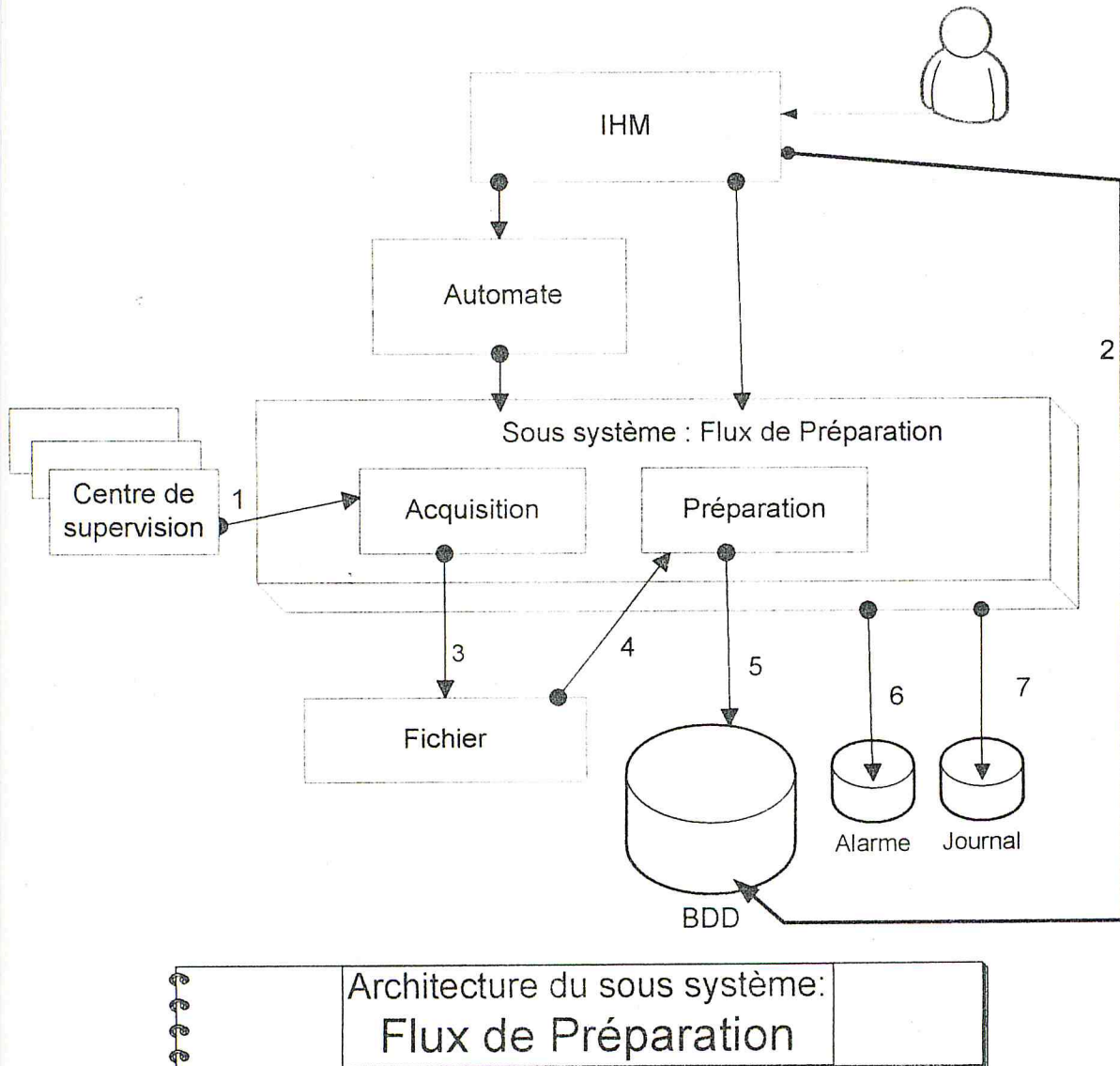
**III Indépendance entre les sous systèmes :**

Les sous systèmes fonctionnent de manière indépendante. Il est possible de démarrer ou d'arrêter un sous système pour effectuer des opérations de maintenance.

**IV Description détaillée du système :**

**I Préparation de flux :**

L'architecture du sous système Préparation de flux :



**Figure III.2** Architecture du Flux de Préparation

**Description :**

- 1 : Récupération des tickets du centre de supervision via le protocole FTP.
- 2 : Saisie les données (Clients, Abonnement, les Tarifs...).
- 3 : Copier les fichiers des tickets dans un fichier interne.
- 4 : Récupération des tickets
- 5 : Stockage dans B.D.D. les tickets harmonisés.
- 6 : les alarmes de sous système.
- 7 : Journal de sous système.

### **Les paramètres :**

Les pré-requis se décomposent en deux parties : les paramètres à valoriser dans la base de données par l'intermédiaire de l'IHM, par exemple (le nombre d'essais maximum de transfert FTP), et les paramètres dits systèmes, contenus dans un fichier d'initialisation d'acquisition, ces paramètres seront initialisés par l'administrateur système, par exemple (le login et le mot de passe utilisé par CFX25 pour lancer la commande FTP sur la station principale (le mot de passe sera saisi par l'administrateur système puis crypté à l'aide d'une fonction qui décrite dans le système ).

### **Le fonctionnement :**

La récupération se fait via des stations de supervision (SVR) (généralement deux). Les deux stations de supervision reçoivent, simultanément, et au fil de l'eau, les tickets de taxation émis par les commutateurs X25.

En fonctionnement normal, les deux stations SVR sont actives, mais l'opérateur du centre de facturation choisit l'une d'elle pour faire la récupération des tickets et l'utiliser systématiquement lors de l'exécution du sous système d'acquisition. Cette station est choisie lors de la phase de paramétrage. On appellera cette station la station «principale», et l'autre station celle de «secours».

La récupération des tickets est réalisée à la demande du Centre de Facturation. La fréquence de fonctionnement de ce processus est paramétrable lors de l'initialisation des paramètres. Ce paramétrage est saisi par l'administrateur et fait partie des paramètres systèmes. Le lancement automatique de l'acquisition peut avoir une périodicité de quelques heures.

## **I.1 Sous système d'Acquisition :**

### **I.1.1 L'Objectif du sous système :**

Les tickets émis par les réseaux X25 sont récupérés sur le centre de facturation par la phase d'acquisition. Ce sous-système est détaillé dans cette section.

Ce sous système est décomposé en trois autres sous-systèmes.

- ◆ Récupération des fichiers de tickets de taxation.
- ◆ Vérification de l'intégralité du transport.
- ◆ Effacement des fichiers tickets de taxation récupérés sur les stations de supervision.

### **I.1.2 Les données :**

- Les données en entrée de ce sous système est un fichier des tickets récupéré et copié dans un répertoire (paramétré par l'administrateur du système).
- Le sous système suivant (préparation) récupère tous les fichiers de ce répertoire, pour les fusionner et les recopier dans un fichier (unique) dans son répertoire de travail.

### **I.1.3 Gestion des alarmes :**

Si une erreur de connexion entre le Centre de Facturation et la station de l'opérateur de télécommunication est détectée, on recommence l'opération un nombre de fois (paramètre saisi par l'administrateur) avant de créer une alarme.

Dans le cas où le système de Facturation constaterait un défaut de fonctionnement sur la station principale, le système inverse le rôle des stations (la principale devient station de secours), et envoie une alarme matérielle. La première station restera station de secours jusqu'à l'apparition d'un autre défaut sur la nouvelle station principale.

## **I.2 sous système de Préparation :**

### **I.2.1 L'Objectif du sous système :**

Ce sous-système a pour but d'uniformiser le format de tous les tickets et de ne garder que les champs utiles au centre de facturation. Ce sous-système est détaillé dans ce document.

Chaque ticket de taxation passe par toutes les phases fusion, control, décodage, simplification, harmonisation et stockage

Le sous-système préparation s'appuie sur les grammaires de descriptions de format général.

- La phase de contrôle vérifie la structure syntaxique des tickets de taxation.
- La phase de décodage, simplification, harmonisation transforme les champs codés utiles à l'harmonisation en ASCII, applique les règles d'harmonisation et de simplification et génère un ticket de taxation de format unique lisible par le sous-système valorisation suivant.
- La phase de fusion sert à fusionner les fichiers harmonisés déjà existés avec sel qui ont harmonisés récemment.

**I.2.2 Les données :**

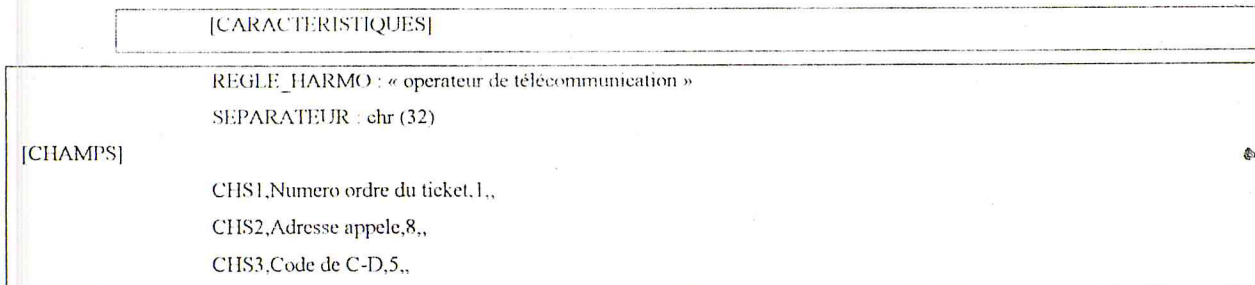
- Les données en entrée du sous-système de préparation sont les fichiers de tickets de taxation acquis par la phase d'acquisition.
- Les données en sortie sont un fichier unique de tickets de taxation harmonisés et avec un indicateur d'origine de l'opérateur pour chaque Ticket de taxation, qui sera enregistré dans la base de données.

**I.2.3 Les grammaires :**

- ◆ Une grammaire est définie pour chaque format de ticket de taxation. Chaque évolution du format implique une modification de la grammaire.

Exemple de grammaire liée au ticket au format variable :

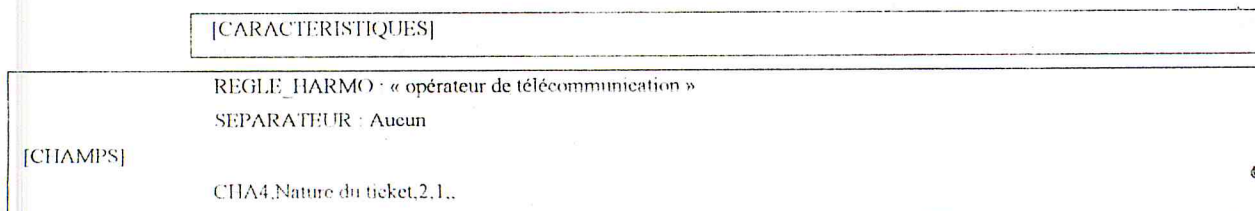
Les champs ont des caractéristiques suivantes: le code du champ défini pour le centre de facturation, le nom du champ, la position du champ (par rapport au séparateur dans ce cas là) et la longueur du champ.



**Figure III.3** Exemple de grammaire liée au ticket au format variable

Exemple de grammaire liée au ticket de format fixe :

Les champs ont des caractéristiques suivantes : le code du champ défini pour le centre de facturation, le nom du champ, la position du champ (en octet), et la longueur du champ.



**Figure III.4** Exemple de grammaire liée au ticket de format fixe

Un champ utile est un champ servant à l'harmonisation directement ou indirectement. Par exemple pour remplir le champ (01) du ticket harmonisé à partir d'un ticket sortant du système d'acquisition on a besoin du champ « Date d'établissement de la connexion du ticket »

Les champs utiles sont décrits dans les tableaux ci-dessous. Leur position dans les tickets d'origine est définie dans les grammaires.

#### I.2.4 Structure du ticket harmonisé :

Champ du ticket	Libellé du champ	Valeurs et Commentaires
_01	Date de début du ticket	AAAAMMJJ pour année, mois jour
_02	Heure de début du ticket	HHmmss pour heure, minute seconde
_03	Numéro d'ordre du champ	000 Champ rempli par la phase de préparation
_04	Marqueur d'origine	Origine opérateur de télécommunication
_05	Type de CV( voir l'annexe)	1 pour CVC 2 pour CVP 3 pour CVC secours 4 pour CVC passerelle
_06	Nature du ticket	Cette nature est le ticket para port au fichier (début, fin ....)
_07	Date de début de connexion	AAAAMMJJ pour année, mois jour
_08	Heure de début de connexion	HHmmss pour heure, minute seconde
_09	Durée du ticket	00000 5 digits en minutes (arrondi supérieur)
_10	Adresse de l'appelant	00000000000000 9 digits représentant le numéro du réseau + 0 à 6 digits maximum représentant l'adresse du sous-réseau.
_11	Adresse de l'appelé	000000000000000 15 digit maximum ( pouvant contenir l'indicatif international et l'indicatif X121 des pays)
_12	Destination	0 pour national 1 pour le groupe 1 2 pour le groupe 2 3 pour le groupe 3 4 pour le groupe 4

_13	Volume échangé	00000000 8 digits en koctets
-----	----------------	------------------------------

**Tableau III.1** Structure du ticket harmonisé

◆ **L'origine du ticket (marqueur d'origine)**

L'origine du ticket peut être :

- Nationale.
- internationale groupe 1,
- internationale groupe 2,
- internationale groupe 3,
- internationale groupe 4.

Un groupe représente un ensemble de pays. Un groupe est défini par au moins un pays. Il n'y a pas d'intersection entre les groupes. C'est à dire qu'un pays ne peut appartenir qu'à un groupe et un seul.

Les propriétés d'un groupe sont le tarif pour un koctet et le volume minimum facturé par communication aboutie.

On détermine l'origine du ticket par l'intermédiaire des champs « *Marqueur d'origine* » et « *Adresse de l'appelé* » du ticket harmonisé.

❖ **Type d'appel**

On détermine si l'appel est en PCV (Voir chapitre I)

Appel direct : ligne spécialisée.

Appel indirect : ligne téléphonique (RTC).

**I.2.5 Stockage des tickets harmonisés :**

Les tickets harmonisés sont stockés dans la base de données. L'ordre des champs lors du stockage est le suivant et il n'y a pas de séparateur de champ. La taille de ticket est fixe (la longueur de chaque champ est fixée).

Champ du ticket	Libellé du champ	Longueur en octet
_01	Date de début du ticket	8
_02	Heure de début du ticket	6
_03	Numéro d'ordre du champ	3
_04	Marqueur d'origine	1
_05	Type de CV	2
_06	Nature du ticket	2
_07	Date de début de connexion	8
_08	Heure de début de connexion	6
_09	Durée du ticket	5
_12	Adresse de l'appelant	15
_13	Adresse de l'appelé	15
_14	Destination	3
_15	Volume échangé	8
<b>TOTAL</b>		<b>82</b>

Tableau III.2 Stockage des tickets harmonisés

### I.2.6 Gestion des Alarmes:

En cas de différence, en ce qui concerne les tickets du premier cas les tickets de taxation est rejetés tels quels dans un fichier dédié. Une alarme (rajout du code alarme) est générée dans ce cas là. En ce qui concerne le deuxième cas, si le fichier d'entrée à une taille incorrecte une alarme est générée.

➤ Les alarmes sont générées (stockées en base) suite à des incidents survenus pendant l'étape de Préparation :

- Mauvaise récupération des variables d'environnement.
- Mauvais format des grammaires.
  - Problème de récupération des fichiers de TT (pour indiquer qu'il n'y a pas de fichiers à traiter).

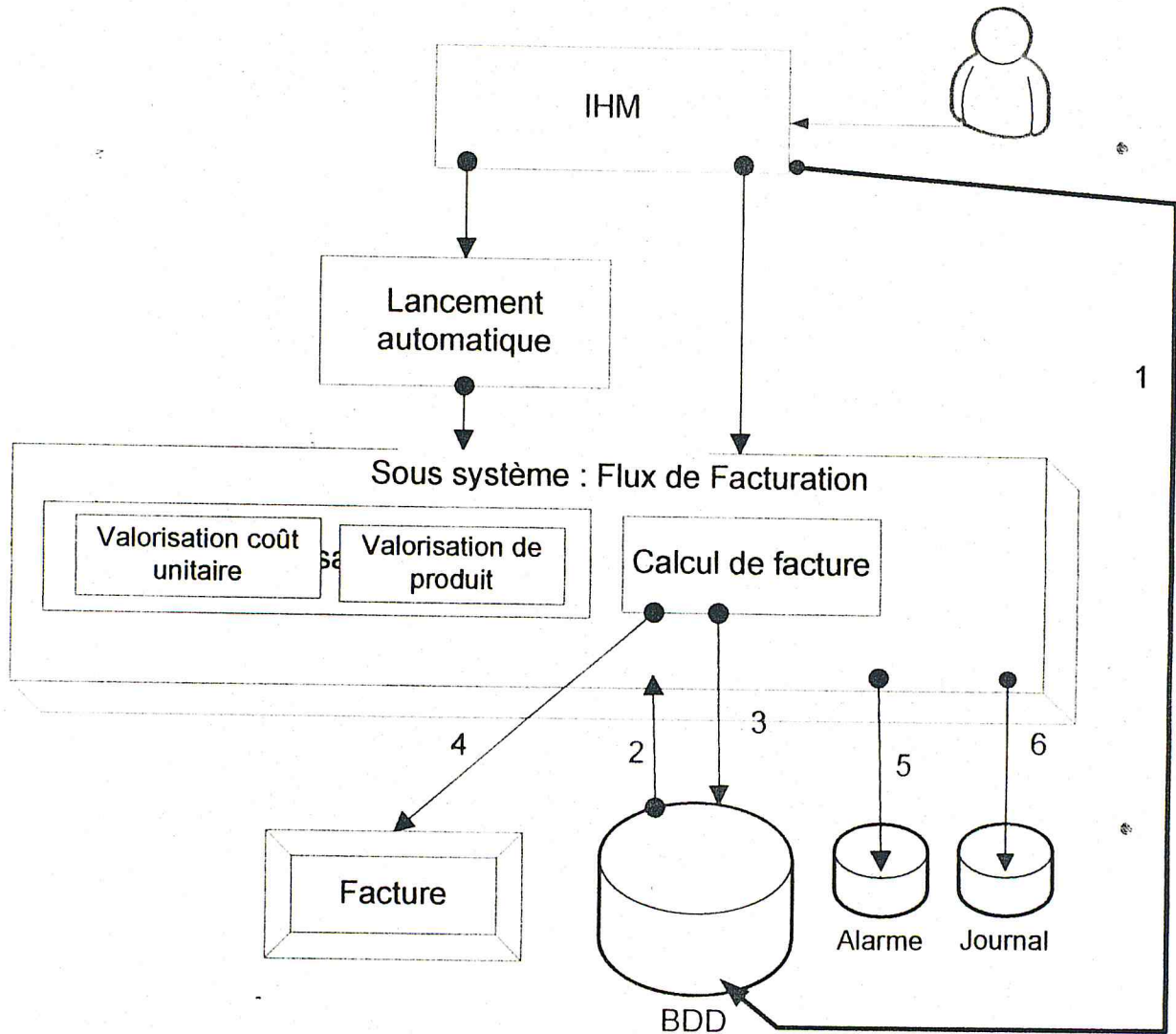


**II Flux du Facturation :**

**L'Objectif du sous système :**

Obtenir dans le détail les coûts de toutes les communications nationales effectuées sur le réseau de l'opérateur dans le but de pouvoir facturer par la suite les abonnés (clients) concernés du réseau X25.

**L'architecture du sous système flux de facturation :**



Architecture du sous système:  
Flux de Facturation

Figure III.5 Architecture de Flux de Facturation

**Description :**

- 1 : Saisie les données.
- 2 : Récupération des données.
- 3 : Stockage des données
- 4 : Edition des factures.
- 5 : Les Alarme du système.
- 6 : Le journal du système.
- 7 : Les données pour le système de comptabilité.
- 8 : Les données pour le suivi de paiement.

**II.1 sous système : Valorisation du coût unitaire:**

Le but de ce sous système est de déterminer le coût unitaire des tickets de taxation. Ce coût dépend des valeurs du ticket comme la durée et le volume échangé lors de la communication, de la destination de l'appel (groupe du ticket), ainsi que des paramètres du plan tarifaire comme les tarifs à la minute, les tarifs au volume, les tranches horaires, le type de tarification de l'accès, les réductions appliquées à l'abonnement et au client.

**II.1.1 Enrichissement des tickets de taxation :**

La phase d'enrichissement analyse le ticket de taxation avec vérification des contenus des champs remplis par le sous système précédent (la préparation).

Un ticket de taxation est émis et enrichi en fonction des données qu'il contient, des données client et des données du plan tarifaire, de l'origine du ticket, et du type d'appel. pour en déduire le champ destination et le volume minimum facturée.

**II.1.2 Valorisation des tickets de taxation :**

Le volume échangé durant la connexion et la durée de la connexion sont analysés pour déterminer les coûts unitaires du ticket.

### **Éléments valorisés :**

La phase de valorisation génère des données servant à la facturation des communications. Ces données sont :

- La mesure du volume échangés pendant la communication en fonction du temps passé sur chaque tranche horaire,
- la durée totale de la communication,
- le coût réel pour le volume sur chaque tranche tarifaire, tenant compte du type de tarification de l'accès (forfait complet, forfait durée, réel)
- le coût pour le volume de la facture détaillée,
- le coût pour la durée de la facture détaillée.

### **II.2 Sous système de Valorisation de produit :**

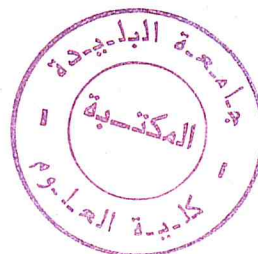
La valorisation de produit est l'étape précédente de la facturation. Elle est indépendante de la valorisation coût unitaire et donc des tickets de taxation. Elle prend en compte les caractéristiques des accès du client. Elle détermine notamment s'il faut éditer une facture pour un client donné et quelle est la période prise en compte pour la facturation. Elle prépare les données pour la facturation des accès des clients à facturer.

- Il faut lancer le calcul de la valorisation de produit uniquement pour les clients :
  - dont l'état est actif ou dont l'état est inactif et la date de résiliation de l'un des abonnements est postérieure ou égale à la date de la dernière facture car le jour de facturation est exclu à la date de dernière facture,
  - ayant au moins un abonnement,

#### **II.2.1 Les données en entrée :**

La date de lancement du calcul de la valorisation de produit dépend des caractéristiques du client:

- périodicité de la facturation (bimestre, 6 mois, ...), jour de facturation, état du client, date de dernière facture.



### **La période de la facture :**

La période de calcul de la valorisation de produit = l'intervalle entre la date de facturation actuelle – nombre de mois défini par la périodicité (inclus) et la date de facturation actuelle (exclus).

Exemple :

⇒ périodicité facturation = « B » (bimensuel)

Date de facturation actuelle = 10/08/2004

La période de calcul de la valorisation de produit est :

[10/06/2004, 10/08/2004[

### **II.3 Sous système de Facturation:**

Le calcul de la facturation se base sur le résultat des autres sous systèmes :

- le résultat de la valorisation de produit (valorisation des accès),
- le résultat de la valorisation de cout unitaire (valorisation des communications),
- les données accordées au client, abonnement et les tarifs.

Ce sous système est composé de:

- 1- **Calcul des valeurs:** en fonction des données clients (facturation au forfait, réduction, location de modem, vitesse de la ligne, facture détaillée ...) et des Tickets.
- 2- **Edition des factures:** en fonction des données calculées et du format paramétré.

### **III Interface Homme Machine :**

C'est l'interaction entre l'utilisateur et le système pour l'échange des données.

- Paramétrage Système;
- Gestion des utilisateurs;
- Gestion Stations d'acquisition;
- Gestion Clients;
- Gestion abonnements;
- Gestion des Adresses;
- Gestion des plans tarifaires;
- Gestion des tranches horaires et type des jours;

- Gestion Services Complémentaires;
- Gestion Alarmes;
- Journal des connexions;
- Visualisation de la base de consultation.
- Suivi de paiement : Combien le client doit à l'opérateur, les dates de paiement, le lieu de paiement, la somme payée etc.. Le paiement effectif pourra aussi se faire selon des politiques dans lesquelles seront prises en comptes certainement spécificités du client, de l'abonnement, de la consommation.. A titre d'exemple une politique de paiement par tranche pourrait être appliqué ne pouvant pas payer d'un seul coup sa redevance.

#### **IV Les données du système :**

##### **IV.1 Données en entrées:**

- ▶ Saisie du paramétrage du système ;
- ▶ Les Tickets de Taxation ;
- ▶ Saisie des données clients, abonnement par IHM ;
- ▶ Saisie des données de tarification :
  - Tarifs,
  - Plages horaires,
  - Calendrier
- ▶ Saisie les données de Paiement.
- ▶ Les autres données en entrée correspondent à la saisie des services ou des divers données de configuration.

##### **IV.2 Données en sortie :**

Les données en sortie du système sont :

- ▶ Factures des clients Globales ;
- ▶ Factures des clients détaillées ;
- ▶ Visualisation des informations des tables (clients, abonnement, les plans tarifaires... etc )
- ▶ Mise à la disposition de la comptabilité, d'informations sur le montant des factures et les clients ;
- ▶ Le journal du système et les alarmes enregistrées.

**Conclusion :**

Dans se présent chapitre nous avons décrit une architecture informelle et décomposer le système en composants plus détaillée ainsi que nous avons décrit les entrés et les sorties du système.

L'avantage de cette architecture est de permettre au lecteur qui a une simple connaissance de mieux comprendre le système.

---

***CHAPITRE IV***  
***LA CONCEPTION***  
***ET LA REALISATION***

## **Introduction :**

La méthode que nous avons utilisés permet de couvrir le cycle de vie d'un logiciel depuis l'analyse du besoin jusqu'au test. Cette notation est d'une très grande richesse. Elle piloté par les cas d'utilisation, centré sur l'architecture, itératif et incrémental [5].

