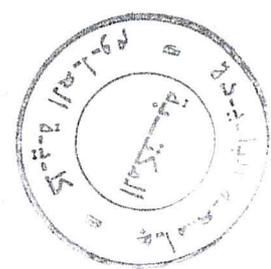


République Algérienne Démocratique et Populaire.
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.



Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.
Option : Système d'information Avancée

Sujet

Conception et Réalisation selon
l'approche composant d'un système de
gestion commerciale des produits de
réseau X25.

Présenté par : Saadi AbdelFetah

Promoteur : Monsieur D.BENNOUAR

Organisme d'accueil : Laboratoire de recherche et de développement des systèmes
Informatisés (LRDSI) – Université de Blida.

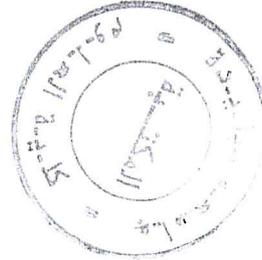
Soutenue le : 26 / 09 /2005 , devant le jury composé de :

- | | |
|-----------------------|-----------|
| Madame Abed | Président |
| Monsieur Boukhlef | Examineur |
| Mademoiselle Aousette | Examineur |

République Algérienne Démocratique et Populaire.
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.



Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.
Option : Système d'information Avancée

Sujet

**Conception et Réalisation selon
l'approche composant d'un système de
gestion commerciale des produits de
réseau X25.**

Présenté par : Saadi AbdelFetah

Promoteur : Monsieur D.BENNOUAR

Organisme d'accueil : Laboratoire de recherche et de développement des systèmes
Informatisés (LRDSI) – Université de Blida.

Soutenue le : 26 / 09 /2005 , devant le jury composé de :

Madame Abed

Président

Monsieur Boukhlef

Examineur

Mademoiselle Aousette

Examineur

*Je dédie cet humble travail
à mes très chers parents
à mes frères et sœurs
à mes amis.*

Remerciements

J'exprime mes plus grande reconnaissances et mes vifs remerciements à mon promoteur **Monsieur D. BENNOUAR** pour le temps et l'attention qu'ils ont bien voulu consacrer au bon déroulement de ce travail.

Je remercie très vivement toute l'équipe du département informatique, et tout le personnel, pour les moyens qu'ils ont mis à notre disposition , tous les conseils et les encouragements qu'ils nous ont prodigués tout au long de ce projet.

Mes gratitudes les plus sincères à tous ceux qui ont voulu juger ce travail.

J'exprime mes plus profonde reconnaissance au **Madame ABED** ainsi qu'aux membres du département informatique et de laboratoire de recherche et de développement des systèmes informatisés (LRDSI) pour leur conseil, leurs suggestions et pour l'aide qu'il me en fournis.

Enfin, je remercie tous les enseignant qui ont contribué à ma formation, **Monsieur BOUKHLEF, Madame OUKID, Madame OUAHRANI**, ainsi que tous ceux qui ont participé de près ou de loin à la réalisation de ce travail.

MIG-004-74-1



RESUME :

Le système de gestion commerciale de réseau X25 a pour objet de générer les factures des clients de l'opérateur utilisant le réseau à partir des tickets de taxations émis et des données clients. De plus, il met à disposition un fichier contenant les informations nécessaires au service comptable, Le but est de mettre en place un site capable de facturer les clients de l'opérateur et ce, quelque soit le réseau utilisé. Ce système à été réalisé selon l'approche composant qui occupe aujourd'hui une partie importante du paysage informatique, notamment dans les domaines du génie logiciel, des systèmes distribués ou encore des réseaux. Cette approche décrit un système logiciel comme un ensemble de composants qui interagissent entre eux par le bais de connecteurs, elle décrit aussi explicitement les composants et définit des mécanismes opérationnels permettant leur réutilisation.

ABSTRACT:

The marketing gestion system, the net X25 is the generator of object and client invoice operate using the net from the tax tickets inited, and give all client information, and give file that includes the essentiel information to the calculating service. The objective is to install a sit that can facturate the clients of the operator. This system has been realised in depend of the composant approche that occupy today an important part of the computer world speacially in the fields of software architecture, the diffused system or also the net works, this approche describes a software system as a set of composant that react between here by the way of connectors, it also describes explicitely the composant and defines operationnel that permit

ملخص

ان نظام التسيير التجاري لشبكة المعلومات يهدف الى استخراج فواتير المشتركين للمتعامل العمومي في مجال الاتصالات انطلاقا من بطاقات تسعيرية و معلومات عن المشتركين بالاضافة الى هذا يضع ملف يحتوي على المعلومات الضرورية لمصلحة المحاسبة الهدف هو وضع نظام قادر على فواتير المشتركين مهما كان نوع الشبكة المستعمل. دا النظام صمم على طريقة العناصر التي هي محل اهتمام اليوم في مجال المعلوماتية خاصة في مجال هندسة البرامج و الانظمة الموزعة و الشبكات. تعتمد هذه الطريقة على تقسيم النظام الى مجموعة عناصر مستقلة و محاولة الربط فيما بينها و جمعها في الاخير للحصول على النظام الكامل.

SOMMAIRE

INTRODUCTION GENERALE.....	1
I. INTRODUCTION AU RESEAU X25	
I.1. Introduction	4
I.2. Présentation de réseau X25	4
I.2.1. Niveau physique.....	5
I.2.2. Niveau trame.....	5
I.2.3. Niveau paquet	5
I.2.3.1. Gestion de circuit virtuel.....	5
I.2.3.2. Voie logique.....	6
I.2.3.3. Contrôle de flux.....	6
I.2.3.4. Multiplexage.....	6
I.3. Structure des paquets	6
I.4. Les Procédures X.25	8
I.4.1. Etablissement d'un circuit virtuel	8
I.4.1.1. Choix des voies logiques	8
I.4.1.2. Demande d'établissement côté appelant.....	8
I.4.1.3. Demande d'établissement côté appelé.....	9
I.4.1.4. Réponse à la demande d'établissement côté appelé....	9
I.4.1.5. Confirmation d'établissement côté appelant.....	9
I.4.2. Libération d'un circuit virtuel	9
I.5. Transfert des données	10
I.6. La supervision de réseau	10
I.6.1. Ticket de taxation	10
I.6.2. Station de supervision	11
I.6.3. Constitution du fichier de ticket de taxation	11
I.6.4. Les commutateurs	11
II. LES LANGAGES DE DESCRIPTION D'ARCHITECTURE (ADL)	
II.1. Introduction.....	12
II.2. Les concepts de base des Langages de Description d'architecture	12
II.2.1. Le composant	13
II.2.1.1. Le type de Composant.....	13
II.2.1.2. La sémantique d'un Composant.....	14
II.2.1.3. Les contraintes d'un Composant.....	14
II.2.2. Le connecteur	14
II.2.2.1. Le type.....	15
II.2.2.2. La sémantique.....	15
II.2.2.3. Les contraintes.....	15
II.2.3. La configuration ou architecture	15
II.2.3.1. Un formalisme Commun.....	16
II.2.3.2. La composition.....	16
II.2.3.3. Le raffinement et la traçabilité.....	17

II.2.3.4. L'hétérogénéité.....	17
II.3. Les principaux ADLs	17
II.4. Les principales phases d'un processus de développement.....	18
II.4.1. La phase d'analyse.....	18
II.4.2. La phase de conception.....	18
II.4.3. La phase d'implémentation.....	18
II.4.4. La phase de déploiement.....	19
II.4.5. La phase de maintenance	19

III. PRESENTATION DE LANGAGE ARCHJAVA

III.1. Introduction.....	20
III.2. Le modèle Archjava	20
III.3. La composition des composants	21
III.4. Les composants et les ports en langage Archjava	21
III.4.1. Pour créer un composant dans le langage Archjava	21
III.4.2. Pour créer un port dans le langage Archjava	22
III.4.3. Interconnexion de ports	22
III.4.3.1. Pour interconnecter deux ports de deux composants différents	22
III.4.3.2. Pour interconnecter un port avec le port d'un composant englobant	23
III.5. Exemple récapitulatif	23
III.6. Les connecteurs personnalisés	25
III.7. La construction d'une architecture dynamique	26

IV. DESCRIPTION DE SYSTEME

IV. Etude existant et analyse	27
IV.1 objectif de système	27
IV.1.1. Objectifs principaux	27
IV.1.2. Objectifs complémentaire	27
IV.2. Elément fondamentaux à la réalisation du système	28
IV.2.1. Les tickets de taxation	28
IV.2.2. La tarification	28
IV.2.3. Validité des tickets	29
IV.2.4. Les tranches horaires	29
IV.2.5. Groupe de ticket	30
IV.2.6. La Grammaire	31
IV.3. Architecture générale du système	31
IV.3.1. L'Acquisition	32
IV.3.2. Pré – traitement.....	33
IV.3.3. Valorisation des tickets	35
IV.3.3.1. Enrichissement des TT.....	35
IV.3.3.2. Valorisation des TT.....	35
IV.3.3.3. Réconciliation.....	36
IV.3.3.4. Stockage.....	36
IV.3.3.5. Rejet et recyclage.....	36

IV.3.3.6. Pondération.....	36
IV.3.4. Sous système coûts des services	36
IV.3.4.1. Lancement de l'opération «coûts des services ».....	37
IV.3.4.2. Période de calcul de l'opération «coûts des services ».....	38
IV.3.5. La Facturation	41
IV.3.5.1. Date de lancement de la facturation	42
IV.3.5.2. Calcule des paramètres de la Facturation	42
IV.3.5.2.1 Phase 1.....	42
IV.3.5.2.2 Phase2.....	43
IV.3.6. Gestion des clients et abonnements	48
IV.3.7. Etat divers	49

V. CONCEPTION DE SYSTEME SELON L'APPROCHE COMPOSANT

V.1 Introduction	50
V.2. Architecture globale de système	50
V.3. Conception des sous composants	52
V.3.1. Sous composant « Pré - traitement»	52
V.3.1.1. Sous composant «Contrôle»	55
V.3.1.2. Sous composant «Décodage »	57
V.3.1.3. Sous composant «Harmonisation»	58
V.3.1.4. Sous composant «Journalisation »	61
V.3.1.5. Sous composant «dé-doublonnage »	61
V.3.2. Sous composant «calcule des coûts»	62
V.3.2.1. Sous composant «Usage Traitement»	64
V.3.2.2. Sous composant «Réconciliation».....	67
V.3.3. Sous composant «Alarme»	69
V.3.4. Sous composant «coûts des services»	69
V.3.4.1. Sous composant «Produit Traitement»	70
V.3.5. Sous composant «base de données»	73
V.3.6. Sous composant «base de Fichiers»	86
V.3.7. Sous composant «Facturation»	88
V.3.7.1. Sous composant «Facturation usage»	89
V.3.7.2. Sous composant «Facturation produit»	92
V.3.8. Sous composant «Acquisition».....	92
V.3.9. Sous composant «IHM»	93
V.4. Conclusion	94

VI. MISE EN ŒUVRE

VI.1. Introduction.....	95
VI.2. La chaîne des traitements	95
VI.2.1. Pré – traitement.....	95
VI.2.2. Calculs des coûts	96
VI.2.3. Coûts des services.....	96
VI.2.4. Facturation	97

VI.3. Application IHM.....	97
VI.3.1. Présentation Générale.....	97
VI.3.2. Présentation détaillée de quelques onglets.....	99
VI.3.2.1. L'onglet client.....	99
VI.3.2.2. L'onglet abonnement.....	100
VI.3.2.3. L'onglet suivi de paiement	101
VI.3.2.4. L'onglet plan tarifaire.....	101
VI.3.2.5. L'onglet archivage et purge.....	102
VI.3.2.6. L'onglet utilisateur.....	103
VI.3.2.7. L'onglet traitements.....	104
VI.3.2.8. L'onglet facturation.....	106
VI.3.3. Les messages.....	107
VI.3.4. Les icône d'opération.....	109
CONCLUSION GENERALE.....	110

LISTE DES TABLEAUX

Tableau I.1 Format d'un paquet X25.....	6
Tableau V.1 Exemple des champs utiles et générique des TT.....	54
Tableau V.2 Format de ticket d'entrée utilisée.....	56
Tableau V.3 Format de ticket après décodage et récupération des CU des TT.....	57
Tableau V.4 Structure du ticket harmonisé.....	59
Tableau V.5 Les règles d'harmonisation.....	60
Tableau V.6 Stockage des tickets harmonisés.....	60
Tableau V.7 Codification des champs.....	65
Tableau V.8 Table Abonnement.....	79
Tableau V.9 Table client.....	79
Tableau V.10 Table Ticket_OK.....	80
Tableau V.11 Table accès X25.....	80
Tableau V.12 Table type de jour.....	81
Tableau V.13 domaine de validité de la table type de jour.....	81
Tableau V.14 table Réduction.....	81
Tableau V.15 Table réseau X25.....	81
Tableau V.16 Domaine de validité de la table réseau X25.....	82
Tableau V.17 Table Nature ticket.....	82
Tableau V.18 Domaine de validité de la table nature ticket.....	82
Tableau V.19 table circuit virtuel.....	82
Tableau V.20 domaine de validité de la table circuit virtuel.....	82
Tableau V.21 table Débit.....	82
Tableau V.22 domaine de validité de la table débit.....	83
Tableau V.23 table modem.....	83
Tableau V.24 domaine de validité de modem.....	83
Tableau V.25 table raccordement.....	83
Tableau V.26 domaine de validité de la table raccordement.....	83
Tableau V.27 Table tarification accès.....	84
Tableau V.28 table groupe pays.....	84
Tableau V.29 table tranche horaire.....	84
Tableau V.30 table réduction tranche horaire.....	84
Tableau V.31 table services complémentaires.....	84
Tableau V.32 table tarification abonnement.....	85
Tableau V.33 table tarification client.....	86
Tableau V.34 Codification proposé au niveau BDD.....	91

LISTE DES FIGURES

Figure I. 1 Couche réseau.....	4
Figure I. 2 Présentation de Réseau X25.....	4
Figure I. 3 Circuit Virtuel.....	5
Figure III. 1 Représentation Graphique d'un composant.....	21
Figure III. 2 Représentation des Ports.....	21
Figure III. 3 Interconnexion des Ports.....	22
Figure III. 4 Connecteur de délégation.....	23
Figure III. 5 Exemple de compilateur simplifié.....	23
Figure IV. 1 Fonctionnement générale de système.....	28
Figure IV. 2 Tranche Horaire.....	30
Figure IV. 3 Groupe Pays.....	30
Figure IV. 4 Architecture générale du système.....	32
Figure IV. 5 Représentation d'opération coûts des service.....	37
Figure V. 1 Architecture globale de système.....	51
Figure V. 2 L'architecture globale de sous composant «Pré – traitement ».....	53
Figure V. 3 L'architecture globale de sous composant «calcul des coûts».....	63
Figure V. 4 Pondération des volume.....	66
Figure V. 5 Sous Composant Alarme.....	69
Figure V. 6 L'architecture globale de sous composant «coûts des services».....	69
Figure V. 7 L'architecture globale de sous composant «base de données ».....	73
Figure V. 8 Diagramme de classe.....	76
Figure V. 9 Le modèle navigationnel.....	78
Figure V. 10 L'architecture globale de sous composant «base de Fichiers».....	86
Figure V. 11 L'architecture globale de sous composant «Facturation».....	88
Figure V. 12 L'architecture globale de sous composant «Acquisition».....	92
Figure VI.1 Exécution de sous composant Pré – traitement.....	95
Figure VI.2 Exécution de sous composant calculs des coûts.....	96
Figure VI.3 Exécution de sous composant coûts des services.....	96
Figure VI.4 Exécution de l'opération de facturation.....	97
Figure VI.5 Fenêtre principale de mot de passe.....	97
Figure VI.6 Fenêtre mot de passe pour un utilisateur.....	98
Figure VI.7 Présentation Générale de l'application.....	98
Figure VI.8 Onglet Client.....	99
Figure VI.9 Création d'un client.....	99
Figure VI.10 Onglet Abonnement.....	100
Figure VI.11 Création d'un abonnement.....	100
Figure VI.12 Onglet Suivi de paiement.....	101
Figure VI.13 Onglet Plan Tarifaire.....	101
Figure VI.14 Tarif des services Complémentaires.....	102
Figure VI.15 Onglet Archivage et purge.....	102
Figure VI.16 Purge des données.....	103
Figure VI.17 Onglet Utilisateur.....	103
Figure VI.18 Gestion des utilisateurs.....	104

Figure VI.19 Onglet Traitement.....	104
Figure VI.20 Lancement des traitements.....	105
Figure VI.21 Historique des traitements.....	105
Figure VI.22 Mot de passe de facturation.....	106
Figure VI.23 Choix de numéro d'abonné.....	106
Figure VI.24 La facture.....	107
Figure VI.25 message d'erreur.....	107
Figure VI.26 Message d'avertissement.....	108
Figure VI.27 Message de confirmation.....	108
Figure VI.28 Message d'erreur.....	108
Figure VI.29 Icône des données archiver.....	109
Figure VI.30 Icône des différents.....	109

Introduction Générale

Introduction Générale :

La notion de composant logiciel occupe aujourd'hui une partie importante du paysage informatique, notamment dans les domaines du génie logiciel, des systèmes distribués ou encore des réseaux. Comme tout concept émergent, l'ambiguïté prévaut lorsqu'il s'agit d'en donner une définition claire. Nous pouvons cependant nous reposer sur une définition relativement consensuelle[RFC10] :

Les composants logiciels sont des unités de composition de logiciels.

Ou bien :

Un composant est une entité logicielle (d'implémentation), élaborée et vendue par une société et utilisée par un (ou des) client(s) pour composer un logiciel.

Cette proposition [RFC10] est bien sûr peu satisfaisante mais elle est un point de convergence de nombreux travaux autour du concept de composant. Dès les années 80, le terme de composant sera souvent employé en informatique.

Le domaine des architectures logicielles propose de décrire la structure d'un logiciel comme un assemblage, ou plutôt une interconnexion, de composants.

Une architecture logicielle modélise un système logiciel en termes de composants et d'interactions entre composants.

Du point de vue industriel, les besoins en terme de composants logiciels s'expriment plutôt par la nécessité de disposer de technologies permettant de composer des applications à partir d'éléments logiciels réutilisables.

aujourd'hui on trouve de nombreuses solutions industrielles à base de composants par exemple : COM, JavaBeans, EJB, Corba Components...

Les principales difficultés rencontrées lors de la conception de logiciel telles que la gestion de complexité, de la conformité ou encore de l'interopérabilité ne trouvent que des réponses partielles dans la programmation orientée objet.

Face à ces difficultés, les réactions sont de natures différentes. Certaines entreprises préfèrent maintenant *acheter des logiciels ou des composants plutôt que développer*. Certains concepteurs favorisent le prototypage rapide et la notion de raffinement. D'autres utilisent des méthodes incrémentales de développement.

Actuellement, un grand intérêt est porté au domaine de l'architecture logicielle et l'approche de conception par assemblage de composant. Cet intérêt est motivé principalement par la réduction des coûts et des délais de développement des applications. En effet, on prend moins de temps à acheter (et donc à réutiliser) un composant que de le concevoir, le coder, le tester, le déboguer et le documenter.

L'architecture logicielle joue le rôle d'une passerelle entre l'expression des besoins du système logiciel et l'étape de codage du logiciel. Elle fournit une description abstraite ou un modèle du système logiciel.

Les études menées dans le domaine du génie logiciel ont montré l'importance de la notion d'architecture logicielle. Cette dernière permet d'exposer de manière compréhensible et synthétique la complexité d'un système logiciel et de faciliter l'assemblage de composants logiciels. Les axes de travail autour de ce thème sont multiples, l'idée fondamentale est de profiter des avantages d'une architecture clairement explicitée pour ne laisser en second plan que la programmation d'application.

Ce travail est une étude de conception et d'implémentation d'un système de gestion commerciale des produits de réseau X25 selon l'approche composant.

Le système a pour objet de générer les factures des clients de l'opérateur utilisant le réseau à partir des tickets de taxations émis et des données clients. De plus, il met à disposition un fichier contenant les informations nécessaires au service comptable, Le but est de mettre en place un site capable de facturer les clients de l'opérateur et ce, quelque soit le réseau utilisé.

Le présent mémoire est structuré comme suit :

- ❖ Le premier chapitre vise à exposer les fonctions de base de réseau X25.
- ❖ Le deuxième chapitre vise à exposer les langages de description d'architecture.

- ❖ Le troisième chapitre présente le langage Archjava et ses concepts de base.
- ❖ Le quatrième chapitre présente la description de système de gestion commerciale des produits de réseau X25 et les différents règles et techniques utilisées.
- ❖ Le chapitre suivant évoque la conception selon l'approche composant de notre solution et la présentation des différents composants et l'interaction entre eux.
- ❖ Le dernier chapitre est consacré à l'implémentation des fonctionnalités de notre solution.

Chapitre I

Introduction au réseau X25

I.1. Introduction :

Le protocole X.25 est un protocole relatif à la commutation de paquets. Il a été adopté en 1976 par le CCITT (consultative committee on international telegraphy and telephony). Parmi les réseaux nationaux utilisant X25 nous citons Transpac en France et géré par France Télécom, EPSS pour TCTS (Trans Canada Telephone System) au Canada. Datapac et Telenet pour Telenet Communication Corps aux USA. En Algérie le premier réseaux X25 porte le nom de **DZPAC** et était géré par le ministère des PTT. Ce réseaux est actuellement géré par Algérie Télécom. Cette dernière à depuis quelques années installé un deuxième réseau X25, et appelé **Mégapac**. Ce dernier offre des débits plus important que le DZPAC.

I.2. Présentation de réseau X25 :

Le protocole X.25 contient les trois premières couches de protocole, à savoir [URL04]:

- ◆ La norme X.21 pour le niveau physique.
- ◆ Le sous-ensemble LAP-B de la norme HDLC pour la couche liaison.
- ◆ La norme X.25 pour la couche réseau.

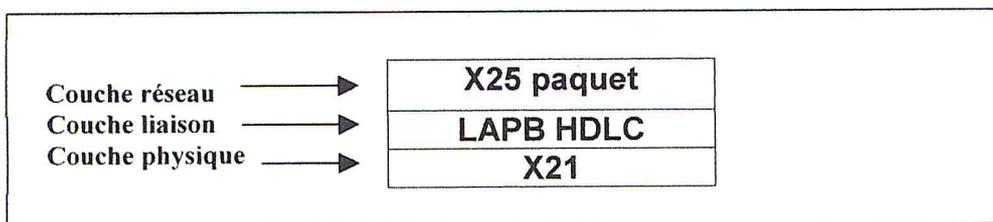


Figure I. 1 Couche réseau

La recommandation X.25 définit un protocole entre l'interface ETTD (Equipement Terminal de Traitement de Donnée) et un ETCD (Equipement de terminaison de circuit de données) pour la transmission de paquets [URL04].

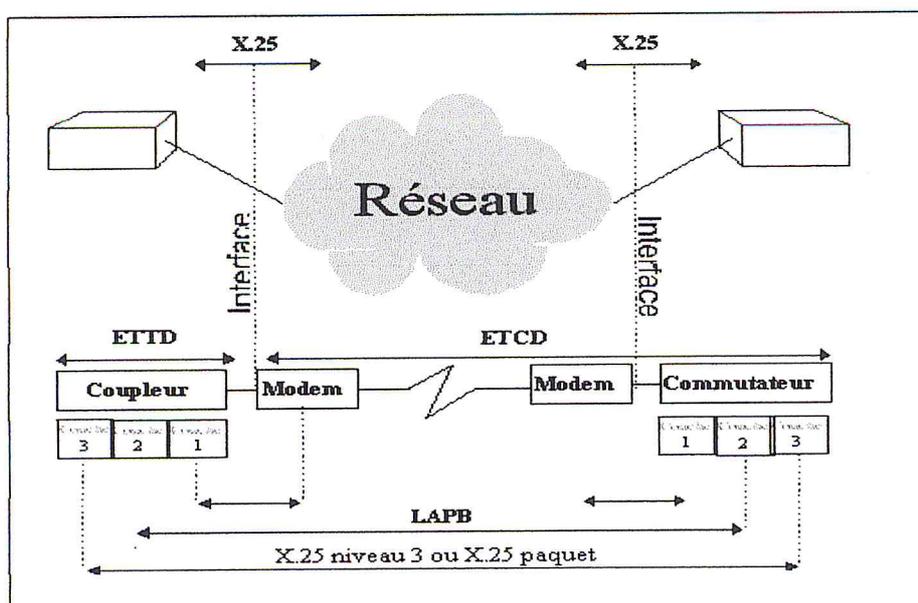


Figure I. 2 Présentation de Réseau X25

Les trois niveaux définis par X.25 correspondent aux trois premières couches du modèle de référence OSI (physique, liaison et réseau) [RFC09].

I.2.1. Niveau physique

Les réseaux à commutation de paquets peuvent offrir une ou plusieurs des interfaces physiques suivantes. Ils peuvent bien sûr en offrir d'autres.

1. Recommandation ITU-T X.21.
2. Recommandation ITU-T X.21bis.
3. Recommandations ITU-T de la série V.
4. Recommandation ITU-T X.31.

I.2.2. Niveau trame

Le niveau trame correspond à la couche liaison de données du modèle de référence OSI. Deux types de procédures sont utilisables à ce niveau :

1. procédure à liaison unique (identique au mode LAPB du protocole HDLC) ;
2. procédure multi – liaison (plusieurs supports physiques).

I.2.3. Niveau paquet :

L'objectif du niveau paquet est de permettre l'interconnexion entre deux usagers du réseau par l'intermédiaire de ce l'on appelle un circuit virtuel. Un usager peut ouvrir (créer) plusieurs circuits virtuels ce qui lui permet de pouvoir gérer plusieurs communications simultanément (multiplexage).

I.2.3.1. Gestion de circuit virtuel

Un circuit virtuel consiste en une mise en communication de deux usagers. On distingue deux types de circuits virtuels :

- **les circuits virtuels permanents (CVP) :** Qui sont des circuits établis et maintenus de façon permanente par le réseau.
- **les circuits virtuels commutés (CVC) :** Qui sont des circuits établis et libérés à chaque communication sur demande d'un usager.

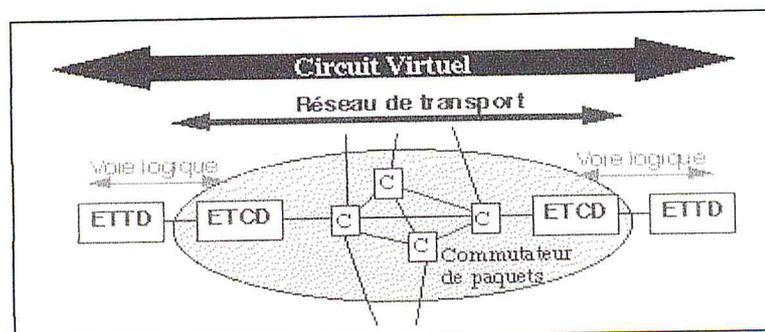


Figure I. 3 Circuit Virtuel

Le niveau paquet gère les circuits virtuels établis par l'utilisateur avec ses correspondants en assurant :

1. l'établissement et la libération des circuits virtuels commutés.
2. le transfert des données sous forme de paquets.
3. le contrôle du transfert des données.
4. la gestion des erreurs.

I.2.3.2. Voie logique

Afin de permettre la coexistence à un instant donné de plusieurs communications virtuelles, on utilise des identifiants de communication appelés numéro de voie logique. Un numéro de voie logique est attribué lors de l'abonnement à chaque circuit virtuel permanent. Dans le cas d'une communication commutée, le numéro de voie logique est attribué lors de la phase d'établissement du circuit virtuel.

I.2.3.3. Contrôle de flux

Le contrôle de flux permet à un récepteur d'information de limiter la vitesse à laquelle un émetteur émet ses paquets de données. Le contrôle de flux ne s'exerce habituellement pas de bout en bout. Lorsque l'abonné récepteur de données ne peut plus continuer à recevoir des données, il signale cet état de fait au nœud du réseau auquel il est rattaché. Ce dernier peut répercuter ce contrôle vers l'abonné émetteur des données.

I.2.3.4. Multiplexage

Le multiplexage permet à un abonné d'établir plusieurs communications (appelées circuits virtuels) sur un même support physique, ou plus précisément sur une même liaison d'accès entre l'abonné et le réseau.

I.3. Structure des paquets :

UN paquet X25 est composé d'un minimum de 3 octets et d'un maximum de n octets. Le format d'un paquets est comme suit [URL04]:

OCTET 1	GFI Q D X X	Numéro de groupe des canaux logiques
OCTET 2	Numéro du canal logique	
OCTET 3	Identification du type de paquet	
OCTET 4		
....	Données du paquet	
OCTET N		

Tableau I.1 Format d'un paquet X25

Chaque paquet qui est transféré à travers l'interface DTE/DCE consiste au minimum à trois octets. Ces octets contiennent une identification de format général du paquet(GFI), une identification du canal logique et une identification de type de paquet.

L'identification de format général (GFI) informe sur le format général du reste de l'entête. Il est composé des bits Q et D et de deux autres bits qui informent du nombre modulo de séquence devant être utilisé.

Le bit Q du GFI est utilisé pour définir de niveau de donnée dans le paquet. Une des utilisations de ce bit est dans les procédures X.29 pour la distinction entre les données et les messages PAD.

N.B :

Le PAD :

- convertit le protocole X25 en mode asynchrone.
- assemble et désassemble les paquets.
- établit, gère et libère les circuits virtuels.

Le bit D du GFI est utilisé pour informer si les procédures de délivrance de confirmation sont à utiliser. Ceci signifie, si l'on doit accuser réception à la fin des paquets de données ou non.

Le numéro des groupes de canaux logique (LCGN) et le numéro de canal logique (LCN) donne l'information sur l'adresse qui dirige les paquets à travers les canaux logiques.

L'identification de type de paquet informe du type spécifique de paquet à transmettre. Il y a cinq types de paquet définis :

- ◆ Initialisation d'une connexion et effacement.
- ◆ Donnée et interruption
- ◆ Contrôle du flux et réinitialisation
- ◆ Reprise
- ◆ Diagnostique

I.4. Les Procédures X.25 :

Les mécanismes fondamentaux que doivent mettre en œuvre les réseaux à commutation de paquets, ainsi que les ETTD (usagers) désirant s'y raccorder. Ce sont [RFC09] :

1. Les procédures d'établissement et de libération des communications ;
2. Les procédures pour le transfert des données ;
3. Les procédures de contrôle de flux ;
4. procédures de gestion des incidents.

A ces procédures gérant le service de commutation, s'ajoutent des procédures permettant la gestion de services complémentaires.

I.4.1. Etablissement d'un circuit virtuel :

L'objectif de l'établissement d'un circuit virtuel est de créer un chemin à travers un réseau à commutation de paquet, de le marquer (de façon à ce que les paquets suivants empruntent ce chemin, quel que soit le sens de transmission), et de réserver des ressources dans les nœuds de commutation de façon à permettre le passage de ces paquets.

I.4.1.1. Choix des voies logiques :

D'une manière générale, un usager dispose d'une gamme de voies logiques convenue avec le réseau auquel il est raccordé. La manière de choisir une voie logique libre est la suivante :

1. un usager doit choisir la première voie logique libre dont le numéro d'identification est le plus élevé possible dans la gamme des numéros qui lui sont attribués.
2. Le réseau doit, par contre, choisir la première voie logique libre dont le numéro d'identification est le plus bas possible dans la gamme des numéros attribués à l'utilisateur.

I.4.1.2. Demande d'établissement côté appelant

Un ETTD désirant établir une communication (autrement dit un circuit virtuel) émet un paquet d'appel sur une voie logique libre, Ce paquet d'appel est souvent appelé paquet d'appel « sortant ». Le paquet d'appel sortant contient les champs suivants :

1. le numéro de voie logique choisi par l'appelant.
2. l'adresse de l'appelé.
3. éventuellement l'adresse de l'appelant.

4. éventuellement un champ de facilités demandant des options ou des services complémentaires.
5. enfin, des données d'appel (facultatives).

I.4.1.3. Demande d'établissement côté appelé

Le réseau remet à l'appelé un paquet d'appel dit «entrant». Le paquet d'appel entrant utilise une voie logique parmi les voies logiques libres. Le paquet d'appel entrant contient les informations suivantes :

1. le numéro de voie logique choisie par le réseau ;
2. l'adresse de l'appelant ;
3. éventuellement l'adresse de l'appelé ;
4. un champ de facilités (facultatif ; dépend du réseau) ;
5. des données d'appel (si elles étaient présentes dans le paquet d'appel émis par l'appelant).

I.4.1.4. Réponse à une demande d'établissement côté appelé

L'appelé peut :

1. accepter l'appel en émettant un paquet de confirmation d'appel, en utilisant la même voie logique indiquée dans le paquet d'appel entrant.
2. refuser l'appel en émettant un paquet de libération.
3. ne pas répondre avant un certain délai. Dans ce cas, le réseau considère que l'appelé est dans un état où il est incapable de répondre et entame une procédure de libération en émettant vers l'appelant un paquet de libération.

I.4.1.5. Confirmation d'établissement côté appelant

Lorsque l'appelant reçoit un paquet de confirmation d'appel sur la voie logique où il avait effectué son appel, il considère que la communication a été acceptée par l'appelé. S'il reçoit autre chose (en particulier un paquet de libération), il doit considérer que le circuit virtuel n'a pas été établi.

I.4.2. Libération d'un circuit virtuel :

Trois cas peuvent se produire conduisant à la libération d'un circuit virtuel :

1. la communication peut être normalement libérée par un des deux usagers.
2. la communication peut également être libérée par un des deux correspondants suite à un refus d'appel.

3. ou enfin, le réseau lui – même peut détruire une communication (généralement suite à une erreur).

I.5. Transfert des données :

La procédure de transfert des données s'applique à chaque voie logique utilisée. Le transfert des données est assuré, dans les deux sens, par les paquets de données et les paquets d'interruption [RFC09].

Caractéristiques générales du transfert :

1. toutes les données d'usager (dans les paquets de données et d'interruption) sont transmises en transparence sans que le réseau introduise d'altération.
2. l'ordre des éléments binaires dans les paquets de données et d'interruption est conservé.
3. les séquences de paquets sont transférées comme des séquences de paquets complètes.

Le transfert des données ne peut s'effectuer que si la voie logique est dans l'état *transfert de données* (la voie logique d'un circuit virtuel permanent se trouve toujours dans cet état, sauf pendant le déroulement de la procédure de reprise).

Durant cette période, un ETTD peut recevoir :

1. des paquets de données transportant des informations.
2. des paquets d'interruption transportant également des informations ; ces paquets ne sont pas soumis au contrôle de flux.
3. des paquets de contrôle de flux limitant la vitesse de transfert des paquets de données.
4. des paquets de réinitialisation utilisés pour régler les différents cas d'erreurs ou d'incidents se produisant au niveau du réseau ou des équipements.

I.6. La supervision de réseau :

I.6.1. Ticket de taxation :

L'opérateur qui exploite le réseaux X25 de constructeurs différents. Chacun de ces constructeurs journalise les communications de son propre réseau à l'aide de tickets de taxation de format spécifique.

un ticket à la fin de chaque communication et un ticket pour chacune des communications établies et pour lesquelles aucun ticket n'a été transmis depuis un temps configurable dans le commutateur.

Les tickets de taxations contient généralement pour chaque communication les informations suivantes :

- Date d'enregistrement sur la station de supervision.
- Date d'enregistrement sur la station de supervision.
- Nom du nœud.
- Numéro d'ordre du ticket.
- Mot de passe composé par l'appelant / user data.
- Date d'établissement de la connexion.
- Heure d'établissement de la connexion.
- Codes de cause et Diagnostic.
- Durée de la connexion minutes / secondes.
- Adresse de l'appelant.
- Adresse de l'appelé.
- Nombre de paquets transmis et reçus ;
- Nombre total d'octets transmis et reçu.

I.6.2. Station de supervision :

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

Généralement on trouve deux stations de supervisions qui sont active en même temps, l'une est choisie principale et l'autre de secours lors de la phase de paramétrage. Dans le cas où le système 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.6.3. Constitution du fichier de taxation :

Les tickets sont regroupés dans des fichiers dont l'extension de nom est «.txt».

I.6.4. Les commutateurs :

La récupération des tickets de taxation est réalisée à la demande; elle se fait via les stations de supervision SVR. Les stations de supervision reçoivent, simultanément, au fil de l'eau les tickets émis par les commutateurs X25.

Chapitre II
LES LANGAGES DE DESCRIPTION
D'ARCHITECTURE (ADL)

II.1. Introduction :

Pour mieux contrôler la complexité du logiciel, il est nécessaire d'avoir un niveau d'abstraction élevé et de disposer de modèles qui s'approchent du modèle mental du développeur. Une réponse possible est la définition d'une architecture du système. **Une architecture logicielle a base de composants** décrit l'ensemble des composants qui la composent, donne la définition de leur assemblage et prend en compte les structures d'accueil nécessaires pour le déploiement et l'exploitation du système résultant[RFC04].

On peut dire que la définition de l'architecture d'un système correspond à l'établissement du plan de construction du logiciel. Elle permet la conception d'applications en se détachant de détails techniques propres à l'environnement et en respectant les conditions fixées par les futurs utilisateurs. En maîtrisant l'architecture conceptuelle, il est alors plus facile de gérer ses éventuelles évolutions. En effet la modification d'un plan est plus simple que la modification d'un système complet.

Les architectures logicielles ont pour origine, à la fois les difficultés rencontrées par les concepteurs de gros logiciels, les caractéristiques de la programmation à grande échelle et également les besoins de réutilisation de logiciel. La plupart du temps, l'architecture logicielle d'un système est spécifiée de manière informelle et intuitive par un diagramme de type *Box – and – line* sans sémantique associée. Ce manque de définition rigoureuse vient essentiellement du fait que peu de démarches de conception intègrent cette étape, soit parce que la structure du logiciel est simple, soit parce qu'au travers de l'analyse, les différentes structures du logiciel ont été appréhendées. Cependant, avec l'apparition de systèmes répartis, cette étape de définition d'architecture devient incontournable [RFC04].

Par cette notion de plan de logiciel, la complexité est prise en compte, les possibilités de réutilisations sont augmentées, et il est possible d'utiliser des outils formels sur certaines parties précises de l'architecture et de générer des parties de code automatiquement.

Les langages de description d'architecture (ou ADL pour *Architecture Description Language*) répondent en partie à cette problématique en permettant la définition d'un vocabulaire précis et commun pour les acteurs devant travailler autour la spécification liée à l'architecture. Ils spécifient les composants de l'architecture de manière abstraite sans entrer dans des détails d'implantation, définissent de manière explicite les interactions entre composants d'un système et fournissent un support de modélisation pour aider les concepteurs à structurer et composer les différents éléments[RFC04].

II.2. Les concepts de base des Langages de Description d'architecture (ADLs) :

On trouve un ensemble des concepts dans les différents langages de description d'architecture. Ces concepts sont au nombre de trois. Le premier est *le composant*, le second *le connecteur*, et finalement le dernier *la configuration* ou encore la topologie.

Medvidovic et Taylor [RFC04] ont défini un gabarit de comparaison pour la plupart des langages de description d'architecture existants dans les milieux académique et

industriel. Leur étude consiste, d'une part à définir les éléments communs d'un grand nombre d'ADLs (composant, connecteur, configuration) et, d'autre part, à proposer un ensemble de caractéristiques globales pour chaque élément fournissant ainsi un cadre commun favorisant la comparaison entre plusieurs ADLs. Ces caractéristiques sont également un moyen d'évaluer un langage de description d'architecture et sont considérées comme un ensemble de critères que doit prendre en compte un langage pour être considéré comme étant un langage de description d'architecture [RFC04].

II.2.1. Le composant :

Un composant [RFC04] est une unité de calcul ou de stockage à laquelle est associée une ou plusieurs unités d'implantation localisée dans un ou plusieurs environnements d'exécution. Il peut ainsi être simple ou composé, on parle alors dans ce dernier cas de composite. Sa taille peut aller de la fonction mathématique à une application complète. On définit deux parties dans un composant.

- ***Une première partie***, dite extérieure, correspond à son interface. Elle comprend la description des interfaces fournies et requises par le composant. Elle définit les interactions du composant avec son environnement.
- ***La seconde partie*** correspond à son implantation et permet la description du fonctionnement interne du composant.

Parmi les caractéristiques globales d'un composant définies par Medvidovic et Taylor [RFC04] :

- Le type d'un composant,
- La sémantique d'un composant.
- Les contraintes d'un composant

II.2.1.1. Le type d'un composant

Le type d'un composant est un concept représentant l'implantation des fonctionnalités fournies par le composant. Il s'apparente à la notion de classe que l'on trouve dans le modèle orienté objet. Ainsi, un type de composant permet la réutilisation d'instances de même fonctionnalité soit dans un même architecture, soit dans des architectures différentes.

En fournissant un moyen de décrire, de manière explicite, les propriétés communes à un ensemble d'instances d'un même composant, la notion de type de composant introduit un classificateur qui favorise la compréhension d'une architecture et sa conception.

II.2.1.2. La sémantique d'un composant

La sémantique du composant est exprimée en partie par son interface. Cependant, l'interface telle que décrite ci-dessus ne permet pas de préciser complètement le comportement du composant. La sémantique doit être enrichie par un modèle plus complet et plus abstrait permettant de spécifier les aspects dynamiques ainsi que les contraintes liées à l'architecture.

Ce modèle doit garantir une projection cohérente de la spécification abstraite de l'architecture vers la description de son implantation avec différents niveaux de raffinements.

La sémantique d'un composant s'apparente à la notion de type dans le modèle orienté objet.

II.2.1.3. Les contraintes d'un composant

Les contraintes définissent les limites d'utilisation d'un composant et ses dépendances intra composants. Une contrainte est une propriété devant être obligatoirement vérifiée sur un système ou une de ces parties. Si celle-ci est violée, le système est considéré comme un système incohérent et inacceptable. Elles permettent ainsi de décrire de manière explicite les dépendances des parties internes d'un composant comme la spécification de la synchronisation entre composants d'une même application (dépendance intra composant).

II.2.2. Le connecteur :

Le connecteur [RFC04] correspond à un élément d'architecture qui modélise de manière explicite les interactions entre un ou plusieurs composants en définissant les règles qui gouvernent ces interactions. Par exemple, un connecteur peut décrire des interactions simples de type appel de procédure ou accès à une variable partagée, mais aussi des interactions complexes telles que des protocoles d'accès à des bases de données avec gestion des transactions, la diffusion d'événements asynchrones ou encore l'échange de données sous forme de flux.

Un connecteur comprend également deux parties. **La première** correspond à la partie visible du connecteur, c'est-à-dire son interface, qui permet la description des rôles des participants à une interaction. **La seconde** partie correspond à la description de son implantation. Il s'agit là de la définition du protocole permettant la mise en œuvre du protocole associé à l'interaction.

Le connecteur permet également de vérifier l'intégrité de la communication, c'est-à-dire de vérifier que les composants peuvent être connectés correctement. Ainsi, il permet la réutilisation et l'adaptation des interfaces de composants déjà existants que l'on cherche à relier.

Six caractéristiques importantes définies par Medvidovic et Taylor [RFC04] sont à prendre en compte pour spécifier de manière exhaustive un connecteur. Parmi ces caractéristiques :

II.2.2.1. Le type

Le type d'un connecteur correspond à sa définition abstraction qui reprend les mécanismes de communication entre composants ou les mécanismes de décision de coordination et de médiation. Il permet la description d'interactions simples ou complexes de manière générique et offre ainsi des possibilités de réutilisation de protocoles.

II.2.2.2. La sémantique

Comme pour les composants, la sémantique des connecteurs est définie par un modèle de haut niveau spécifiant le comportement du connecteur. A l'opposé de la sémantique du composant qui doit exprimer les fonctionnalités déduites des buts ou des besoins de l'application, la sémantique du connecteur doit spécifier le protocole d'interaction. De plus, celui-ci doit pouvoir être modélisé et raffiné lors du passage d'un niveau de description abstraite à un niveau d'implantation.

II.2.2.3. Les contraintes

Les contraintes permettent de définir les limites d'utilisation d'un connecteur, c'est-à-dire les limites d'utilisation du protocole de communication associé. Une contrainte est une propriété devant être vérifiée sur un système ou sur l'une de ses parties. Si celle-ci est violée, le système est considéré comme un système incohérent et inacceptable. Par exemple, le nombre maximum de composants interconnectés à travers le connecteur peut être fixé et correspond alors à une contrainte.

II.2.3. La configuration ou architecture :

Les configurations d'application, encore appelées topologies [RFC04] définissent les propriétés architecturales de connectivité et de conformité aux heuristiques de conception, ainsi que des propriétés de concurrence et de répartition (dans certains ADLs).

Une configuration définit la structure et le comportement d'une application formée de composants et de connecteurs. Une composition de composants, appelée dans certains contextes composites, est une configuration.

La configuration structurelle de l'application correspond à un graphe connexe des composants et des connecteurs formant l'application. Elle détermine les composants et

les connecteurs appropriés à l'application et vérifie la correspondance entre les interfaces des composants et des connecteurs.

La configuration comportementale, quant à elle, modélise le comportement en décrivant l'évolution des liens entre composants et connecteurs. Elle définit également le schéma d'instanciation des composants au moment de l'initialisation de l'application, ainsi que le placement des composants sur les sites au moment du démarrage du système et leur évolution pendant la vie de l'application.

Neuf caractéristiques sont précisées par Medvidovic et Taylor [RFC04] pour évaluer la configuration d'un ADL. Parmi ces caractéristiques :

II.2.3.1. Un formalisme commun

Une configuration doit permettre de fournir une syntaxe simple et une sémantique permettant de

- ✓ Faciliter la communication entre les différents partenaires d'un projet (concepteurs, développeurs, testeurs, architectes),
- ✓ Rendre compréhensible la structure d'une application à partir de la configuration sans entrer dans le détail de chaque composant et de chaque connecteur,
- ✓ Spécifier la dynamique d'un système, c'est-à-dire l'évolution de celui-ci au cours de son exécution.

II.2.3.2. La composition

La définition de la configuration d'une application doit permettre la modélisation et la représentation de la composition à différents niveaux de détail.

La notion de configuration spécifie une application par composition hiérarchique. Ainsi un composant peut être composé de composants, chaque composant étant spécifié lui-même de la même manière, jusqu'au composant dit primitif, c'est-à-dire non décomposable.

L'intérêt de ce concept est qu'il permet la spécification de l'application **par une approche descendante par raffinement**, allant du niveau le plus général formé par les composants et les connecteurs principaux, définis eux mêmes par des groupes de composants et de connecteurs, jusqu'aux détails de chaque composant et de chaque connecteur primitifs.

II.2.3.3. Le raffinement et la traçabilité

La configuration est également un moyen de permettre le raffinement de l'application d'un niveau abstrait de description général vers un niveau de description de plus en plus détaillé, et, ceci, à chaque étape du processus de développement (conception, implantation, déploiement). Ainsi il est possible, par la définition de la configuration, de garder trace de ce qui a été fait en amont, et de créer des liens entre les différents niveaux de description de l'application. Cette caractéristique permet le rapprochement entre les modèles de haut niveau et le code.

II.2.3.4. L'hétérogénéité

La configuration d'un ADL doit permettre le développement de grands systèmes avec des éléments préexistants de caractéristiques différentes. L'ADL doit être capable de spécifier une application indépendamment du langage de programmation, du système d'exploitation et du langage de modélisation.

II.3. Les principaux ADLs :

Les langages de description d'architecture sont des langages dits déclaratifs. Ils peuvent être classés en deux grandes familles. La première correspond aux langages qui privilégient la description des éléments de l'architecture et leur assemblage structurel, la seconde définit les langages qui se centrent sur la description de la configuration d'une architecture et sur la dynamique du système [RFC04].

La première famille de langages correspond à une famille de langages qui est accompagnée d'outils de modélisation, d'analyseur syntaxique et de générateur de code.

La deuxième famille de langages regroupe des langages accompagnés d'outils de modélisation et de génération de code mais aussi d'une plate-forme d'exécution ou de simulation d'un système, voire de modification dynamique pendant l'exécution. La particularité de ces langages est de définir un élément d'une architecture (composant ou connecteur) comme une instance. Il devient alors facile et simple de spécifier l'évolution dynamique d'une application au cours de son exécution.

Les ADLs les plus connus, sont : UniCon, Aesop, Darwin, C2, Wright, Rapide, OLAN et ACME. Chaque ADL présente des avantages et des inconvénients [RFC04].

II.4. Les principales phases d'un processus de développement :

Les principales phases d'un processus de développement [RFC04] sont les suivantes :

- la phase d'analyse du système,
- la phase de conception du système,
- la phase d'implantation du système,
- la phase de déploiement du système,
- la phase de maintenance du système.

II.4.1. La phase d'analyse :

La phase d'analyse du système consiste à analyser les besoins fonctionnels de la future application. Elle représente l'étape zéro de tout processus de construction de logiciel. En effet, tous les ADLs sont des outils qui n'ont pas pour objectif de définir ou de modéliser des besoins.

II.4.2. La phase de conception :

La phase de conception du système consiste à définir les éléments ou les composants d'un système et leurs interactions (assemblage) et à définir une architecture répondant aux besoins modélisés dans la phase d'analyse.

II.4.3. La phase d'implantation :

La phase d'implémentation consiste à coder et à tester le système. Dans cette phase, les points suivants sont importants :

- une projection des éléments définis de manière abstraite dans la phase de conception sur des éléments d'implantation nouveaux ou existants doit pouvoir s'effectuer simplement, quel que soit le langage de programmation utilisé et en restant cohérent avec la spécification conceptuelle,
- l'assemblage des éléments de l'architecture doit pouvoir être testé sur une structure d'accueil.

II.4.4. La phase de déploiement :

La phase de déploiement consiste à configurer le système avant son exécution. Ainsi, les composants sont configurés en vue de leur déploiement sur des structures d'accueil.

II.4.5. La phase de maintenance :

Elle consiste :

- à maintenir le système de façon à ce qu'il s'exécute de la manière la plus optimisée.
- à assurer l'évolution du système et de son architecture (maintenance évolutive) suite à l'ajout de nouveaux besoins ou à la modification d'un ou plusieurs besoins sans modifier ou détruire ce qui existe.

Chapitre III
PRESENTATION DE LANGAGE
« ARCHJAVA »

III.1. Introduction :

Archjava est un langage de description d'architecture créée par l'université de Washington Département of Computer Science and Engineering USA dont les concepteurs principaux sont JONATHAN et CRAIG CHAMBERS.

Archjava travaille [RFC03] sur une partie encore peu explorée des langages de description d'architecture (ADL). En effet beaucoup d'ADLs décrivent l'architecture logicielle d'une application dans leur propre langage, **au contraire Archjava étend l'implantation d'un langage**, en l'occurrence **JAVA** afin d'incorporer les éléments d'architectures à l'intérieur de code.