Elle permet de couvrir toutes les phases du développement :

- Les besoins des utilisateurs du futur système, exprimés à l'aide de cas d'utilisation.
- La spécification complète du système, sous forme de diagrammes couvrant les parties statiques et dynamiques du système.
- La conception détaillée, jusqu'à un niveau très proche du code en langage objet de l'implémentation.
- Les suites de tests permettant de s'assurer qu'une implémentation candidate est effectivement conforme à la spécification élaborée.

## **I La Vue logique (Diagramme de cas d'utilisation):**

Le but de cette vue est de comprendre et structurer les besoins du client. Il faut d'abord 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.

Les besoins des clients sont des éléments de traçabilité dans un processus intégrant UML.

Ce modèle conceptuel joue un rôle central dans notre méthode, il est capital de bien le définir.

Les fonctionnalités du système sont décrites comme un ensemble de cas d'utilisation.

- Chaque cas représente un flot spécifique d'événements vers le système.
- La description du cas d'utilisation définit ce qui arrive dans le système lors de sa réalisation.



## **I Diagramme de cas d'utilisation:**

Premièrement il faut définir tous les acteurs possibles du système.

### **I.1 Définition des acteurs :**

**I.1.1 Administrateur :** est un utilisateur du système qui est chargé d'effectuer plusieurs fonctions :

- Paramétrage (Système, SVR supervision),
- Gestion Profils utilisateurs,
- Saisie les plans tarifaires.

**I.1.2 Opérateur :** est un utilisateur initial, cet acteur est chargé de :

- Saisie les plages horaires ;
- Saisie les plans tarifaires ;
- Saisie les clients ;
- Saisie les abonnés ;
- Saisie abonnements ;
- Saisie les tarifs;
- Edition des factures;
- Suivi de paiement ;
- Consultation des factures stockées.

### **I.1.3 Observateur :**

Il ne peut que consulter les données de notre système.

**II Le cas d'utilisation de haut niveau** : Il décrit les interactions essentielles entre le système et les utilisateurs.

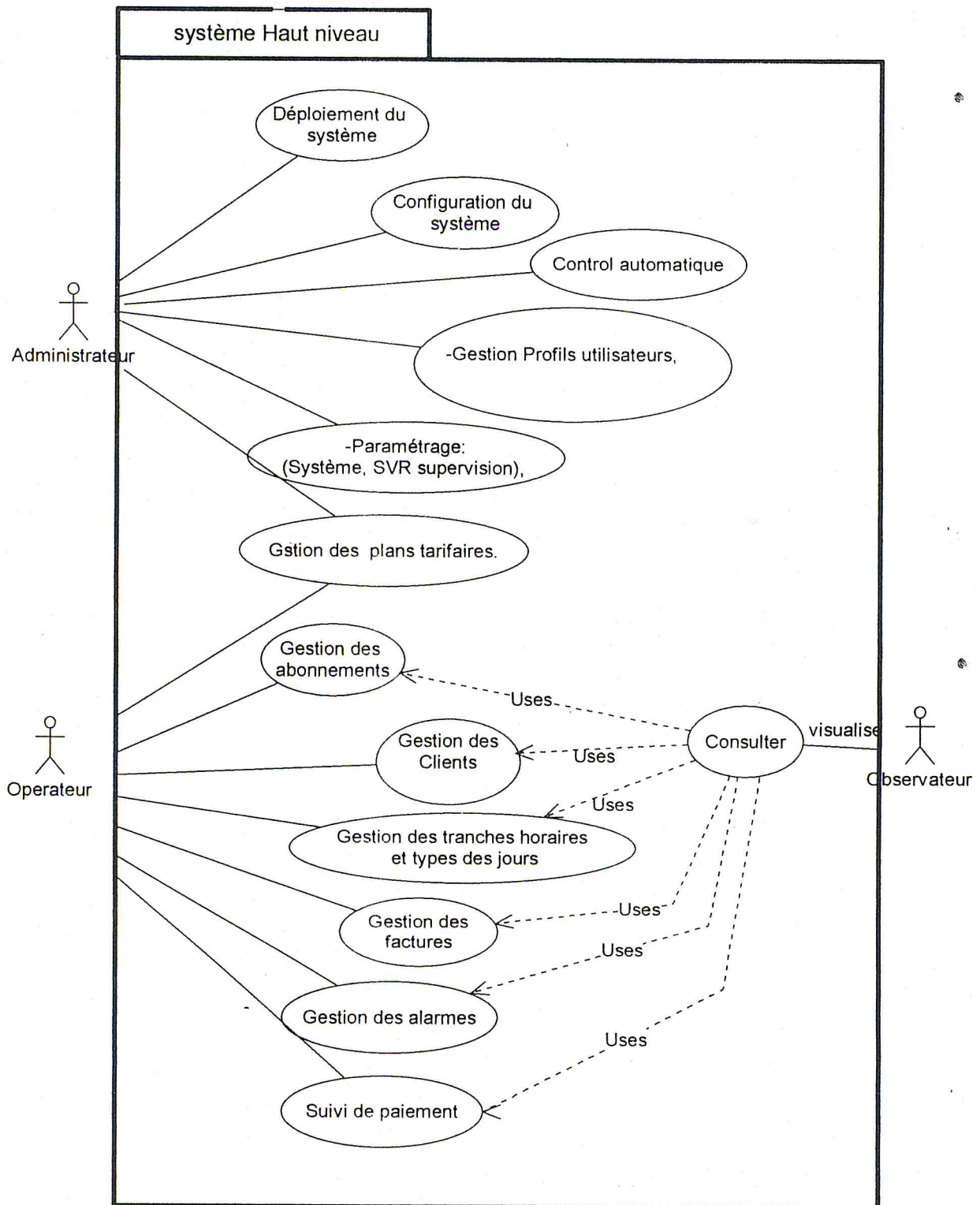


Figure IV.1 Cas d'utilisation de haut niveau

Il existe plusieurs cas d'utilisation qui sont regroupées selon les paquetages suivants :

- ▶ **Gestion des utilisateurs;**
- ▶ **Gestion des clients du réseau X25;**
- ▶ **Gestion des abonnements;**
- ▶ **Configuration des plans tarifaires et tranches horaires;**
- ▶ **Configuration des tarifications;**
- ▶ **Visualisation et Impression des factures stockées sur le serveur.**

### **III Les interactions entre le système et les utilisateurs :**

#### **III.1 Les modes**

Chaque cas d'utilisation comprend des modes d'accès aux données suivant les habilitations de l'utilisateur et les opérations qu'il désire réaliser, le mode d'accès aux données peut être différent. Il existe 4 modes d'accès aux données :

- ▶ Création,
- ▶ Consultation ;
- ▶ Modification;
- ▶ Suppression.

##### **III.1.1- Création :**

Ce cas d'utilisation est utilisé lorsqu'il s'agit d'ajouter une (ou plusieurs) occurrence dans une table de la base de donnée.

Les deux seules fonctionnalités possibles lors de la création sont : sauvegarder et annuler.

##### **III.1.2- Consultation :**

Ce cas d'utilisation est utilisé lorsqu'il s'agit de visualiser toute ou partie des données d'une table ; ce mode ne permet pas la modification des données.

Les fonctionnalités proposées, suivant les cas, sont : rechercher, naviguer.

### III.1.3- Modification des données :

Ce cas d'utilisation est utilisé pour modifier les données ; Il dispose de toutes les fonctionnalités présentes dans le mode de consultation, l'utilisateur disposera de deux fonctionnalités supplémentaires : sauvegarder et annuler, mais certaines données restent en lecture seule.

### III.1.4- suppression:

Il est utilisé pour effectuer une inactivation d'un client ou suppression d'un abonnement ; Il dispose de toutes les fonctionnalités présentes dans le mode de consultation normale, afin de se positionner sur la donnée à supprimer. Mais il dispose en plus d'une fonctionnalité permettant de préciser que l'enregistrement courant doit être résilié.

Donc les actions sont : rechercher, naviguer, supprimer, annuler

## III.2 Fonctionnalités génériques :

Cette partie a pour objectif de décrire les actions et les fonctionnalités que les utilisateurs peuvent interagir avec le système :

### III.2.1 Ajouter :

Il permet d'ajouter une nouvelle occurrence parmi les données déjà existantes. La zone de détails s'initialise avec des valeurs vierges ou bien choix parmi des listes défilant et l'utilisateur peut y saisir les valeurs de son choix.

### III.2.2 Rechercher :

Elle permet de se positionner sur une donnée au attribut d'une classe en ayant spécifié la valeur à laquelle une des données doit correspondre.

### III.2.3 Sauvegarder :

Ce cas d'utilisation est valide après une procédure d'ajout ou de modification de données. Son action est d'enregistrer les données situées dans la zone de détails, toutefois après une demande de confirmation de sauvegarde faite à l'utilisateur.

### III.2.4 Annuler :

Dans le cas d'une création ou modification, l'utilisateur peut réinitialise toutes les données qui viennent d'être saisies.

### III.2.5 Supprimer :

L'utilisateur peut effacé certaines données après la vérification, sinon il est prévenu de l'impossibilité de la suppression par un message explicatif.

Une demande de confirmation sera, demandée à l'utilisateur afin de ne pas supprimer involontairement des données.

## IV Gestion des utilisateurs et l'administration :

L'acteur unique et principal dans cette section est l'administrateur.

### IV.1 Paramétrage du système :

Le but de ce cas d'utilisation est de définir les paramètres du système.

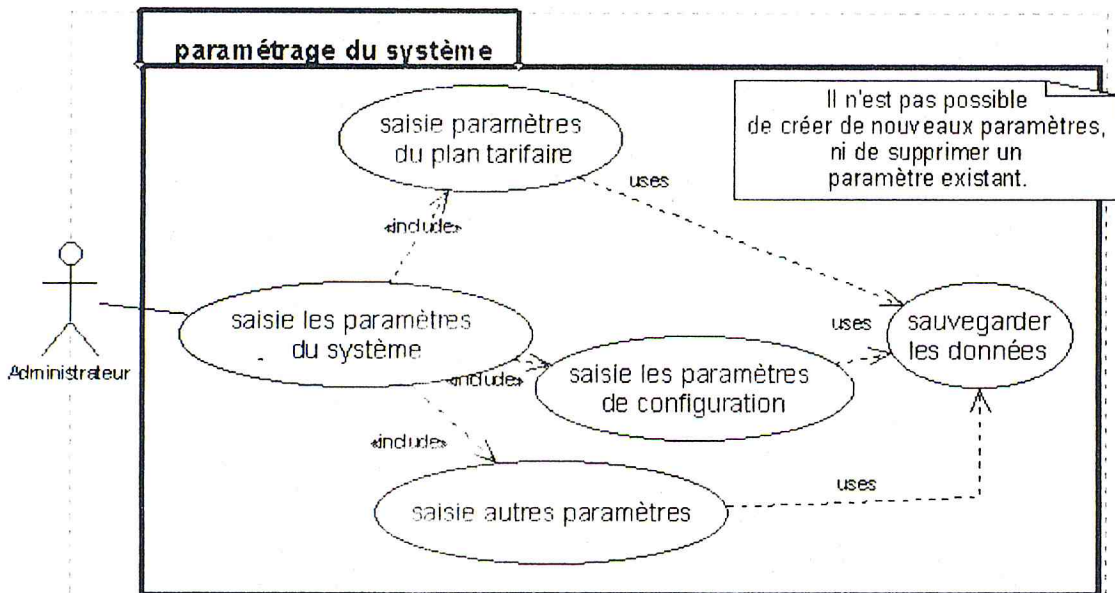


Figure IV.2 Cas d'utilisation paramétrage du système

- La seule action permise à l'utilisateur est la modification des valeurs liées à un paramètre.
- Le nom du paramètre n'est pas modifiable.
- Il n'est pas possible de créer de nouveaux paramètres, ni de supprimer un paramètre existant.

#### Liste des paramètres :

- Paramètres du plan tarifaire : TVA, GFA (Groupe Fermé Abonné) ;
- Paramètres de configuration ;
- Autres paramètres : date de restauration, date de l'archivage, taux maximal...

#### IV.2 Gestion des utilisateurs :

Ce cas d'utilisation permet à un administrateur de définir les profils (habilitations) des groupes d'utilisateurs, les groupes d'utilisateurs et les utilisateurs eux-mêmes.

Dans le paquetage de gestion des utilisateurs, il existe trois cas d'utilisations :

- ▶ Cas d'utilisations gestion des groupes : Nouveau groupe, Supprimer un groupe
  - ▶ Cas d'utilisations gestion des utilisateurs : Nouvel utilisateur, Modifier un utilisateur, Ajouter un utilisateur à un groupe, Enlever un utilisateur de son groupe, Supprimer un utilisateur
  - ▶ Cas d'utilisations gestion des permissions : Donner à un groupe les droits d'accès, Supprimer pour un groupe les droits d'accès.
- 
- ▶ Lors de la création d'un nouvel utilisateur il faut saisir les paramètres suivantes : nom, prénom, téléphone, fax, E-mail.
  - ▶ Un utilisateur ne peut appartenir qu'à un et un seul des groupes d'utilisateurs définis dans la table des groupes d'utilisateurs.

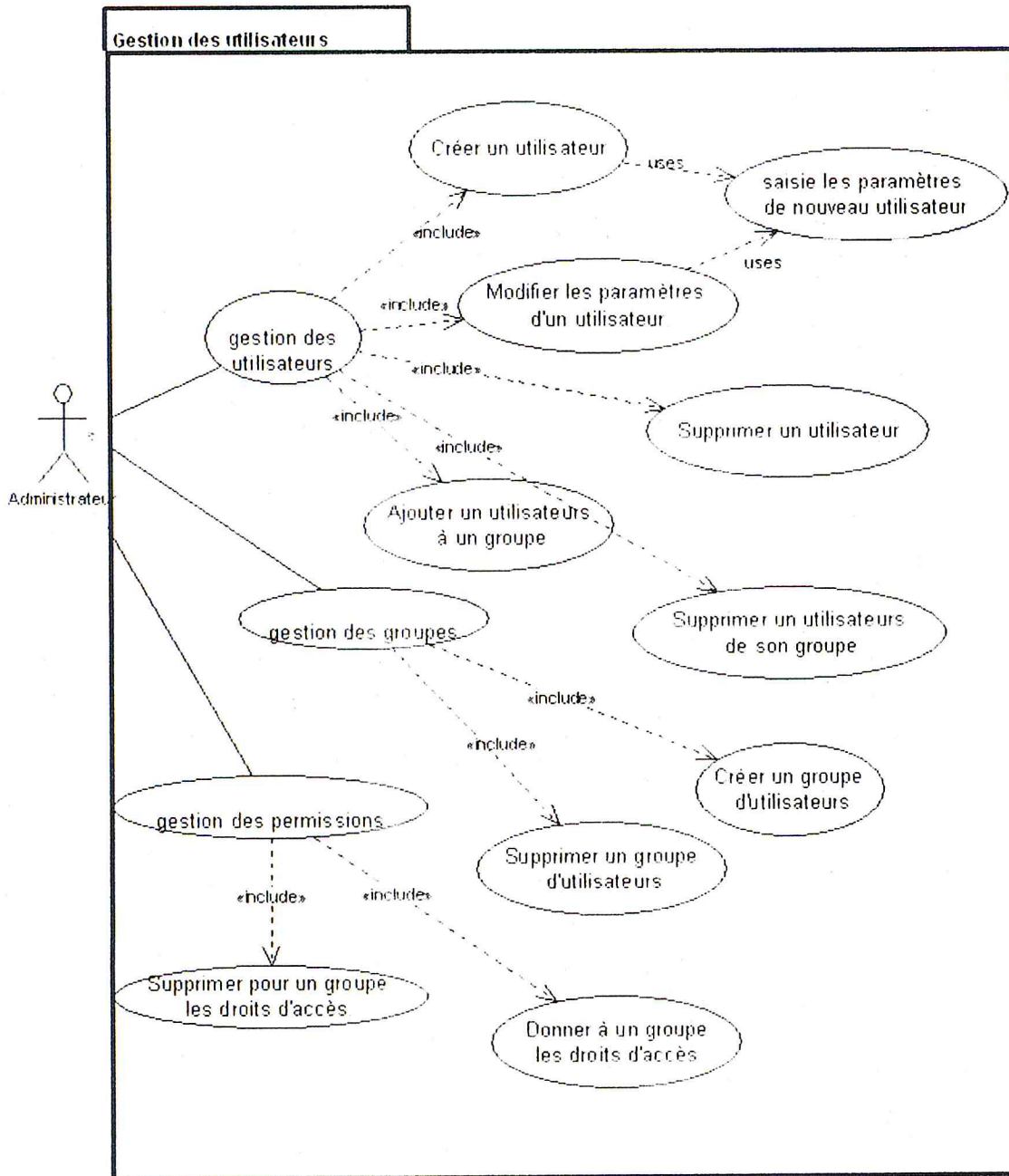
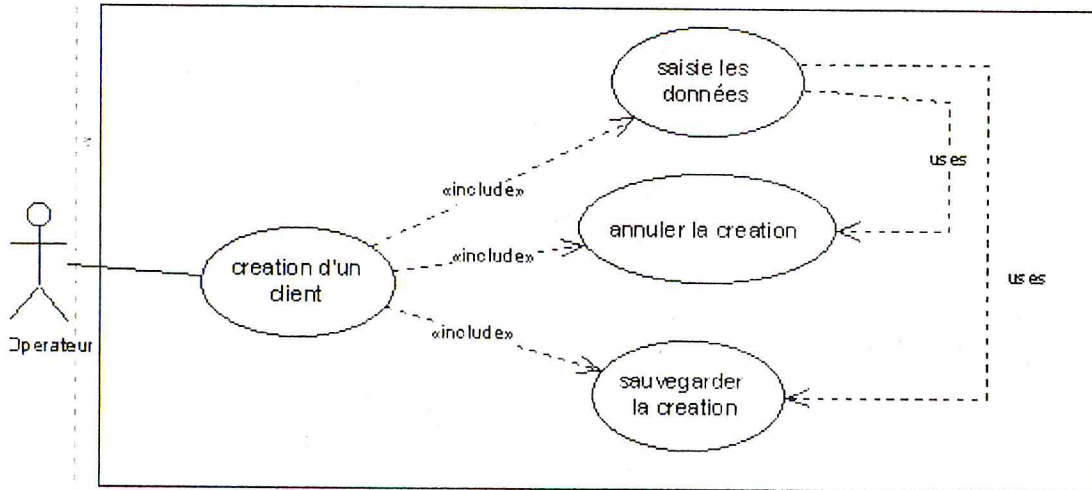


Figure IV.3 Cas d'utilisation gestion d'utilisateur.

## V. Gestion des clients :

### V.1 Création d'un client :

Ce cas d'utilisation est utilisé lorsqu'il s'agit d'ajouter une (ou plusieurs) occurrence dans une table.



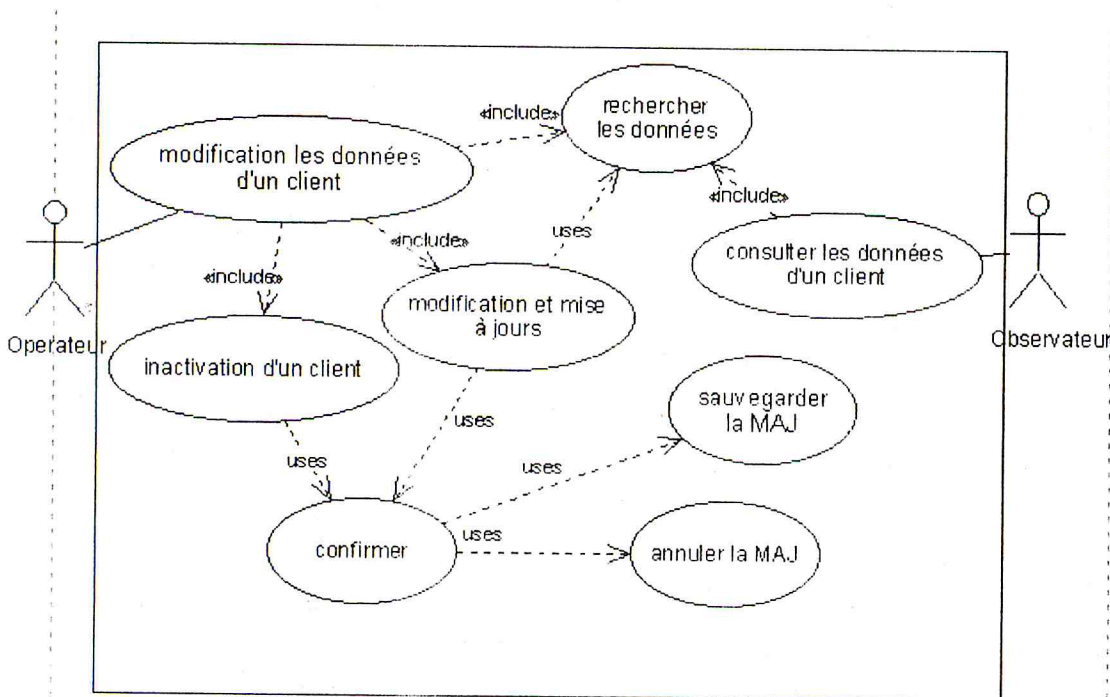
**Figure IV.4** Cas d'utilisation création d'un client

- ▶ Dans le cas où le mode est « Création », le numéro de client n'est pas affiché, il est attribué automatiquement.
- ▶ La date de création du client est calculée automatiquement lorsqu'un utilisateur créer un nouveau client et elle est positionnée à la valeur de la date système.
- ▶ la périodicité de la facturation du client dépend du choix du client.



## V.2 Consultation et Modification des données d'un client :

Ce cas d'utilisation montre le traitement de consultation et MAJ sur le client :



**Figure IV.5** Cas d'utilisation consultation et modification des données de client.

- ▶ La Modification ne permet pas la mise à jour des données suivantes :  
Numéro du client, Type de client, Date de création...
- ▶ L'état du client peut être Actif ou Inactif et est calculé à partir du nombre d'abonnements du client.
- ▶ Inactivation directe du client : la date d'inactivation du client prend la valeur de la date système lorsque l'état du client est positionné à Inactif.
- ▶ Inactivation automatique du client : l'inactivation est automatiquement réalisée suite à la résiliation de tous ses abonnements.