Un important travail à été réalisé sur le compilateur Archjava afin de respecter l'intégrité des communications entre les composants. L'approche architecture logicielle au travers d'Archjava à pour objectif d'améliorer la compréhension des programmes, de garantir l'architecture de l'application, de permettre une meilleure évolutivité des applications et d'encourager les développeurs à se servir des avantages offerts par l'architecture logicielle [URL01].

III.2. Le modèle Archjava :

Dans la plupart des ADLs [RFC03] les trois concepts de composants, connecteur et configuration sont généralement acceptés comme essentiels. Archjava reprend ces concepts, en les ajoutant comme nouveaux éléments de *syntaxe au langage JAVA*.

Un composant Archjava est un objet spécial qui communique avec les autres composant sous une forme structuré et bien défini.

Un composant interagit avec le monde extérieur via des ports. Chaque ports correspond à une interface à travers laquelle un ensemble de services fournis et requis sont décrit. Ces services sont représentées par un ensemble de méthodes.

Deux types de composants existent :

- **Les composites** : Les composants composites sont les unités de configuration, en effet ce sont eux qui contiennent les descriptions des composants d'une application ainsi que leur inter – connexions, **une application est donc un composant composite**, la communication entre les composants se fait par appel de méthodes, Archjava offre la possibilité de décrire la création dynamique de composants. Cette caractéristique offre la possibilité de représenter le comportement d'une application
- **Les primitifs** : Les primitifs sont des composants atomiques. Il correspondent par exemple à un composant en librairie, à un exécutable, ou à un module écrit dans un langage de programmation, dans notre cas à une classe JAVA

III.3. La composition des composants :

L'architecture logicielle en Archjava [RFC07] se fait par composition des composants, Un composant composite est une entité de description de configuration et une entité de structuration d'une application de composants coopérants. Ainsi en Archjava, les différents composants peuvent contenir eux même des sous composants connectés ensemble, Un sous composant est déclaré à l'intérieur d'un autre composant, un composant peut invoquer des méthodes directement sur un des ses sous composants. Un sous composant ne peut communiquer avec d'autres composants externes à son composant container. Ceci permet d'introduire une notion de hiérarchie dans l'architecture logicielle. Les sous composants sont automatiquement instanciés quand leur container est instancié.

III.4. Les composants et les ports en langage Archjava :

Les composants [RFC03] se définissent par des classes **component** et sont créés par instantiation. Une instance de composants communique avec les autres instances par l'intermédiaire de ports, un port est donc *un canal de communication*, il spécifie donc les services offerts par un composant et les services dont il a besoin pour fonctionner.

Convention graphique :

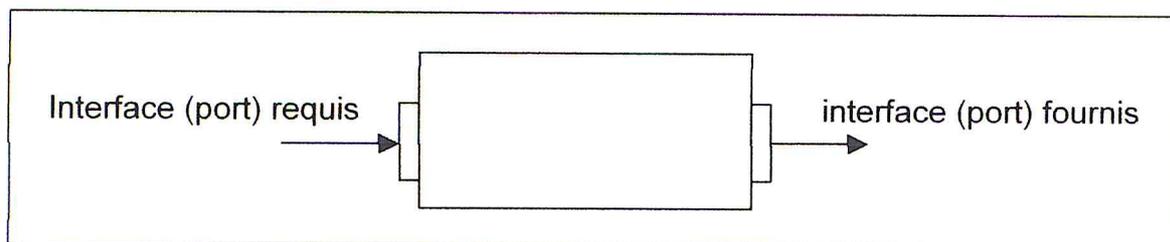


Figure III. 1 Représentation Graphique d'un composant.

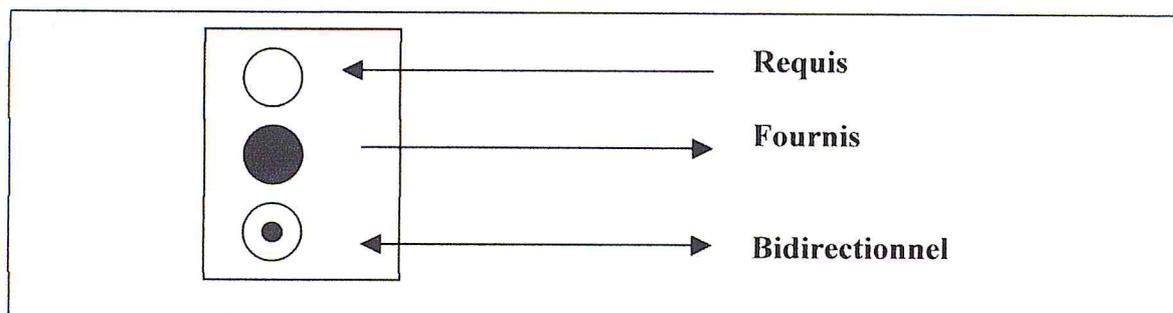


Figure III. 2 Représentation des Ports

1. Pour créer un composant dans le langage Archjava :

```
Public component class [nomComposant]
{
    // Corps du composant
}
```

Les ports [RFC08] déclarent trois types de méthodes, spécifiées en utilisant les mots clé **provides**, **requiers** et **braodcasts**, les méthodes **provides** peuvent être invoquées par d'autres instances de composants connectés aux ports. Cela correspond aux services offerts par le composant, le composant peut invoquer des méthodes à travers le port, ces méthodes sont déclarées par le mot clé **requiers**, cela correspond aux services dont le composant a besoin pour fonctionner. Le mot clé **braodcasts** n'est qu'une extension du mot clé **requiers**, il est utilisé dans les cas où la méthode retourne void et quand plusieurs composants connectés implémentent ce service.

2. Pour créer un port dans le langage Archjava :

```

Public component class [nomComposant] {
    public port [nomPort] { ....
        requires SignatureMethod1 ;
        provides SignatureMethod2 ;
        .....
    }
    SignatureMethod1
    {
        // corps de la méthode1
    }
    .....
}

```

Le corps des méthodes **provides** doit être écrit dans le code du composant.

3. Interconnexion de ports :

3.1. Pour interconnecter deux ports de deux composants différents :

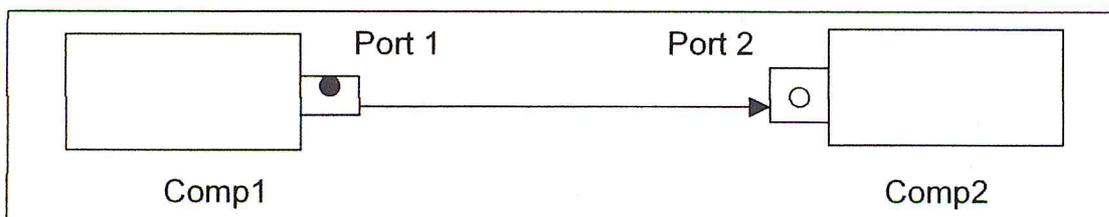


Figure III. 3 Interconnexion des ports

InstComp1, instComp2 des instances de composant Comp1 et Comp2 respectivement.

```
connect instComp1.port1, instComp2.port2 ;
```

Remarque : on peut interconnecter plus de deux ports.

le mot clé **connect** permet ainsi de connecter deux ports de sous composants ou plus de deux.

```
connect instComp1.port1, instComp2.port2, ..., instCompi.porti ;
```

3.2. Pour interconnecter un port avec le port d'un composant englobant : (Connecteur de Délégation)

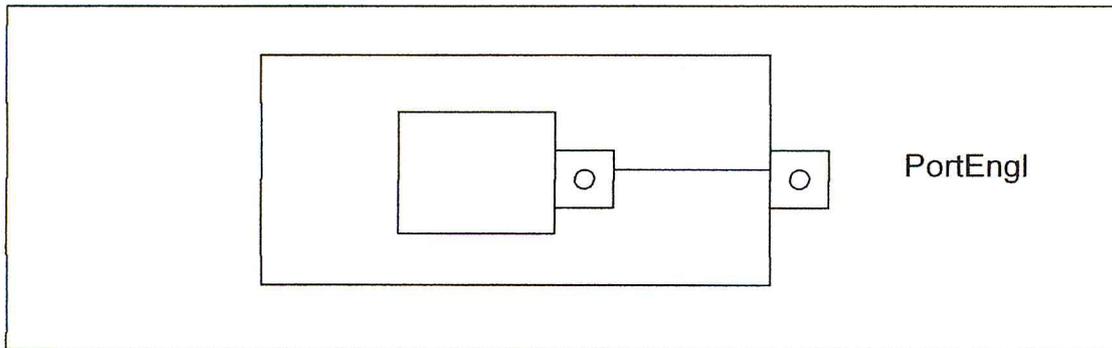


Figure III. 4 Connecteur de délégation

```
glue portEngl to instComp.port ;
```

III.5. Exemple récapitulatif :

(Exemple : Référence [RFC03])

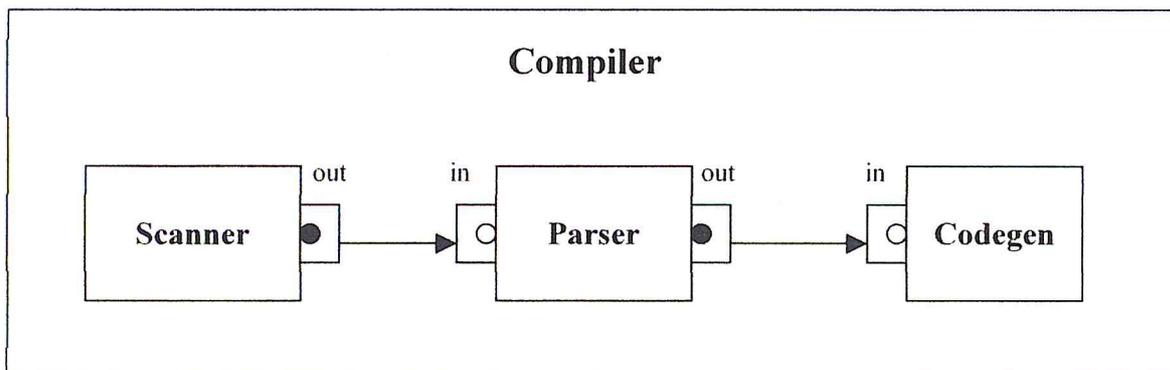


Figure III. 5 Exemple de compilateur simplifié

```
public component class Compiler
{
  Private final Scanner scanner = new Scanner() ;
  Private final Parser parser = new Parser() ;
  Private final CodeGen codegen = new CodeGen() ;

  connect scanner.out, parser.in ;
  connect parser.out, codegen.in ;
}
```

```
public component class Scanner
{
    public port in
    {
        // Le corps
    }

    public port out
    {
        provides Token nextToken() ;
    }
}

public component class Parser
{
    public port in {
        requires Token nextToken() ;
    }

    public port out
    {
        provides AST parse() ;
    }

    AST parse()
    {
        Token tok = in.nextToken() ;
        return parseExpr(tok) ;
    }
    AST parseExpr(Token tok)
    { // le corps de la méthode. }
}

public component class CodeGen {
    public port in
    {
        requires AST parse() ;
    }

    public port out
    {
        provides AsmCode generateCode() ;
    }

    AsmCode generateCode()
    {
        // le corps de la méthode.
    }
}
```

Analyse du Code Source de l'Exemple

- Le mot – clé `component` indique que nous créons une classe de type architectural composant.
- `Compiler` est le composant principal de l'application : le compilateur.
- `Compiler` est constitué de plusieurs composants dont il suffit d'instancier les classes.
- L'architecture pipe-line est définie : la sortie du `Scanner` est connectée à l'entrée du `Parser` et la sortie du `Parser` est connectée à l'entrée du `CodeGen`.
- Le composant `Scanner` est défini. Le composant `Parser`, qui est plus détaillé sert de référence pour les commentaires.
- Le mot – clé `port` indique que la définition suivante est celle d'un port de connexion entre composants.
- Un port de connexion peut fournir et/ou nécessiter certains types de messages. Comme nous travaillons avec Java, les types de messages entre composants ne sont autres que des méthodes Java ordinaires. Les mots – clés `Archjava provides` et `requires` sont respectivement utilisés afin de différencier la fourniture et la nécessité d'un type de message.
- Les méthodes de la classe `Parser` sont implémentées de façon habituelle. Les méthodes appartenant à des ports sont définies et implémentées comme les autres méthodes.
- Le port d'entrée `in` est accessible comme toute classe interne Java conventionnelle. Les méthodes sont les siennes, pas celles du composant agrégé.
- Le composant `CodeGen` est implémenté de la même façon que les autres composants.

III.6. Les connecteurs personnalisés :

Pour créer un propre connecteur on doit utiliser la bibliothèque ***archjava.reflec.****, Les principaux class de cette bibliothèque sont : `Connector`, `Port`, `Type`, `Methode`, `Error`, `Connection`, et `Call`.

Tout nouveau connecteur doit hériter de la class `Connector`.

Quelques méthodes :

- ✓ La méthode `Error[] typecheck (Connection c)` de la class `Connector` est très importante. Son rôle est la vérification des connexions lors de la compilation : la méthode `typechecking` par défaut vérifie que pour toute méthode `requires` il y a une et une seule méthode `provides` et vis versa.

- ✓ La méthode `Objet invoc (Call call)` de la class `Connector` est aussi très importante. Son rôle est de chercher la méthode `provides` correspondante à l'appelant, et passe les résultats à ce dernier.

III.7. La construction d'une architecture dynamique :

Dans la plupart des applications, une architecture logicielle statique est suffisante. Cependant, certaines applications peuvent nécessiter de créer et de connecter dynamiquement certains composants. Archjava offre donc un mécanisme qui permet de modifier dynamiquement l'architecture logicielle d'une application.

Quatre fonctions sont offertes par Archjava pour gérer le comportement dynamique d'une application :

- ✓ la création de composants au `Runtime`,
- ✓ la connexion de composants,
- ✓ la connexion de nouveaux composants sur un même composant.
- ✓ la destruction de composants.

La création dynamique de composant se fait grâce à la syntaxe **NEW** identique à la POO (Programmation Orientée Objet). Des composants peuvent être connectés, par une instance de port, pendant l'exécution grâce à l'expression **CONNECT**. La destruction de composants n'est pas générée de façon explicite. Mais comme en java, les composants sont ramassés par le ramasse miettes (garbage collector) quand ils ne sont plus référencés ni connectés à d'autres composants.

Dans de nombreuses applications, il peut être nécessaire de connecter plusieurs composants sur un même port sans savoir au départ combien. Archjava permet alors de définir une interface de port. Une nouvelle instance de port sera alors créée à chaque connexion.

Chapitre IV
DESCRIPTION DE SYSTEME

IV. Etude existant et analyse :

IV.1 objectif de système :

IV.1.1 Objectifs principaux :

Les objectifs principaux du Système de gestion commerciale des produits du réseau X25 sont :

- De générer les factures des clients du réseaux à partir des tickets de taxations émis et des données clients.
- Met à disposition un fichier contenant les informations nécessaires au service comptable.
- Réalise le suivi des paiements.

Les autres objectifs sont :

- Le suivi des opérations anormales du système par la mise en place de politique de déclenchement d'alarme et de niveau de gravité de ces alarmes.
- L'établissements de diverses statistiques concernant l'exploitation du réseau.

IV.1.2. Objectifs complémentaire :

Pour arriver à ces objectifs il faudrait les éléments complémentaires suivants :

- La gestion des clients.
- La gestion des abonnements. Un client peut avoir 0, 1 ou plusieurs abonnements.
- La définition des divers produits à commercialiser : ce sont notamment des services et la location d'équipements.
- La définition des tarifs des produits à commercialiser.
- La définitions de la manière avec laquelle un produit est commercialisable : Volume d'information, temps de connexion, location à des prix forfaitaires etc.
- La détermination des produits consommés. Ceci se fait essentiellement à partir de deux éléments fondamentaux :
 - Les tarifs des tickets de taxations. Ces derniers indiquent les volumes, durée et qualité des services consommés
 - Les tarifs fixes liés aux droits d'abonnement , à la location des équipements et aux services complémentaires

Le schéma suivant récapitule les fonctionnalités générales du système visé :

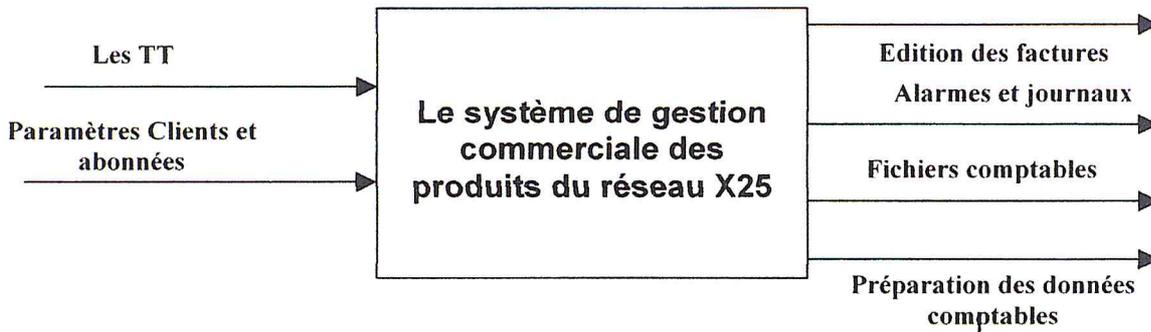


Figure IV. 1 Fonctionnement générale de système

IV.2. Elément fondamentaux à la réalisation du système :

IV.2.1. Les tickets de taxation :

Les Tickets de taxation qui sont émis par les commutateurs X25 sont stockés dans l'une des deux station de supervision du réseau X25. cela nécessite une opération de transfert pour la récupération des fichiers de TT.

Les fichiers de TT contient plusieurs TT de différents formats (différents constructeurs), Les origines de chaque fichier sont connues, cela nécessite de rendre notre système indépendant des constructeurs ce qui implique une couche d'isolation de notre systèmes vis-à-vis des spécificité des constructeurs. Cette isolation consiste à définir un format unique pour notre système et utilisé pour chaque constructeur un convertisseur de format.

Si un TT d'un constructeur donnée ne possède pas un format reconnaissable par l'application en référence à la grammaire associée, il est rejeté.

Ces problèmes nécessitent un pré – traitement qui sera composé de plusieurs phases afin d'obtenir un fichier contenant les TT sous format unique.

IV.2.2. La tarification :

La tarification nécessite une étape de valorisation des tickets, Ces dernier en fin de traitement doivent être mis en base de données. Le ticket en base de données doit contenir des informations supplémentaires, On trouve plusieurs type de tarification.

➤ Pour la tarification à la durée, 2 sortes de tarif sont définies :

- Le tarif à la minute national et le tarif à la minute international.
- Le tarif minute national est dépendant du débit de l'abonnement et d'une date de valeur.

- Le tarif minute international est dépendant du débit de l'abonnement et d'une date de valeur et du groupe.
- Pour la tarification au volume un type de tarif est défini, c'est le tarif pour un koctet, le tarif pour un koctet est dépendant du groupe du ticket et d'une date de valeur.
- Pour la tarification de l'accès on a trois types :
 - **Le forfait complet** : qui représente un forfait illimité sur les communications nationales au volume et à la durée. Les communications internationales restent redevables.
 - **Le forfait sur la durée** : qui représente un forfait illimité sur les communications nationales à la durée. Les communications internationales et les communications nationales sur le volume restent redevables.
 - **La tarification réelle** : qui tient compte de toutes les communications pour les facturer.

IV.2.3. Validité des tickets :

Les tickets de taxations sont Valide pour la réalisation d'une facturation pour une période bien précise, souvent deux mois.

Après réalisation des factures ces tickets n'ont plus raisons d'exister. Cependant il seront laissé pour un certain temps nécessaire à la correction d'erreur après réclamation. Ainsi après que le temps légal de réclamation soit dépassé les tickets doivent alors être éliminé du système ou archivé.

En pratique soit il sont purement éliminé soit il reste pour un certain temps supplémentaire mais dans les sauvegarde du système (Sauvegarde de la base ou du système entier, ces opérations sont réalisé par l'administrateur des machines)

IV.2.4. Les 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).
- Une tranche horaire est définie pour un groupe unique et pour un type de jour unique (ouvré, ouvrable, férié).
- Pour chaque tranche horaire, on attribue une réduction.

- Il y a au 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.
- Une tranche tarifaire correspond à une réduction, Par exemple, une tranche tarifaire 1 correspond à un code réduction 1 (tranche pleine), Il existe trois tranches tarifaires au maximum.

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.

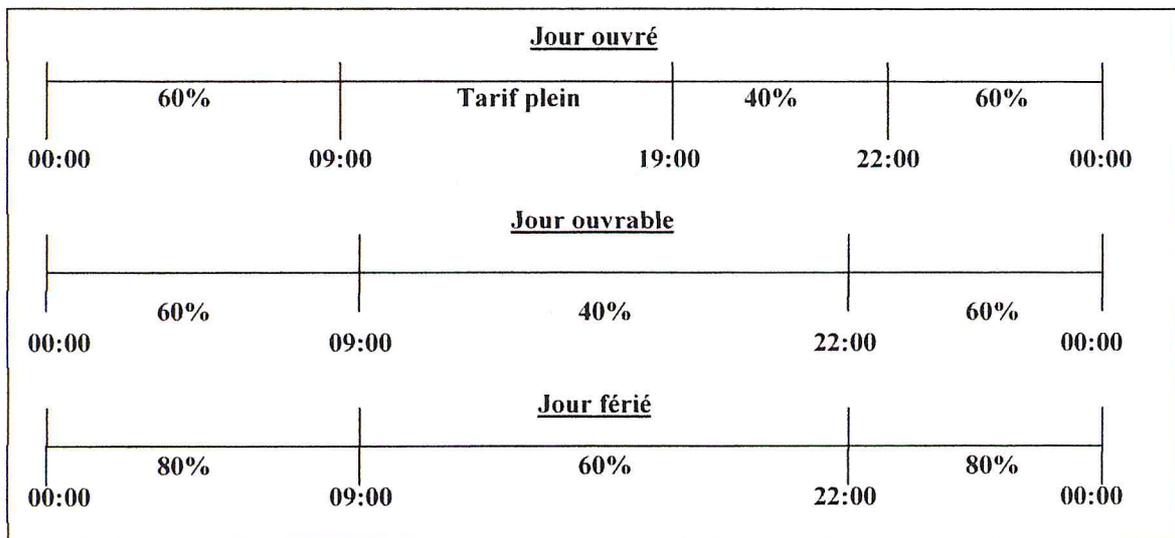


Figure IV. 2 Tranche Horaire

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.

IV.2.5. Groupe de ticket :

chaque ticket de taxation appartient à un groupe national ou international.
Les pays sont composés selon le tableau ci – dissous.

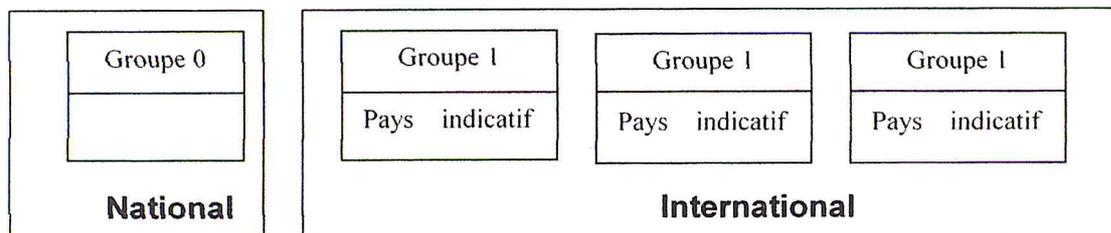


Figure IV. 3 Groupe Pays

IV.2.6. La Grammaire :

Une grammaire est définie pour chaque format de TT. Chaque évolution du format implique une modification de la grammaire.

La grammaires ci-dessous illustrent la structure des grammaires qui pourraient être utilisée pour le Centre de Facturation.

Exemple de grammaire liée a un ticket ci-dessus :

Les champs ont pour caractéristiques : 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 (vide dans ce cas là car inutile).