La figure suivante décrit le packaging de cas d'utilisation la gestion des clients :

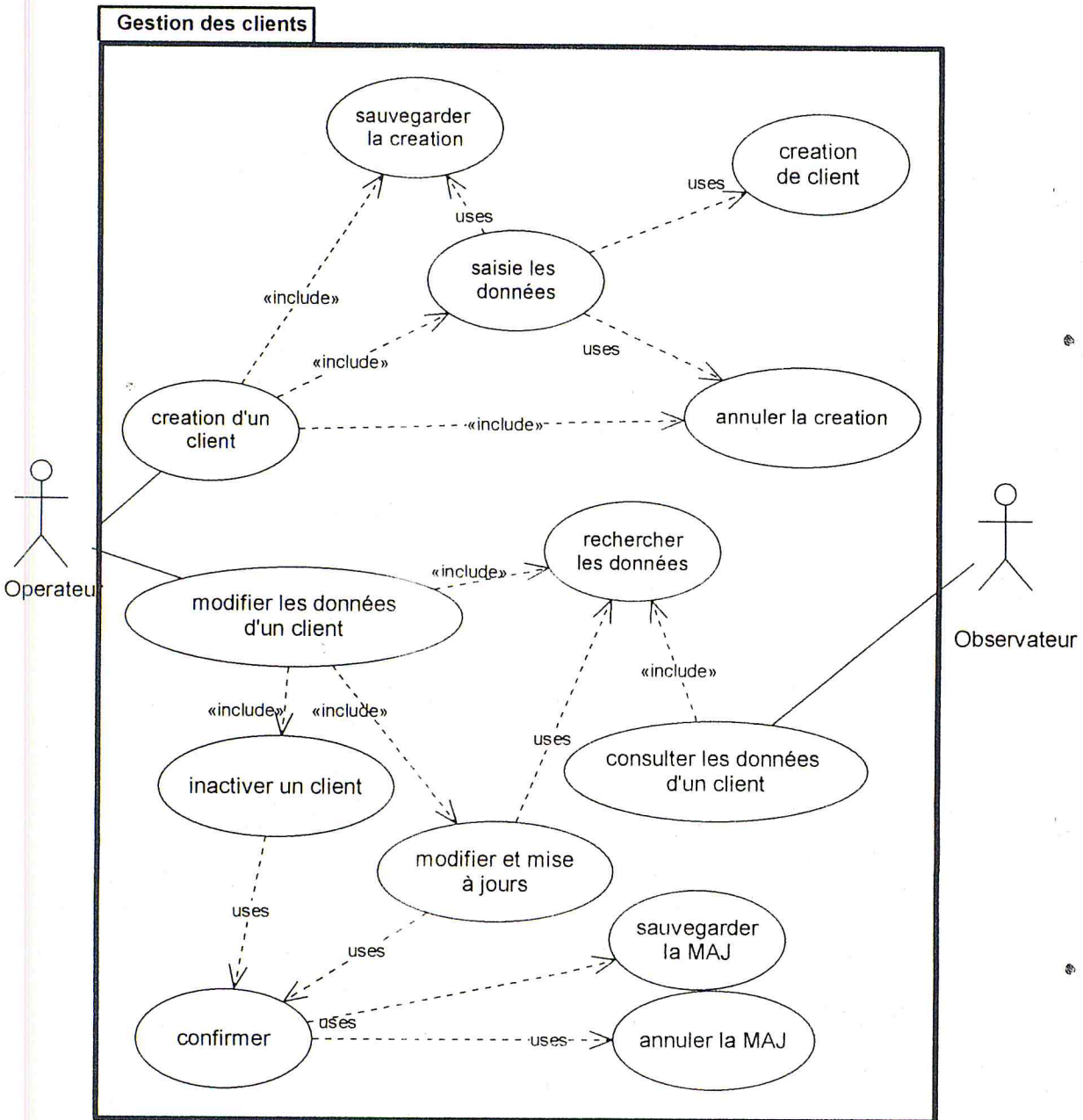


Figure IV.6 Packaging de cas d'utilisation gestion des clients

La figure suivante décrit le packaging de cas d'utilisation la gestion des clients :

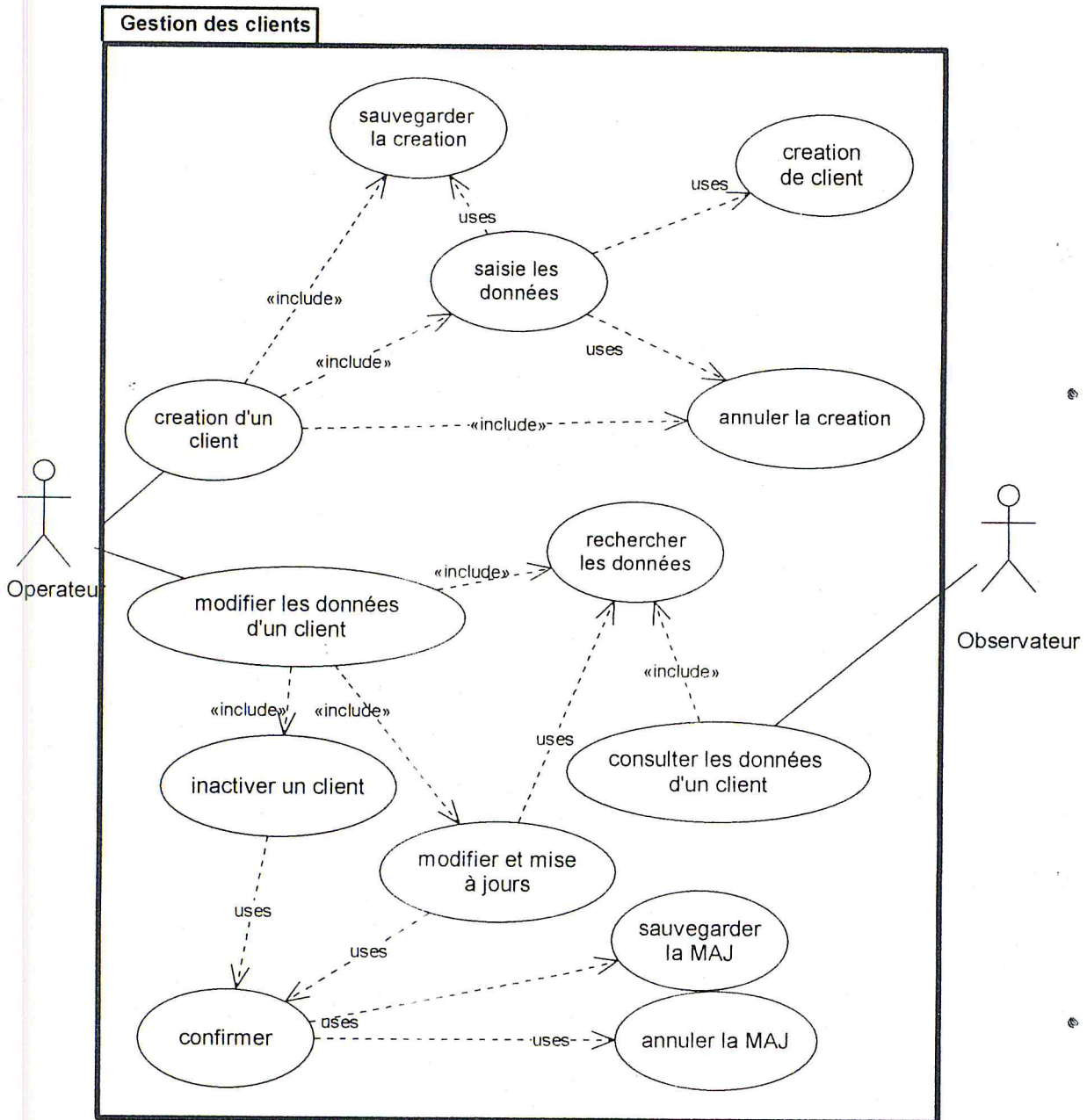


Figure IV.6 Packaging de cas d'utilisation gestion des clients

- ▶ Les cas d'utilisation des abonnements ne peuvent se baser que sur un client du réseau X25. Les données accessibles ne seront donc pas la liste de tous les abonnements, mais seulement la liste de tous les abonnements du client courant.
- ▶ L'ajout des abonnements au client ne peut être fait qu'après la validation et l'enregistrement du client dans la base de données.
- ▶ A la résiliation d'un abonnement, le système va vérifier que le client possède au moins un autre abonnement actif, sinon, le client est automatiquement inactivé.

## VII Gestion des types de jour et tranches horaires :

### VII.1 les types des jours :

Ce cas d'utilisation a pour objectif de permettre à un utilisateur d'affecter à chaque jour de la semaine un type de jour.

Les valeurs possibles pour chaque type de jour sont :

- ▶ Jour Ouvrable
- ▶ Jour Ouvré
- ▶ Jour Férié

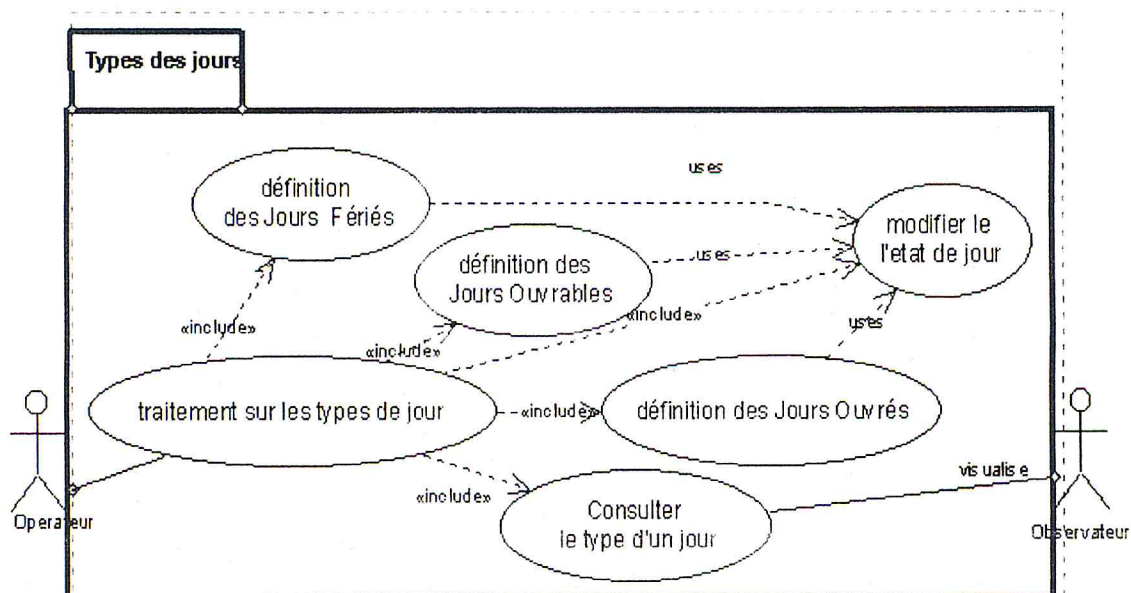


Figure IV.8 Cas d'utilisation type de jours

- ▶ Les cas d'utilisation des abonnements ne peuvent se baser que sur un client du réseau X25. Les données accessibles ne seront donc pas la liste de tous les abonnements, mais seulement la liste de tous les abonnements du client courant.
- ▶ L'ajout des abonnements au client ne peut être fait qu'après la validation et l'enregistrement du client dans la base de données.
- ▶ A la résiliation d'un abonnement, le système va vérifier que le client possède au moins un autre abonnement actif, sinon, le client est automatiquement inactivé.

## VII Gestion des types de jour et tranches horaires :

### VII.1 les types des jours :

Ce cas d'utilisation a pour objectif de permettre à un utilisateur d'affecter à chaque jour de la semaine un type de jour.

Les valeurs possibles pour chaque type de jour sont :

- ▶ Jour Ouvrable
- ▶ Jour Ouvré
- ▶ Jour Férié

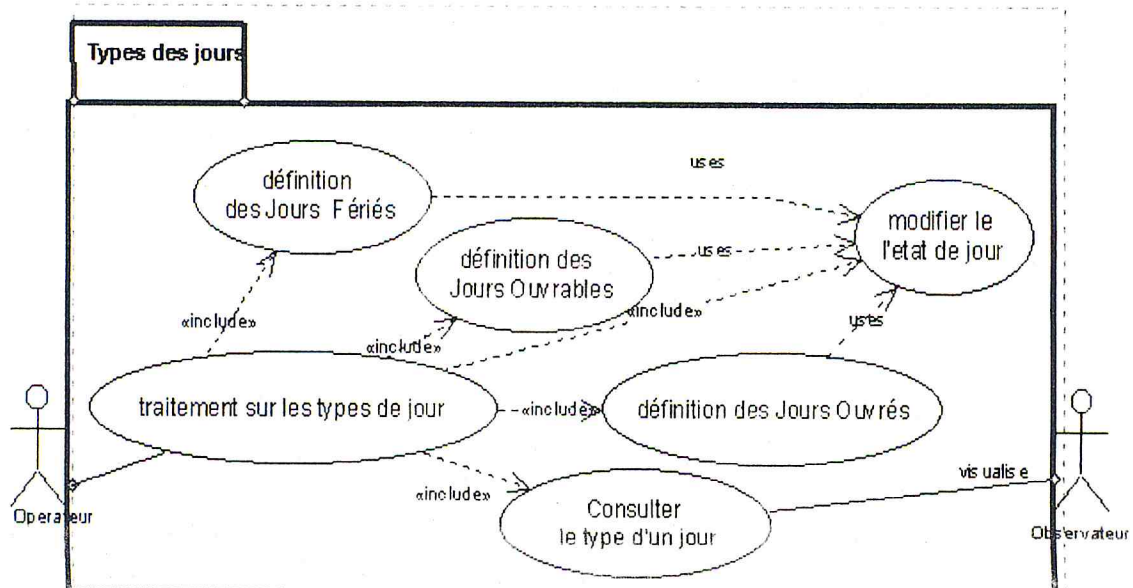


Figure IV.8 Cas d'utilisation type de jours

- ▶ L'affichage du type général de chaque jour de la semaine se fait par l'intermédiaire de 3 listes, une pour chaque type de jour.
- ▶ L'utilisateur peut ainsi visualiser tous les jours ouvrés, tous les jours ouvrables et tous les jours fériés.
- ▶ Chaque jour ne correspond qu'à un et un seul type de jour.
- ▶ Lors de l'ouverture du système, les valeurs par défaut suivantes sont proposées : Le jeudi est ouvrable, le vendredi est férié et tous les autres jours sont ouvrés.

## VII.2 Les tranches horaires :

Les tranches horaires, ne devront pas être définies individuellement, mais par jour entier, c'est à dire définies à partir des types de jour.

Une tranche horaire est constituée par une heure de début et une heure de fin.

Il existe trois types de réductions pour les tranches horaires : fortes, moyennes et faibles.



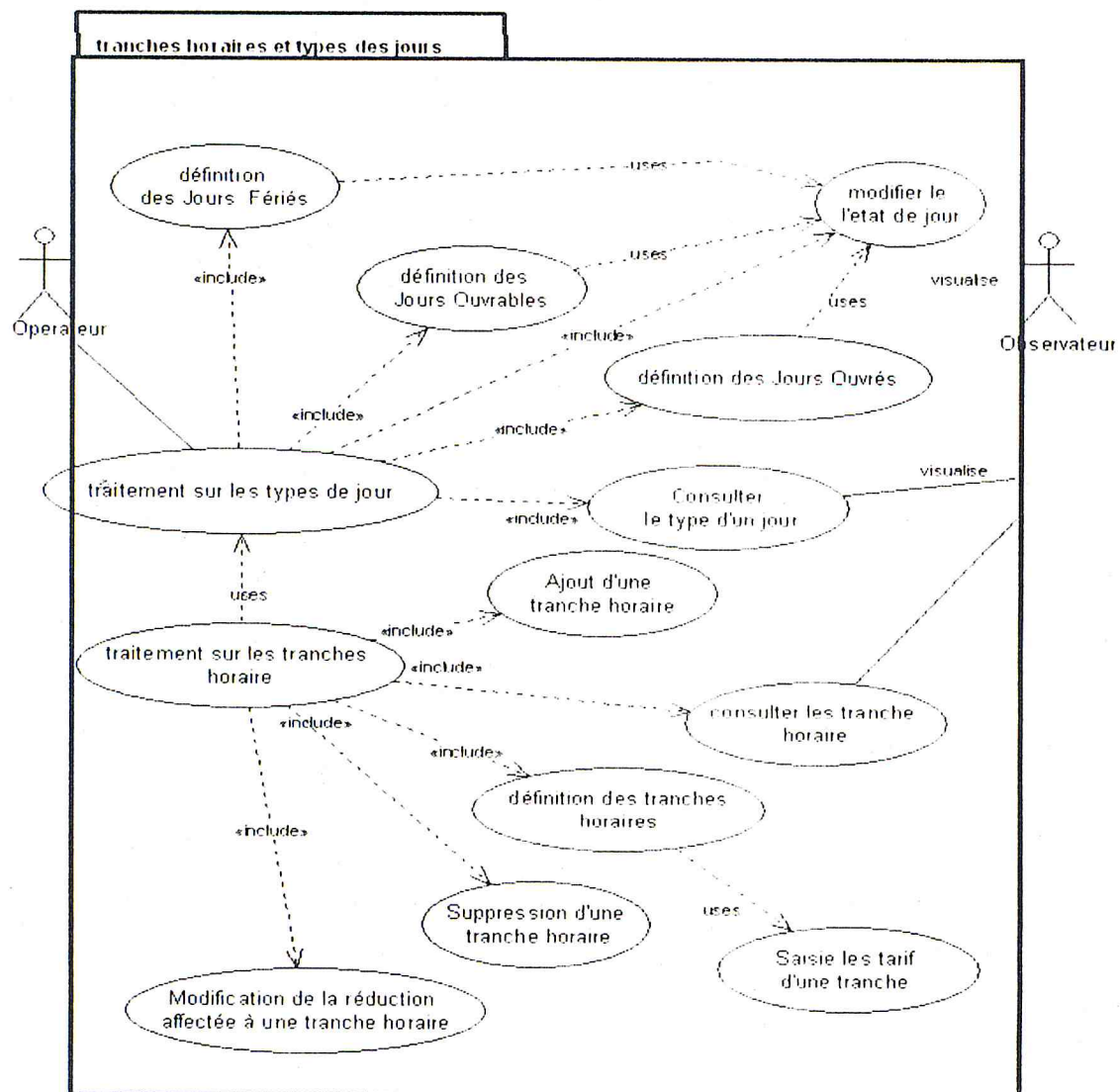


Figure IV.9 Cas d'utilisation type des jours et tranches horaires.

L'utilisateur peut effectuer les actions suivantes sur la liste des tranches horaires :

- ▶ Ajout d'une tranche horaire,
- ▶ Suppression d'une tranche horaire,
- ▶ Modification de la réduction affectée à une tranche horaire.

Et il peut aussi modifier les réductions fortes, moyennes et faibles utilisés pour définir les valeurs des réductions associées à chaque tranche.

**Remarque :** Deux tranches horaires ne peuvent se chevaucher (superposer).

## VIII Les tarifications :

### VIII.1 Traitement des tarifications:

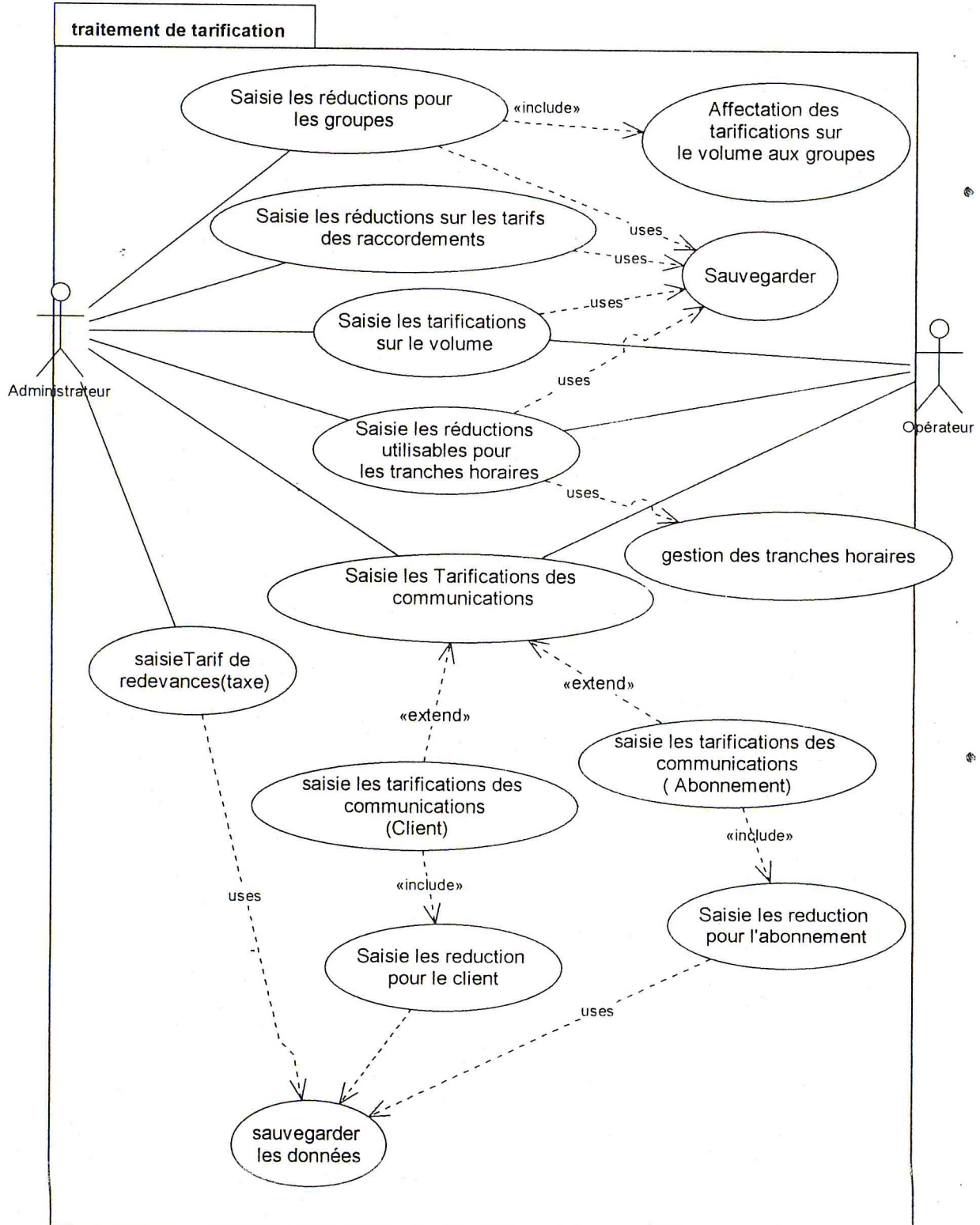


Figure IV.10 Cas d'utilisation traitement des tarifications



Il n'est pas possible de supprimer un type de tarif s'il est utilisé par au moins un des clients ou un des abonnements. La suppression n'est autorisée que si le type de tarif est inutilisé

### VIII.2 Réductions et tarifs spécifiques :

Les tarifications spécifiques aux clients et aux abonnements disposent :

- Réduction à la durée
- Réduction au volume
- Réduction pour un forfait complet

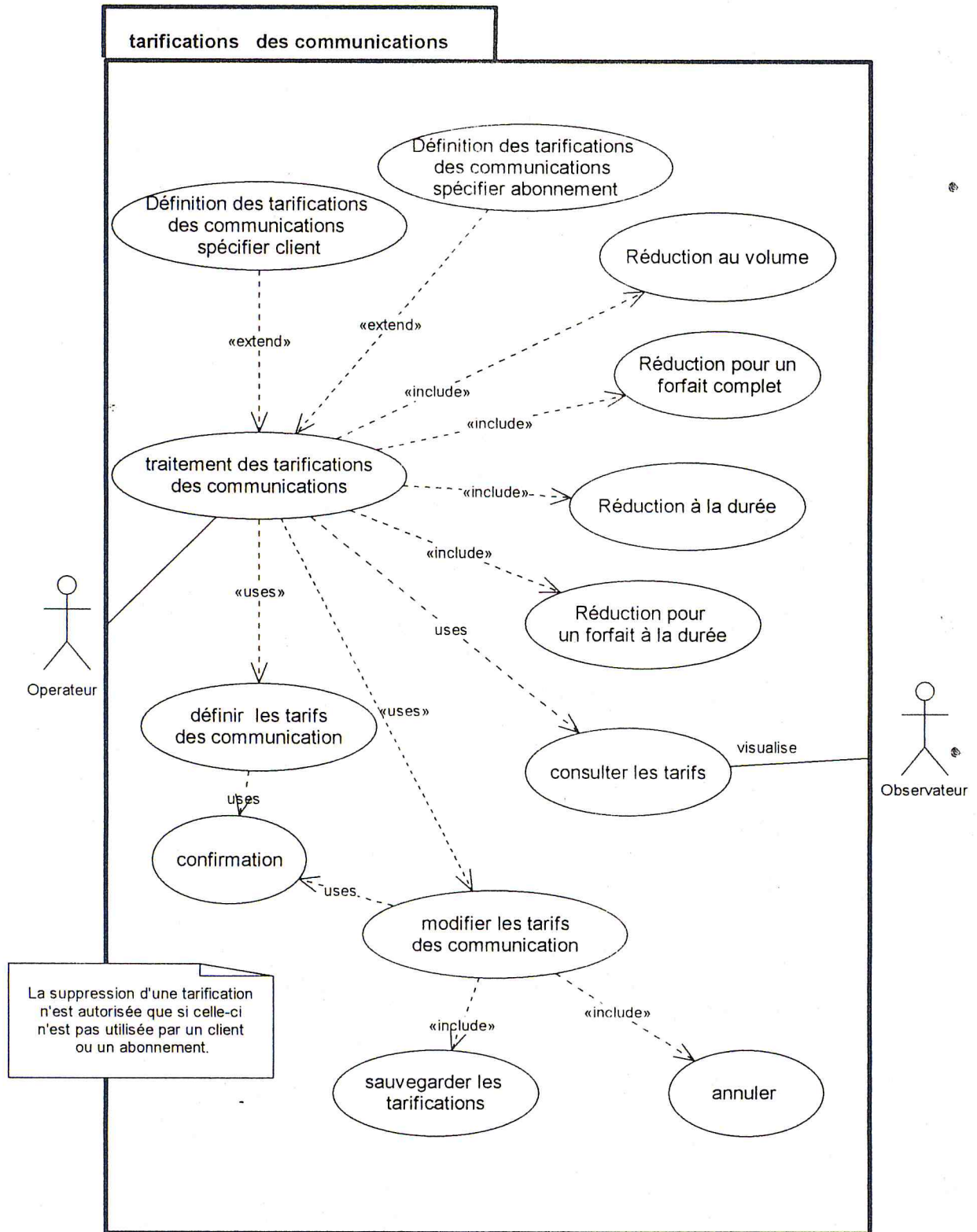
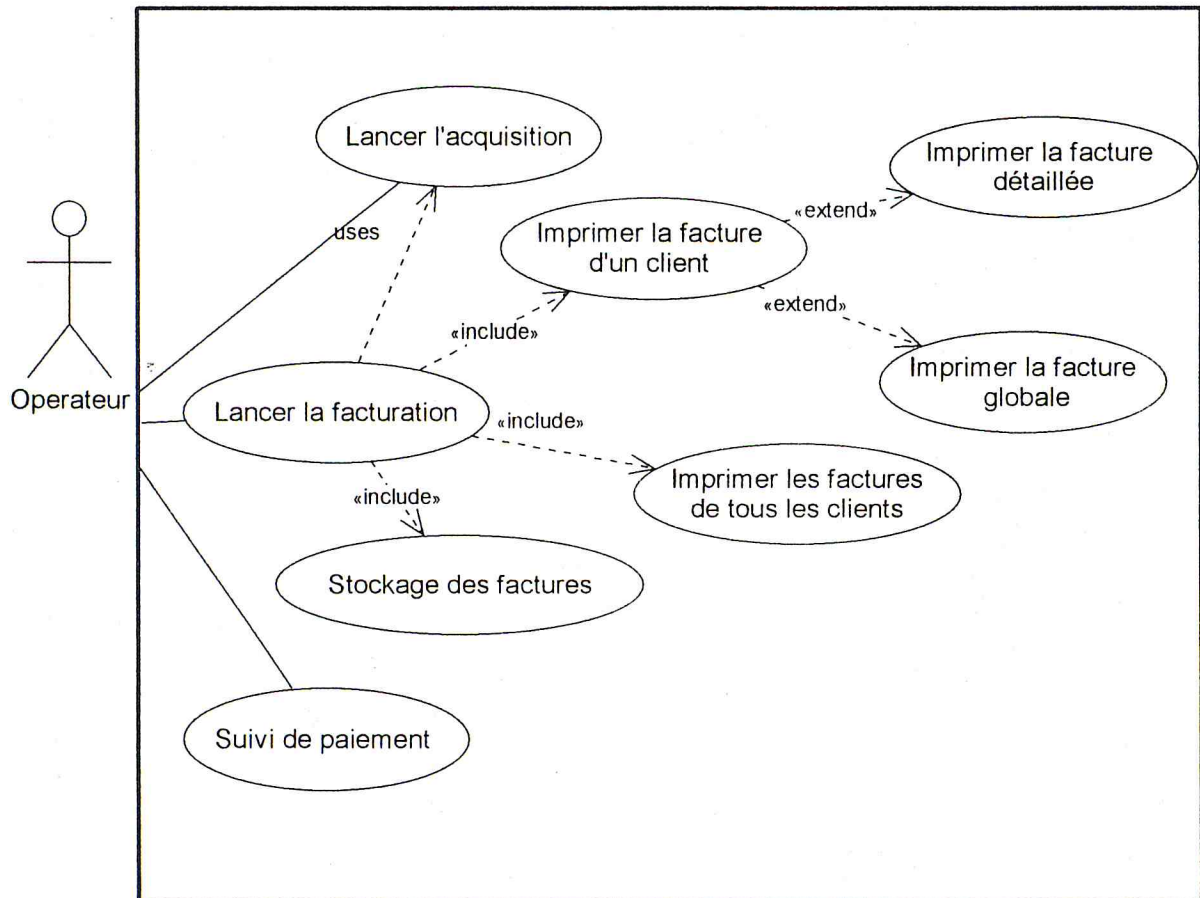


Figure IV.11 Cas d'utilisation réductions et tarifs spécifiques

## IX Facturation et Suivi de paiement :

La figure suivante décrit le cas d'utilisation de l'impression des factures :



**Figure IV.12** Cas d'utilisation impression des factures.

- L'édition des factures peut être :
  1. Automatique : début de chaque bimestre.
  2. A la demande de l'utilisateur du système.