[CARACTERISTIQUES]

	REGLE_HARMO : SAGEM
	SEPARATEUR : TAB
[CHAMPS]	CHS1,Numero ordre du ticket,1,,
	CHS2,Adresse appele,8,,
	CHS3,Code de C-D,5,,

IV.3. Architecture générale du système :

A la base des études précédentes, le système a réaliser comportera les éléments fondamentaux suivants (schéma informel):

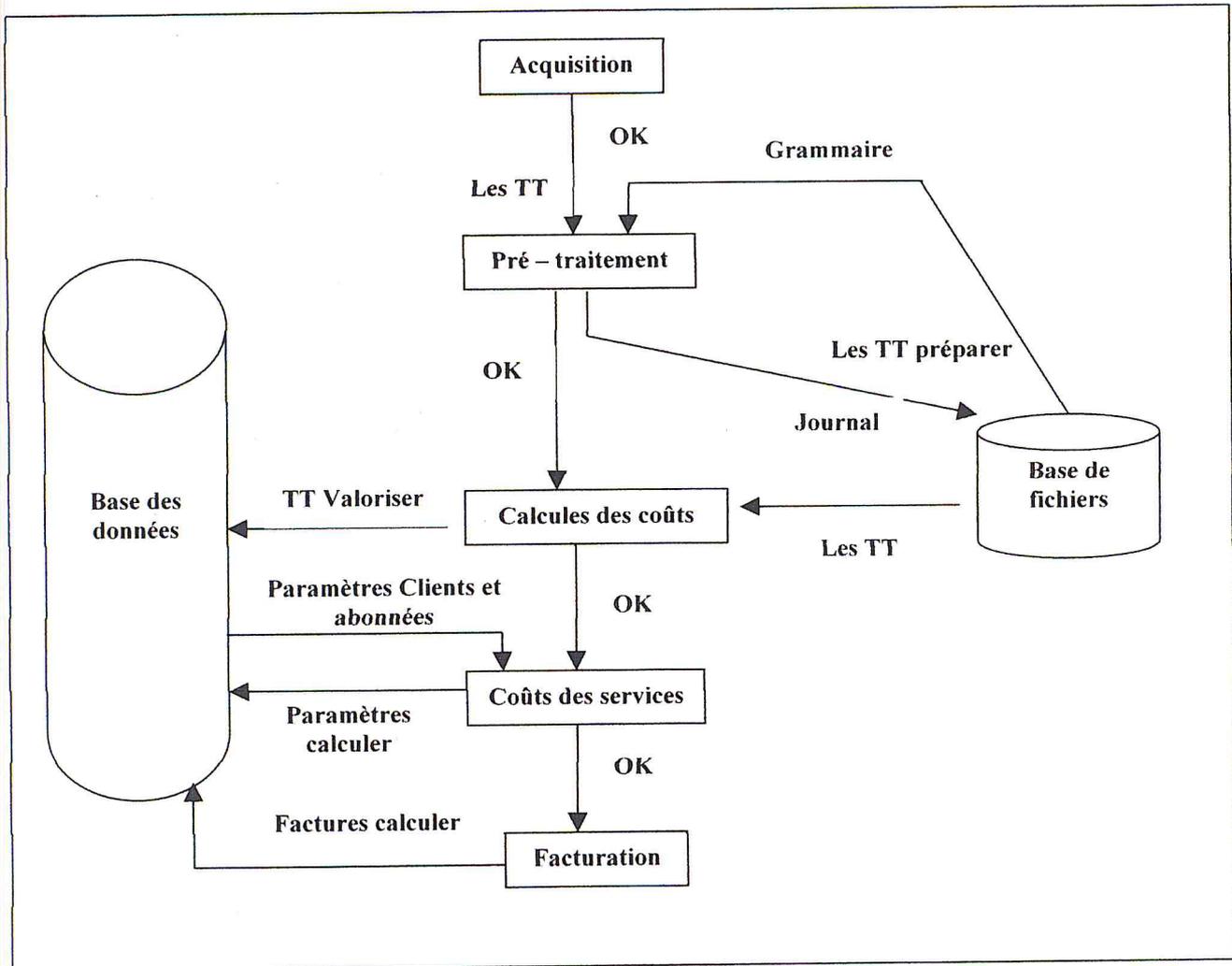


Figure IV. 4 Architecture générale du système

Le système est composé des sous-systèmes suivants :

IV.3.1. L'Acquisition :

Ce module se chargera de la :

Récupération des fichiers de TT en provenance d'une des stations de supervision du réseau X25. La récupération se fait via 2 stations de supervision (SVR). Les 2 stations de supervision reçoivent, simultanément, et au fil de l'eau, les TT émis par les commutateurs X25.

Sur une station de supervision, le fichier de stockage des TT au fil de l'eau 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.

En fonctionnement normal, les 2 stations SVR sont actives, mais l'opérateur du centre choisit l'une d'elle pour faire la récupération des TT et l'utiliser

systématiquement lors de l'exécution du sous système d'acquisition. cette station est appelée la station «principale», et l'autre station celle de «secours».

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.

Si une erreur de connexion entre le Centre de Système de gestion commercial des produits du réseau X25 et une station de supervision est détectée, on recommence l'opération un nombre de fois avant de créer une alarme.

Si la station principale ne retourne pas le fichier des nouveaux TT, et que la station de secours en retourne un, on inverse le rôle des 2 stations (la station de secours devient station principale et inversement).

Le sous système d'acquisition fonctionne de manière indépendante des autres sous-systèmes. Il est possible de le démarrer ou de l'arrêter pour effectuer des opérations de maintenance ou de reprise.

Dans ce sous système, il n'y a que deux sortes d'alarmes : l'alarme provoquée par un problème de connexion est de niveau «majeure ou mineure» puisqu'il n'y a jamais perte de données. On gère donc la date et l'heure de disparition de ce type d'alarme. Si une alarme a déjà été créée pour ce problème, on laisse l'alarme telle que, sinon on la crée.

L'alarme est mineure s'il y a problème de connexion ou de transfert avec une seule des deux stations déclarées dans le paramétrage.

L'alarme est majeure s'il y a impossibilité de récupérer les TT sur les stations déclarées c'est à dire problème sur la station principale, ou un problème sur les deux stations déclarées.

IV.3.2. Pré – traitement :

Le Pré – traitement de tickets consiste à :

- Contrôler s'ils sont sans erreurs dans leur format initial.
- Transformer les ticket dans le format interne du système et indépendant de tout réseau X25.
- Contrôler le cas ou un fichier de ticket provient d'une station de secours. Dans ce cas il est possible d'avoir des doublons. Il faut donc réaliser une opération de Dé – doublonnage des TT récupérés à partir des deux stations de supervision.

Le but de ce sous-système est de préparer les tickets de taxation, reçus des réseaux X25 de l'opérateur, pour le sous-système calcule des coûts.

L'opérateur exploite le réseau X25 de plusieurs constructeurs différents. Chacun de ces constructeurs journalise les communications de son propre réseau à l'aide de tickets de taxation de format spécifique, Un problème sur les SVR peut engendrer des doublons dans les fichiers de TT reçus par le responsable de facturation.

La finalité du sous-système de préparation est d'obtenir des tickets de format unique, indépendant de tout constructeur, sans doublons.

Les données en entrée du sous-système de pré – traitement sont les fichiers de TT acquis par la phase d'acquisition. Les différents fichiers sont bien distincts, Les origines de chaque fichier sont connues.

chaque ticket de taxation appartient à un groupe national ou international.

Après la fusion les fichiers acquis sont effacés, Les données en sortie sont un fichier unique de TT harmonisés et dé – doublonnés avec un indicateur d'origine pour chaque TT.

Chaque TT passe par toutes les phases FUSION, CONTROLE, DECODAGE (SIMPLIFICATION, HARMONISATION et STOCKAGE), DE-DOUBLONNAGE. Le sous-système pré - traitement s'appuie sur les grammaires de descriptions de format des divers constructeurs.

- La phase de contrôle vérifie la structure syntaxique des TT. Si un TT ne possède pas un format reconnaissable par l'application en référence à la grammaire associée, il est rejeté.
- La phase de décodage, simplification, harmonisation transforme les champs codés utiles à l'harmonisation, applique les règles d'harmonisation et de simplification et génère un TT de format unique lisible par le sous-système calcule des coûts.
- La phase de dé – doublonnage élimine les doublons éventuellement présents dans les fichiers harmonisés.

Chaque TT subit en premier lieu un contrôle syntaxique global avant d'être décodé et harmonisé.

Les champs utiles des tickets sont récupérés et décodés.

Et en dernier lieu, les tickets sont harmonisés en un ticket unique à l'aide de règles d'harmonisation et à partir des champs utiles.

Les TT dont les champs ne sont conformes à la grammaire correspondante sont rejetés dans un fichier dédié stocké dans un répertoire dédié aux TT rejetés.

IV.3.3. Valorisation des tickets :

La finalité de ce sous-système est d'obtenir dans le détail les coûts de toutes les connexions aux réseaux de l'opérateur dans le but de pouvoir facturer par la suite les clients concernés.

La prochaine étape consiste à valoriser chaque ticket de taxation en se basant sur un plan tarifaire qui aurait été établi au préalable.

Cette étape devra se terminer par la production de ticket contenant les informations nécessaire pour être pris en charge par un opération de calcul de la facture. Le ticket dans cette nouvelle forme (nous parlons de tickets enrichi) sera alors stocké dans une base de données dédié au système de gestion commerciales des produits de X25. Ainsi les étapes principales de cette phase sont :

IV.3.3.1. Enrichissement des TT :

un ticket de taxation est émis, représentant une partie de la communication ou la communication entière. Ce TT est enrichi en fonction des données qu'il contient, des données client et des données du plan tarifaire.

IV.3.3.2. Valorisation des TT :

La valorisation permet de déterminer les coûts unitaires des communications. Ce coût dépend des valeurs du ticket comme la durée et le volume échangé lors de la communication 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, Le volume échangé durant la connexion et la durée de la connexion sont analysés pour déterminer les coûts unitaires du ticket.

Les éléments valorisés sont :

- Le volume sur les tranches tarifaires 1,2 et 3.
- Le volume réel.
- Coût volume sur les tranches tarifaires 1,2 et 3.
- Coût durée réel.
- Coût durée détail.
- Coût volume détail.
- Coût réel servant pour les calcules des montants de la facture globales, les coûts détails servent pour les lignes de la facture détaillée.

IV.3.3.3. Réconciliation :

Tous les TT enrichis et valorisés constituant une communication ouverte sont réconciliés afin de vérifier l'intégrité de la communication fermée qu'ils représentent.

IV.3.3.4. Stockage :

Les TT enrichis, valorisés et réconciliés sont stockés.

IV.3.3.5. Rejet et recyclage :

un ticket de taxe sortant de la phase de préparation n'est pas forcément valorisable, il est dans ce cas contrôlé et éventuellement rejeté au moment de la valorisation d'usage dans la base pour modification manuelle ultérieure sous IHM.

IV.3.3.6. Pondération :

Pendant la durée d'une connexion réseau un abonné 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 consiste à pondérer le volume total échangé sur chaque tranche en fonction du temps passé dans chaque tranche horaire.

IV.3.4. Sous système coûts des services :

Il consiste à :

- Analyse des données d'abonnement.
- Valorisation des mises en service associées au client.
- Valorisation des modifications d'abonnement associées au client.
- Valorisation des abonnements associés au client.

L'opération «Coût de service» est l'étape précédant la facturation. Elle est indépendante de l'opération calcule des coûts 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.

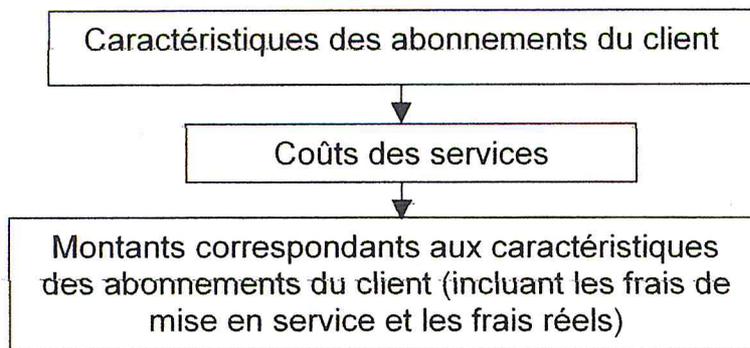


Figure IV. 5 Représentation d'opération coûts des service

IV.3.4.1. Lancement de l'opération «coûts des services » :

La date de lancement de l'opération «coûts des services ». Dépend des caractéristiques du client :

- Périodicité de la facturation.
- Parité de la périodicité.
- Jour de facturation.
- Etat du client.
- Date de dernière facture.

Il faut lancer l'opération «coûts des services ». 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 postérieur ou égal 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.
- Dont la date de facturation correspond à la date de facturation actuelle.

Exemples de calcul de date de lancement de l'opération «coûts des services ».:

⇒ périodicité de la facturation = «B» (bimensuel)

parité de la périodicité = «P» (paire)

jour de facturation = 10

défini les jours de lancement suivants :

10/02/2000, 10/04/2000, 10/06/2000, 10/08/2000, 10/10/2000, 10/12/2000 ...

- ⇒ périodicité facturation = «B» (bimensuel)
parité de la périodicité = «I» (impaire)
jour de facturation = 15
définit les jours de lancement suivants :
15/01/2000, 15/03/2000, 15/05/2000, 15/07/2000, 15/09/2000, 15/11/2000 ...

Le jour de facturation est toujours compris entre le 1^{er} et le 28^{ème} jours du mois, pour éviter, les problèmes de débordement.

IV.3.4.2. Période de calcul de l'opération «coûts des services » :

La période prise en compte dépend de la date de la précédente facture :

- Première facturation : la date de précédente facturation n'est pas renseignée.
la période de calcul = l'intervalle entre la date de facturation actuelle – nombre de mois défini par la périodicité (inclus) et la date de facturation actuelle (exclus).

Par exemple,

- ⇒ périodicité facturation = «B» (bimensuel)
date de facturation actuelle = 10/08/2000
la période de calcul est : [10/06/2000, 10/08/2000[

- Facturation autre que la première : la date de précédente facture est renseignée.
la période de calcul = l'intervalle entre la date de précédente facture (inclus) et la date de facturation actuelle (exclus).

Par exemple :

- ⇒ date de précédente facture = 10/07/2000
date de facturation actuelle = 10/08/2000
la période de calcul est : [10/07/2000, 10/08/2000[

Dans cette opération les éléments suivantes doivent être calculés :

1. Nombre de mois de la redevance d'un abonnement :

Le nombre de mois de la redevance dépend de la périodicité de la facturation.

a) Première facturation :

Il s'agit de la première facturation si la date de la précédente facture n'est pas renseignée.

Dans le cas de la première facturation :

Nombre de mois de redevance = nombre de mois correspondant à la période [date d'ouverture de l'abonnement date de facturation actuelle] + nombre de mois correspondant à la périodicité de la facturation.

Attention :

s'il s'agit à la fois de la première et dernière facturation, il ne faut pas ajouter le nombre de mois correspondant à la périodicité de facturation.

b) Dernière facturation :

Il s'agit de la dernière facturation si la date de résiliation de l'abonnement est saisie et que celle-ci est antérieure à la date de facturation actuelle.

Dans le cas de la dernière facturation, il ne faut pas facturer la redevance. **Le nombre de mois de redevance = 0.**

c) Facturation autre que la première et la dernière (fonctionnement nominal) :

Le nombre de mois de la redevance = nombre de mois correspondant à la périodicité de facturation.

Par exemple,

périodicité facturation = «B» (bimensuel)

date de facturation actuelle = 10/08/2000

Le nombre de mois de la redevance est 2 et couvre la période [10/08/2000, 10/10/2000[.

2. L'accès d'un abonnement :

On distingue les frais de mise en service et les frais d'évolution durant la période de l'abonnement.

Les frais de mise en service correspondent à la première mise en service réalisée sur l'abonnement.

Les frais d'évolution correspondent aux frais liés aux évolutions de certains paramètres de raccordement qui ont suivies la première mise en service (Vitesse de la ligne, modem, type de raccordement).

Les frais d'évolutions sont facturés à chaque nouvelle évolution effectuée sur la période à facturer.

(a) 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 = $(100 - \text{réduction sur les frais de mise en service}) / 100$ pour l'abonnement.

⇒ Réduction client sur les frais de mise en service = $(100 - \text{réduction sur les frais de mise en service}) / 100$ pour le client

Frais de mise en service = (Frais de mise en service X Réduction abonnement sur les frais de mise en service X Réduction client sur les frais de mise en service) si la date d'ouverture de l'abonnement appartient à la période de calcul de l'opération « coûts des services », sinon c'est égal à zéro.

(b) 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 = $(100 - \text{réduction sur les frais d'évolution}) / 100$ pour l'abonnement

⇒ Réduction client sur les frais d'évolution = $(100 - \text{réduction sur les frais d'évolution}) / 100$ pour le client

Frais d'évolution = $\Sigma(\text{Frais d'évolution X Réduction abonnement sur les frais d'évolution X Réduction client sur les frais d'évolution})$ sur la période de calcul de l'opération « coûts des services ».

3. L'abonnement :

L'abonnement comprend :

- ❑ La redevance des accès liés au type de raccordement, au modem et au débit.
- ❑ La redevance des services complémentaires.

3.1. Redevance des accès :

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

- ❖ Réduction abonnement sur la redevance = $(100 - \text{réduction sur la redevance}) / 100$ pour l'abonnement.
- ❖ Réduction client sur la redevance = $(100 - \text{réduction sur la redevance}) / 100$ pour le client

Redevance des accès = (Redevance abonnement X Nombre de mois redevance X Réduction abonnement sur la redevance X Réduction client sur la redevance) pour le service actif à la date de facturation actuelle.

On choisit la bonne occurrence en fonction du débit de l'accès, de la fourniture du modem et du type de raccordement.

3.2. Redevance des services complémentaires :

Les services complémentaires comprennent :

- Le service GFA.
- Les services VL.
- Les autres services complémentaires (incluant les frais de facture détaillée et les frais de facture mensuelle).

3.2.1. Redevance des services GFA :

Redevance des services GFA = Redevance service GFA X Nombre de mois redevance X Nombre de services pour le service actif à la date de facturation actuelle.

3.2.2. Redevance des services VL :

Redevance des services VL = Redevance service VL X Nombre de mois redevance X Nombre de services pour le service actif à la date de facturation actuelle.

3.2.3. Frais de la facture détaillée :

Frais de la facture détaillée pour le code «DETAIL».

3.2.4. Frais de la facture mensuelle :

Frais de la facture mensuelle pour le code «MENSUEL».

3.2.5. Redevance des autres services complémentaires :

Redevance des autres services complémentaires = Frais de la facture détaillée + Frais de la facture mensuelle + Σ (Redevance autre service complémentaire X Nombre de mois redevance X Nombre de services pour le service actif à la date actuelle).

IV.3.5. La Facturation :

Le sous système à pour rôle :

- Calcul des valeurs en fonction des données clients (facturation au forfait, réduction, location de modem, vitesse de la ligne, facture détaillée ...) et des TT.
- Edition des factures en fonction des données calculées.
- Stockage des factures telles qu'elles sont éditées.
- Constitution d'un fichier récapitulatif des données calculées. Ce fichier sera récupéré par le système comptable.

IV.3.5.1. Date de lancement de la facturation :

Tous les jours, l'exploitant lance la facturation incluant l'opération « coûts des services. Afin de sécuriser le lancement de la facturation, les dates de facturation sont stockées dans une table, Un seul enregistrement dans la table ou une liste de toutes les dates de facturation dans l'état IV. Cette table permet de suivre l'état des facturations journée par journée. Elle permet aussi de ne pas oublier de facturer des clients notamment dans le cas où un problème système persiste pendant plusieurs jours empêchant ainsi le calcul et l'édition des factures.

Le mécanisme des dates de facturation permet de gérer les cas de reprise : si la dernière date de facturation n'est pas la date du jour, cela signifie qu'il reste une ou plusieurs journées de facturation à traiter. On ne passe à la journée suivante que si toutes les factures ont bien été éditées. En cas d'anomalie, un redémarrage en mode reprise permet de reprendre le traitement sur la dernière date de facturation enregistrée et de terminer les facturations en attente.

Au lancement de la facturation, il faut contrôler la dernière date de facturation.

Si son état est 'A' (à valoriser), cela signifie que la facturation a été interrompue et que l'on est dans une phase de reprise.

Si son état est 'V' (valorisé), cela signifie que la facturation du jour s'est bien passée. Si cette date est antérieure à la date du jour, c'est qu'il y a au moins une facturation d'un jour à effectuer. Il faut donc insérer une nouvelle date correspondant à la dernière date + 1 jour. Son état est 'A' (à valoriser). Le traitement de cette journée peut commencer. Lorsque ce traitement sera terminé, il faudra mettre l'état de cette date à 'V' (valorisé).

IV.3.5.2. Calcule des paramètres de la Facturation :

La facturation se base sur le calcule des paramètres selon les deux phases :

1. Phase1 :

1.1. Frais total de mise en service des accès :

Frais total de mise en service des accès = Σ (Frais de mise en service des accès) pour tous les abonnements du client.

1.2. Frais total d'évolution des accès :

Frais total d'évolution des accès = Σ (Frais d'évolution des accès) pour tous les abonnements du client.

1.3. Redevance totale des accès :

Redevance totale des accès = Σ (Redevance des accès) pour tous les abonnements du client.

1.4. Redevance totale des services complémentaires :

Redevance totale des services complémentaires =
 Σ (Redevance des services GFA
+ Redevance des services VL
+ Redevance des autres services complémentaires) pour tous les abonnements du client.

1.5. Redevance groupe GFA :

Redevance groupe GFA = Frais de création X (le nombre de GFA créés + le nombre de GFA supprimés sur la période de valorisation du produit)+ Nombre de GFA client X Tarif redevance GFA client

2. Phase 2 :

a. Définition des calculs par accès :

Les calculs par accès prennent en compte la période de validité de l'accès, La recherche de tickets pour un accès donné se fait en considérant la date de début du ticket.

Cette date doit être incluse dans la période de validité de l'accès.
De plus, l'adresse d'imputation de taxe des tickets doit faire référence au numéro X25 de l'accès.

Le numéro X25 de l'accès constitue les 9 caractères de l'adresse d'imputation de taxe des tickets.

b. Définition des tickets à facturer :

Les tickets à prendre en compte pour la facturation sont tous les tickets valorisés non encore facturés, antérieurs à la date de facturation demandée par le client.

En cas de réédition de facture, Les tickets à prendre en compte pour la facturation sont les tickets dont la date de facturation vaut la date de réédition souhaitée.

2.1. Valorisation des communications internationales non abouties :

l'opération «calcule des coûts» d'un tel ticket doit donner coût volume = zéro DA coût durée détail = zéro DA.

Faire coût durée réelle = coût durée détail = Tarif d'une communication internationale du groupe de pays où est rattaché le ticket.

Les communications non abouties seront détaillées sur la facture.

2.2. Contrôle du volume des communications pour un accès :

Avant de calculer les coûts des communications de l'accès à traiter, il faut effectuer un contrôle du volume des communications concernées par cet accès. En effet, il faut prendre en compte la notion de volume minimal facturé pour une communication aboutie (c'est à dire ayant une durée différente de zéro) donnée. Ainsi, tous les tickets dont le volume de communication est inférieur à un volume minimal subissent un ajustement de leur volume.

2.2.a. Définition du volume minimal :

Ce volume minimal est établi en fonction de la destination de la communication c'est-à-dire du groupe de pays auquel est attaché le ticket.

2.2.b. Conditions d'application du contrôle du volume des communications :

Le contrôle du volume des communications s'applique aux tickets à facturer c'est-à-dire dont la date de facturation n'a pas été renseignée. Il faut aussi que ces tickets représentent des communications abouties c'est-à-dire que les durées des tickets sont non nulles.

2.2.c. Conséquences du changement de volume dans les communications :

Le volume transmis et reçu courant de la communication du ticket est comparé au volume minimal facturé pour le groupe auquel appartient le ticket. Si ce volume est inférieur au volume minimal, alors le volume transmis et reçu devient égal au volume minimal.

2.2.d. Calculs liés au changement de volume des communications

Etant donné que les volumes peuvent être nuls dans certains cas, le calcul des volumes sur les tranches est basé sur les durées des tickets calculées et contrôlées par la valorisation d'usage. Ainsi les volumes sont déterminés par :

- Nouveau volume tranche 1 = ancien volume tranche 1 X durée tranche 1 / durée ticket.
- Nouveau volume tranche 1 = volume minimum facturé pour le groupe X volume tranche 1 / volume ticket.
- Nouveau volume tranche 2 = ancien volume tranche 2 X durée tranche 2 / durée ticket.
- Nouveau volume tranche 2 = volume minimum facturé pour le groupe X volume tranche 2 / volume ticket.

- Nouveau volume tranche 3 = ancien volume tranche 3 X durée tranche 3 / durée ticket.
- Nouveau volume tranche 3 = volume minimum facturé pour le groupe X volume tranche 3 / volume ticket.

La durée du ticket doit être contrôlée, différente de 0.

- Nouveau coût volume = [nouveau volume tranche 1 X tarif tranche 1] + [nouveau volume tranche 2 X tarif tranche 2] + [nouveau volume tranche 3 X tarif tranche 3]

S'il ne s'agit pas d'une communication nationale associée à un forfait complet, il faut recalculer les nouveaux coûts réels pour chaque tranche :

- Nouveau coût réel tranche 1 = nouveau volume tranche 1 X tarif tranche 1
- Nouveau coût réel tranche 2 = nouveau volume tranche 2 X tarif tranche 2
- Nouveau coût réel tranche 3 = nouveau volume tranche 3 X tarif tranche 3

2.3. Nombre de communications nationales pour un accès :

Nombre de communications nationales pour un accès = Σ (nombre de tickets à facturer) pour les tickets nationaux et pour l'accès considéré.

2.4. Nombre de communications internationales pour un accès :

Nombre de communications internationales pour un accès = Σ (nombre de tickets à facturer) pour les tickets internationaux et pour l'accès considéré. Les communications non abouties sont prises en compte.

2.5. Volume des communications nationales sur chaque tranche tarifaire pour un accès :

- ❖ Volume en Koctets des communications nationales pour la tranche tarifaire 1 pour un accès = Σ (Volume sur la tranche tarifaire 1) pour les tickets à facturer, pour les tickets nationaux et pour l'accès considéré.
- ❖ Volume en Koctets des communications nationales pour la tranche tarifaire 2 pour un accès = Σ (Volume sur la tranche tarifaire 1) pour les tickets à facturer, pour les tickets nationaux et pour l'accès considéré.
- ❖ Volume en Koctets des communications nationales pour la tranche tarifaire 3 pour un accès = Σ (Volume sur la tranche tarifaire 1) pour les tickets à facturer, pour les tickets nationaux et pour l'accès considéré.