## **II La Vue Logique (Diagramme de classes):**

Afin de permettre au lecteur de bien comprendre notre système, dans cette section, nous utilisons la vue logique qui est représentée par le diagramme de classes. Nous présentons notre système par un ensemble de paquetage chacun correspond à un sous système. Ces paquetages sont répond aux besoins des cas d'utilisation. Ensuite nous allons décomposer et détailler chaque paquetage de classes à part.

### **I - Diagramme de classe :**

Le diagramme de classe nous a permet de modéliser, d'une manière logique, les relations qui existent entre l'ensemble de classes. Il développe d'une part la structure des entités du système et d'autre part celle d'un code orienté objet.

La figure suivante présente les paquetages des diagrammes de classes et les relations entre eux :

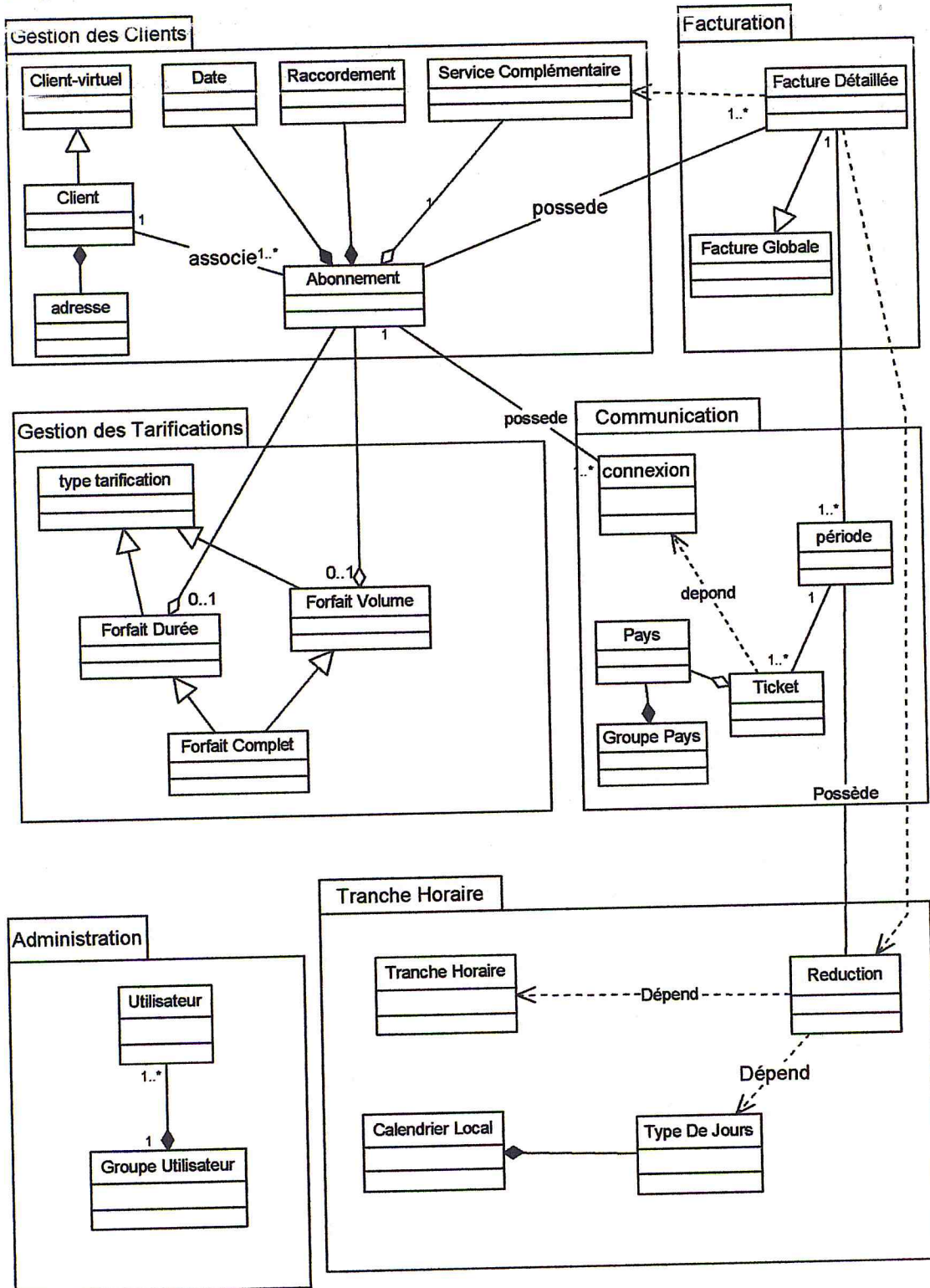


Figure IV.1 Paquetage de diagramme des classes

## **II Paquetage de gestion des clients :**

Ce paquetage présente les classes : client, abonnement...etc.

La classe client est une abstraction du client qui utilise la connexion avec le protocole X25.

Chaque client peut avoir un ou plusieurs abonnements.

Chaque abonnement possède plusieurs factures. La classe facture est depend de la classe 'services complémentaire'.

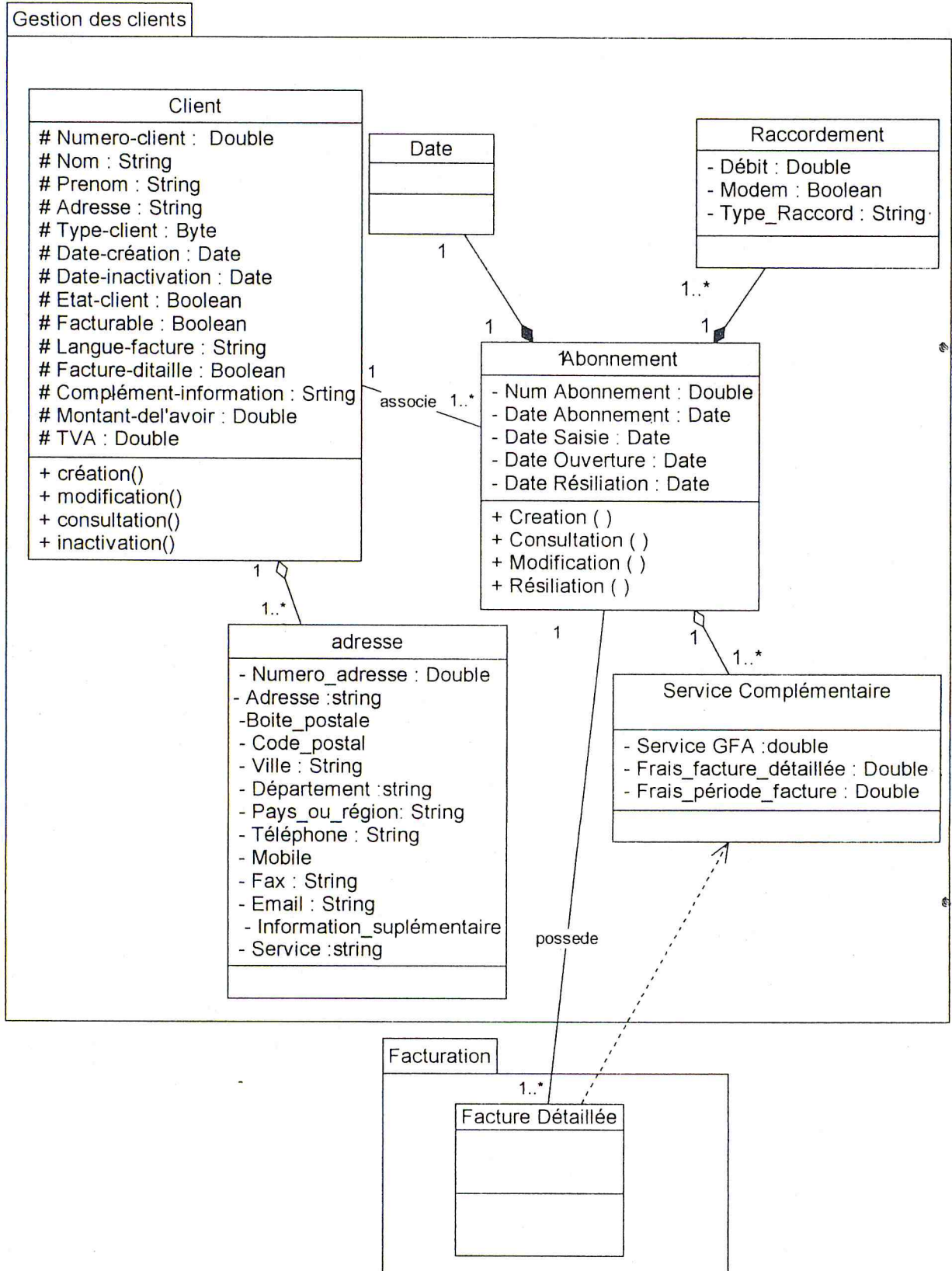


Figure IV.2 Paquetage de gestion des clients

### III Paquetage de l'administration :

Ce paquetage contient les classes des utilisateurs et les paramètres du système.

La classe Parametres\_système a tous les attributs statiques car ils sont des variables de classe.

Caque groupe d'utilisateur est composé de plusieurs utilisateurs.

Un utilisateur ne peut appartenir qu'un seul groupe.

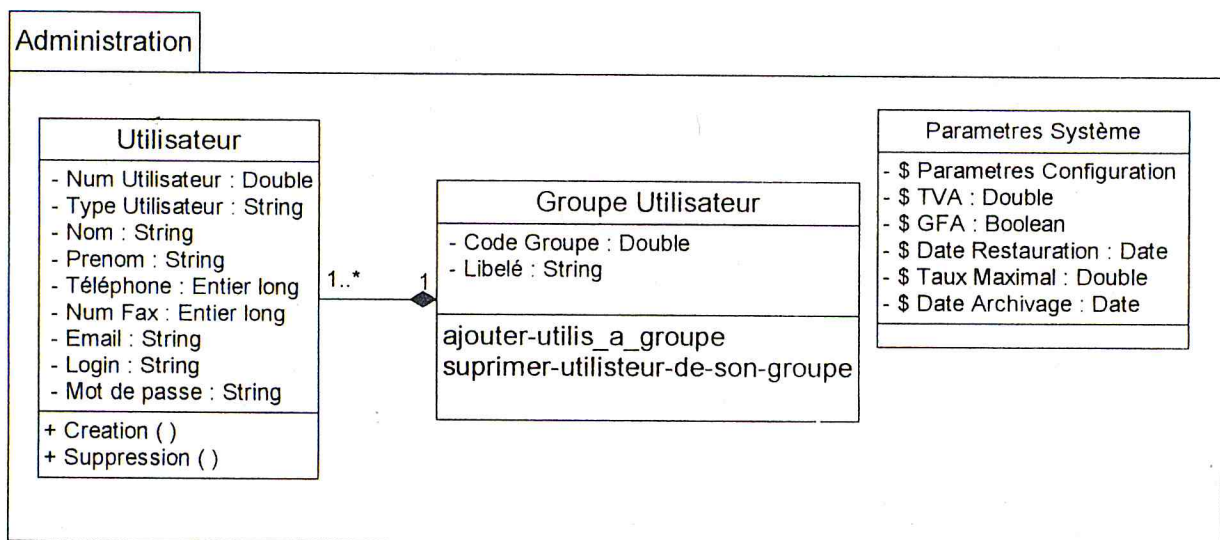


Figure IV.3 Paquetage de l'administration

### IV Le paquetage de tranches horaires :

Un groupe donné peut détenir plusieurs tarifications pour le volume en fonction de tranches horaires. Pour le groupe national, il faut aussi tenir compte du type de jour (ouvré, ouvrable, férié).

#### La classe tranche horaire :

Elle définit la durée de connexion par jour associé à un ticket.

Le jour de la connexion peut avoir l'un des trois types (ouvrés, ouvrable fériés définis sur un calendrier dédié). qui sont représentés par la classe types de jour.

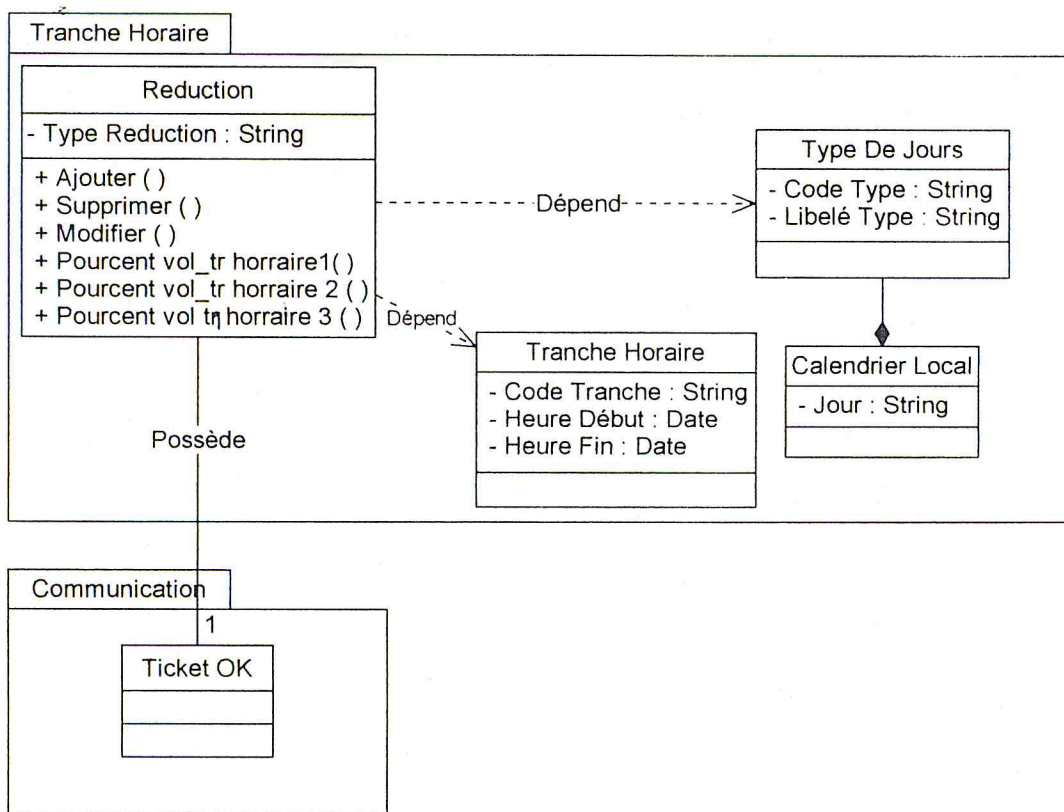


La classe réduction :

Pour chaque tranche horaire, on attribue une réduction. Il y a maximum trois réductions possibles par groupe de pays ( ce qui ne limite pas le nombre de tranches horaires possibles). Ces réductions sont nommées tranches tarifaires.

La classe réduction est dépendante de la classe type des jours et la classe tranche horaire.

Le calendrier est composé des jours.



**Figure IV.4** Paquetage des tranches horaires

Dans l'exemple ci-dessous, la tranche horaire 2 (entre 09h00 et 19h00) est au tarif plein. Par contre la tranche 1 est à 60% de réduction.

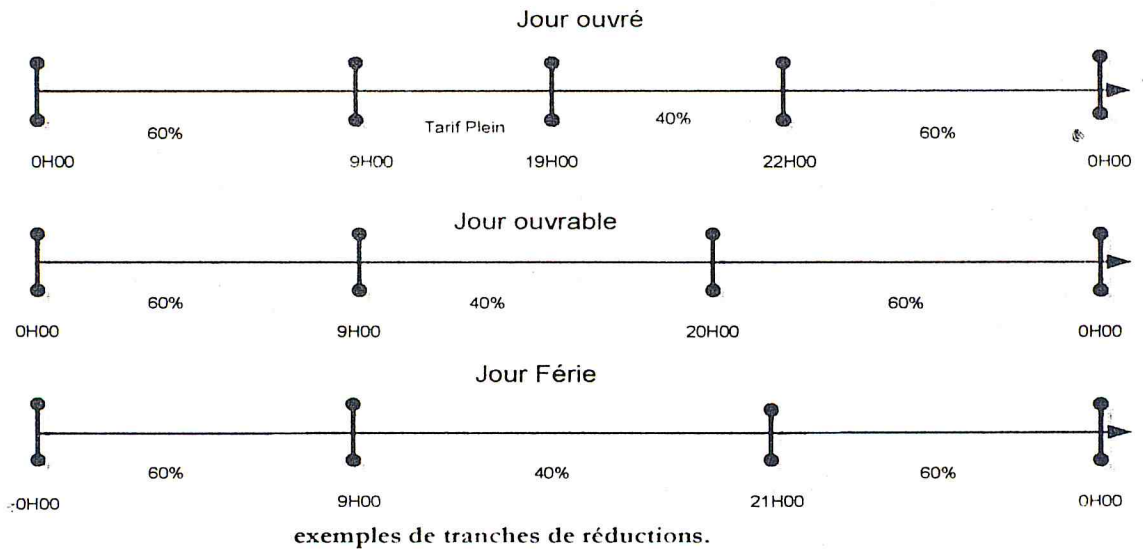


Figure IV.5 Exemple des tranches horaires

Dans l'exemple précédent on définit pour le groupe national :

- ▶ 4 tranches et 3 tarifs pour les jours ouvrés,
- ▶ 3 tranches et 2 tarifs pour les jours ouvrables,
- ▶ 3 tranches et 2 tarifs pour les jours fériés.

Les codes tranches devront être définis de manière explicite de façon à bien les différencier des autres tranches des autres groupes et types de jours.

## V -Paquetage de communication :

Ce paquetage de classes contient essentiellement les classes de ticket possible

La classe ticket a les méthodes de vérification\_des\_grammaire (tic : ticket)

Un pays est défini par :-

- ▶ Code\_indicatif : pays correspondant à l'indication X121,
- ▶ libellé.

Un groupe est défini par :

- ▶ Code de groupe,
- ▶ des tarifs horaires pour un ticket,
- ▶ un volume minimum facturé par communication aboutie.

Chaque ticket possède une période qui est caractérisée par 'Date; heure début' et la durée.

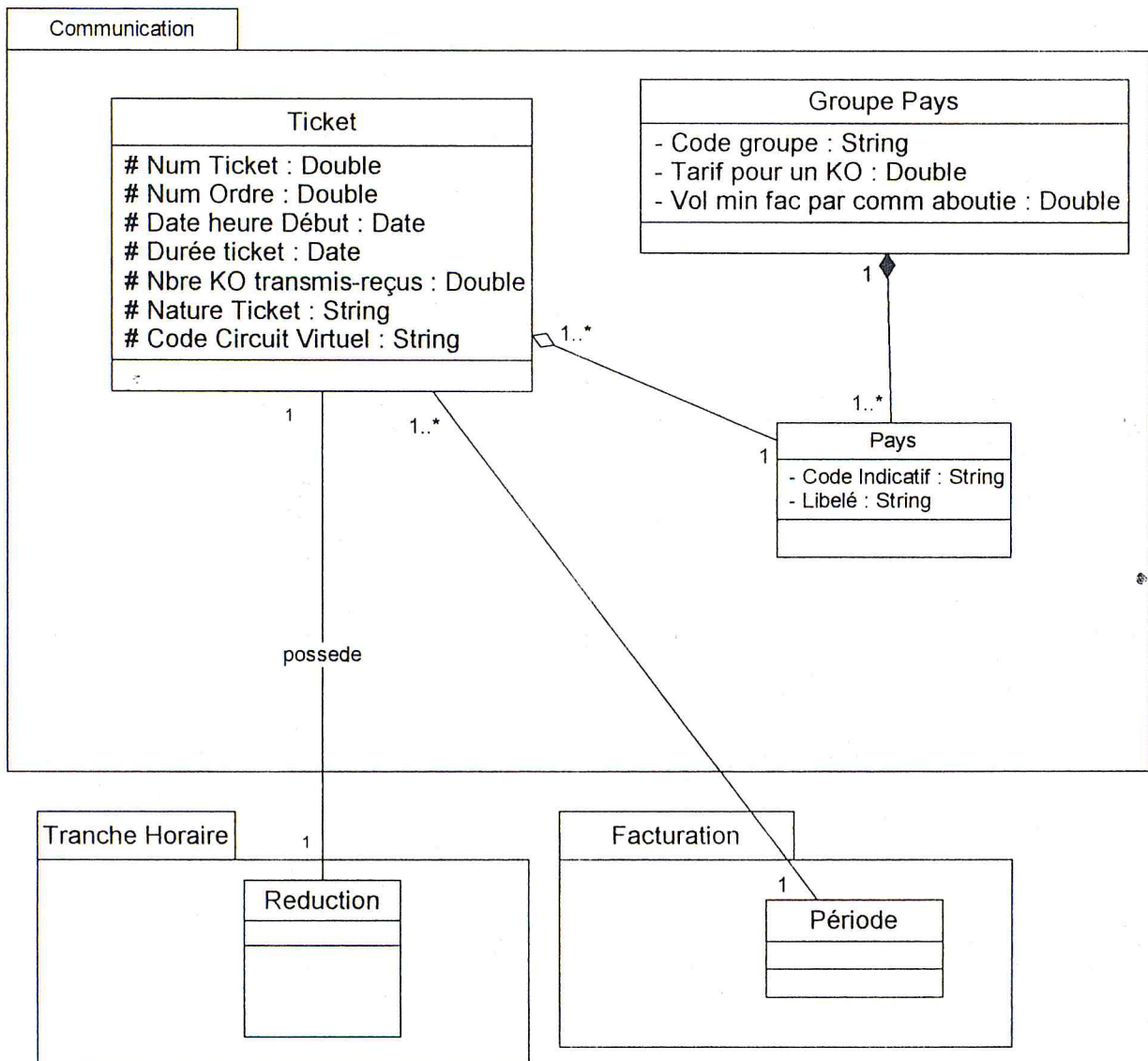


Figure IV.6 Paquetage de communication

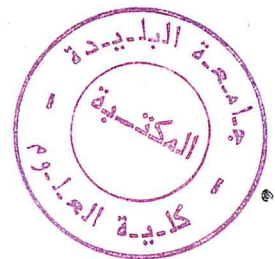
## VI Le paquetage de gestion des tarifications :

La classe 'type de tarification' est une abstraction des tarifs.

Les deux classes 'forfait durée', 'forfait volume' héritent de la classe 'type de tarification', chaque classe hérite les attribut de la classe mère et contient des autres attribut en plus.

La classe 'tarif complet' possède les attribut de la classe 'forfait durée', 'forfait volume'.

Chaque abonnement possède une tarification.



- le forfait complet qui représente un forfait illimité sur les communications nationales au volume et à la durée.

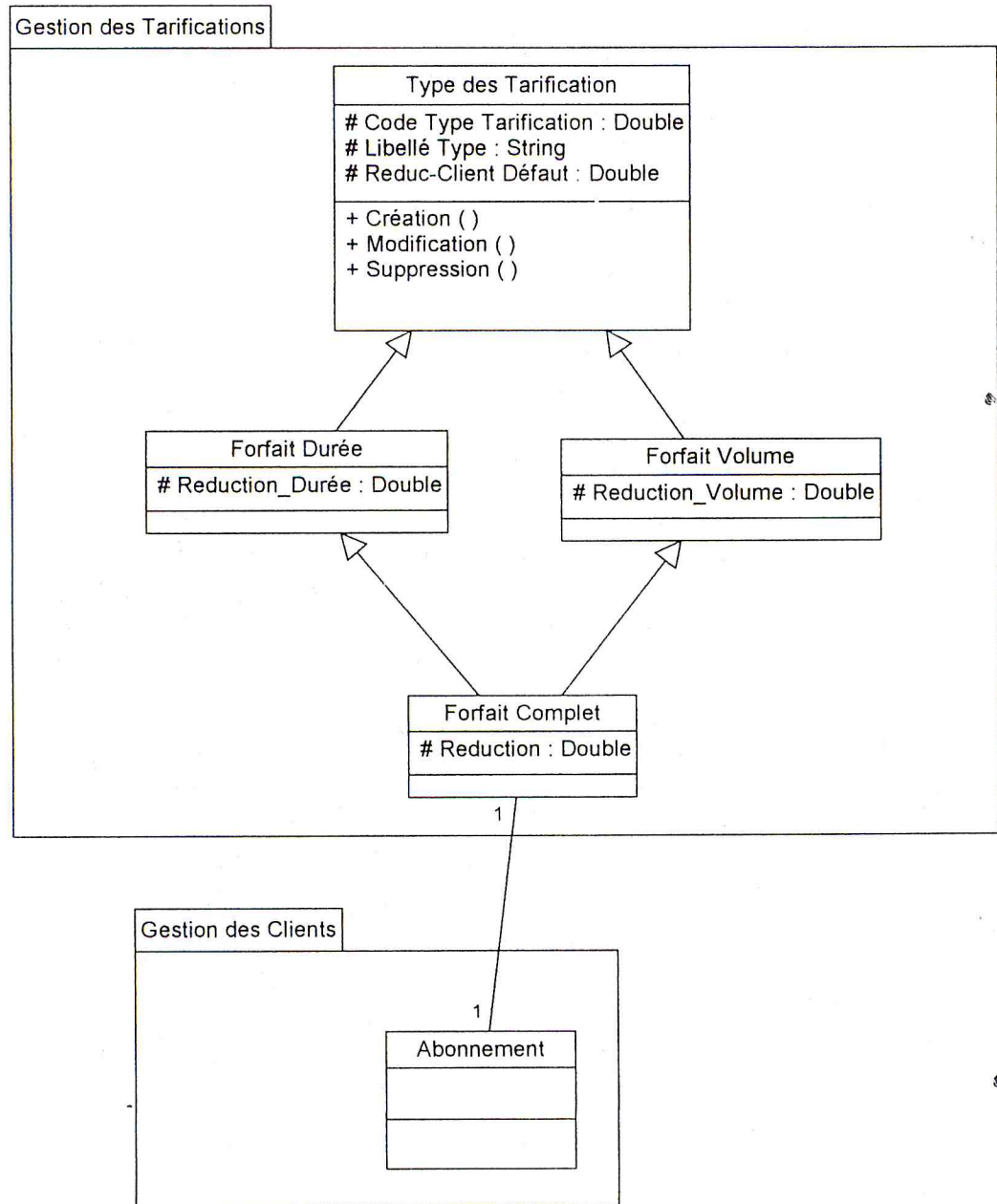


Figure IV.7 Paquetage de gestion des tarifications

## VII Paquetage de la facturation :

La classe facture détaillée hérite les propriétés de la classe Facture Globale

Chaque abonnement possède une ou plusieurs factures.