2.6. Volume des communications internationales sur chaque tranche tarifaire pour un accès :

- ❖ Volume en Koctets des communications internationales pour la tranche tarifaire 1 pour un accès = $\Sigma(\text{Volume sur la tranche tarifaire 1})$ pour les tickets à facturer, pour les tickets internationaux et pour l'accès considéré.
- ❖ Volume en Koctets des communications internationales pour la tranche tarifaire 2 pour un accès = $\Sigma(\text{Volume sur la tranche tarifaire 2})$ pour les tickets à facturer, pour les tickets internationaux et pour l'accès considéré.
- ❖ Volume en Koctets des communications internationales pour la tranche tarifaire 3 pour un accès = $\Sigma(\text{Volume sur la tranche tarifaire 3})$ pour les tickets à facturer, pour les tickets internationaux et pour l'accès considéré.

2.7. Coût des communications nationales par tranche tarifaire et par accès :

- ❖ Coût des communications nationales pour la tranche tarifaire 1 et par accès = $\Sigma(\text{Coût ticket pour la tranche 1})$ pour les tickets à facturer, pour les tickets nationaux et pour l'accès considéré.
- ❖ Coût des communications nationales pour la tranche tarifaire 2 et par accès = $\Sigma(\text{Coût ticket pour la tranche 2})$ pour les tickets à facturer, pour les tickets nationaux et pour l'accès considéré.
- ❖ Coût des communications nationales pour la tranche tarifaire 3 et par accès = $\Sigma(\text{Coût ticket pour la tranche 3})$ pour les tickets à facturer, pour les tickets nationaux et pour l'accès considéré.

2.8. Durée totale de communications nationales pour un accès :

Durée totale de communications nationales pour un accès = $\Sigma(\text{durée des tickets})$ pour les tickets à facturer, pour les tickets nationaux et pour l'accès considéré.

2.9. Coût pour la durée des communications nationales pour un accès :

Coût pour la durée des communications nationales pour un accès = $\Sigma(\text{coût durée réel des tickets})$ pour les tickets à facturer, pour les tickets nationaux et pour l'accès considéré.

2.10. Nombre de communications internationales non abouties pour un accès

Nombre de communications internationales non abouties pour un accès = $\Sigma(\text{nombre de tickets à facturer dont le volume et la durée sont nuls})$ pour les tickets internationaux et pour l'accès considéré.

2.11. Coût des communications internationales par tranche tarifaire et par accès :

- ❖ Coût des communications internationales pour la tranche tarifaire 1 et par accès = $\Sigma(\text{Coût ticket pour la tranche 1})$ pour les tickets à facturer, pour les tickets internationaux et pour l'accès considéré.
- ❖ Coût des communications internationales pour la tranche tarifaire 2 et par accès = $\Sigma(\text{Coût ticket pour la tranche 2})$ pour les tickets à facturer, pour les tickets internationaux et pour l'accès considéré.
- ❖ Coût des communications internationales pour la tranche tarifaire 3 et par accès = $\Sigma(\text{Coût ticket pour la tranche 3})$ pour les tickets à facturer, pour les tickets internationaux et pour l'accès considéré.

2.12. Coût total pour la durée des communications internationales pour un accès :

Coût pour la durée des communications internationales pour un accès = $\Sigma(\text{coût durée réel des tickets})$ pour les tickets à facturer, pour les tickets internationaux et pour l'accès considéré.

2.13. Coût total forfait complet :

- ❖ Réduction client forfait complet = $(100 - \text{réduction forfait complet pour le client}) / 100$ pour le client.
- ❖ Réduction abonnement forfait complet = $(100 - \text{réduction forfait complet pour l'abonnée}) / 100$ pour l'abonnement.
- ❖ Coût total forfait complet = Réduction client forfait complet X $\Sigma(\text{Coût forfait complet X Réduction abonnement forfait complet X nombre de mois de la redevance})$ pour tous les accès forfait complet.

2.14. Coût total forfait à la durée :

- ❖ Réduction client forfait durée = $(100 - \text{réduction forfait à la durée pour le client}) / 100$ pour le client.
- ❖ Réduction abonnement forfait durée = $(100 - \text{réduction forfait à la durée pour l'abonnée}) / 100$ pour l'abonnement.
- ❖ Coût total forfait à la durée = Réduction client forfait durée X $\Sigma(\text{Coût forfait durée})$ X Réduction abonnement forfait durée X nombre de mois de la redevance) pour tous les accès forfait durée.

2.15. Valeur de remise dégressivité au volume :

- ❖ Volume total des communications hors forfait complet = $\Sigma(\text{volume réel hors forfait complet des tickets})$ pour les tickets nationaux à facturer, pour tous les accès concernés par la dégressivité.
- ❖ Valeur de remise dégressivité au volume = $((100 - \text{réduction de dégressivité}) / 100)$ pour la tranche dont le volume des communications hors forfait complet

2.16. Coût total des communications réelles nationales :

Coût total des communications réelles nationales = $\Sigma(\text{Coût pour la durée des communications nationales pour un accès})$ pour tous les accès + Valeur remise dégressivité au volume X $\Sigma(\text{Coût des communications nationales pour la tranche tarifaire 1} + \text{Coût des communications nationales pour la tranche tarifaire 2} + \text{Coût des communications nationales pour la tranche tarifaire 3})$ pour tous les accès concernés par la dégressivité renseigné + $\Sigma(\text{Coût des communications nationales pour la tranche tarifaire 1} + \text{Coût des communications nationales pour la tranche tarifaire 2} + \text{Coût des communications nationales pour la tranche tarifaire 3})$ pour tous les accès concernés par la dégressivité non renseigné.

2.17. Coût total des communications internationales :

Coût total des communications internationales = $\Sigma(\text{Coût total pour la durée des communications internationales pour un accès})$ pour tous les accès + $\Sigma(\text{Coût des communications internationales pour la tranche tarifaire 1 pour un accès} + \text{Coût des communications internationales pour la tranche tarifaire 2 pour un accès} + \text{Coût des communications internationales pour la tranche tarifaire 3 pour un accès})$ pour tous les accès.

IV.3.6. Gestion des clients et abonnements :

L'aboutissement de la création d'un client est la création d'abonnement.
La démarche commerciale d'un client est une action sur son (ou ses) abonnements.
Cette action peut être une création, une modification ou une résiliation d'abonnement.

- Un abonnement est toujours rattaché à un client unique.
- Un client peut souscrire à plusieurs abonnements.
- Un client possède au moins un abonnement.
- Un abonnement peut exister sans être rattaché à un accès réseau.
- Un client n'ayant aucun abonnement valide devient invalide.
- Un abonnement est en Attente de validation (attend un numéro d'accès), Confirmé ou Résilié.

IV.3.6. Etat divers :

La partie transactionnelle est composé de plusieurs transactions pour le paramétrage et la consultation :

- ◆ Fonctionnel :
 - Données clients.
 - Données abonnements.
 - Données accès réseau et adresse géographiques.
 - Plan tarifaire.
 - Tranches horaires.
 - Services complémentaires...

- ◆ Système et administration.
 - Paramétrage de configuration système.
 - Gestion des utilisateurs.
 - Gestion des alarmes
 - Visualisation de la base de consultation.

Chapitre V

**CONCEPTION DE SYSTEME
SELON L'APPROCHE
COMPOSANT**

V.1. Introduction :

Le processus de développement consiste à :

- ❑ Déterminer les composantes d'un système.
- ❑ Les interconnecter (Assembler) pour former une configuration pour l'architecture.
- ❑ Raffiner en Répétant le processus pour chaque composant.
- ❑ Le processus s'arrêtera quand le composant utilisé existe ou doit être directement programmé dans un langage de programmation.

Le système de gestion commerciale de réseau X25 a pour objet de générer les factures des clients de l'opérateur utilisant le réseau à partir des tickets de taxations émis et des données clients. De plus, il met à disposition un fichier contenant les informations nécessaires au service comptable.

Le but est de mettre en place un site capable de facturer les clients de l'opérateur et ce, quelque soit le réseau utilisé.

V.2. Architecture globale de système :

A partie de schéma informel décrit en chapitre IV on peut tirer l'architecture globale de système suivante :

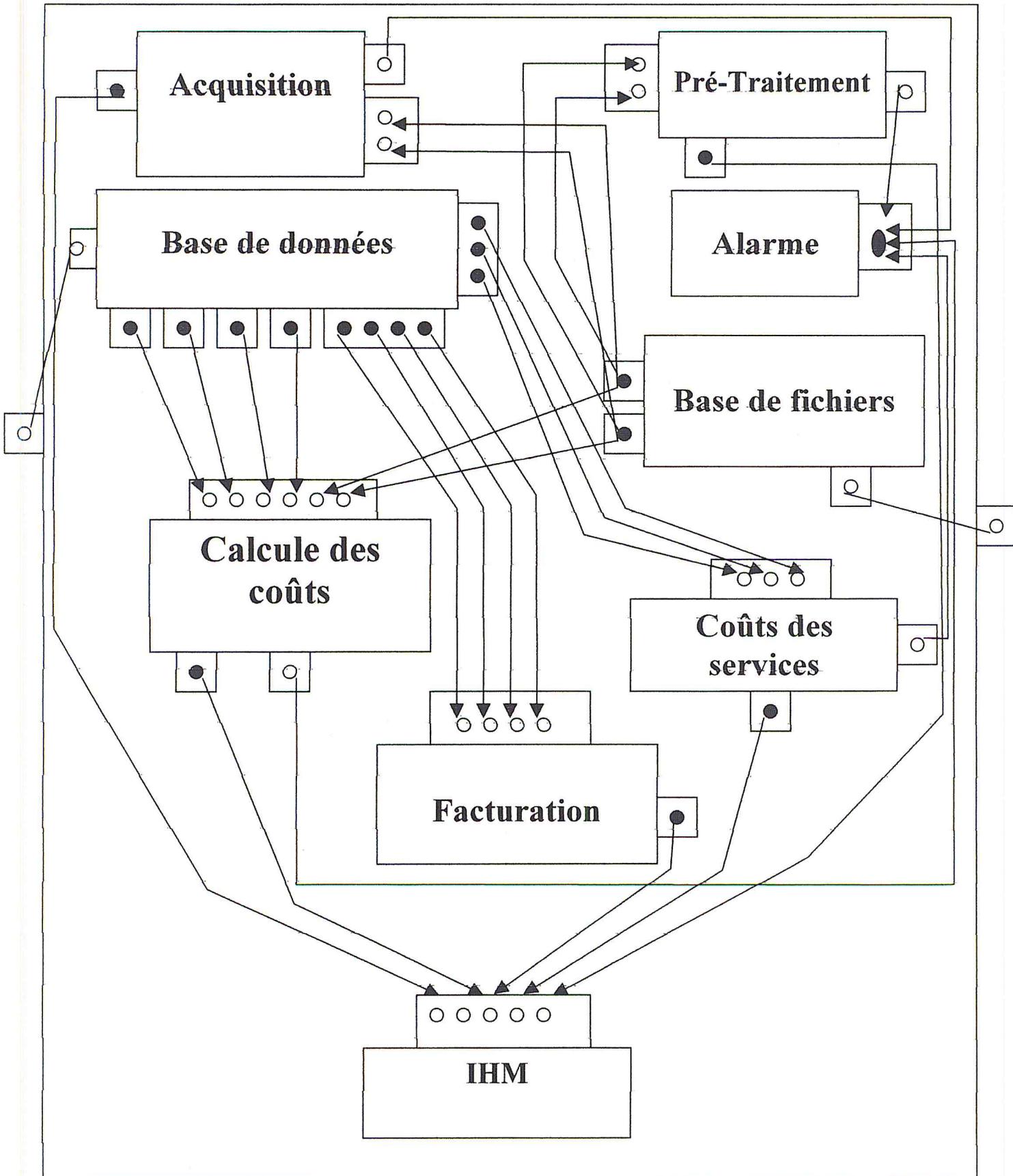


Figure V. 1 Architecture globale de système

Les données en entrée du système sont :

- Les TT de divers constructeurs
- Les données clients saisis par l'interface de l'application.
- Le plan tarifaire (tarifs, plages horaires...).
- Le paramétrage du système.
- Les autres données en entrée correspondent à la saisie des services ou des diverses données de configuration.

Remarques :

Les formats de ticket, peuvent contenir des communications nationales ou internationales.

Données intermédiaires :

Les types de données intermédiaires présentes dans le système :

- TT réceptionnés,
- TT valorisés et réconciliés.
- Les TT valorisés sont conservés puis stockés pendant une période donnée.

Les données en sortie du système sont :

- Impressions des factures des clients facturables.
- La facture telle qu'elle a été envoyée au client (visualisable par un opérateur).
- La facture détaillée de chaque client facturé (visualisable par un opérateur).
- 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.

V.3. Conception des sous composants :

V.3.1. Sous composant «Pré – traitement» :

L'architecture globale de sous composant «Pré – traitement » :

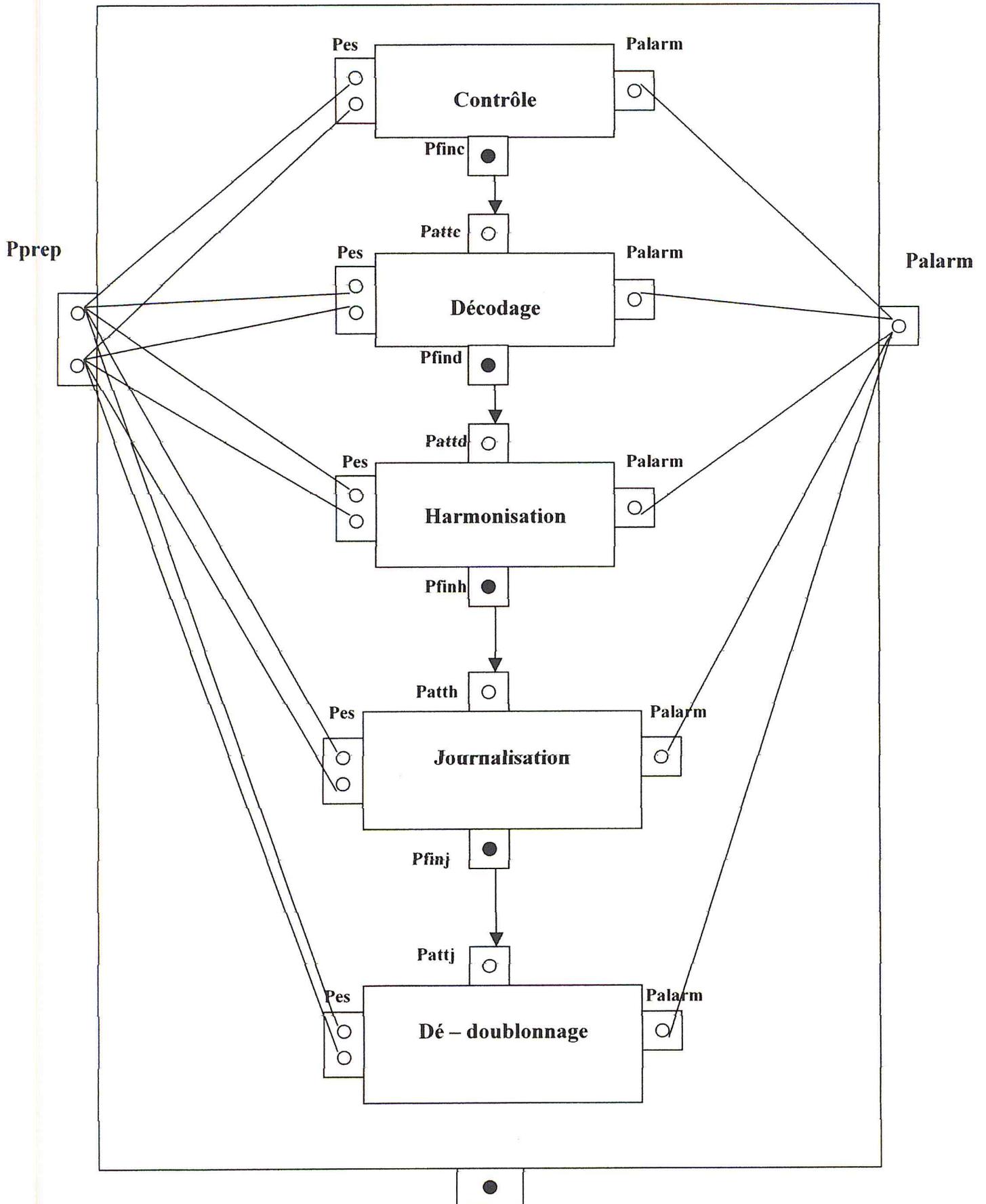


Figure V. 2 L'architecture globale de sous composant «Pré-traitement»

Le sous-système pré – traitement se décompose en cinq traitements :

- Contrôle.
- Décodage.
- Harmonisation.
- Dédoublonnage.
- Journalisation.

Les TT sont contrôlés, décodés, harmonisés et dé-doublonnés. Ce sous-système a pour but d'uniformiser le format de tous les TT et de ne garder que les champs utiles au centre de facturation, il prépare les tickets de taxation, reçus des réseaux X25 de l'opérateur, pour le sous-système «calculs des coûts ».

Exemple des champs utiles et générique des TT :

(CG pour champ générique et CUS pour champ utile)

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

Code champ générique	Champs ticket	Code champ utile	Champs ticket
CG_01	Date de début du ticket	CUS_03	Date d'établissement de la connexion
CG_02	Heure de début du ticket	CUS_04	Heure d'établissement de la connexion
CG_03	Numéro d'ordre du ticket	-	-
CG_04	Marqueur d'origine	-	-
CG_05	Type de CV	-	-
CG_06	Nature du ticket	CUS_02	Cause – Diagnostic
CG_07	Date de début de communication	CUS_03	Date d'établissement de la connexion
CG_08	Heure de début de communication	CUS_04	Heure d'établissement de la connexion
CG_09	Durée du ticket	CUS_05	Durée de la connexion
CG_10	Imputation de taxes	CUS_02	- Taxation au demandé
CG_11	Adresse d'imputation des taxes	CUS_06 CUS_07	- 9 premiers digits Adresse appelé - 9 premiers digits Adresse appelant
CG_12	Adresse de l'appelant complète	CUS_07	Adresse appelant
CG_13	Adresse de l'appelé complète	CUS_06	Adresse appelé
CG_14	Destination	-	-
CG_15	Nombre d'octets transmis + reçus	CUS_08	Nombre total de paquets transmis et reçus et nombre total d'octets transmis et reçus

Tableau V. 1 Exemple des champs utiles et générique des TT

En Archjava :

```
public component class pré_traitement
{
  private final Controle Ctrl=new Controle();
  private final Decodage Dec=new Decodage();
  private final Journalisation Jour=new Journalisation();
  private final Harmonisation Harm=new Harmonisation();
  private final dedoublonnage Dd=new dedoublonnage();
  public port pprep
    { requires String[] getFile(String nom);
      requires void saveTable(String[] tab,String nom); }
  public port Palarme
    { requiers void put(Alarm a) ; }
  public port Pplane
    { provide void execute() ; }
  connect Ctrl.Pfinc,Dec.Pattc;
  connect Dec.Pfind,Harm.Pattd;
  connect Harm.Pfinh,Jour.Patth;
  connect Jour.Pfinj,Dd.Pattj;
  glue Palarm to Jour.Palarm, Dec.Palarm, Dd.Palarm, Ctrl.Palarm, Harm.Palarm ;
  glue pprep to Ctrl.Pes,Dec.Pes,Harm.Pes,Jour.Pes,Dd.Pes;
  // les procédures de traitement
  public void Execute()
  { // permet de faire les traitement nécessaire de l'opération complète de «pré –
    // traitement » }
}
```

V.3.1.1 Sous composant «Contrôle» :

Entrer : Fichier «TT_Constructeur.txt ».

- Chaque TT subit un contrôle syntaxique global avant d'être décodé et harmonisé.
- La phase de contrôle vérifie la structure syntaxique des TT, Si un TT ne possède pas un format reconnaissable par l'application en référence à la grammaire associée, il est rejeté.

Sortie : deux Fichiers «TT_Correct.Txt » et «TT_Rejet.Txt »

Format de ticket d'entrée utilisée :

Rang (CUS)	Rang	Exemple	Signification	Nombre de caractères
	1	10/12/99	Date d'enregistrement sur SVR	8
	2	16:08	Date d'enregistrement sur SVR	5
	3	MEGAPAC2	Nom du nœud	8
	4	{00}	Numéro d'ordre du ticket	0-3
	5	jean	Mot de passe composé par l'appelant / user data	8
3	6	10/12/99	Date d'établissement de la connexion	5-8
4	7	16:08	Heure d'établissement de la connexion	3-5
2	8	0/0-	Codes de cause et Diagnostic Taxation au demandé R présente - absente	1-3/1-3 1
5	9	1/55	Durée de la connexion minutes / secondes	1-n/1-n
7	10	EI00	Adresse de l'appelant	9-15
6	11	BE00	Adresse de l'appelé	9-15
8	12	0001 ;0002 ; 0013;0085	Nombre de paquets transmis et reçus ; Nombre total d'octets transmis et reçu	2-n ;2-n 2-n ;2-n

Tableau V. 2 Format de ticket d'entrée utilisée

Rejets des TT :

Les TT dont les champs ne sont conformes à la grammaire correspondante sont rejetés dans un fichier dédié stocké dans un répertoire dédié aux TT rejetés (C:/.../Fichiers/TT_Rejet.Txt).

En Archjava :

```
public component class Controle
{
    public port Pes
        { requires String[] getFile(String nom);
          requires void saveTab(String[] tab,String nom);    }
    public port Pfinc
        { Provides boolean Fin_Contrôle () ; }
    public port Palarme
        { requiers void put(Alarm a) ; }
    // les procédures de traitement
    public boolean Fin_Contrôle ()
        { // Indiquer par un booléen le bon déroulement de l'opération(pas d'exception).}
    public void Controle()
        { // Exécution de l'opération de contrôle des tickets.}
    }
}
```

V.3.1.2. Sous composant «Décodage » :

Entrer : Fichiers «TT_Correct.txt » et «Gram_Constructeur.txt ».

- Elle transforme les champs codés utiles à l'harmonisation en ASCII.
- Elle applique les règles de simplification.
- Elle génère un TT de format unique lisible par le sous-système « calcule des coûts ».
- Consulter la grammaire et les tickets pour déterminer les champs d'origine et leur emplacement.
- Charger le fichier de grammaire et applique les règles de codage.

Sortie : Fichier «Fichier_Decode.Txt »

Format de ticket après décodage et récupération des champs utiles des TT :

Code du champ	Longueur du champ utile en octet	Nom du champ du ticket	Valeurs champ utile et commentaire
CUS_02	8 max.	Cause – Diagnostic	1-3/1-3- ou 1-3/1-3R Exemples : 0/0- ou 0/0R ou 255/255-
CUS_03	8 max.	Date d'établissement de la connexion	JJ/MM/A ou JJ/MM/AA ou J/MM/AA ou J/MM/A
CUS_04	5	Heure d'établissement de la connexion	HH:mm
CUS_05	pas de longueur maximum	Durée de la connexion minutes/secondes	1-n/1-n Exemples :0/0,120/45,5/55
CUS_06	15 max.	Adresse de l'appelé	Exemples : 110100002151515
CUS_07	15 max.	Adresse de l'appelant	Exemples : 110100002151515
CUS_08	pas de longueur maximum	Nombre total de paquets transmis et reçus et nombre total d'octets transmis et reçus	2-n;2-n ;2-n ;2-n

Tableau V. 3 Format de ticket après décodage et récupération des champs utiles des TT

En Archjava :

```
public component class Decodage
{
public port Pes
{ requires String[] getFile(String nom);
requires void saveTab(String[] tab,String nom); }
public port Pattc
{ requires boolean Fin_Contrôle () ; }
public port Pfind
```

```

    { Provides boolean Fin_Décodage () ; }
public port.Palarme
    { requiers void put(Alarm a) ; }
// les procédures de traitement
public boolean Fin_Décodage ()
    { // indiquer par un booléen le bon déroulement de l'opération(pas d'exception). }
public void Decodage()
    { // Exécution de l'opération de décodage des tickets.}
}
    
```

V.3.1.3. Sous composant «Harmonisation» :

Utilisation de grammaires de description de format pour harmoniser les TT en un format unique, On ne conserve que les champs utiles à la valorisation.

Le but est d'obtenir les champs génériques du ticket harmonisé.

Entrer : Fichiers «Param_Constructeur.txt » et «Fichier_Decode.txt » et «Gram_Constructeur.txt».

Descriptions des règles fonctionnelles du Composant d'harmonisation :

- Une grammaire est définie pour chaque format de TT. Chaque évolution du format implique une modification de la grammaire.
- Les champs du ticket harmonisé sont prédéfinis.
- Les règles d'harmonisation sont prédéfinies.
- Les champs utiles aux règles d'harmonisation sont prédéfinis

Sortie : Fichier «Fichier_Harmonise.Txt » et «TT_Rejet.Txt ».

Structure du ticket harmonisé :

Champ du ticket	Libellé du champ	Valeurs et Commentaires
CG_01	Date de début du ticket	AAAAMMJJ pour année, mois jour
CG_02	Heure de début du ticket	HHmmss pour heure, minute seconde
CG_03	Numéro d'ordre du champ	000 Champ rempli par la phase de préparation
CG_04	Marqueur d'origine	S pour Constructeur1 A pour Constructeur1
CG_05	Type de CV	1 pour CVC 2 pour CVP 3 pour CVC secours 4 pour CVC passerelle
CG_06	Nature du ticket	TUA pour ticket unique constructeur N°2 TDA pour ticket début constructeur N°2 TSA pour ticket suite constructeur N°2 TFA pour ticket fin constructeur N°2 TUS pour ticket unique ou fin constructeur N°1 TDS pour ticket suite ou début constructeur N°1

Conception de Système selon l'approche Composant

		TKR ticket réconcilié (cette valeur est utilisée uniquement au moment de calcul des coûts lors de la réconciliation)
CG_07	Date de début de connexion	AAAAMMJJ pour année, mois jour
CG_08	Heure de début de connexion	HHmmss pour heure, minute seconde
CG_09	Durée du ticket	00000 5 digits en minutes (arrondi supérieur)
CG_10	Imputation de taxe	0 pour Appellant pour les appels normaux 1 pour Appelé pour les appels en PCV
CG_11	Adresse de l'imputation de taxe	00000000 9 digits représentant le numéro du réseau
CG_12	Adresse de l'appelant	000000000000000 9 digits représentant le numéro du réseau + 0 à 6 digits maximum représentant l'adresse du sous-réseau.
CG_13	Adresse de l'appelé	000000000000000 15 digit maximum (pouvant contenir l'indicatif international et l'indicatif X121 des pays)
CG_14	Destination	0 pour national 1 pour le groupe 1 2 pour le groupe 2 3 pour le groupe 3 4 pour le groupe 4
CG_15	Volume échangé	00000000 8 digits en koctets

Tableau V. 4 Structure du ticket harmonisé

Les champs utiles sont harmonisés en vue d'obtenir les champs génériques du ticket harmonisé.

Les règles d'harmonisation sont les suivantes :

Champ du ticket à obtenir	Libellé du champ	Règles d'harmonisation
CG_01	Date de début du ticket	Récupérer la valeur du champ CUS_03 transformé au format AAAAMMJJ
CG_02	Heure de début du ticket	Récupérer la valeur du champ CUS_04 transformé au format HHmmss en considérant ss=0 seconde.
CG_03	Numéro d'ordre du champ	Non rempli
CG_04	Marqueur d'origine	Rempli avec la valeur S
CG_05	Type de CV	1
CG_06	Nature du ticket	TDS si les 7 premiers digits du champ CUS_02 sont égaux à 255/255 TUS si les premier digits du champ CUS_02 différents de 255/255
CG_07	Date de début de connexion	égal au champ CG_01
CG_08	Heure de début de connexion	Egal au champ CG_02.
CG_09	Durée du ticket	Récupérer la valeur du champ CUS_05 au format minutes/secondes et le transformer en minute en arrondissant à la minute supérieure. Si CUS_05 > 99999 minutes alors CG_09 = 99999 minutes.
CG_11	Adresse de l'imputation de taxe	Récupérer les 9 premiers digits de la valeur du champ CUS_06 si le dernier digit du champ CUS_02 est à R Récupérer les 9 premiers digits de la valeur du champ CUS_07 sinon
CG_12	Adresse de l'appelant	Récupérer la valeur du champ CUS_07.

CG_13	Adresse de l'appelé	Récupérer la valeur du champ CUS_06.
CG_10	Imputation de taxe	1 le dernier digit du champ CUS_02 est à R 0 sinon
CG_14	Destination	Non rempli par ce traitement
CG_15	Volume échangé	Somme sur les deux derniers champs de CUS_08 séparés par un point virgule. Champ transformé en Koctet arrondi au Koctet supérieur (1Koctet=1024 octets)

Tableau V. 5 Les règles d'harmonisation

En Archjava :

```
public component class Harmonisation
{
public port Pes
    {requires String[] getFile(String nom);
    requires void saveTab(String[] tab,String nom); }
public port attd
    {requires boolean Fin_Décodage () }
public port Pfinh
    {Provides boolean Fin_harmonisation () ; }
public port Palarme
    {requiers void put(Alarm a) ; }
// les procédures de traitement
public boolean Fin_harmonisation ()
    { // indiquer par un booléen le bon déroulement de l'opération(pas d'exception). }
public void Harmonisation()
    { // Exécution de l'opération de Harmonisation des tickets. }
}
}
```

Stockage des tickets harmonisés :

Les tickets harmonisés sont stockés dans un seul fichier. L'ordre des champs lors du stockage est le suivant :

Champ du ticket	Libellé du champ
CG_01	Date de début du ticket
CG_02	Heure de début du ticket
CG_03	Numéro d'ordre du champ
CG_04	Marqueur d'origine
CG_05	Type de CV
CG_06	Nature du ticket
CG_07	Date de début de connexion
CG_08	Heure de début de connexion
CG_09	Durée du ticket
CG_10	Imputation de taxe
CG_11	Adresse de l'imputation de taxe
CG_12	Adresse de l'appelant
CG_13	Adresse de l'appelé
CG_14	Destination
CG_15	Volume échangé
Retour chariot + Saut de ligne	

Tableau V. 6 Stockage des tickets harmonisés

V.3.1.4. Sous composant «Journalisation » :

Entrer : Fichier «Fichier_Harmonise.Txt », «TT_Rejet.Txt » et «TT_Constructeur.txt »
Les données journalisées dans le fichier «Journal_Constructeur1.txt » sont :

- Le nombre de tickets en entrée de la préparation.
- Le nombre de tickets harmonisés.
- Le nombre de tickets rejetés.

Sortie : Fichier «Journal_Constructeur1.txt ».

En Archjava :

```
public component class journalisation
{
    public port Pes
        {requires String[] getFile(String nom);
         requires void saveTab(String[] tab,String nom);    }
    public port atth
        { requires boolean Fin_harmonisation () ; }
    public port Palarme
        { requiers void put(Alarm a) ; }
    public port pfinj
        { provide boolean Fin_Journalisation () ; }
    // Les procédures de traitement.
    public boolean Fin_Journalisation ()
        { // Indiquer par un booléen le bon déroulement de l'opération(pas d'exception). }
    public void Journalisation()
        { // Exécution de l'opération de journalisation des tickets. }
}
```

V.3.1.5. Sous composant «dé-doublonnage » :

Le rôle de composant «dé-doublonnage» :

Le fichier généré par le décodage et l'harmonisation est triée et dé – doublonné sur lui-même et par rapport au fichier des tickets de la précédente phase de préparation.

But : éliminer les doublons.

Méthode de traitement :

- Soit fich1 le nom du fichier de TT harmonisés à dé-doublonner.
- Soit fich2 le nom du fichier de TT harmonisés et dé-doublonnés de la précédente

préparation des tickets ayant le même marqueur d'origine.

- 1) le fichier fich1 est trié dans fich1bis.
- 2) les fichiers fich1bis et fich2 sont fusionnés ensemble dans un fichier fich3.
- 3) le fichier fich3 est dé-doublonné sur lui même dans un fichier fich4.
- 4) les doublons du fichier fich3 sont identifiés en faisant la différence entre fich3 et fich4 dans un fichier fich5 (qui contient alors les doublons).

le fichier fich1bis est dé-doublonné en faisant la différence entre le fichier fich1bis et fich5.

En Archjava :

```
public component class dedoublonnage
{
    public port Pes
        { requires String[] getFile(String nom);
          requires void saveTab(String[] tab,String nom);    }
    public port attj
        { requires boolean Fin_Journalisation ()    }

    public port Palarme
        { requiers void put(Alarm a) ; }
    // les procédures de traitement
    public void dedoublonnage()
    { // Exécution de l'opération de dé – doublonnage des tickets. }
}
```

V.3.2 Sous composant «calcule des coûts» :

L'architecture globale de sous composant «calcule des coûts» :

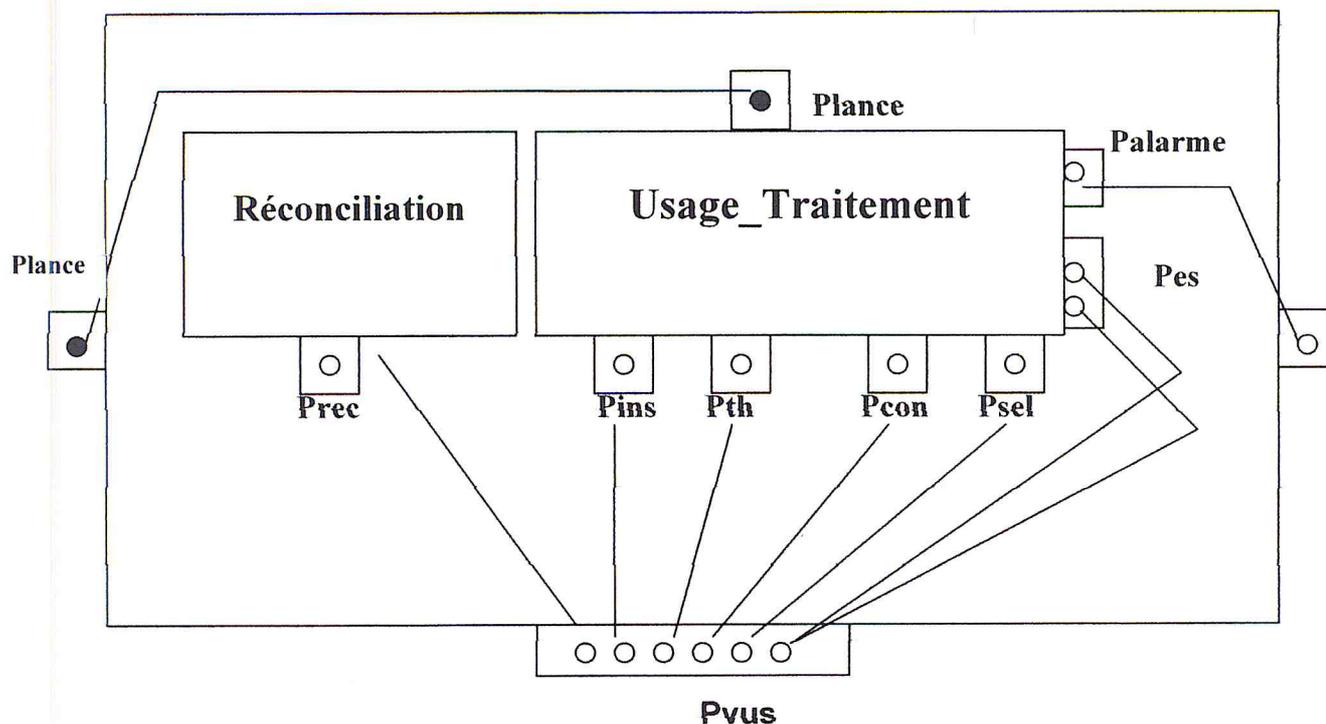


Figure V. 3 L'architecture globale de sous composant «calcul des coûts»

But :

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

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

Il est nécessaire de faire les vérifications de cohérence suivantes :

Mois : entre 1 et 12

Jour : entre 1 et 28, 29, 30, 31 suivant le numéro du mois

Heure : entre 0 et 23

Minute : entre 0 et 59

Durée du ticket : <= à 24h

L'adresse d'imputation associée à un abonnement valide

En Archjava :

```
public component class calcules_coûts
{
private final Usage_Traitement us = new Usage_Traitement ();
private final Réconciliation rec = new Réconciliation ();
```

```
public port pvus
    { requires void Inserter(String s) ;
      requires void Connecter() ;
      requires String[][] Selection() ;
      requires int[][] Select_TH() ;
      requires String[] getFile(String nom);
      requires void saveTable(String[] tab,String nom); }
public port Palarm { requiers void put(Alarm a) ; }
public port Plance { provide void execute() ; }
glue Palarm to us.Palarm ;
glue Plance to us.Plance ;
glue pvus to us.pins,us.pcon,us.psel,us.pth,us.pes,rec.Prec;
// les procédures de traitement
public void execute()
    { // Traitement et valorisation des tickets . }
}
```

V.3.2.1. Sous composant «Usage Traitement» :

Le sous composant «Usage_Traitement» génère des données servant à la facturation des communications. Ces données sont :

- la pondération du volume échangé 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 réel pour la durée, coût tenant compte du type de tarification de l'accès (forfait complet, forfait durée, réel).
- le coût pour le volume pour la facture détaillée, coût ne tenant pas compte du type de tarification de l'accès (forfait complet, forfait durée, réel)
- le coût pour la durée pour la facture détaillée, coût ne tenant pas compte du type de tarification de l'accès (forfait complet, forfait durée, réel)

Les paramètres à déterminer lors de la valorisation sont :

- Le volume sur la tranche tarifaire 1.
- Le volume sur la tranche tarifaire 2.
- Le volume sur la tranche tarifaire 3.
- Le volume réel.

Coût volume sur la tranche tarifaire 1 réel (dans ce cas on tient compte du type de tarification de l'accès, c'est à dire que si l'accès est au forfait complet et si la communication est nationale, le coût sur la tranche tarifaire 1 est nul).

- Coût volume sur la tranche tarifaire 2 réel (dans ce cas on tient compte du type de tarification de l'accès, c'est à dire que si l'accès est au forfait complet et si la communication est nationale, le coût sur la tranche tarifaire 2 est nul).
- Coût volume sur la tranche tarifaire 3 réel (dans ce cas on tient compte du type de tarification de l'accès, c'est à dire que si l'accès est au forfait complet et si la communication est nationale, le coût sur la tranche tarifaire 3 est nul).
- Coût durée réel (dans ce cas on tient compte du type de tarification de l'accès, c'est à dire que si l'accès est au forfait complet ou au forfait sur la durée et si la communication est nationale, le coût durée réelle est nul).
- Coût durée détail (dans ce cas on ne tient pas compte du type de tarification de l'accès, c'est à dire que le coût est systématiquement calculé. Cette valeur sert à la facture détaillée).
- Coût volume détail (dans ce cas on ne tient pas compte du type de tarification de l'accès, c'est à dire que le coût est systématiquement calculé. Cette valeur sert à la facture détaillée).

Codification :

Libellé	Code
Date heure début TT	Tic_date_deb_tt
Date heure début com	Tic_heure_deb_tt
Durée tic	Tic_duree
Imputation de taxe	Tic_imp_taxe
Adr imputation de taxe	Tic_adr_taxe
Adr appelant complète	Tic_adr_applant
Adr appelé complète	Tic_adr_apellee
Nbre Ko transmis reçus	Tic_vol_echange
Vol sur tranche tarifaire 1	Tic_vol1
Vol sur tranche tarifaire 2	Tic_vol2
Vol sur tranche tarifaire 3	Tic_vol3
Coût tranche 1	Tic_cout1
Coût tranche 2	Tic_cout2
Coût tranche 3	Tic_cout3
Coût durée réel	Tic_coût_dur_reel
Coût durée détail	Tic_coût_dur_detail
Coût volume détail	Tic_coût_vol
Durée tranche 1	Tic_dur1
Durée tranche 2	Tic_dur2
Durée tranche 3	Tic_dur3
Tarif appliqué tranche 1	Tic_tar1
Tarif appliqué tranche 2	Tic_tar2
Tarif appliqué tranche 3	Tic_tar3
Type tarif appliqué	Tic_type

Tableau V. 7 Codification des champs.

Les coûts réels servent pour les calculs des montants de la facture globale. Les coûts détails servent pour les lignes de la facture détaillée.

Le groupe du ticket est déterminé par l'intermédiaire du champ **CG_13** :

- Si le premier digit de «Adresse de l'imputation de taxe » est 0, la communication est internationale. Les trois digits suivants du champ «Adresse de l'imputation de taxe » correspondent à l'indicatif pays.
- Si le premier digit de «Adresse de l'imputation de taxe » est différent de 0, la communication est nationale.

Exemple de pondération :

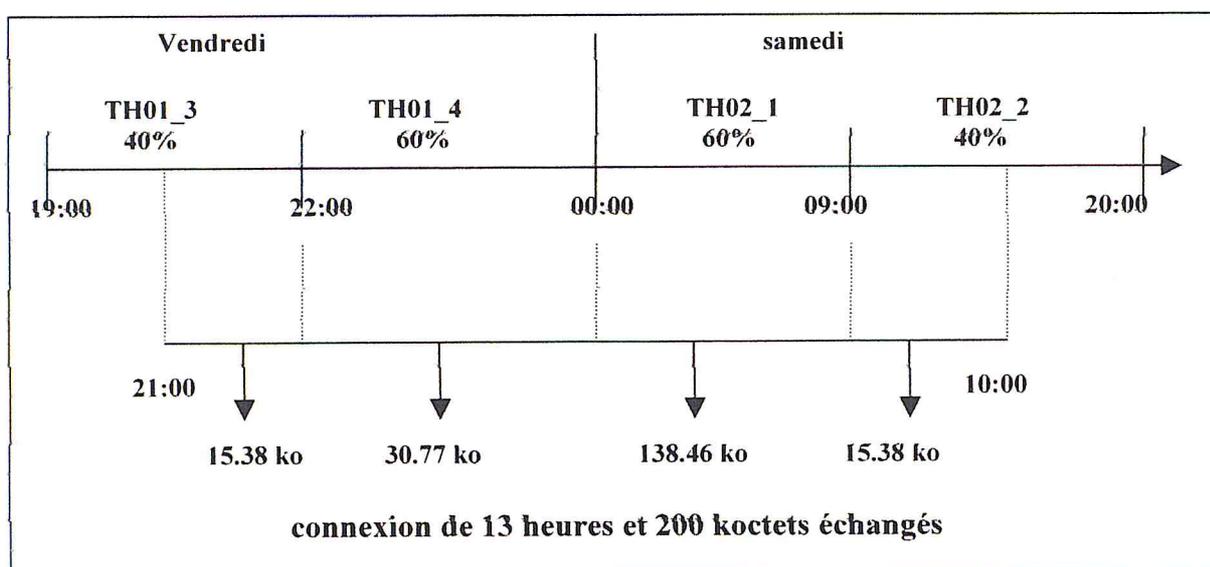


Figure V. 4 Pondération des volume

La pondération se calcule de la manière suivante :

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

Dans notre exemple cela donne :

Volume sur tranche THO1_3 = (1 heure / 13 heures) x 200 koctets = 15.38 koctets.
 Volume sur tranche THO1_4 = (2 heures / 13 heures) x 200 koctets = 30.77 koctets.
 Volume sur tranche THO2_1 = (9 heures / 13 heures) x 200 koctets = 138.46 koctets.
 Volume sur tranche THO2_2 = (1 heure / 13 heures) x 200 koctets = 15.38 koctets.

Le volume sur chaque tranche horaire est ensuite cumulé par tranche tarifaire :

Volume sur tranche tarifaire 1 (Tarif plein) = 0 koctets.

Volume sur tranche tarifaire 2 (40%) = volume THO1_3 + volume THO2_2 = 30.76.

Volume sur tranche tarifaire 3 (60%) = volume THO1_4 + volume THO2_1 = 169.23.

En Archjava :

```
public component class Usage_Traitement
{
    public port pes
        { requires String[] getFile(String nom);
          requires void saveTable(String[] tab,String nom); }
    public port pins
        { requires void Inserer(String s) ; }
    public port pcon
        { requires void Connecter() ; }
    public port psel
        { requires String[][] Selection() ; }
    public port pth
        { requires int[][] Select_TH() ; }
    public port Palarm
        { requiers void put(Alarm a) ; }
    public port plance
        { provides void execute() ; }
    public void Traitement()
    { }
}
```

V.3.2.2. Sous composant «Réconciliation» :

Cette phase stocke les valeurs obtenues par la valorisation et les tickets de taxation proprement dite. Avant la phase de stockage, les tickets sont analysés pour déterminer s'ils représentent une communication complète. Si ce n'est pas le cas ils sont réconciliés avec les tickets déjà stockés.

Le ticket réconciliable est réconcilié avec son homologue dans la table qui stock les tichet_ok les champs du TT de la table ne varient pas sauf :

- Le champ durée est sommé avec le champ «Durée du ticket » du ticket.
- Le champ volume échangé est sommé avec le champ «Volume échangé » du ticket.
- les champs TIC_VOL1, TIC_VOL2, TIC_VOL3, TIC_COUT1, TIC_COUT2, TIC_COUT3, TIC_COUT_DURR, TIC_COUT_DURD, TIC_COUT_VOL, TIC_VOL_REL, TIC_DUR1, TIC_DUR2 et TIC_DUR3 sont cumulés avec les valeurs obtenues par le sous système «calculs des coûts».
- Si le champ «Nature du ticket » a pour valeur «TDS », alors le champ nature ticket de la base reste à « TDS ».
- Si le champ «Nature du ticket » a pour valeur «TUS », alors le champ nature ticket passe à « TKR ».

Le champ **CG_06** «*Nature du ticket* » détermine si la communication représentée par le TT est complète, Quelle que soit la valeur du champ **CG_06**, il faut analyser le TT.

```
public component class réconciliation
{
    public port prec
        { requires String [][] gettickets ();}
    public void Traitement()
    { }
}
```

Les ports :

Le port requiers «Pins » :

avec ce port le composant envoie les requête d'insertion vers le composant «BDD», ce dernier exécute la requête.

Le port requiers «Pth » :

demande au composant 'BDD' de charger la table contenue les tranches horaires et les réductions sur chaque tranche horaire.

Le port requiers «Pcon » :

demande au composant 'BDD' d'établir la connexion de la base de données.

Le port requiers «Psel » :

demande au composant 'BDD' de fournir les informations concernant les abonnés et les clients (Numéro accès X25, débit, Réduction durée de l'abonnement, Réduction durée de client, Réduction volume de l'abonnement, Réduction volume de client, Type de tarification de l'abonnement.).

Le port requiers «Prec » :

avec ce port le composant envoie les requête d'insertion des ticket réconcilie vers le composant «BDD», ce dernier exécute la requête.

Le port requiers «Palrme » :

insertion des alarmes.

Le port requiers «Pes » :

traite les opérations d'entrée sortie.

Le port requiers «Plance» :

permet de lancer le composant.

V.3.3. Sous composant «Alarme» :

L'architecture globale de sous composant «Alarme» :

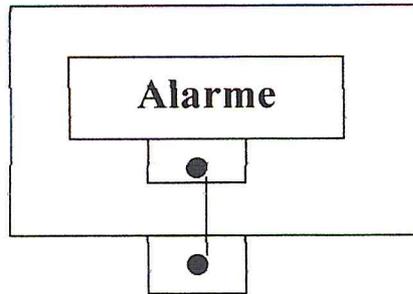


Figure V. 5 Sous Composant Alarme

Gestion des Alarmes :

Les alarmes sont générées (stockées) 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.
- Rejet des TT.

V.3.4. Sous composant «coûts des services» :

L'architecture globale de sous composant «coûts des services» :

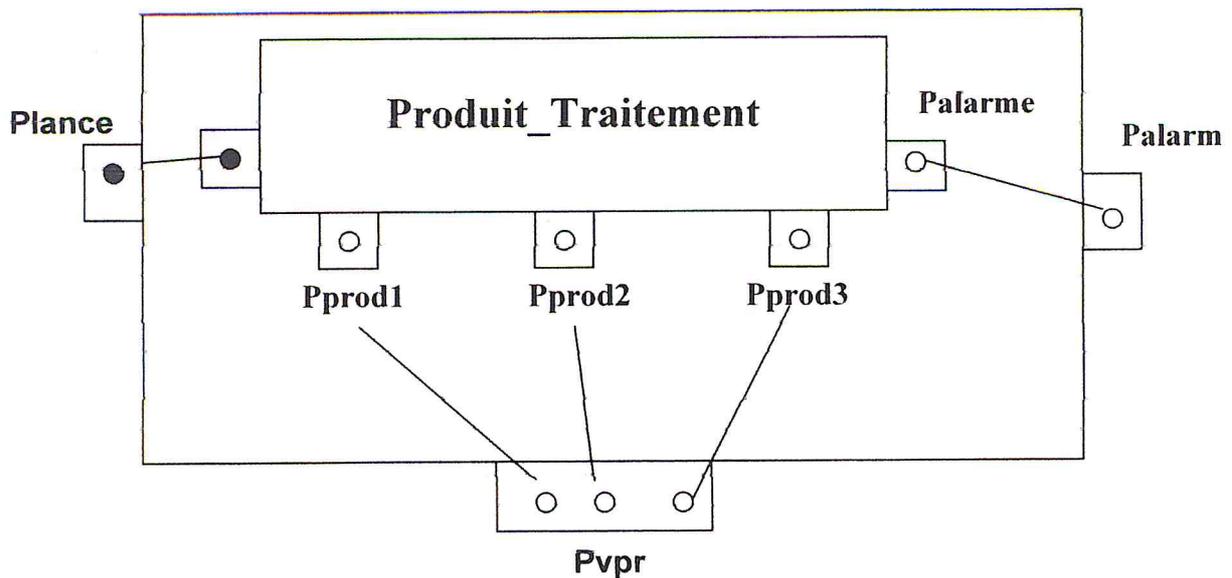


Figure V. 6 L'architecture globale de sous composant «coûts des services»

But :

L'opération coûts des services est l'étape précédant la facturation. Elle est indépendante de sous système «calcule des coûts» et donc des tickets de taxation. Elle prend en compte les caractéristiques des accès du client et des abonnés. 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.

En Archjava :

```
public component class valorisation_produit
{
private final Produit_Traitement pr = new Produit_Traitement ();
public port pvpr
    {requires void inserer_produit(String Req);
    requires String [][] Charger_info_produit();
    requires String date_exploite(); }
public port Palarm
    { requiers void put(Alarm a) ; }
public port Plance
    { provides void execute() ; }
glue Palarm to pr.Palarm ;
glue Plance to pr.Plance;
glue pvpr to pr.pprod1,pr.pprod2,pr.pprod3;

// Les procédures de traitements.
public void execute()
    { }
}
```

V.3.4.1. Sous composant «Produit Traitement» :

son rôle est :

1. Analyse des données d'abonnement.
2. Valorisation des mises en service associées au client.
3. Valorisation des modifications d'abonnement associées au client (Abonné).
4. Valorisation des abonnements associés au client.
5. Calcul des valeurs en fonction des données clients :
 - Facturation au forfait,
 - Réductions,
 - Location de modem, vitesse de la ligne, facture détaillée, services complémentaires...