La facture est dépend des valeurs de réduction accordée.

La facture possède une période qui est caractérisée par 'Date; heure début' et la durée.

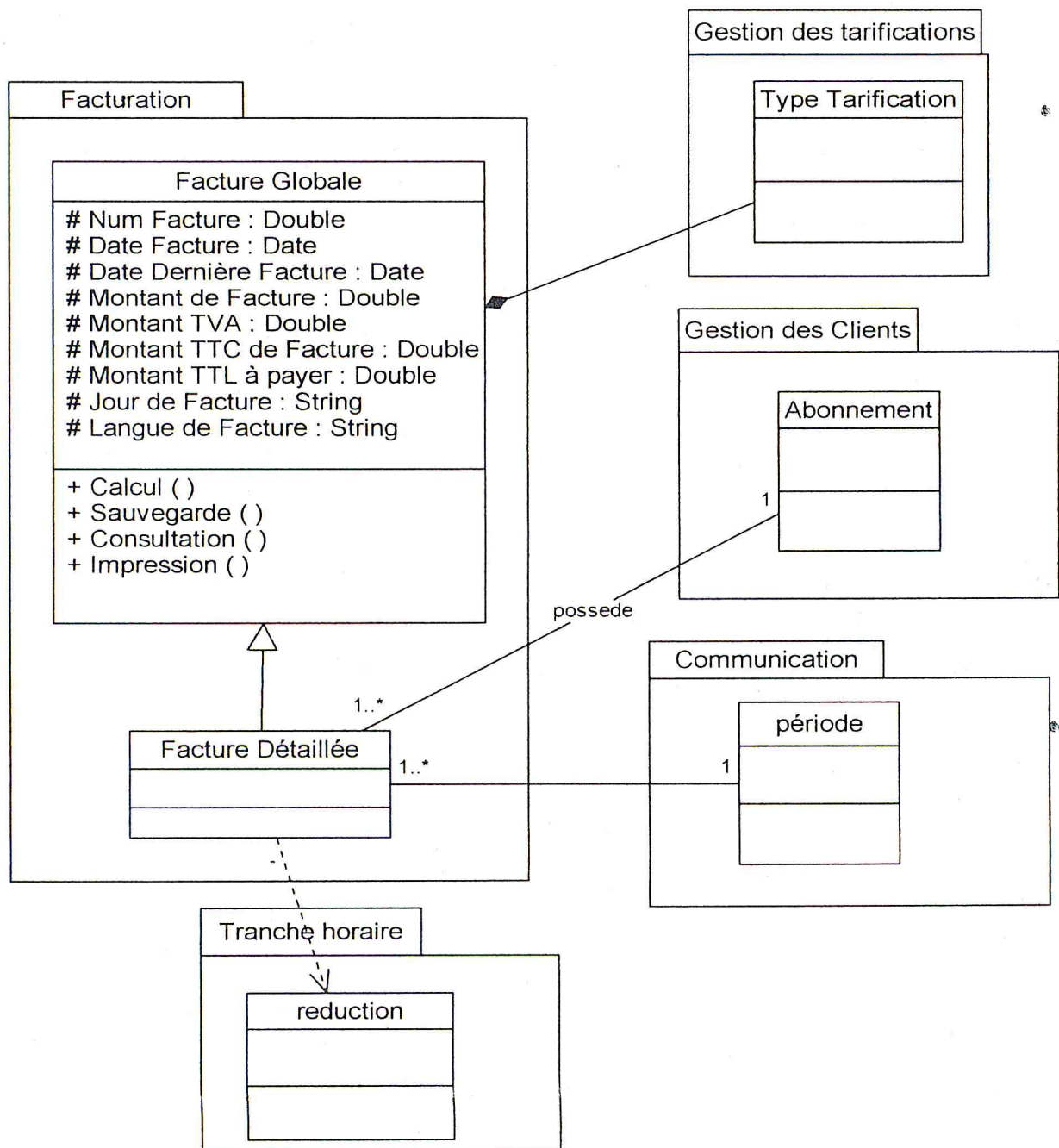


Figure IV.8 Paquetage de la facturation

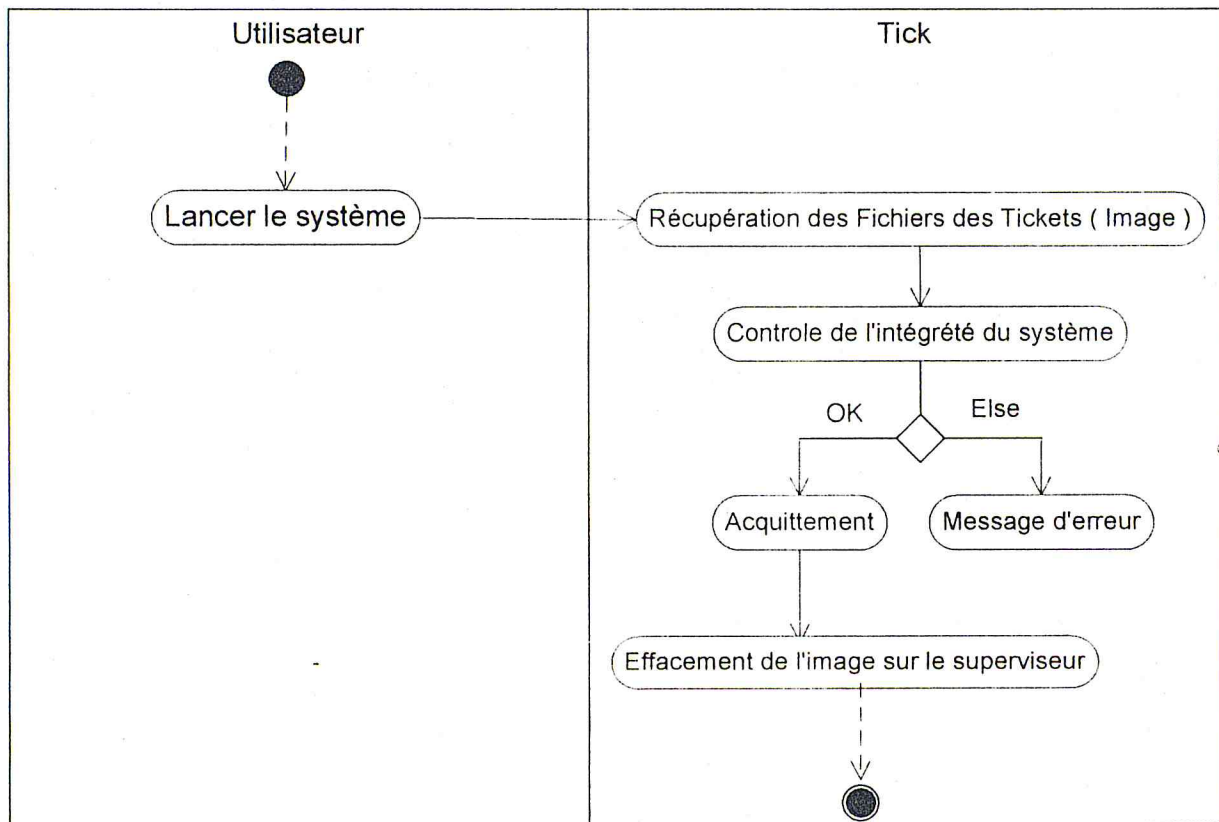
### III La Vue Dynamique (Diagramme d'activités) :

Cette vue est très importante dans les environnements multitâches; elle montre la décomposition du système en terme de processus et elle permet de définir les fonctions du système.

La vue dynamique est représentée par le diagramme d'activités pour identifier, à partir du cahier des charges, les principales composantes du système à construire.

#### I Diagramme d'activités:

##### I.1 Diagramme d'activité du sous système d'acquisition :



**Figure IV.9** Diagramme d'activité de sous système d'acquisition

**1. Activité : lancement du système :** l'acquisition est lancée soit par un ordre d'acquisition dont la période peut être paramétrée via le moniteur d'exploitation, soit par l'intervention de l'utilisateur à partir de l'IHM ;

**2. Activité :Récupération des fichiers de tickets de taxation :** Sur une station de supervision, le fichier de stockage des tickets de taxation ne peut pas être lu directement par le centre de facturation : il faut passer par un fichier intermédiaire, qui représente « l'image » de ce fichier de stockage, pris à un instant t.

**3. Activité :Vérification de l'intégralité du transport :** En raison du temps nécessaire au dépôt des tickets de facturation (manuelle ou automatique) et afin de préserver l'intégrité des données, il est nécessaire de synchroniser les traitements de récupération des ticket avec les transferts FTP.

**4. Activité :Acquittement :** le système efface l'image du fichier sur la station de supervision si la réception a été établit son erreur.

I.2 Diagramme d'activités du sous système préparation :

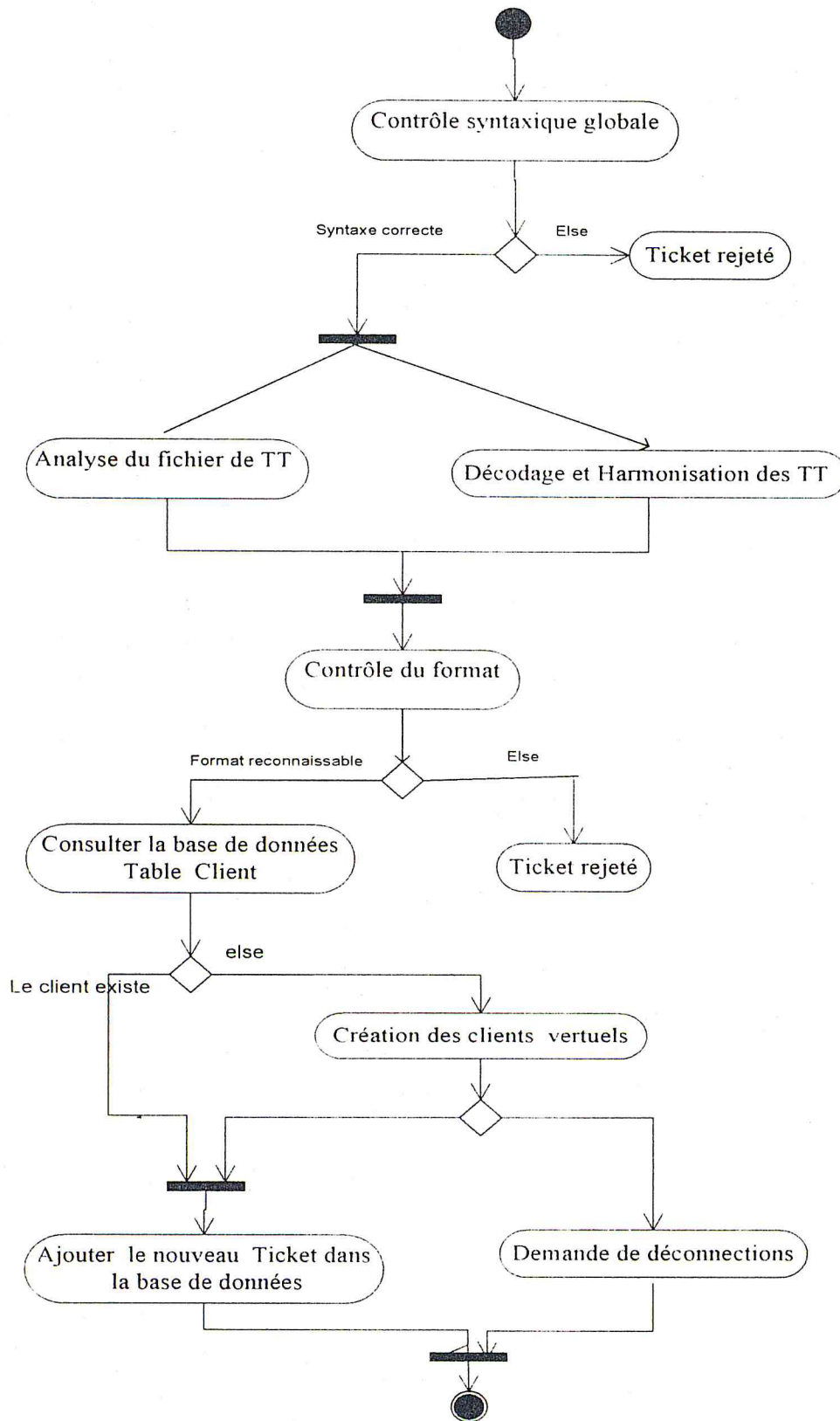


Figure IV.10 Diagramme d'activité sous système préparation



### 1. Activité : Contrôle syntaxique global :

Chaque TT subit un contrôle syntaxique global avant d'être décodé et harmonisé. Pour chaque ticket, il s'agit de contrôler que le nombre de champs est égal au nombre précisé si la taille du ticket est fixé, et de contrôler le nombre d'octets du fichier à traiter si la taille du ticket est variée.

**2. Activité : Décodage et Harmonisation des TT :** Utilisation de grammaires de description de format pour harmoniser les tickets en un format unique. Les champs utiles des tickets sont récupérés et décodés.

### 3. Activité : Contrôle du format et création des clients virtuels :

Description de règles fonctionnelle du processus de contrôle:

- ◆ Le contrôle vérifie la structure syntaxique des tickets de taxation.
  - Si un ticket de taxation ne possède pas un format reconnaissable par l'application en référence à la grammaire associée, il est rejeté dans un fichier et non récupéré par le système.
- ◆ Le sous système de préparation effectue une deuxième phase de vérification afin de déterminer si les données contenues dans le ticket de taxation sont correctes et correspondent à un client déjà existé : à ce niveau il n'y a pas de rejets de ticket ; le système va créer un **client virtuel**.
- ◆ **Création d'un client virtuel** Un ticket de taxe sortant de la phase de préparation n'est pas forcément valorisable. Il est dans ce cas contrôlé et stocké dans la base de données pour un identificateur de client virtuel ou bien il récupère une autre fois à partir des fichiers de TT (le traitement est fait à partir de la nature d'erreur).

### 4. Activité : Enregistrement des tickets harmonisés :

Dans cette phase nous allons ajouter le ticket harmonisé dans la base de données (table ticket).

## I.3 Diagramme d'Activités sous système valorisation du coût unitaire:

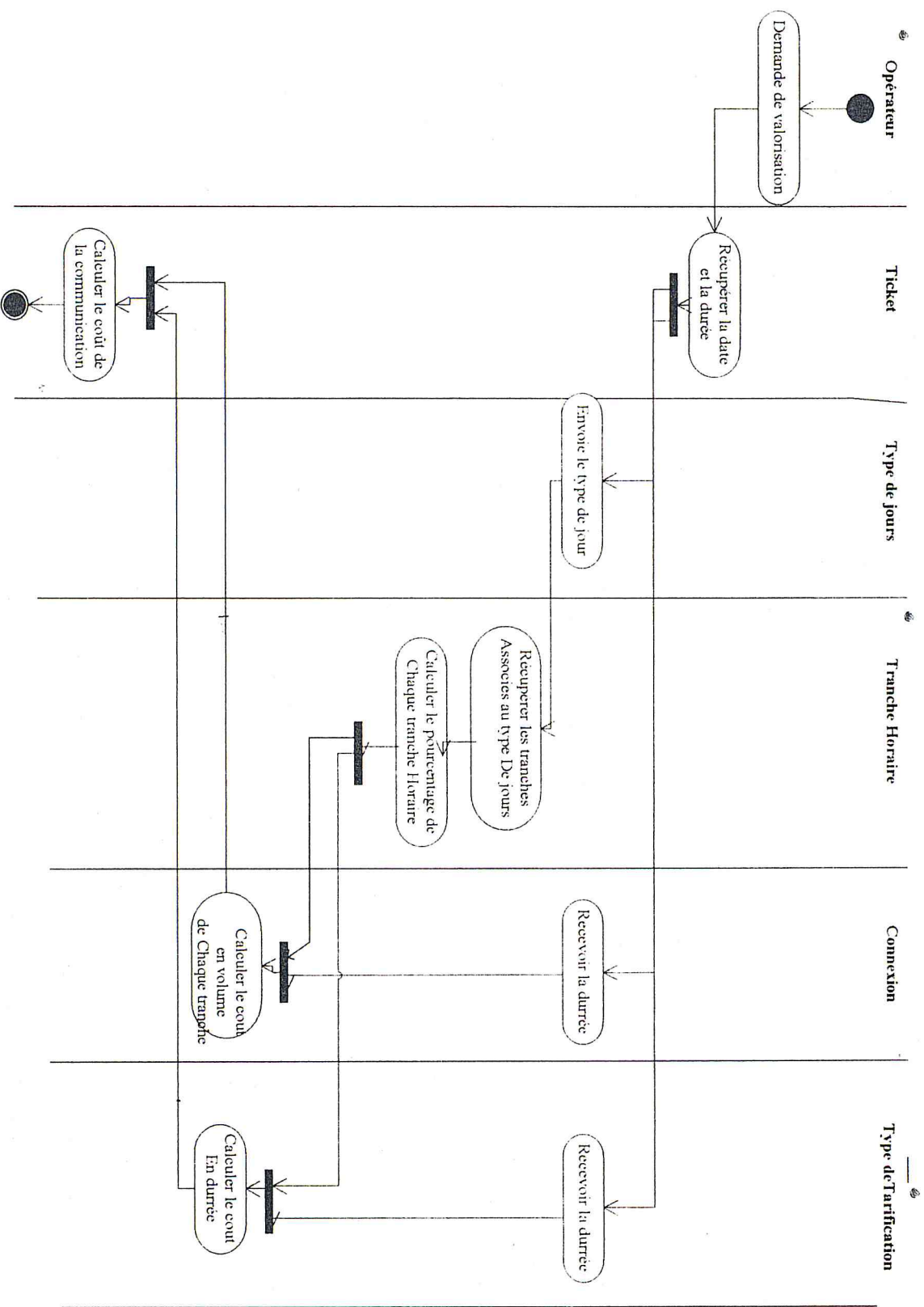


Figure IV.11 Diagramme d'activité du sous système de valorisation de coût unitaire.

### 1. Activité : Calculer le coût en volume de Chaque tranche :

Pendant la durée d'une connexion réseau un client échange un certain volume de données. Cette connexion peut s'étendre sur plusieurs tranches horaires et sur plusieurs types de jour. L'information remontée sur cet aspect au niveau du ticket est le nombre de KOctets total échangés durant la communication. A ce niveau, il est impossible de savoir le nombre de KOctets échangés dans chaque tranche horaire. Une solution de contournement consiste à pondérer le volume total échangé sur chaque tranche en fonction du temps passé dans chaque tranche horaire

$$\text{Volume sur tranche} = (\text{Temps passé sur la tranche} / \text{Temps total}) \times \text{Volume total}$$

### 2. Activité : Calculer le coût En durée :

Cette phase calcule le coût à la durée du ticket en tenant compte des réductions spécifiques abonnements et clients.

Les phases de traitement sont les suivantes :

- 1- Récupération de la durée sur le champ du ticket « *Durée du ticket* ».
- 2- Récupération du tarif à la minute :
  - On récupère le tarif concerné et valide pour la date de début de ticket de taxation .
- 3- Calcule du Coût durée détail :
  - on multiplie la durée et le tarif à la minute récupéré en 2- en tenant compte de la réduction spécifique abonnement sur la durée et de la réduction spécifique client sur la durée.

### 3. Activité : Calcul des coûts :

Les activités détaillées sont les suivantes :

Coût sur la tranche tarifaire n :

- Tarif récupéré multiplié par le volume sur la tranche en tenant compte de la réduction sur la tranche de la réduction spécifique abonnement sur le volume et de la réduction spécifique client sur le volume.

Coût sur la tranche tarifaire n réel :

- Si le type de tarification de l'accès est au forfait complet et si la communication est nationale, alors ces valeurs sont nulles
- sinon elles sont égales au coût sur la tranche tarifaire n déterminé ci-dessus.

Coût volume détail :

- On somme les Coût sur les trois tranches tarifaires .

#### **4. Activité : Stockage des données valorisées**

Les Tickets sont stockés dans la base avec toutes les informations produites par la valorisation d'usage :

- Groupe, Type d'appel ;
- Volume sur la tranche tarifaire 1, 2 et 3.
- Coût sur la tranche tarifaire 1, 2 et 3.

1.4 Diagramme d'activités du sous système Calcul des frais :

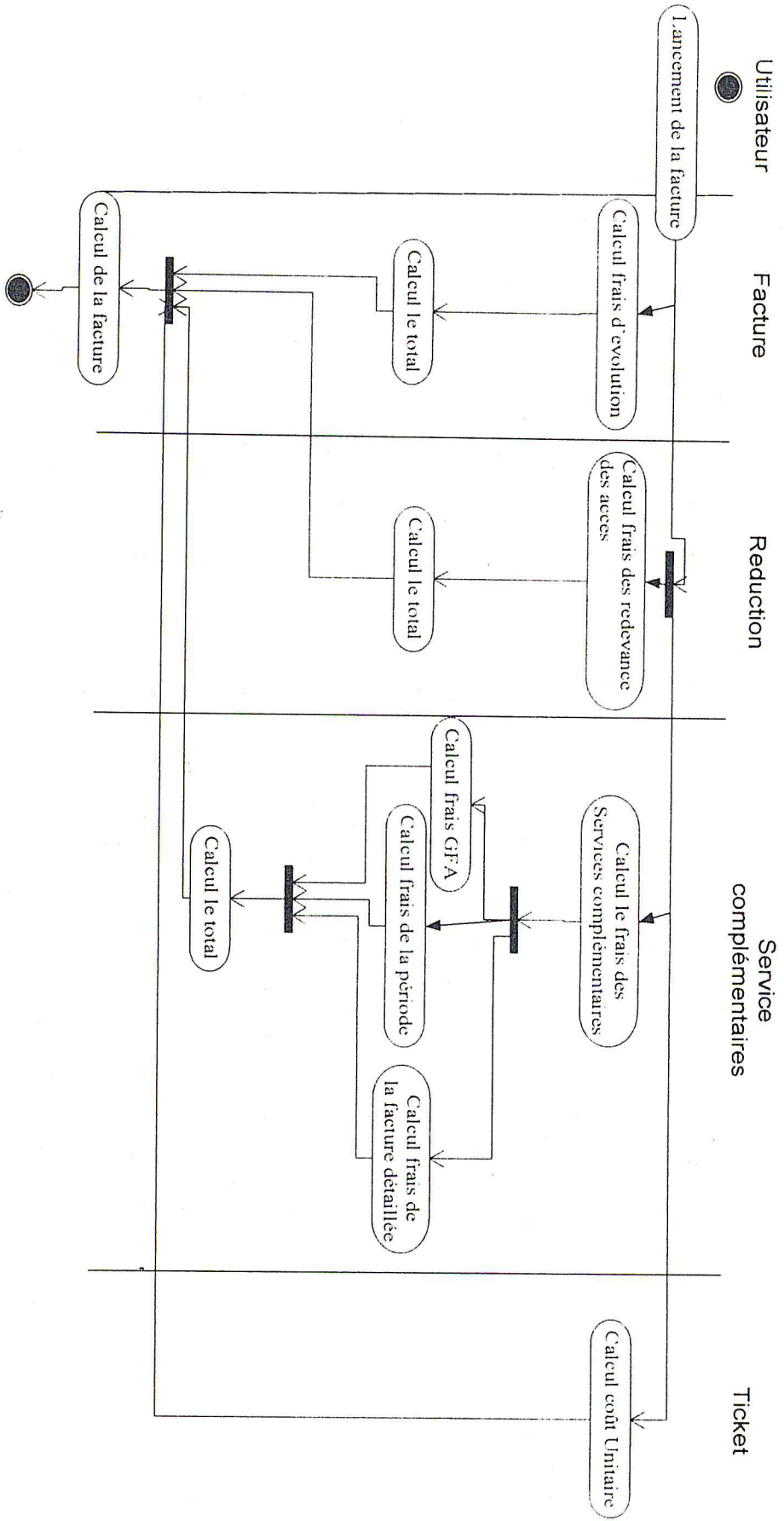


Figure V.12 Diagramme d'activité calculs de frais

### 1. Activité: Calcul frais de mise en service des accès :

Deux types de réductions sont à déduire sur les frais de mise en service des accès :

- Réduction abonnement sur les frais de mise en service (**RAFMES**) = (100 - réduction sur les frais de mise en service / 100 pour l'abonnement).
- Réduction client sur les frais de mise en service (**RCFMES**) = (100 - réduction sur les frais de mise en service) / 100 pour le client.