A partir des paramètres suivants :

- Périodicité de la facturation.
- Parité de la périodicité.
- Jour de facturation.
- Etat du client.
- Date de dernière facture.

Et les caractéristiques du client La date de lancement du calcul de l'opération « coûts des services » est déterminer.

Le sous composant «Produit Traitement» permet de calculer :

➤ **Nombre de mois de la redevance d'un abonnement**

➤ **L'accès d'un abonnement**

(a) Frais de mise en service des accès :

- ⇒ Réduction abonnement sur les frais de mise en service = $(100 - \text{réduction sur les frais de mise en service}) / 100$ pour l'abonnement.
- ⇒ Réduction client sur les frais de mise en service = $(100 - \text{réduction sur les frais de mise en service}) / 100$ pour le client.
- ⇒ Frais de mise en service .

(b) Frais d'évolution des accès :

- ⇒ Réduction abonnement sur les frais d'évolution = $(100 - \text{réduction sur les frais d'évolution}) / 100$ pour l'abonnement
- ⇒ Réduction client sur les frais d'évolution = $(100 - \text{réduction sur les frais d'évolution}) / 100$ pour le client
- ⇒ Frais d'évolution.

➤ **L'abonnement**

(a) Redevance des accès

- ⇒ Réduction abonnement sur la redevance = $(100 - \text{réduction sur la redevance}) / 100$ pour l'abonnement.
- ⇒ Réduction client sur la redevance = $(100 - \text{réduction sur la redevance}) / 100$ pour le client
- ⇒ Redevance des accès.

(b) Redevance des services complémentaires

- ⇒ Redevance des services GFA
- ⇒ Redevance des services VL
- ⇒ Redevance des autres services complémentaires
 - Frais de la facture détaillée
 - Frais de la facture mensuelle

Les ports :

Le port requiers «Pprod1 » :

demande au composant 'BDD' d'insérer les résultats calculer par le composant 'Produit_Traitement'.

Le port requiers «Pprod2 » :

Demande au composant 'BDD' de fournis la table contenue les informations concernant les clients et les abonnés(type de tarification, périodicité, état.....).

Le port requiers «Pprod3 » :

Demande au 'BDD' de fournir la date d'exploitation.

Le port requiers «Palrme » :

Insertion des alarmes.

Le port requiers «Plance » :

Lance le composant.

En Archjava :

```
public component class Produit
{
    public port pprod1
        { requires void inserer_produit(String Req); }
    public port pprod2
        { requires String [][] Charger_info_produit(); }
    public port pprod3
        { requires String date_exploite(); }
    public port Palarm
        { requiers void put(Alarm a) ; }
    public port Plance
        { provides void execute() ; }
    public void Traitement()
    { }
}
```

V.3.5. Sous composant «base de données» :

L'architecture globale de sous composant «base de données » :

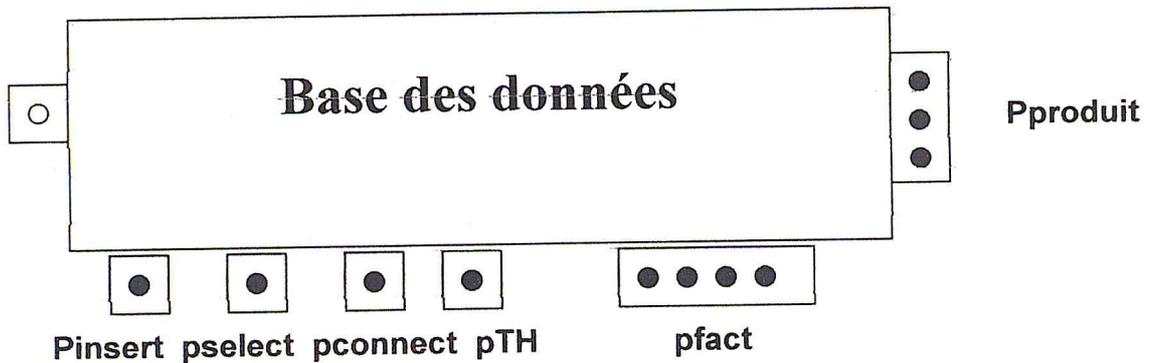


Figure V. 7 L'architecture globale de sous composant «base de données »

Le composant «base de données » a pour rôle de faire tous les traitements (selection, insertion, suppression) sur les tables de la base de donnée et fourni les résultats d'exécution des requêtes demander par les autres composants.

La base des données est construite selon le **modèle objet – relationnelle** à partir de diagramme de classe UML et le passage de diagramme de classe UML vers les tables de la base de donnée et faite selon le **modèle navigationnel. (voir annexe).**

Remarque :

Le SGBD utilisé est ORACLE 8.

En Archjava :

```
public component class BDD
{
public port pinsert
    { provides void Inserer(String Req) ; }
public port pselect
    { provides String [][] Selection() ; }
public port pTH
    { provides int [][] Select_TH() ; }
public port pconnect
    { provides void Connecter() ; }
public port pproduit
    {provides void inserer_produit(String Req) ;
    provides String [][] Charger_info_produit();
    provides String date_exploite(); }
public port pfact
    { provides void Fact_usage();
    provides void Changement_Volume();
    provides void Calcule();
```

```
        provides void Fact_val_produit(); }
// Les procédures de traitements.
public void Connecter()
{ // exécution des requêtes de connexion }
public void Insérer(String Req)
{ // exécution des requêtes d'insertion. }

public String[][] Selection()
{ // exécution des requêtes de selection. }
public int[][] Select_TH()
{ // exécution des requêtes de selection de la table de tranche horaire. }
public void inserer_produit(String Req)
{ // exécution des requêtes d'insertion. }
public String[][] Charger_info_produit()
{ }
public String date_exploite()
{ }
}
```

Les ports :

Le port provides «pinsert » :

Permet l'exécution des instructions SQL d'insertion.

Le port provides «pselect » :

Permet la sélection des informations demander par les autres composants.

Le port provides «pconnect » :

Permet l'établissement de la connexion vers la base des données.

Le port provides «pTH » :

Fournit la table tranche horaire.

Le port provides «pproduit » :

Exécute les requêtes demander par le composant «coûts des services».

Le port provides «pfact » :

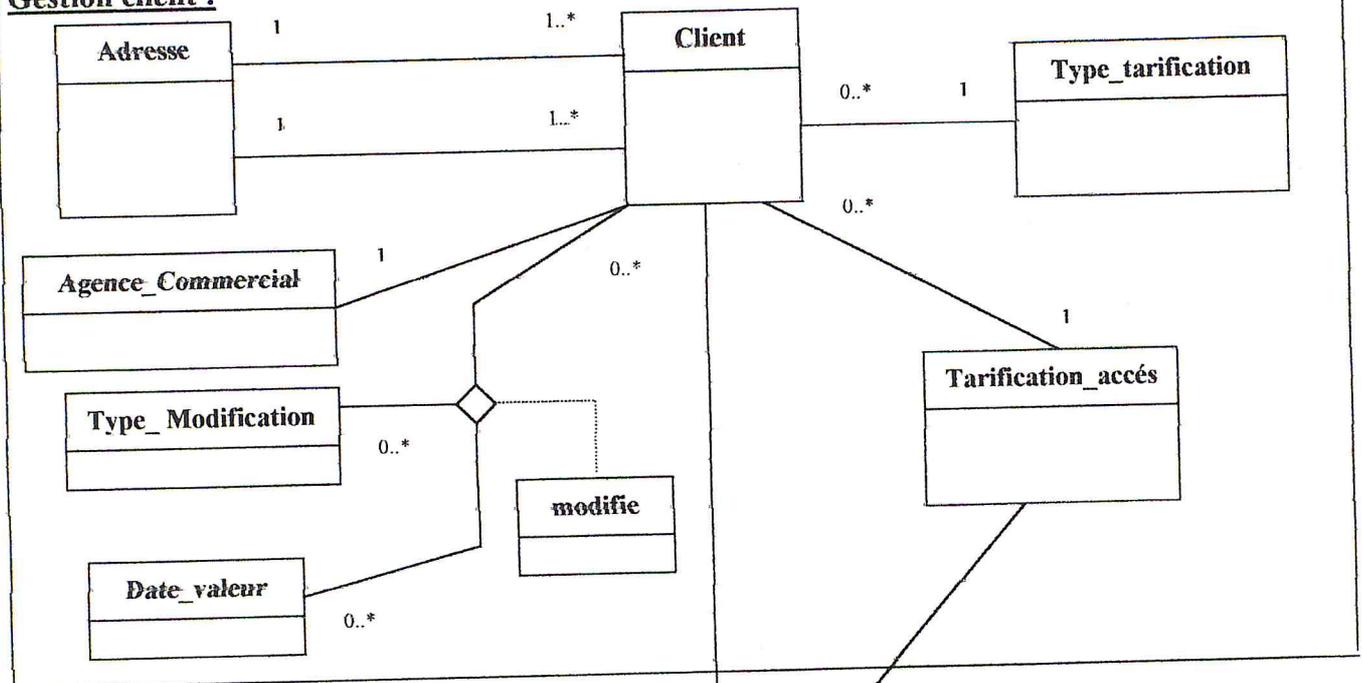
Exécute les requêtes demander par le composant «Facturation».

Conception de la base des données :

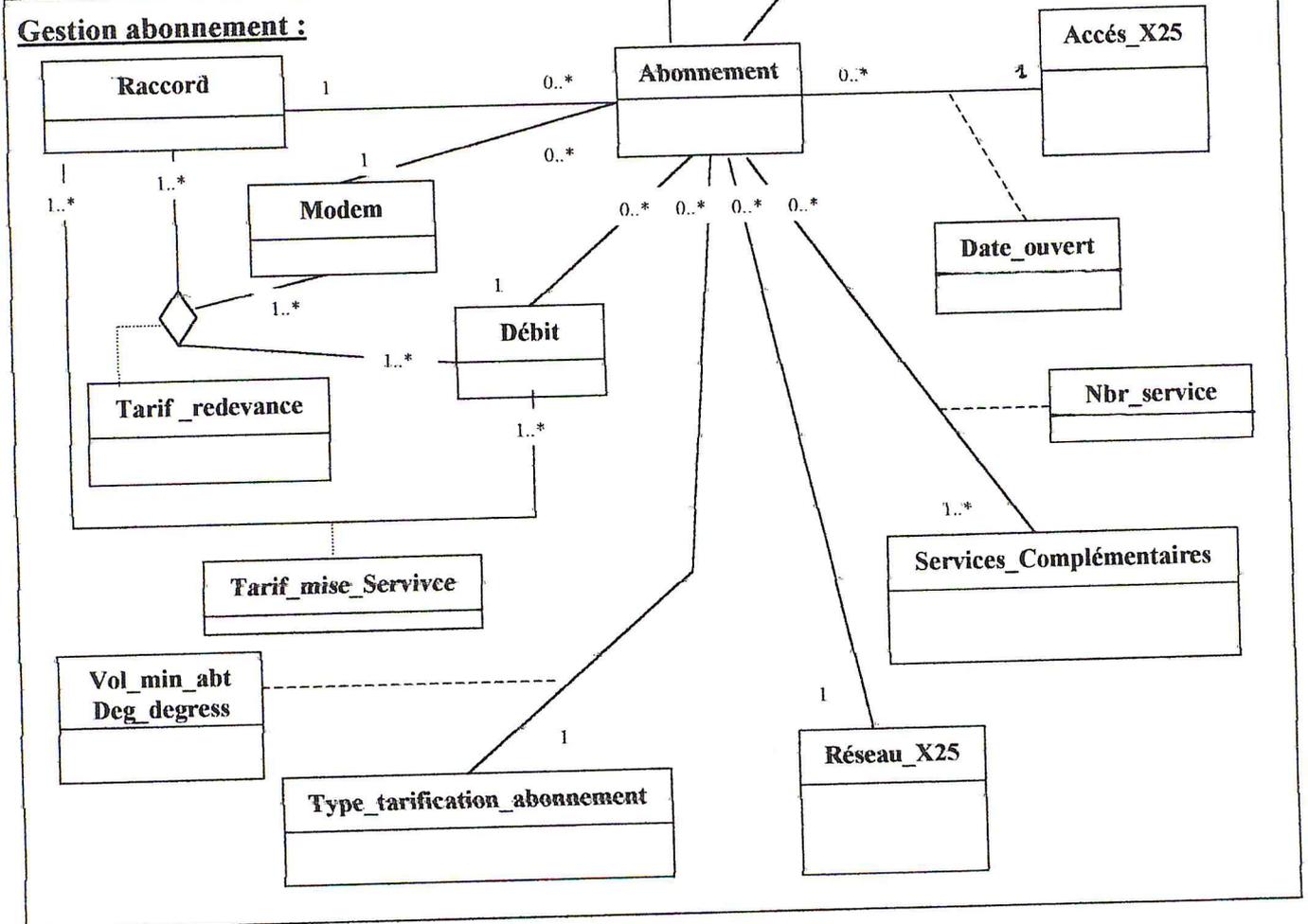
Le diagramme de classe UML de la base de données :

Conception de Système selon l'approche Composant

Gestion client :



Gestion abonnement :



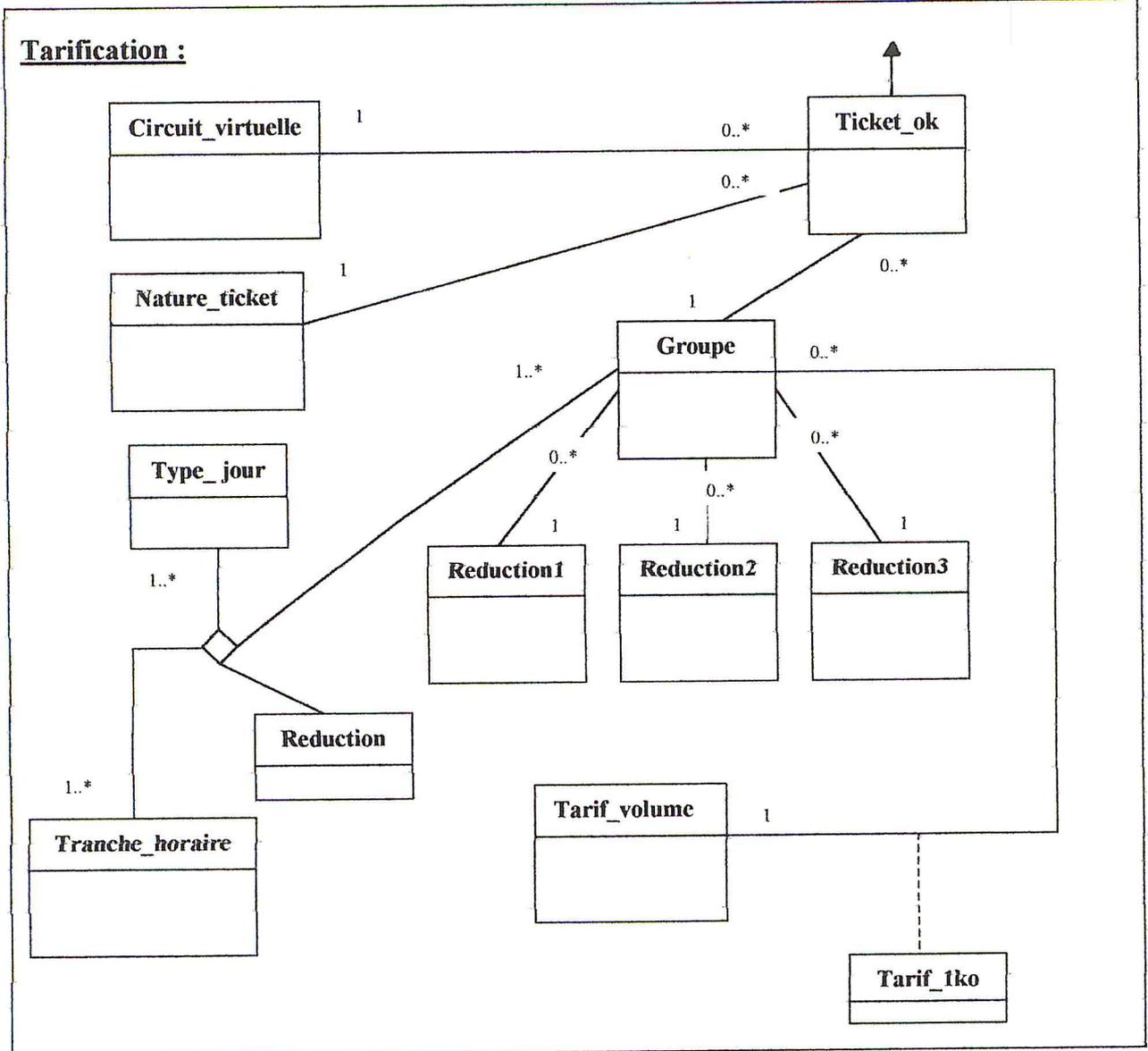
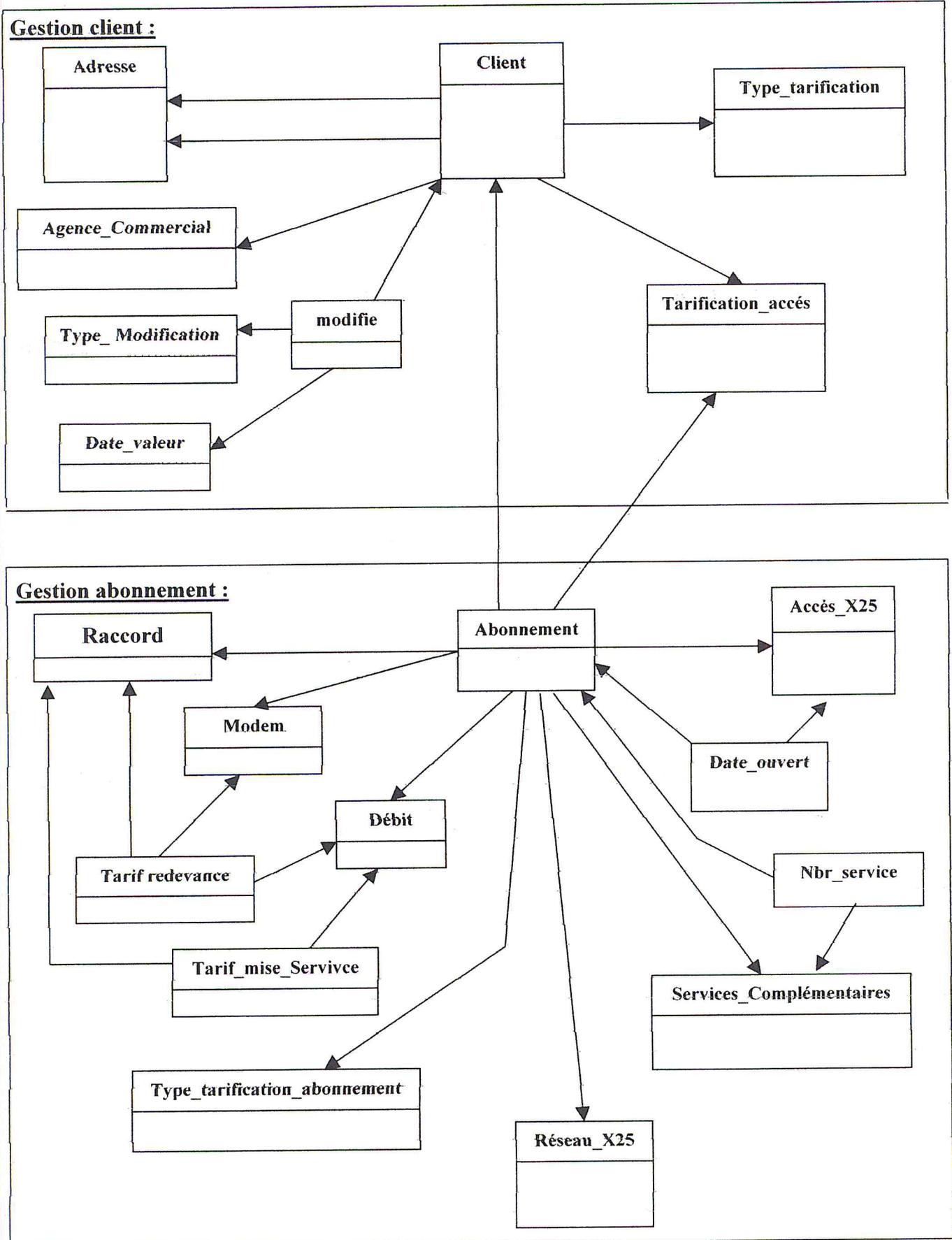


Figure V.8 Diagramme de classe

Le modèle navigationnel :



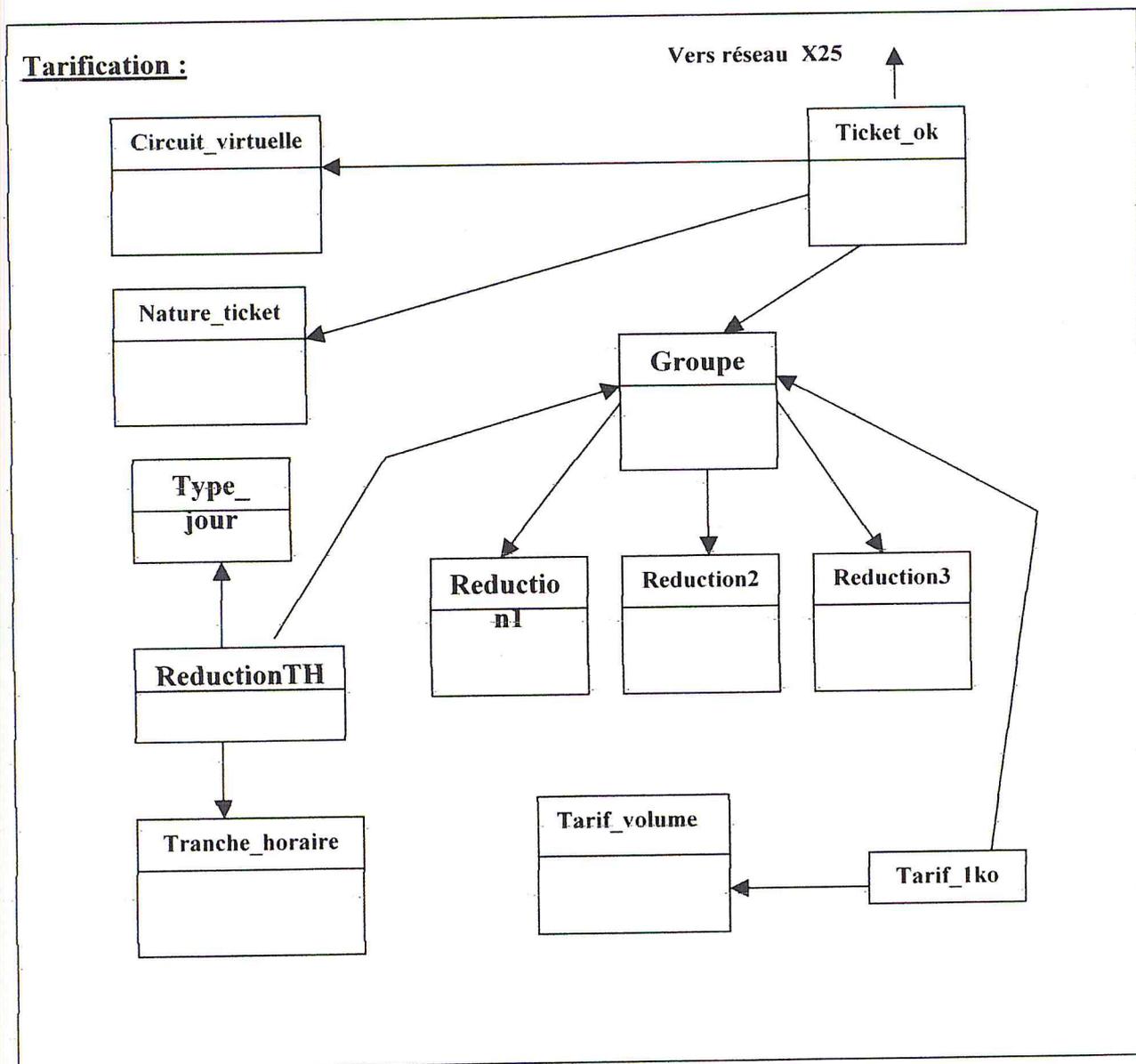


Figure V. 9 Le modèle navigationnel

Description des tables objet – relationnelle de la base des données :

Abonnement :

Libellé	Code	Type
Numéro de l'abonnement	Abo_num	Numérique
Date de saisie	Abo_date_sai	Date
Date d'ouverture souhaitée	Abo_date_ouv	Date
Date d'ouverture d'accès X25	Abo_date_ouv_accX25	Date
Date de résiliation abonnement	Abo_date_resl_accX25	Date
Référence client	Abo_client	Pointeur vers Client
Référence accès X25	Abo_num_accx25	Pointeur vers accès X25
Référence réseau X25	Abo_reseau	Pointeur vers réseau X25
Référence type de tarification accès	Abo_type_tarif_acc	Pointeur vers tarification accès
Référence type de tarification	Abo_type_tarif	Pointeur vers type de tarification

Référence débit	Abo_debit	Pointeur vers débit
Référence modem	Abo_modem	Pointeur vers modem
Référence raccordement	Abo_raccord	Pointeur vers raccordement
Référence services complémentaires	Abo_svc_compl	Tables des Pointeurs services complémentaires

Tableau V. 8 Table Abonnement

Client :

Libellé	Code	Type
Numéro du client	Clt_num	Numérique
Nom	Clt_nom	Caractère
Prénom	Clt_prenom	Caractère
Civilité	Clt_civilité	Caractère
Raison Sociale	Clt_raison	Caractère
Type de client	Clt_type	Caractère
Numéro fiscal	Clt_fis_num	Caractère
Complément d'information	Clt_info	Caractère
Date de création	Clt_date_cree	Date
Date d'inactivation	Clt_date_supp	Date
Etat du client	Clt_etat	Caractère
Facturable (O/N)	Clt_fact	Numérique
Périodicité de la facturation	Clt_per_fact	Caractère
Parité de la périodicité	Clt_parite	Caractère
Date de la précédente facture	Clt_prec_fact	Date
Jour de facturation	Clt_jour_fact	Numérique
Langue de facturation	Clt_lang_fact	Caractère
Détail de la facture (O/N)	Clt_det_fact	Caractère
Nombre de GFA	Clt_nbr_gfa	Numérique
Nombre de création de GFA	Clt_nbr_cree_gfa	Numérique
Nombre de suppression de GFA	Clt_nbr_supp_gfa	Numérique
TVA Applicable (O/N)	Clt_tva_app	Caractère
Référence adresse commerciale	Clt_adr_num_cial	Pointeur vers adresse
Référence adresse de facturation	Clt_adr_num_fact	Pointeur vers adresse
Référence type tarif	Clt_type_tarif	Pointeur vers type tarification
Référence type tarif accès	Clt_tarif_acc	Pointeur vers type tarification accès
Référence agence	Clt_pay_agence	Pointeur vers agence

Tableau V. 9 Table client

Domaine de validité de certaines données :

Les champs suivants n'admettent que des valeurs qui sont fixées par l'application elle-même :

Type de client : Le type du client peut être une Société ou un Particulier.