- Frais de mise en service (**FMES**) = Frais de mise en service X **RAFMES** X **RCFMES**

Si la date d'ouverture de l'abonnement appartient à la période de calcul de la valorisation de produit,

Sinon c'est égal à zéro.

### 2. Activité: Calcul frais d'évolution des accès :

Deux types de réductions sont à déduire sur les frais d'évolution des accès :

- Réduction abonnement sur les frais d'évolution (**RAFE**) = (100 - réduction sur les frais d'évolution) / 100 pour l'abonnement
- Réduction client sur les frais d'évolution (**RCFE**) = (100 - réduction sur les frais d'évolution) / 100 pour le client

Frais d'évolution (**FE**) =  $\sum$ (Frais d'évolution X **RAFE** X **RCFE**) sur la période de calcul de la valorisation de produit.

### 3. Activité: Calcul frais de redevance des accès :

Deux types de réductions sont à déduire sur la redevance des accès :

- Réduction abonnement sur la redevance (**RAR**) = (100 - réduction sur la redevance) / 100 pour l'abonnement
- Réduction client sur la redevance (**RCR**) = (100 - réduction sur la redevance) / 100 pour le client

#### 4. Activité: Calcul les frais des services complémentaires :

Les services complémentaires :

- le service GFA (voir le chapitre I) ;
- Autres services complémentaires ( le frais de facture détaillée, le frais de la périodicité de la facture... etc.

##### 4.1. Activité:Calcul frais des services GFA :

Redevance des services GFA (*RGFA*)= Redevance service GFA X Nombre de mois redevance (*NMR*) X Nombre de services pour le service actif à la date de facturation actuelle (*DE*)

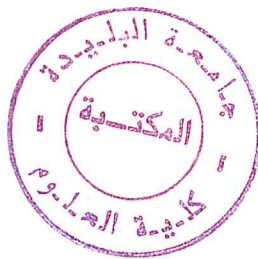
(Il est dans le champ service\_GFA= 1 de la classes services complémentaires.

##### 4.2. Activité :Calcul frais de la facture détaillée :

Frais de la facture détaillée (*FFD*) = pour le code « DETAIL ».

##### 4.3. Activité :Calcul frais de la périodicité de la facture :

Frais de la facture mensuelle (*FFM*) = pour le code « MENSUEL ».



**IV. La vue des composants :**

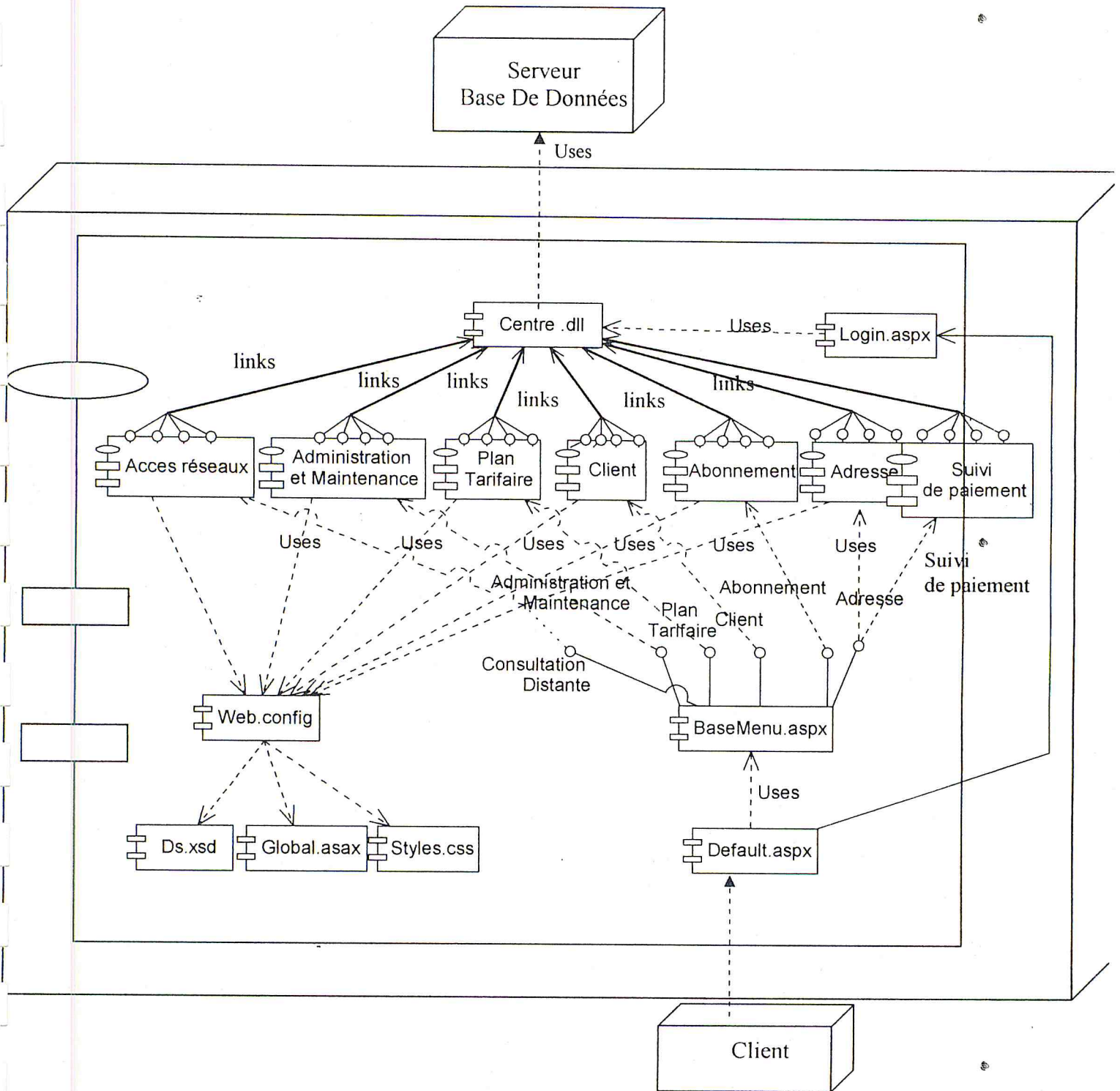


Figure IV.13 Diagramme de composants



Le client accède au système via la page d'accueil de l'application identifiée par le fichier **Default.aspx**, ce dernier utilise deux fichiers **Login.aspx** et **BaseMenu.aspx**.

Le fichier **Login.aspx** qui permet aux utilisateurs (clients qui sont sur d'autres machines) d'ouvrir ou fermer une session. Pour identifier un utilisateur le fichier **Login.aspx** doit se connecter à la base de données par l'intermédiaire de la classe **Class\_Data.vb** qui est dans le fichier **Centre.dll** qui accède au serveur de la base de données (qui est sur une autre machine).

Le fichier **BaseMenu.aspx** représente l'interface qui contient les liens hypertextes vers les différentes pages contenues dans les différents packages (Consultation Distant, Administration et Maintenance, Plan Tarifaire, Client, Abonnement, Adresse). Chacun de ces derniers packages est décrits comme suit (voir figure IV.14) :

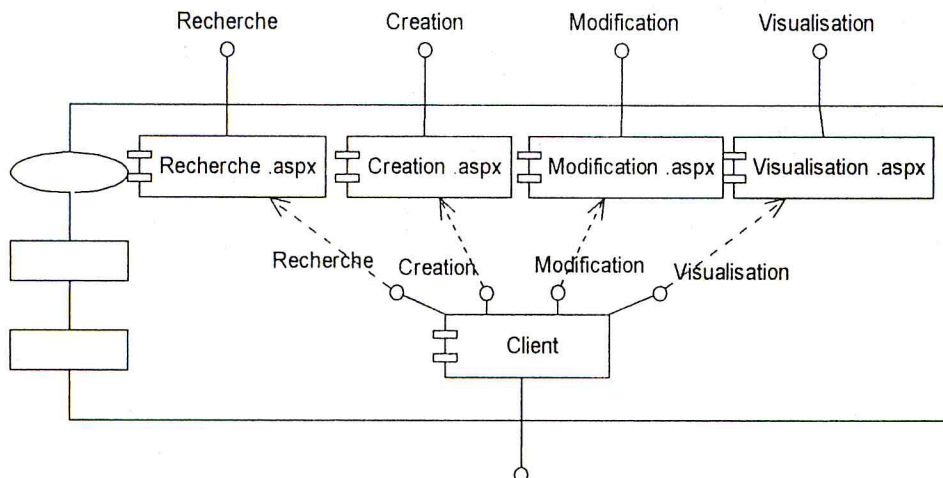


Figure IV.14 Description des composants de paquetage client

Chaque page (\*.aspx) se connecte à la base de données par l'intermédiaire de la classe **Class\_Data.vb** qui est dans le fichier **Centre.dll** qui accède au serveur de la base de données. Elle utilise aussi le fichier **Web.config** qui contient les informations de la configuration du site web de l'application. Ce dernier utilise les fichiers suivants :

- **Ds.xsd** qui représente le cache du système.
- **Global.ascx** qui représente le Proxy du site web.
- **Style.css** qui représente la feuille de style de l'interface utilisateur.

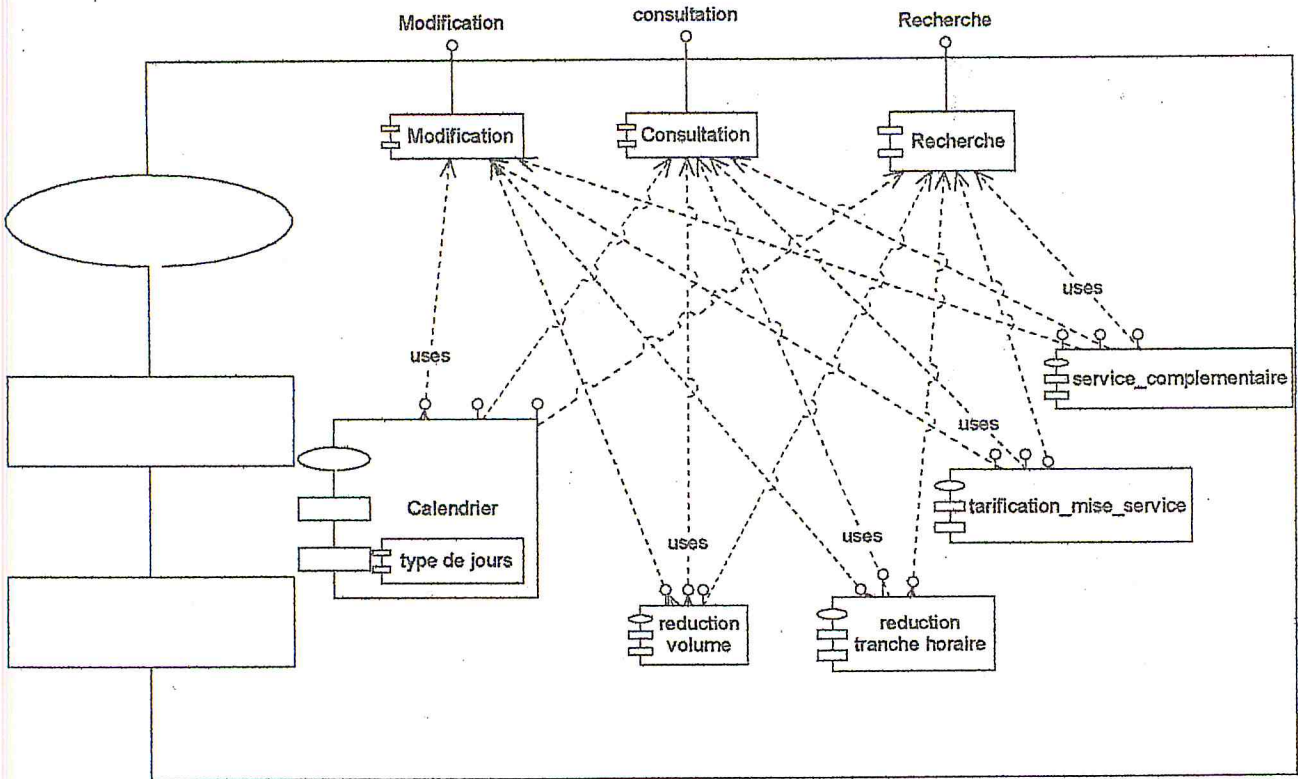


Figure IV.15 Description des composants de paquetage plan tarifaire

**IV. La vue de déploiement :**

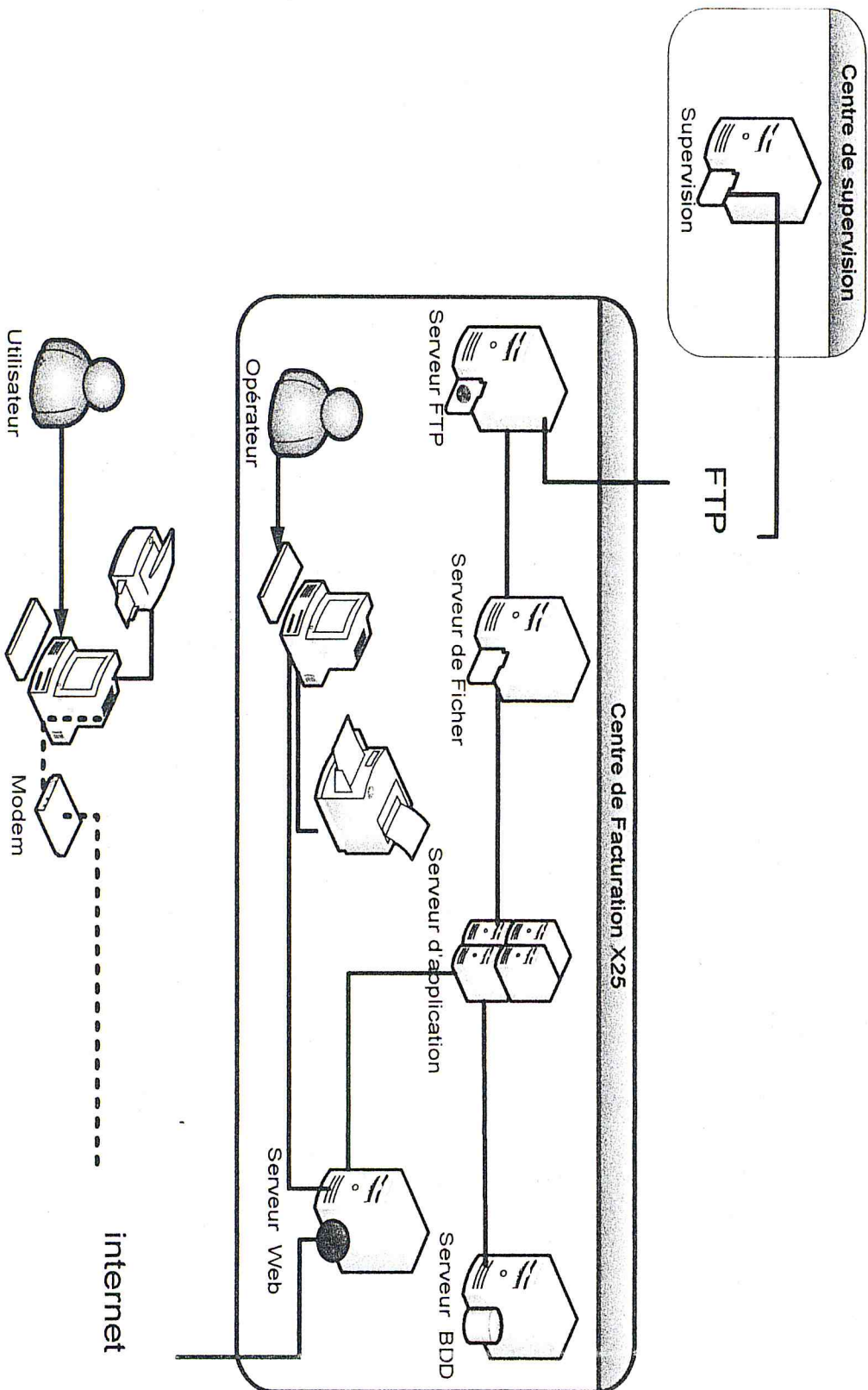


Figure IV.16 Diagramme de déploiement

### Description du diagramme de déploiement :

Notre Système est déployé en architecture *3-tiers* :

1. L'utilisateur (le demandeur de ressources) connecte avec le serveur Web via l'Internet , il possède un modem .
2. d'autre part la connexion entre le serveur FTP et la station de supervision est réalisée à travers le protocole FTP.
3. Le serveur d'application (appelé aussi **middleware**): chargé de fournir les services faisant appel au serveur secondaire (serveur de base de données).

### **Conclusion :**

Dans ce présent chapitre nous avons présenté, les étapes de développement de notre système en utilisant une démarche dans un environnement orienté objet à savoir la modélisation et la conception. Les logiciels objet sont plus extensibles, plus proches des utilisateurs, plus faciles à maintenir, mais ils restent toujours difficiles à réaliser. Dans le développement de notre système le modèle **4+1** présente la démarche suivie, et l'utilisation de la notation UML montre l'environnement.

Pour développer notre système nous avons commencé par la spécification des besoins, qui décrit les fonctionnalités du système, présentés par les diagrammes des cas d'utilisation, puis nous avons entamé le diagramme de classe et le diagramme d'activités pour représenter la vue logique et la vue dynamique, enfin diagramme de composant et diagramme déploiement.



***CHAPITRE V***  
***L'IMPLEMENTATION***

## **1. Introduction:**

A ce niveau, il s'agit d'implémenter la solution retenue au niveau de la phase de conception, c'est la phase au cours de laquelle les structures et les algorithmes définis pendant la conception sont traduits dans un langage de programmation et / ou une BDD.

### **1.1 . Choix de la technologie:**

Dans notre cas l'architecture qui devait être choisi parmi les trois technologies à partir du cahier de charge est: EJB( Entreprise Java Beans), CORBA et DCOM( ). Notre choix s'est porté sur l'architecture DCOM qui est basé sur le langage de programmation Visuel Studio.Net consiste en un ensemble de technologies visant à transformer Internet en véritable plate-forme informatique distribuée. Elle offre de nouvelles méthodes de création d'application à partir d'une série de services Web.(Voir la figure V.1)

### **1.2 . Choix du langage de programmation:**

Une des décisions les plus importantes dans la conception et la réalisation d'un logiciel est le choix du langage de programmation. Notre choix de langage de programmation est: Visuel Basic et C# .

## **1. Introduction:**

A ce niveau, il s'agit d'implémenter la solution retenue au niveau de la phase de conception, c'est la phase au cours de laquelle les structures et les algorithmes définis pendant la conception sont traduits dans un langage de programmation et / ou une BDD.

### **1.1 . Choix de la technologie:**

Dans notre cas l'architecture qui devait être choisi parmi les trois technologies à partir du cahier de charge est: EJB( Entreprise Java Beans), CORBA et DCOM( ). Notre choix s'est porté sur l'architecture DCOM qui est basé sur le langage de programmation Visuel Studio.Net consiste en un ensemble de technologies visant à transformer Internet en véritable plate-forme informatique distribuée. Elle offre de nouvelles méthodes de création d'application à partir d'une série de services Web.(Voir la figure V.1)

### **1.2 . Choix du langage de programmation:**

Une des décisions les plus importantes dans la conception et la réalisation d'un logiciel est le choix du langage de programmation. Notre choix de langage de programmation est: Visuel Basic et C# .



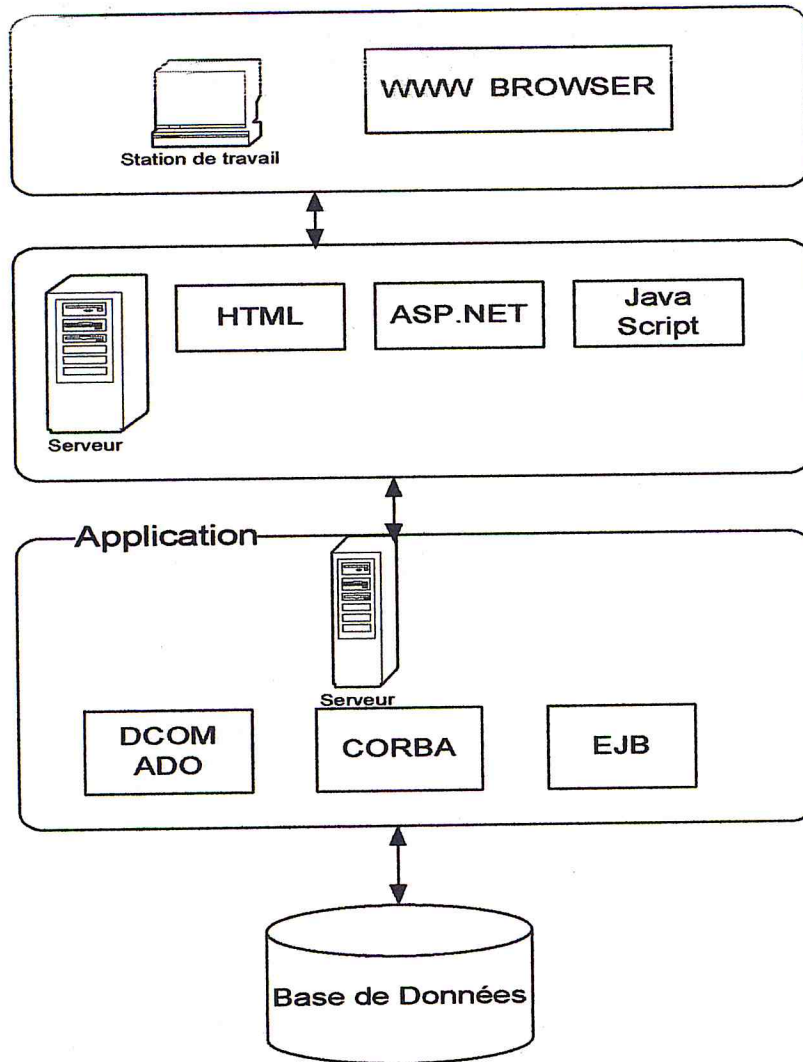


Figure V.1 : Architecture physique de l'Application

### 1.3 Création de la base de données

L'objectif primordial de la modélisation conceptuelle de données est d'aboutir à un modèle logique de données qui permettra ensuite la création d'une base de données.

Chaque approche de modélisation est plus adaptée à une catégorie spécifique de base de données. C'est ainsi que les modèles entité-association sont plus destinés aux bases de données relationnelles et les modèles conceptuels objets sont les mieux adaptés aux bases de données objets étant donné qu'ils ont des concepts similaires (héritages, polymorphisme, ...etc.). Dans notre projet nous avons développé une BDD relationnel, utilisant SQL Server (d'après le cahier de charge).

Pour créer notre base de données nous avons respecté la règle de passage de UML vers SQL Server (le passage orienté objet vers SGBDR), qui remplace l'identificateur d'objet par la clé de la table de la base donnée.

## Création Abonnement

Informations d'abonnement	
client	mezaourou
Mise en Service	
Date de saisie de l'abonnement	07/10/2004
Date d'ouverture souhaitée	10/10/2004
Adresse X25	011000020
Type de raccordement	ligne réservée <input checked="" type="checkbox"/>
Modem	Oui <input checked="" type="checkbox"/>
Débit	256 ko/s <input checked="" type="checkbox"/>
Paramètre	
Dégressivité au Volume National	Oui <input checked="" type="checkbox"/>
Volume Minimum par Transaction Nationale Aboutie	Oui <input checked="" type="checkbox"/>
Réseau X25 de Raccordement	operateur 0 <input checked="" type="checkbox"/>
Réduction sur les Redevances	60 <input type="checkbox"/> %
Réduction sur les communications nationales	60 <input type="checkbox"/> %
Reduction sur les communications internationales	<input type="checkbox"/> %
<input type="button" value="Sauvegarder"/> <input type="button" value="Annuler"/>	

Figure V.7

Figure V.7 représente le fenêtré de création d'abonnement associé à un client spécifique, elle contient les informations suivantes: -Information d'abonnement;

- Mise en service;
- Paramètre.

## Modification Evolution en fonction du Volume

**Detail**

Code

Volume Min

Volume Max

pourcentage

	Code	volume min	Volume max	pourcentage
<input type="button" value="Edit"/>	t1	1	3	80
<input type="button" value="Edit"/>	1	0	45	0
<input type="button" value="Edit"/>	2	45	53	2
<input type="button" value="Edit"/>	3	53	61	23
<input type="button" value="Edit"/>	4	61	82	50
<input type="button" value="Edit"/>	5	82	91	56
<input type="button" value="Edit"/>	6	91	99	60
<input type="button" value="Edit"/>	7	99	infini	72

Figure V.8

Figure V.8 est la fenêtre de modification Evolution en Fonction du Volume.

## Modification de Tarif de mise en service

**Detail**

Raccordement  Débit

Tarif des frais de mise en service (en DA)

Tarif des frais d'évolution

	Raccordement	Débit	tarif des frais de mise en service(en DA)	tarif des frais d'évolution (en DA)
<input type="button" value="Edit"/>	Accès direct	128	1000	2000
<input type="button" value="Edit"/>	Accès direct	256	700	610
<input type="button" value="Edit"/>	Accès direct	64	150	120
<input type="button" value="Edit"/>	Accès indirect par ligne banalisée	128	0	500
<input type="button" value="Edit"/>	Accès indirect par ligne Réservee	256	310	500

Figure V.9

Figure V.9 est la fenêtre de modification de Tarif mise en service.

## Réduction Utilisables pour les tranches Horaires

Detail				
Code	base	Pourcentage de Réduction	0	00.00
Type de réduction	Réduction de type 2			
Libelle	pas de réduction			
<input type="button" value="Ajouter"/> <input type="button" value="Mise à jour"/> <input type="button" value="Supprimer"/>				
Edit	Code	Libelle	Pourcentage	Type
<input type="button"/>	Base	10 Cv Supplémentaires	00.00	Réduction de type 0

Figure V.10

Figure V.10 est la fenêtre de réduction Utilisables pour les tranches Horaires.

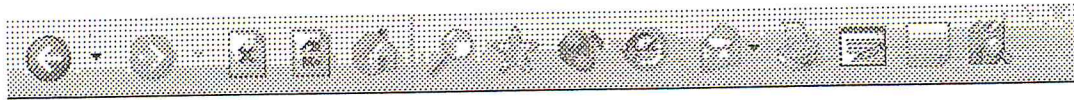
## Modification de Services Complémentaires

Detail			
Libellé	paquets de longueur 10	Tarif	75.00
Code <input type="text"/>			
<input type="button" value="Ajouter"/> <input type="button" value="Mise à jour"/> <input type="button" value="Supprimer"/>			
Edit	Code	libelle	Tarif
<input type="button"/>	1	10 Cv Supplémentaires	10.00
<input type="button"/>	2	numéro abrégé	3.00
<input type="button"/>	3	accès aux GFA	5.00

Figure V.11

Figure V.11 est la fenêtre de modification de services Complémentaires.





## Initialisation des types de jour

jour ouvrable		jour ouvré		jour férié
Jeu	ouvré	Sam	ouvré	Ven
		Dim		
	ouvrable	Lun	fé	
		Mar		
		Mer		

saisir l'année jusqu'à laquelle le calendrier sera définie

2050

Sauvegarder

Figure V.12

Figure V.12 est la fenêtre d'Initialisation de Type de jour.

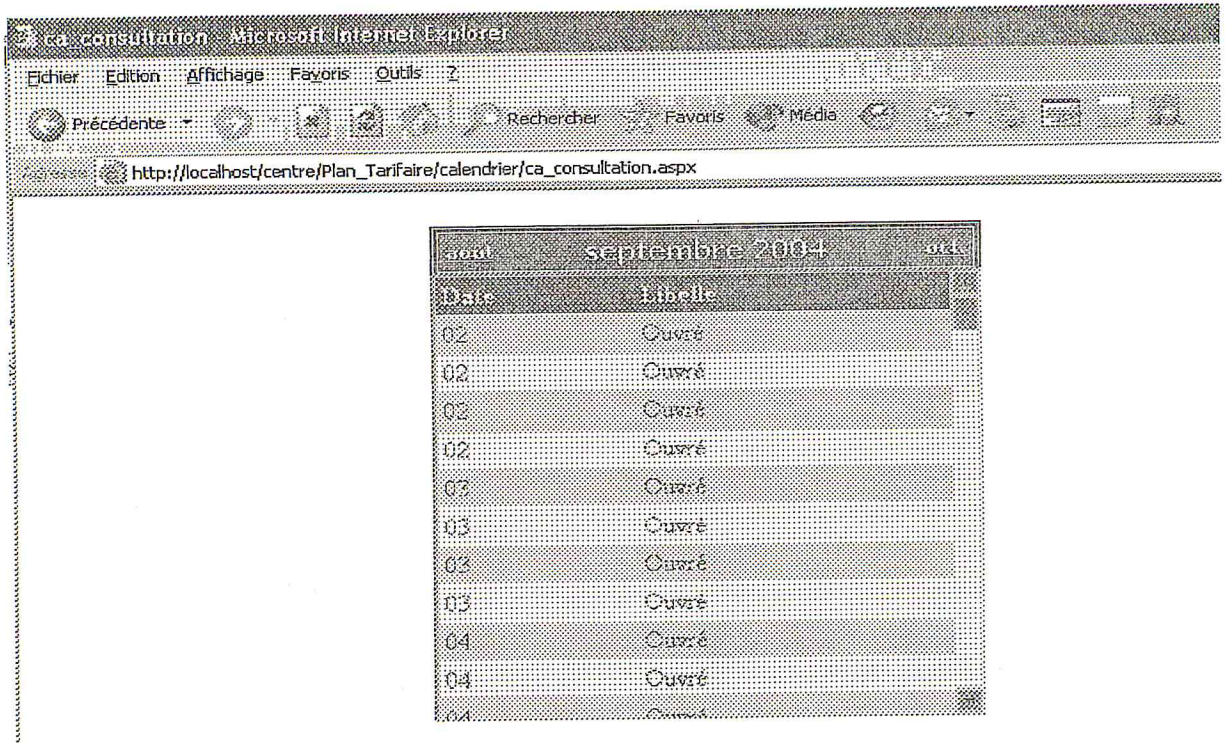


Figure V.13

Figure V.13 est la fenêtre de Calendrier.

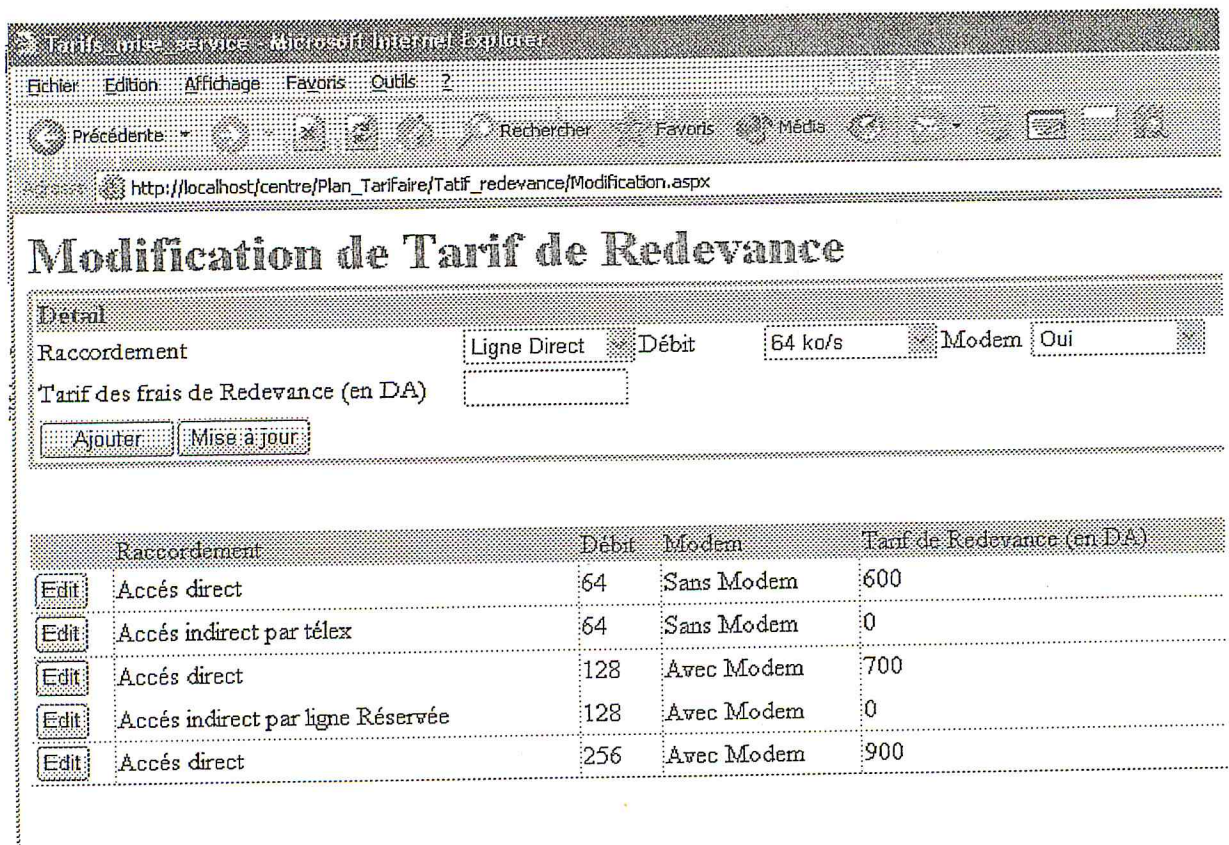


Figure V.14

Figure V.14 est la fenêtre qui permet de modifier le tarif redevance.

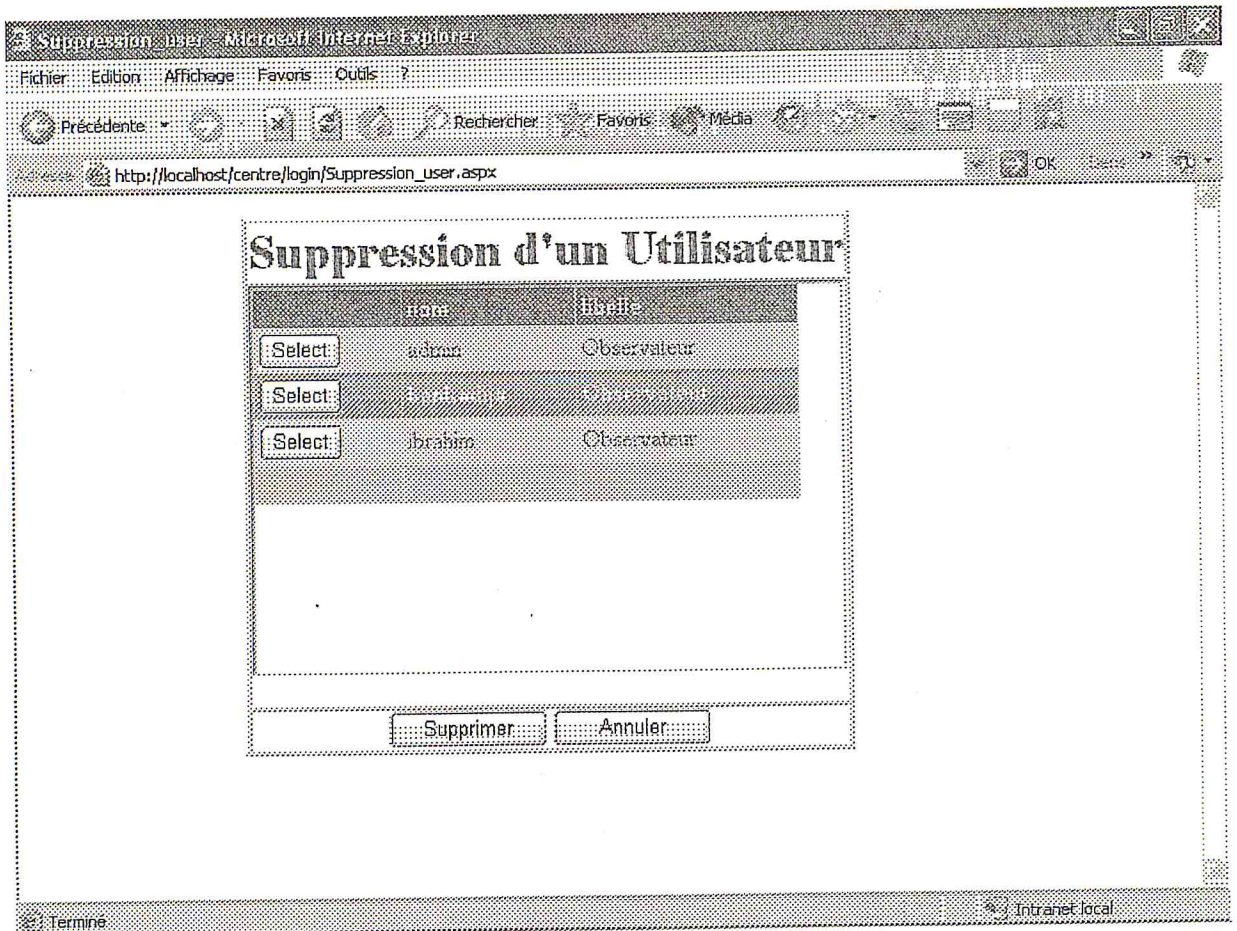


Figure V.15

Figure V.15 est la fenêtre qui permet de supprimer un utilisateur.

Agence commerciale:	Agence principale d'Alger	Numéro de téléphone:	021305360
Adresse:	Alger centre		
Numéro du client:	1	Numéro de la facture:	1
Raison sociale:	efim	Date d'édition:	08/10/04
Code postal:	09100	Date de la précédente facture:	08/10/04
Adresse:	Boudjetou	Date limite de règlement:	25/10/04
Numéro de téléphone:	024421120	Numéro fiscal:	14051982
Adresse destinataire:	intelsa-tigaza		
Numero d'accès:	123456789	Nombre de VL:	12
Frais de mise en service de l'accès:	123456789.90	Débit:	64 Kbit/s
Frais d'évolution de l'accès:	123456789.90	Redevance nombre de VL:	55335533.33
GFA: Redevance abonnement:	129236789.90	Redevance GFA:	55335533.33
Nombre de communications:	111111	Redevance autres services complémentaires:	11111111141.33
Volume sur tranche tarifaire 1 (kOctet):	4444444	Coût tranche	1:123446789.90
Volume sur tranche tarifaire 2 (kOctet):	32323232	Coût tranche 2:	123456789.90
Volume sur tranche tarifaire 3 (kOctet):	66666666	Coût tranche 3:	123456789.90
Durée totale de communications (mn):	666654	Coût pour la durée:	123456789.90
Nombre de communications:			
Volume sur tranche tarifaire 1 (kOctet):	111111	Coût tranche 1:	123456789.90
Volume sur tranche tarifaire 2 (kOctet):	12121212	Coût tranche 2:	123456789.90
Volume sur tranche tarifaire 3 (kOctet):	66666666	Coût tranche 3:	123456789.90
Durée totale de communications (mn):	986532	Coût pour la durée:	123456789.90
Frais totaux de mise en service des accès:	443366887.88	Forfait complet:	443366887.88
Frais d'évolution des accès:	443366887.88	Forfait à la durée:	443366887.88
Redevance totale des abonnements:	443366887.88	Communications nationales:	443366887.88
Redevance totale des services complémentaires:	443366887.88	Communications internationales:	443366887.88
Redevance grosse GFA:	443366887.88		
Numéro du client:	1	Numéro de la facture:	20001228321
Raison sociale/Nom du client:	7777777777	Date d'édition:	11/01/2020
Montant H.T. de la facture:	666666666.66	Date limite de règlement:	22/02/2020
Montant de la T.V.A. (DA):	933333333.33		
Montant T.T.C. de la facture (DA):	222222222.33		
Régularisation des précédentes factures (dt ou avoir T.T.C. DA):	333333333.66		
NET A PAYER (DA):	123361233.33		

Figure V.16

Figure V.16 est présente la Facture Globale.



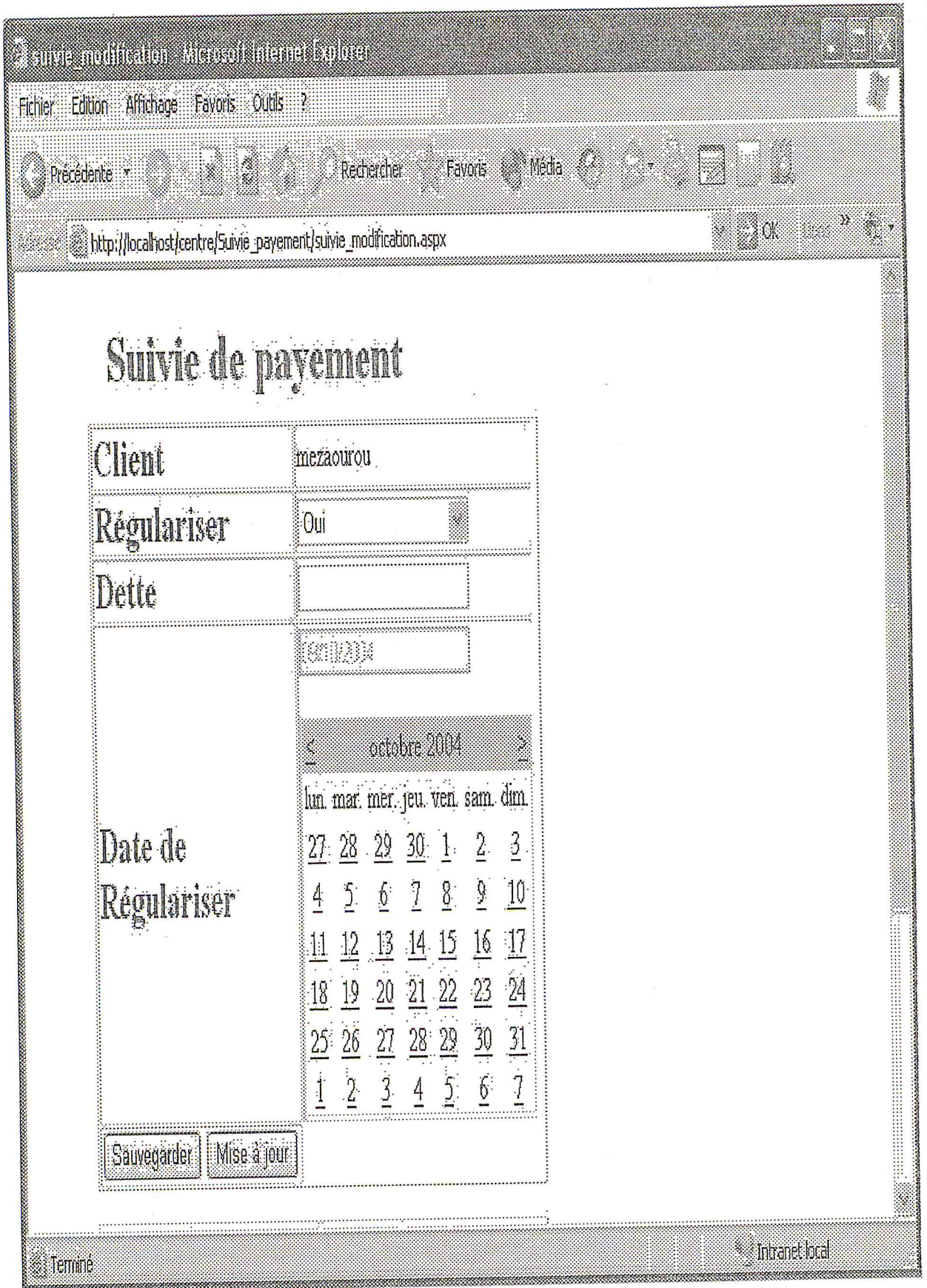


Figure V.17

Figure V.17 présente Suivi des paiements.

L'objectif primordial de ce mémoire consiste à analyser, concevoir et réaliser un système de gestion commercial d'un réseau de données pour un opérateur de télécommunication.

La présence d'un processus de développement formalisé, bien défini et bien géré est un facteur de réussite d'un projet. Pour cette raison nous avons suivi le modèle 4+1, utilisant UML comme un langage de modélisation.

Au cours de notre projet nous avons développé La gestion commerciale des produits d'un réseau de données qui couvre toutes les activités liées à son exploitation tels que: la gestion des abonnés, des abonnements, des produits, des tarifs, des politiques tarifaires, l'ouverture effective d'un abonnement, la résiliation effective d'abonnements, la reprise d'un abonnement, l'ajout ou retrait de service, la facturation des produits et usages, le suivi des paiements, l'ouverture sur d'autres logiciels, notamment ceux de comptabilité, le suivi de l'exploitation par l'établissement de statistiques etc..

Ce système est simple, accessible et convivial d'utilisation, fournir aux usagers des factures de clients donnés pour des périodes données, Divers états (exemple: visualisation des tables de la base de données), Les messages d'erreurs (ou alarmes). Nous avons interprété également le méta modèle UML pour la modélisation orienté objet aboutissant à un schéma d'implémentation en utilisant l'environnement Visuel Studio.Net, il s'agit d'un architecture DCOM ainsi que la création de la base de données par SQL Server.

Ce projet très enrichissant nous a donné l'occasion de nous plonger dans un problème concret et pratique, ce qui nous a permis d'affiner les connaissances acquises lors de notre projet. Nous avons eu la chance d'être intégré à un projet important impliquant un grand nombre de personnes et faisant appel à des technologies novatrices.

Le début a été difficile, ce qui est principalement dû à la multiplicité des concepts d'architecture DCOM et le langage de programmation Visuel Studio.Net et des notations UML. Cependant, ces difficultés nous ont permis d'acquérir un esprit de synthèse, ce qui est loin d'être négligeable.

Ce projet nous a aussi permis de progresser en terme de rigueur, de méthodologie et aussi bien concernant la conduite de projet que la programmation. Le projet nous a de plus permis d'approfondir nos connaissances sur la gestion des réseaux, sur le la plate forme .Net et UML, sur le langage visuel Basic et C#, ainsi que l'acquisition de connaissance sur des techniques récentes de modélisation de données et tout particulièrement la méta modélisation.

---

## ANNEXE

---

### I - Présentation de la plate-forme Microsoft .NET :

#### I.1- La plate-forme .NET :

Cette section présente les grandes lignes de l'architecture de la plate-forme .NET. La plate-forme .NET se compose de plusieurs fonctionnalités et services de la base, comme l'illustre la figure A.1, l'un des objectifs de cette nouvelle plate-forme est de simplifier le développement Web. [8], [9]

#### I.2- Technologie de base de la plate-forme .NET :

Les technologies de base qui composent la plate-forme .NET sont les suivantes : [8], [9]

- .NET Framework :

Cette technologie se fonde sur un nouveau Common Language Runtime. Celui-ci fournit un ensemble commun de services pour les projets créés avec Visual Studio.NET, indépendamment du langage. Ces services fournissent des blocs modulaires de base pour les applications de tous types, utilisables à tous les niveaux des applications.

Microsoft Visual Basic, Microsoft Visual C++, Microsoft Visual C# et d'autres langage de programmation Microsoft ont été améliorés pour tirer profit de ces services.

- .NET Building Block Services :

C'est un ensemble de services programmables distribués disponibles à la fois en ligne et hors connexion. Un service peut être appelé sur un ordinateur autonome non connecté à Internet : il peut également être fourni par un serveur local fonctionnant au sein d'une entreprise. .NET Building Block Services peut être utilisé à partir de n'importe quelle plate-forme prenant en charge SOAP. Au nombre de ces services, citons : les calendrier, les annuaire, la notification et la messagerie... etc.

- Visual Studio .NET :

Constitue un environnement de développement de haut niveau, destiné à la création d'application sur le .NET Framework. Il fournit des technologies clés afin de simplifier la création.

- .NET Enterprise Server :

Les produits .NET Enterprise Server permettent une évolutivité, une fiabilité, une gestion de l'intégration. Ils offrent en outre un grand nombre de fonctionnalité par exemple : Microsoft SQL Server 2000, Microsoft Commerce Server 2000.

La figure suivante décrit les technologies de base du plate forme Microsoft.NET

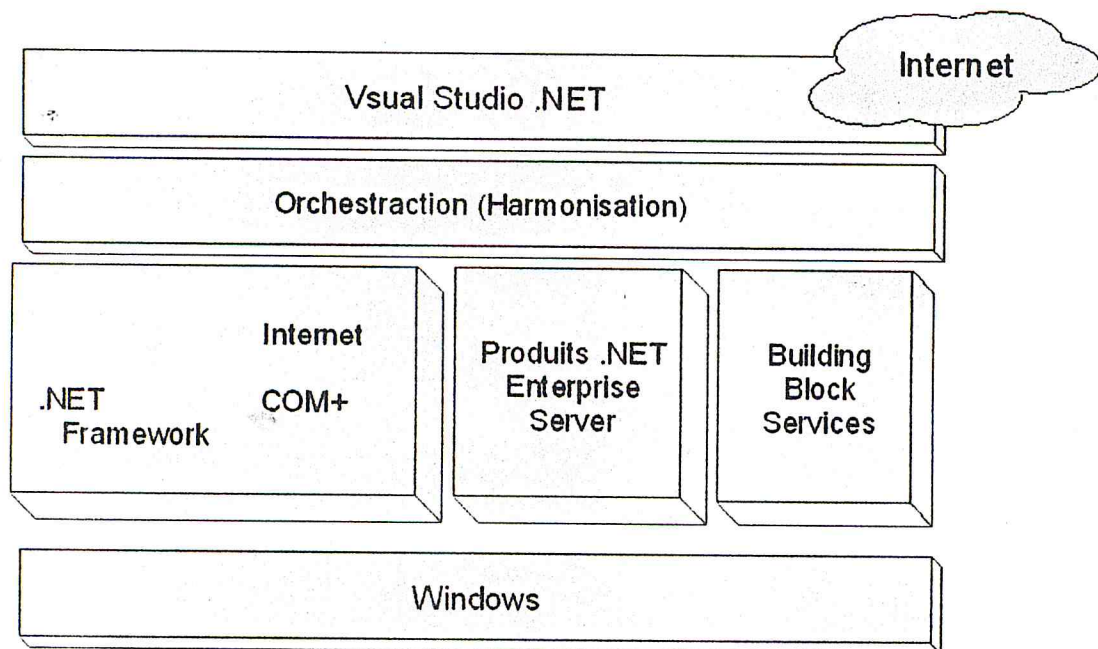


Figure A.1 Plate forme Microsoft .NET

### I.3- Les avantages de la nouvelle technologie :

La plate-forme .NET offre les avantages suivants : [8], [9]

- Modèle de programmation cohérent et indépendant du langage, utilisable à tous les niveaux d'une application ;
- Interopabilité parfaite entre technologies ;
- Migration aisée à partir des technologies existantes ;
- Prise en charge totale des technologies Internet fondées sur des standards et indépendantes des plate-forme, telles que [http<sup>1</sup>](#), XML et SOAP.

<sup>1</sup> [http](#) : Hypertext Transfer Protocol

- Grâce au Common Language Runtime, tous les langages compatibles avec la plate-forme .NET vont utiliser les mêmes fichiers d'exécution. Il n'est donc plus nécessaire de distribuer des bibliothèques d'exécution spécifiques à un seul langage, parce que les fichiers d'exécution .NET seront installés automatiquement dans les versions de Microsoft.

## II -Présentation du .NET Framework :

Le .NET Framework fournit tous les services communs nécessaire pour l'exécution de nos applications ; Ces services sont disponibles dans tous les langages compatibles avec .NET grâce à la spécification CLS<sup>2</sup>. [8], [9]

Cette figure décrit ces services :

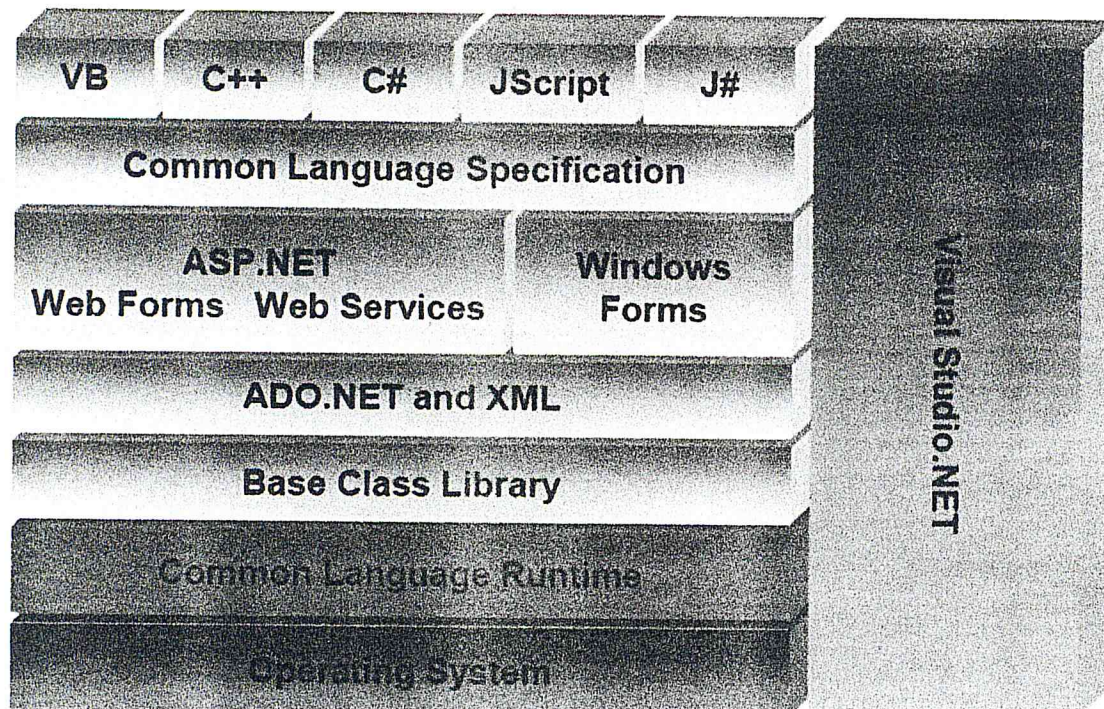


Figure A.2 Services communs pour l'exécution des applications

<sup>2</sup> CLS :Common Language Specification

## II.1 - Création de composants dans le .NET Framework :

Avant l'avènement de COM, les applications étaient des entités totalement séparées, sans aucune intégration ou très peu. Grâce à COM, nous pouvons intégrer des composants au sein d'une même application et à plusieurs, en exposant des interfaces communes. Dans le .NET Framework, les composants possèdent une base commune. Il n'est plus nécessaire d'écrire le code visant à permettre aux objets d'interagir directement les uns avec les autres. Dans notre environnement le .NET Framework prend totalement en charge les classes, l'héritage, les méthodes, les propriétés, le polymorphisme, les constructeurs et d'autres constructions orientées objet. [8], [9]

## II.2- Spécification CLS (Common Language Specification) :

La spécification CLS définit les standards communs que doivent respecter les langages et les développeurs pour que leurs composants et applications puissent être largement utilisés par d'autres langages compatibles avec le modèle .NET. Elle permet aux développeurs Visual Basic .NET, Visual C++ ou d'autres langages de créer des applications dans le cadre d'une équipe multi-langage, avec l'assurance que l'intégration des différents langages s'effectuera sans problème. CLS permet même aux développeurs Visual Basic .NET ou Visual C++ ...etc d'hériter de classes définies dans des langages différents. [8], [9]

## II.3 Visual Studio .NET:

Dans le .NET Framework, Visual Studio .NET fournit les outils servant au développement rapide d'applications. [8], [9]

## II.4 Les Langages du .NET Framework :

Cette section présente les langages que Microsoft fournit avec Visual Studio .NET pour le .NET Framework. [8], [9]

- **Visual Basic .NET** : Nouvelle version de Visual Basic avec des innovations substantielles en terme de langage.
- **C#**: Il s'agit du premier langage moderne orienté composant.
- **Extensions C++**: offre plus de puissance et de contrôle.

- **J# .NET** : est un langage pour les développeurs java qui souhaitent créer des applications et des services pour le .NET.
- **Langages tiers** : divers langage tiers prennent en charge le .NET : COBOL, Pascal, SmallTalk... etc.

### III Présentation des composants .NET Framework :

Les composants du .NET Framework sont les suivants :

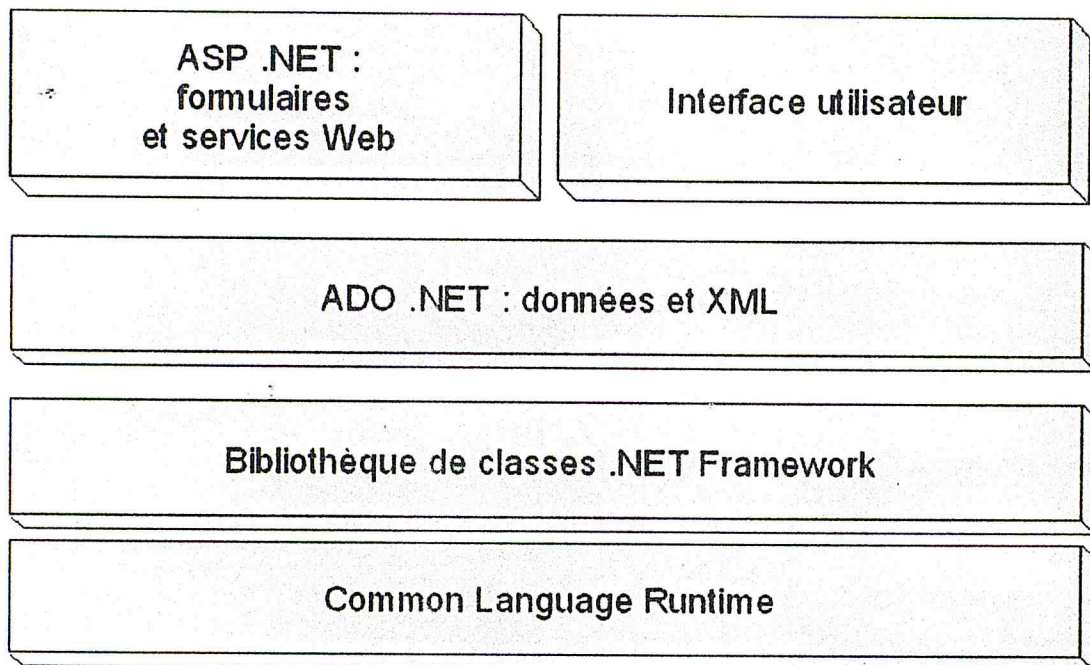


Figure A.3 Composants du .NET Framework

#### III.1- Common Langage Runtime :

Il simplifie le développement d'application, fournit un environnement d'exécution robuste et sécurisé, prend en charge plusieurs langages, simplifie le déploiement et la gestion des applications et offre u environnement géré

##### III.1.1- Composants du Common Langage Runtime :

Les fonctionnalités du Common Langage Runtime sont décrites dans la figure suivante :



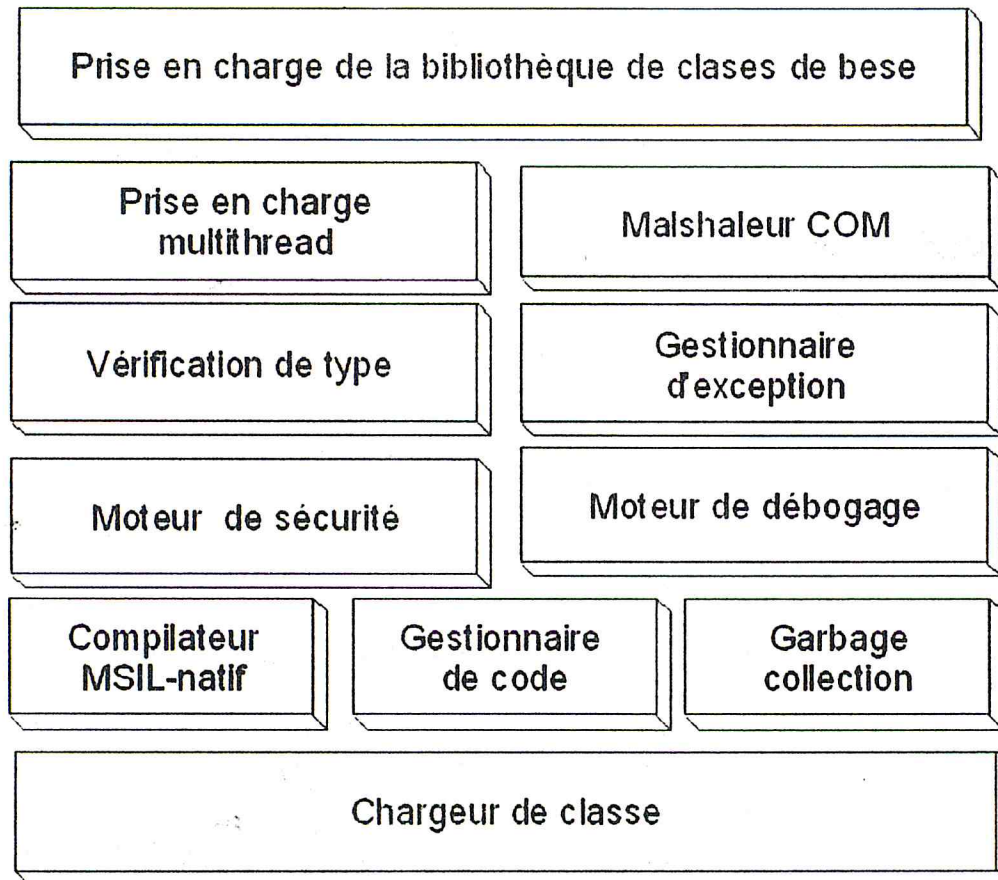


Figure A.4 Composants du Common Language Runtime.

- **Chargeur de classe** : charge en mémoire l'implémentation d'un type chargeable et le répare à l'exécution.
- **Compilateur MSIL<sup>3</sup> - natif** : convertit MSIL en un code natif.
- **Gestionnaire de code** : gère l'exécution du code.
- **Garbage collection** : fournit une gestion automatique de la durée de vie de tous nos objets.
- **Moteur de sécurité** : fournit une sécurité par preuve, fondée sur l'origine du code en plus de l'utilisateur.
- **Moteur de débogage** : permet de déboguer l'application et de tracer l'exécution du code.
- **Vérification de type** : n'autorisera pas les conversions non sécurisées ou les variables non initialisées.

<sup>3</sup> MSIL : Microsoft Intermediate Language

- **Gestionnaire d'exceptions** : fournit un traitement structuré des exception
- **Prise en charge multithread** : fournit des classes et des interface qui permettent la programmation multithread.
- **Marchaleur COM** : fournit le marshaling à partir et à destination de COM.
- **Prise en charge de la bibliothèque de classes.NET Framework** : intègre du code au runtime qui prend en charge la bibliothèque de classes.

### III.2-Bibliothèque de classes .NET Framework :

Elle fournit de nombreuses nouvelles fonctionnalités puissantes du runtime et d'autres services essentiels de haut niveau via une hiérarchie d'objets qui s'appelle un espace de nom. [8], [9]

La figure suivante décrit ces espaces de nom :

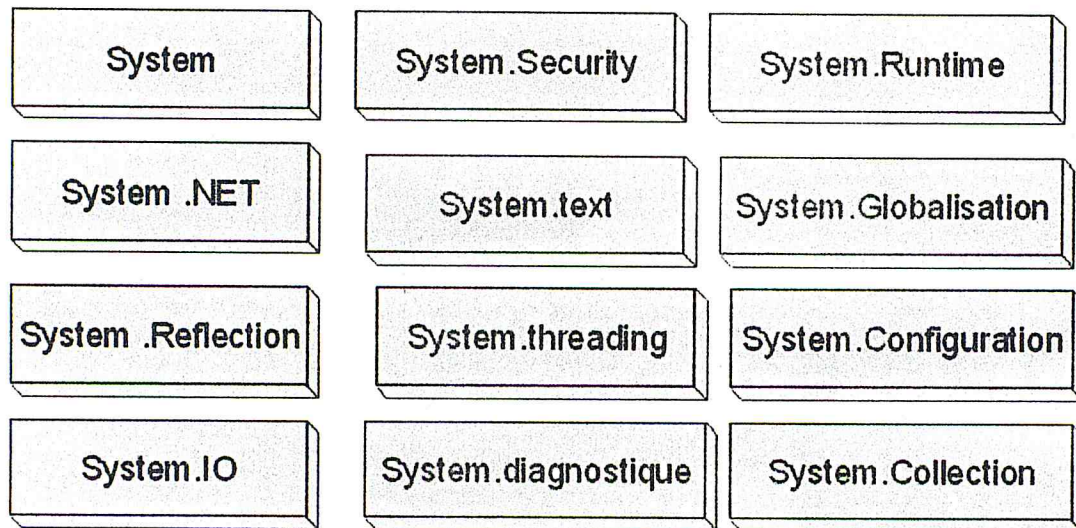


Figure A.5 Bibliothèque de classes .NET

- **System** : contient des classes fondamentales et de base qui définissent les types de données, les événements, les interfaces, les attributs... etc.
- **System.Collection** : fournit des listes, des tables et d'autres méthodes de regroupement de données.
- **System.IO** : il s'agit d'entrée /sortie et flux de fichiers.

- **System.NET** : fournit une prise en charge des sockets et de TCP/IP.

Pour plus d'information consulter la documentation du SDK Microsoft .NET Framework.

### III.3- ADO .NET données et XML :

ADO.NET est la nouvelle génération de la technologie ADO<sup>4</sup>. Son but est l'amélioration du modèle de programmation déconnecté, ainsi elle est riche de XML. [8], [9]

- **System.Data** : comprend la classe **DataSet** qui représente des tables multiples et leur relations.
- **System.XML** : il comprend un outil d'écriture et un analyseur XML.

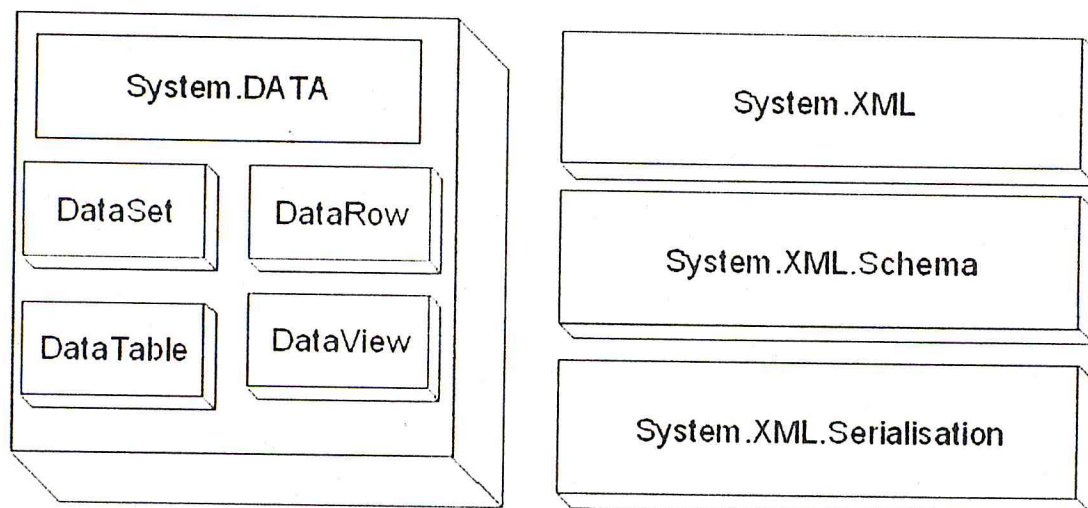


Figure A.6 ADO.NET

### III.4- ASP .NET formulaires et services Web ( Active Server Pages ) :

ASP.NET est un cadre de programmation élaboré sur la base du Common Language Runtime et qui peut être employé sur un serveur pour créer des applications Web puissantes. Les formulaires ASP.NET sont des outils d'emploi pour la création d'interface utilisateur Web dynamiques. [8], [9]

<sup>4</sup> ADO : ActiveX Data Object

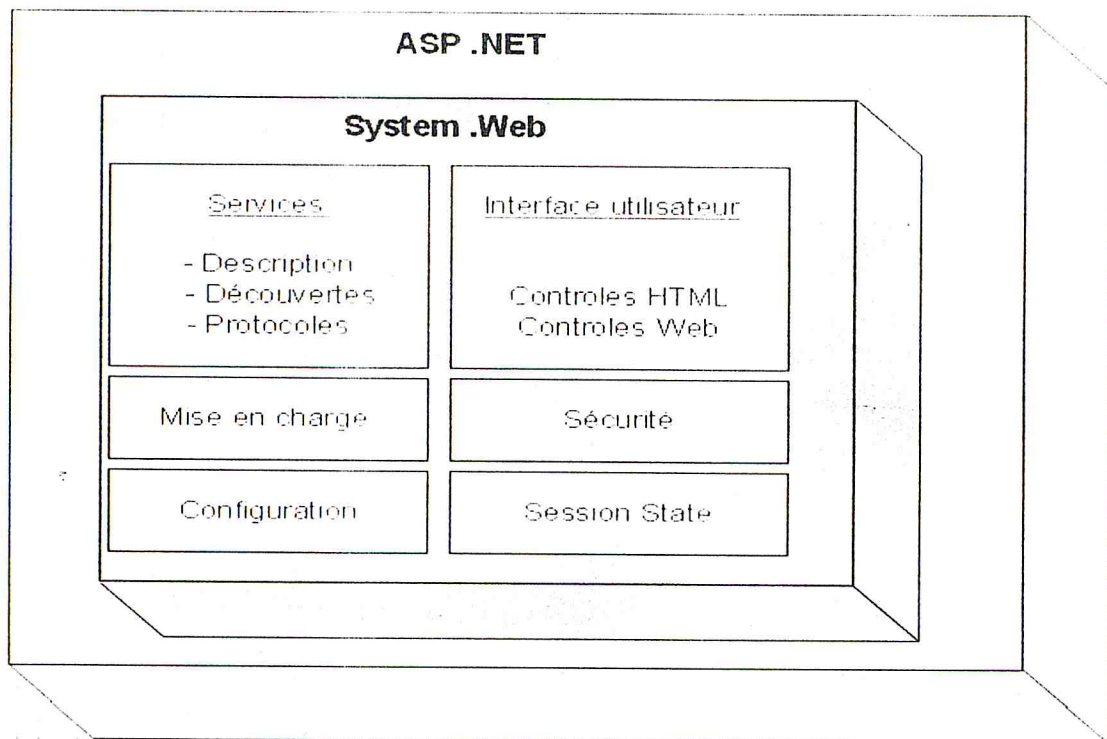


Figure A.7 Présentation des formulaire et services Web : ASP .NET

Dans **System.Web**, certains services tels que la mise en charge, la sécurité ou la configuration sont partagés par les services Web et l'interface.

### III.5 Interface utilisateur :

La figure suivante explique comment le .NET gère l'interface Framework des applications Windows traditionnelles :

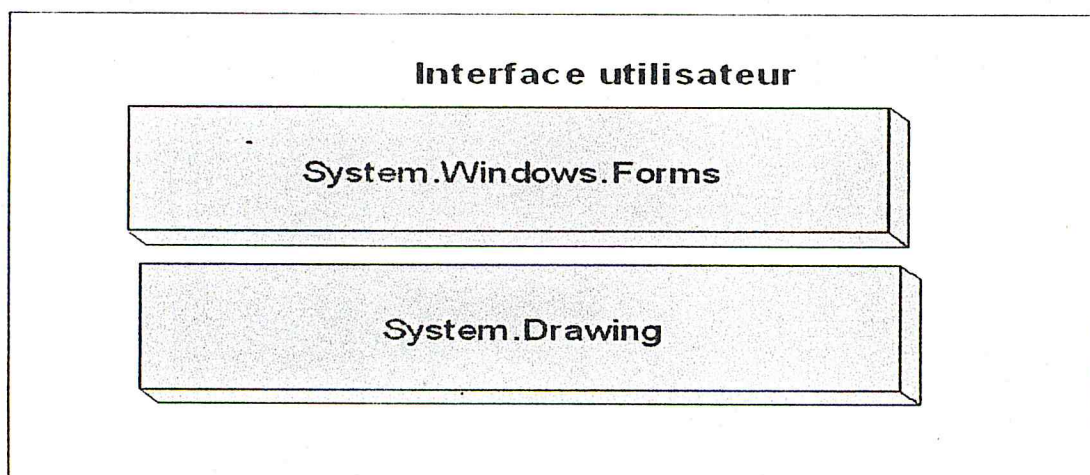


Figure A.8 Interface utilisateur

- **System.Windows.Forms** représente l'interface utilisateur côté client, tandis que **System.Drawing** représente la nouvelle génération de services GDI+ (Graphic Device Interface Plus).

## **Internet Information Serveur (I.I.S.).**

L'un des problèmes de base des services W.W.W. avec la programmation de documents HTML est l'aspect statique des pages qui ne permettent que la consultation d'informations.

La professionnalisation d'Internet et du World Wide Web (WWW) a permis la création de serveur WWW assurant l'exécution de programme, et l'accès aux données. Les documents visualisés par les navigateurs Web sont ainsi devenus dynamiques.

Cette présentation repose sur le système propriétaire MicroSoft, avec l'utilisation de Windows NT server 4.0 qui offre un serveur W.W.W. à l'installation par l'intermédiaire d'Internet Information Serveur (version 3 après correctif du service pack 3 I.I.S.3), qui permettra l'exécution de programme ou de gérer l'accès à des bases de données.

En préalable on considère l'installation de Windows NT et donc du serveur WWW complète, ainsi que le serveur WWW démarré.

Le serveur World Wide Web I.I.S. fournit avec Windows NT 4.0 est à la base du développement et la gestion de site Internet ou Intranet : c'est à dire qu'il va permettre la publication de documents au format ASPX, la gestion de formulaires interactifs avec des applications, des bases de données, ... ainsi que la gestion de transferts de fichiers, ou encore la publication d'archives d'informations.

### **Les composants de IIS :**

Internet Information Server inclut les composants suivants :

- les services Internet : **WWW** (World Wide Web) est le service qui permet la gestion des documents HTML, mais aussi certains accès aux bases de données comme on pourra le voir ultérieurement.
- GOPHER** est un service plus ancien et aujourd'hui très peu utilisé qui permet d'organiser un système d'information où les fichiers sont classés en fonction d'une arborescence (sur le modèle des répertoires sous DOS).
- FTP** (File Transfert Protocol) est le service qui va permettre les transferts de fichiers.

## Bibliographie :

<i>Référence</i>	<i>Description</i>
[1]	Guy Pujolle, « Les Réseaux », Eyrolles, 2003
[2]	, « Concepts fondamentaux des réseaux»,
[3]	Didier Donsez, « Architecture des Réseaux », Université de Valenciennes, Institut des Sciences et techniques de Valenciennes
[4]	
[5]	Pierre-Yves Cloux, « RUP, XP Architectures et outils 'industrialiser le processus de développement' », Dunod, Paris, 2003.
[6]	Philippe Kruschten, « THE 4+1 VIEW MODEL OF ARCHITECTURE », IEEE, November 1995.
[7]	Rady Booch, «Software Architecture and the UML», Rational, 1999
[8]	Microsoft, Programmation avec Microsoft Visual Basic.net, 2003 Cours 2374A ref n X08-80353
[9]	Microsoft, Présentation de Microsoft ASP.NET, 2002 Cours N°2680A ref n x08-84792
[10]	Christian Soutou, « Conception de base de données », Eyrolles, Paris, 2002
[11]	James Rumbaugh et al, « OMT : Modélisation et conception », Masson, 2000
[12]	Pierre-Alain Muller, « Modélisation objet avec UML », Eyrolles, 1997

- [13] F. Bernardi, « Méthode d'analyse orienté objet UML », Dunod, 2002
- [14] Pierre-Alain Muller, « Modélisation objet avec UML », Eyrolles, 1997
  
- [15] Rémy Fannader, Hervé Lerroux, « UML principes de modélisation », Dunod, 2000
  
- [16] Joseph Gabay, « Merise vers OMT et UML », Masson, 1998
  
- [17] Conférence « ingénierie des systèmes d'information et ingénierie des connaissances », 2et 3 décembre 1999, Paris.
  
- [18] Richard C.LEE et William M.Tepfenhart, « UML et C++ guide pratique du développement orienté objet », CROSS,1998.
  
- [19] Grady Booch, James Rumbaugh, Ivar Jacobson. "Le guide de l'utilisateur UML", EYROLLES, 2003
  
- [20] Georges Gardarin, « BASE de données Objet et relationnel », Eyrolles, paris 2002
  
- [21] BRICE CHARVIER, « Editeur de diagrammes de séquence UML Rapport- version 02 », Organisme ESSI. 2004
  
- [22] Grady Booch, Object-Oriented Analysis and Design with Applications, 2<sup>nd</sup> edition,IEEE, 1995

## Les Sites Utiles :

<i>Référence</i>	<i>Description</i>
[1]	<a href="http://www.fdigallo.online.fr">http://www.fdigallo.online.fr</a>
[2]	<a href="http://www.infeig.unique.ch">http://www.infeig.unique.ch</a>
[3]	<a href="http://www.jerome.capirrossi.org">http://www.jerome.capirrossi.org</a>
[4]	<a href="http://www.commentcamarche.net">http://www.commentcamarche.net</a>
[5]	<a href="http://www.rational.com/uml">http://www.rational.com/uml</a>
[6]	<a href="http://www.labo-net.com">http://www.labo-net.com</a>