Civilité : Si le client est un Particulier, il est possible de définir pour lui un titre de civilité. Ce titre peut être : Monsieur (par défaut), Madame ou Mademoiselle.

Etat du client : Un client peut être Actif ou Inactif.

Client facturable : La valeur peut être Oui ou Non.

Périodicité de la facturation : Elle peut être Mensuelle ou Bimestrielle

Parité de la périodicité : Si la périodicité est Bimestrielle, il faut définir le type de mois correspondant à l'envoi d'une facture. Ce type peut être Pair ou Impair. Ainsi un client sera facturable tous les mois pairs ou tous les mois impairs.

Jour de la facturation : Les seuls jours acceptables sont les entiers compris entre 1 et 28.

Langue de facturation : Ce champ définit si la facture est éditée en Français ou en Arabe.

Type de facture : Les factures peuvent être Globales ou Détaillées.

TVA applicable : Ce champ indique si la TVA est applicable : Oui ou Non.

Ticket ok :

Libellé	Code	Type
Numéro ticket	Tic_num	Numérique
Référence réseau	Tic_res_cod	Pointeur vers réseau X25
Référence nature tic	Tic_nat_cod	Pointeur vers nature ticket
Référence CV	Tic_cv_cod	Pointeur vers circuit virtuel
Référence groupe	Tic_grp_cod	Pointeur vers groupe pays
Date heure début TT	Tic_date_deb_tt	DATE
Date heure début com	Tic_heure_deb_tt	caractère
Durée tic	Tic_duree	Numérique
Imputation de taxe	Tic_imp_taxe	Numérique
Adr imputation de taxe	Tic_adr_taxe	Caractère
Adr appelant complète	Tic_adr_applant	Caractère
Adr appelé complète	Tic_adr_apellee	Caractère
Nbre Ko transmis reçus	Tic_vol_echange	Numérique
Vol sur tranche tarifaire 1	Tic_vol1	Numérique
Vol sur tranche tarifaire 2	Tic_vol2	Numérique
Vol sur tranche tarifaire 3	Tic_vol3	Numérique
Coût tranche 1	Tic_cout1	Numérique
Coût tranche 2	Tic_cout2	Numérique
Coût tranche 3	Tic_cout3	Numérique
Coût durée réel	Tic_coût_dur_reel	Numérique
Coût durée détail	Tic_coût_dur_detail	Numérique
Coût volume détail	Tic_coût_vol	Numérique
Durée tranche 1	Tic_dur1	Numérique
Durée tranche 2	Tic_dur2	Numérique
Durée tranche 3	Tic_dur3	Numérique
Tarif appliqué tranche 1	Tic_tar1	Numérique
Tarif appliqué tranche 2	Tic_tar2	Numérique
Tarif appliqué tranche 3	Tic_tar3	Numérique
Type tarif appliqué	Tic_type	Caractère

Tableau V. 10 Table Ticket_OK

Accès X25 :

Libellé	Code	Type
Numéro X25	Acc_num	Caractère
Disponibilité	Acc_dispo	Numérique

Tableau V. 11 Table accès X25

type jour :

Libellé	Code	Type
Code du type de jour	Tjo_cod	Caractère
Libellé du type de jour	Tjo_lib	Caractère

Tableau V. 12 Table type de jour

Domaine de validité :

Tjo_cod	Tjo_lib
1	Jours ouvrables
2	Jours ouvrés
3	Jours fériés

Tableau V. 13 domaine de validité de la table type de jour

Réduction1, Réduction2, Réduction3 :

Libellé	Code	Type
Code réduction	Red_cod	Caractère
Libellé réduction	Red_lib	Caractère
Valeur réduction	Red_val	Numérique

Tableau V. 14 table Réduction

La valeur de la réduction est un pourcentage : 0% pour aucune réduction et 100% pour totalement gratuit.

Les réductions :

- Réduction normale,
- Aucune réduction
- Totalement gratuite.

Réseau X25 :

Libellé	Code	Type
Code réseau	Res_cod	Caractère
Libellé réseau	Res_lib	Caractère

Tableau V. 15 table réseau X25

Domaine de validité :

RES_COD	RES_LIB
S	Réseau Constructeur1
A	Réseau Constructeur2

Tableau V. 16 Domaine de validité de la table réseau X25

Nature ticket :

Libellé	Code	Type
Code nature tic	Nat_cod	Caractère
Libellé nature tic	Nat_lib	Caractère

Tableau V. 17 table Nature ticket

Domaine de validité :

Nat_cod	Nat_lib
TUA	Ticket Unique Constructeur2
TDA	Ticket Début Constructeur2
TSA	Ticket Suite Constructeur2
TFA	Ticket Fin Constructeur2
TUS	Ticket Unique ou fin Constructeur1
TDS	Ticket suite ou Début Constructeur1
TKR	Ticket réconcilié

Tableau V. 18 Domaine de validité de la table nature ticket

Circuit virtuel :

Libellé	Code	Type
Code CV	CV_cod	Numérique
Libellé CV	CV_lib	Caractère

Tableau V. 19 table circuit virtuel

Domaine de validité :

CV_cod	CV_lib
1	CVC (Circuit virtuel commuté)
2	CVP (Circuit virtuel permanent)
3	CVC secours
4	CVC passerelle

Tableau V. 20 domaine de validité de la table circuit virtuel

Débit :

Libellé	Code	Type
Code débit	Deb_cod	Caractère
Valeur débit maximum	Deb_max	Caractère

Tableau V. 21 table Débit

Domaine de validité :

Deb_cod	Deb_max
D1	1200 bits/s
D2	2400 bits/s
.....
D37	31 fois 64 kbits/s

Tableau V. 1 domaine de validité de la table débit

Modem :

Libellé	Code	Type
Code modem	Mod_cod	Numérique
Libellé modem	Mod_lib	Caractère

Tableau V. 2 table modem

Domaine de validité :

Mod_cod	Mod_lib
0	Sans Modem
1	Avec Modem

Tableau V. 3 domaine de validité de modem

Raccordement :

Libellé	Code	Type
Code Type raccord	Raccord_cod	Caractère
Libellé type raccord	Raccord_lib	Caractère

Tableau V. 4 table raccordement

Domaine de validité :

Raccord_cod	Raccord_lib
R1	Accès Direct
R2	Accès indirect par ligne banalisée
R3	Accès indirect par ligne réservée
R4	Accès indirect par telexe

Tableau V. 5 domaine de validité de la table raccordement

Tarification accès :

Libellé	Code	Type
Code tarification accès	Tar_acc_cod	Numérique
libellé tarification accès	Tar_acc_lib	Caractère
Réduction tarif de redevance	Tar_acc_red_redv	Numérique
Réduction tarif de mise en service	Tar_acc_red_mis_svc	Numérique
Tarif frais d'évolution	Tar_acc_frai_evolt	Numérique

Tableau V. 6 Table tarification accès

Les tarifications :

- Tarification normale
- Réduction 10%
- Moitié Prix

Groupe pays :

Libellé	Code	Type
Code groupe	Grp_cod	Numérique
Libellé groupe	Grp_lib	Caractère
Référence réduction 1	Red1	Pointeur
Référence réduction 2	Red2	Pointeur
Référence réduction 3	Red3	Pointeur

Tableau V. 28 table groupe pays

Tranche horaire :

Libellé	Code	Type
Code tranche horaire	Tranch_cod	Numérique
Heure de debut	Tranch_hd	Numérique
Heure de fin	Tranch_hf	Numérique

Tableau V. 29 table tranche horaire

NB : Les heures sont converties en minute (1 H = 60 min).

Réduction TH :

Libellé	Code	Type
Réduction	red	Numérique
Référence tranche horaire	Ref_th	Pointeur
Référence type de jour	Ref_tjo	Pointeur
Référence groupe	Ref_grp	Pointeur

Tableau V. 30 table réduction tranche horaire

Services complémentaires :

Libellé	Code	Type
Code services complémentaires	Svc_cod	Caractère
Libellé services complémentaires	Svc_lib	Caractère
Tarif services complémentaires	Svc_tarif	Numérique
Type services complémentaires	Svc_type	Numérique

Tableau V. 31 table services complémentaires

Les services complémentaires :

- Voies GFA
- VL Supplémentaires
- Services complémentaires type 1
- Services complémentaires type 2
- Services complémentaires type 3
- Services complémentaires type 4
- Longueurs de packets 64 Octets
- Longueurs de packets 128 Octets
- Longueurs de packets 256 Octets
- Longueurs de packets 512 Octets
- Numéro Abrégé
- Acceptation des appels en PVC

Tarification abonnement :

Libellé	Code	Type
Code de tarification	Abn_cod	Numérique
Libellé de tarification	Abn_lib	Caractère
Réduction durée	Abn_red_dur	Numérique
Réduction volume	Abn_red_vol	Numérique
Réduction forfait complet	Abn_red_for_fc	Numérique
Réduction forfait durée	Abn_red_for_dr	Numérique
Type de tarification	Abn_type_tarif	Numérique

Tableau V. 32 table tarification abonnement

NB : La valeur de la réduction est un pourcentage : 0% pour aucune réduction et 100% pour totalement gratuit.

Les tarifications abonnement :

- Gratuit.
- Tarif Complet Normal.
- Durée Moitié Prix.
- Tarification Réelle Normale.
- Tarification Complet Normale.
- Tarif Durée Normale.
- Complet Moitié Prix.
- Réel Moitié Prix.

Les types de tarifications :

- Coût Réel.
- Forfait Complet.
- Forfait a la durée.
- Coût Réel.
- Forfait Complet.
- Forfait a la durée.
- Forfait Complet.
- Coût Réel.

Tarification client :

Libellé	Code	Type
Code tarification	Cod	Numérique
Libellé tarification	Lib	Caractère
Réduction forfait complet	Red_fcom	Numérique
Réduction forfait durée	Red_fdur	Numérique
Réduction volume	Red_vol	Numérique
Réduction durée	Red_dur	Numérique

Tableau V. 33 table tarification client

NB. : La valeur de la réduction est un pourcentage.

V.3.6 Sous composant «base de Fichiers» :

L'architecture globale de sous composant «base de Fichiers» :

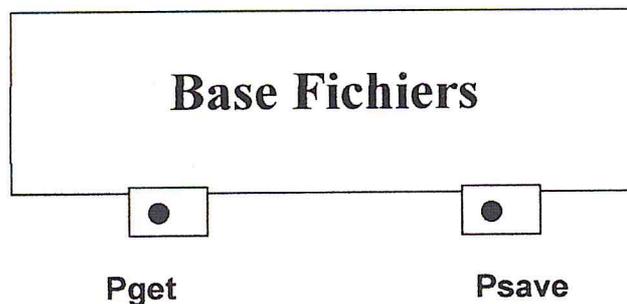


Figure V. 10 L'architecture globale de sous composant «base de Fichiers»

Le composant «base fichiers » assure toutes les opérations d'entré sortie et la manipulation des différents fichiers.

En Archjava :

```
public component class Fichiers
{
    public port psave
        { provides void saveTable(String[] tab,String nom);    }
    public port pget
        { provides String[] getFile(String nom);    }
    // Les procédures de traitement.
    public String[] getFile(String nom)
    { // retourner un tableau a partir d'un fichier }
    public void saveTable(String[] tab,String nom)
    { // save table dans un fichier }
}
```

Définition de la base de Fichiers :

La base de fichiers contient les fichiers suivants :

- TT_Constructeur : contient les tickets Constructeur.
- TT_Correct : contient que les tickets Constructeur1 Correct.
- TT_Rejet : contient les tickets rejeter
- Fichier_Decode : contient les tickets décoder.
- Fichier_Harmonise : contient les tickets harmoniser.
- Journal_Constructeur1 : contient les informations sur le nombre des fichiers traiter et rejeter.
- Param_Constructeur1 : contient les paramètres nécessaires de réseau Constructeur1.
- Gram_Constructeur1 : contient la grammaire de l'opération d'harmonisation.
- Jour_ferie : contient les dates et les jours férie (1^{er} Novembre, 5 Juillet.....)

Les autres Fichiers :

Journal_Constructeur , Param_Constructeur, Gram_Constructeur, Jour_ferie sont au format Texte.

Les ports :

Le port provides « Pget » : récupère le fichier correspondant

Le port provides « Psave » : save le fichiers donnée

V.3.7. Sous composant «Facturation» :

L'architecture globale de sous composant «Facturation» :

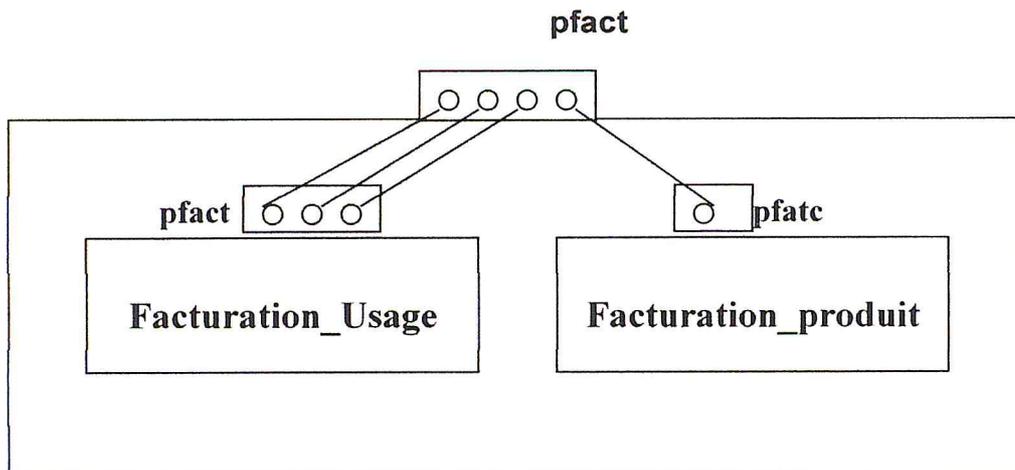


Figure V. 11 L'architecture globale de sous composant «Facturation»

Le calcul de la facturation se base sur :

- Le résultat de sous système «coûts des services » (valorisation des accès),
- Le résultat de sous système «calcule des coûts » (valorisation des communications),
- Les remises de dégressivité éventuellement accordées au client.

En Archjava :

```
public component class Facturation
{
    private final Facturation_produit Fvp = new Facturation_produit();
    private final Facturation_usage Fvu = new Facturation_usage();
    public port pfact
    {
        requires void Fact_usage();
        requires void Changement_Volume();
        requires void Calcule();
        requires void Fact_val_produit(); }
    glue pfact to Fvp.pfact,Fvu.pfact;
    public void Traitement()
    {}
}
```

V.3.7.1. Sous composant «Facturation usage» :

c'est une opération de facturation liée au sous système "calculs des coûts " elle permet de calculer les facteurs suivants :

- 1. Valorisation des communications internationales non abouties**
- 2. Contrôle du volume des communications pour un accès**
- 3. Nombre de communications nationales pour un accès**
- 4. Nombre de communications internationales pour un accès**
- 5. Volume des communications nationales sur chaque tranche tarifaire pour un accès**
 - ✓ Volume en Ko des communications nationales pour la tranche tarifaire 1 pour un accès
 - ✓ Volume en Ko des communications nationales pour la tranche tarifaire 2 pour un accès
 - ✓ Volume en Ko des communications nationales pour la tranche tarifaire 3 pour un accès
- 6. Volume des communications internationales sur chaque tranche tarifaire pour un accès**
 - ✓ Volume en Koctets des communications internationales pour la tranche tarifaire 1 pour un accès
 - ✓ Volume en Koctets des communications internationales pour la tranche tarifaire 2 pour un accès
 - ✓ Volume en Koctets des communications internationales pour la tranche tarifaire 3 pour un accès
- 7. Coût des communications nationales par tranche tarifaire et par accès**
 - ✓ Coût des communications nationales pour la tranche tarifaire 1 et par accès.
 - ✓ Coût des communications nationales pour la tranche tarifaire 2 et par accès
 - ✓ Coût des communications nationales pour la tranche tarifaire 3 et par accès
- 8. Durée totale de communications nationales pour un accès**
- 9. Coût pour la durée des communications nationales pour un accès**
- 10. Nombre de communications internationales non abouties pour un accès**
- 11. Coût des communications internationales par tranche tarifaire et par accès**
 - ✓ Coût des communications internationales pour la tranche tarifaire 1 et par accès
 - ✓ Coût des communications internationales pour la tranche tarifaire 2 et par accès.

- ✓ Coût des communications internationales pour la tranche tarifaire 3 et par accès

12. Coût total pour la durée des communications internationales pour un accès

13. Coût total forfait complet

- ✓ Réduction client forfait complet
- ✓ Réduction abonnement forfait complet
- ✓ Coût total forfait complet

14. Coût total forfait à la durée

- ✓ Réduction client forfait durée
- ✓ Réduction abonnement forfait durée.
- ✓ Coût total forfait à la durée

15. Valeur de remise dégressivité au volume

- ✓ Volume total des communications hors forfait complet
- ✓ Valeur de remise dégressivité au volume

16. Coût total des communications réelles nationales

17. Coût total des communications internationales

Codification Proposé au niveau BDD:

Abréviation	Intitulé
DE	Date d'exploitation
PC	Période de calcul de la valorisation de produit
NMR	Nombre de mois de redevance
FMES	Frais de mise en service
RAFMES	Réduction abonnement sur les frais de mise en service
RCFMES	Réduction client sur les frais de mise en service
FE	Frais d'évolution
RAFE	Réduction abonnement sur les frais d'évolution
RCFE	Réduction client sur les frais d'évolution
RA	Redevance des accès
RAR	Réduction abonnement sur la redevance
RCR	Réduction client sur la redevance
RGFA	Redevance des services GFA
RVL	Redevance des services VL
RSC	Redevance des autres services complémentaires
FTMES	Frais total de mise en service des accès
FTE	Frais total d'évolution des accès
RTA	Redevance total des accès
RTSC	Redevance totale des services complémentaires
NCN	Nombre de communications nationales pour un accès
NCI	Nombre de communications internationales pour un accès
VCN1	Volume des communications nationales sur tranche tarifaire 1
VCN2	Volume des communications nationales sur tranche tarifaire 2
VCN3	Volume des communications nationales sur tranche tarifaire 3
VCI1	Volume des communications internationales sur tranche tarifaire 1
VCI2	Volume des communications internationales sur tranche tarifaire 2
VCI3	Volume des communications internationales sur tranche tarifaire 3

Conception de Système selon l'approche Composant

CCN1	Coût des communications nationales pour la tranche tarifaire 1
CCN2	Coût des communications nationales pour la tranche tarifaire 2
CCN3	Coût des communications nationales pour la tranche tarifaire 3
DCN	Durée des communications nationales
CDCN	Coût durée des communications nationales
VRDV	Valeur de remise dégressivité au volume
CCI1	Coût des communications internationales pour la tranche tarifaire 1
CCI2	Coût des communications internationales pour la tranche tarifaire 2
CCI3	Coût des communications internationales pour la tranche tarifaire 3
CDCI	Coût durée des communications internationales
CTDCI	Coût total pour la durée des communications internationales pour un accès
FFD	Frais de la facture détaillée
FFM	Frais de la facture mensuelle
RCFC	Réduction client forfait complet
RAFC	Réduction abonnement forfait complet
CTFC	Coût total forfait complet
RCFD	Réduction client forfait durée
RAFD	Réduction abonnement forfait durée
CTFD	Coût total forfait durée
CTCRN	Coût total des communications réelles nationales
RGGFA	Redevance groupe GFA
NCINA	Nombre de communications internationales non aboutis pour un accès
CTCI	Coût total des communications internationales
TC	Total communications
MHT	Montant HT de la facture
MTVA	Montant TVA
MTTC	Montant TTC de la facture
MTTCP	Montant TTC à payer
RPF	Régularisation des précédentes factures
DLR	Date limite de règlement
DPF	Date de précédente facture
NF	Numéro de facture
NC	Numéro client
NTC	Numéro de téléphone commercial
RS	Raison social
CPC	Code postal commercial
NUF	Numéro fiscal
NAC	Nom agence commerciale
AA	Adresse de l'agence
NUMAPP	Numéro appelé
JOUR	Jour communication
TYPE	Type communication
VOL	Volume communication
DUR	Durée communication
COUT	Coût communication

Tableau V. 34 Codification proposé au niveau BDD

En Archjava:

```
public component class Facturation_usage
{
    public port pfact
    { requires void Fact_usage();
      requires void Changement_Volume();
      requires void Calcule(); }
}
```

```
public void Execute()  
{ }  
}
```

V.3.7.2. Sous composant «Facturation produit» :

c'est une opération de facturation liée au sous système "coûts des services" elle permet de calculer les facteurs suivants :

La facturation de liée a l'opération «coûts des services » se base sur l'agrégation des données calculées lors de l'opération «coûts des services ».

- ❑ Frais total de mise en service des accès
- ❑ Frais total d'évolution des accès
- ❑ Redevance totale des accès
- ❑ Redevance totale des services complémentaires
- ❑ Redevance groupe GFA_

En Archjava:

```
public component class Facturation_produit  
{  
    public port pfact  
        { requires void Fact_val_produit(); }  
    public void Execute()  
    {}  
}
```



V.3.8. Sous composant « Acquisition » :

L'architecture globale de sous composant «Acquisition» :



Figure V. 12 L'architecture globale de sous composant «Acquisition»

Le but est de récupérer les fichiers des tickets qui se trouve sur l'un des deux station de supervision.

A l'aide de la classe FTPConnection (télécharger à partir d'Internet) on développe les fonctions suivantes :

```
public boolean Init(String hot,String user,String password) : initialise la connexion avec le serveur.
```

public boolean ListeFichiers (String dossier) : retourne la liste des fichier disponible sur la station de supervision.

public boolean Telecharger (String[] t) : lance le transfert FTP des fichiers.

public boolean SupprimerFichiers (): supprimer le fichier sur les station de supervision.

public void Deconnection() : fermé la connexion.

public Exception getException() : gestion des exception.

Les ports :

Trois ports requires deux pour les opération sur les fichiers et un pour les alarmes.

Un port provides pour le lancement de sous composant.

En Archjava :

```
public component class acquisition
{
public port pes
    { requires String[] getFile(String nom);
      requires void saveTable(String[] tab,String nom); }
public port Palarme
    { requiers void put(Alarm a) ;    }
public port Pliance
    { provide void execute() ; }
// les procédures de traitement
public void Execute()
{ // permet de faire les traitement nécessaire de l'opération complète de «acquisition » }.
```

V.3.9. Sous composant « IHM » :

Ce sous – composant a été développe en JAVA puisque il nécessite l'utilisation des composants graphiques (Fenêtre, Bouton, Forme, champs des textes,.....).

IHM permet de lancer tous les opération précédentes et d'introduire les différents données nécessaire a la base des données .

Insérée, supprimer, modifier les données concernant :

- ✓ Les clients.
- ✓ Les abonnements.
- ✓ Les utilisateurs.
- ✓ Les tranches horaires.
- ✓ Les adresse X25.
- ✓ Le plan tarifaire
- ✓ L'archivage et la purge des données.
- ✓ Lancer l'édition des factures.

Elle permet aussi de faire quelque opérations supplémentaire tel que le suivi de paiement.

Conclusion :

Le développement de ce système à l'aide de Archjava à améliorer la compréhension des programmes, et a garanti l'architecture de l'application, il permet aussi une meilleure évolutivité d ce système.

Le seul composant IHM a été developper en JAVA puisque il nécessite l'utilisation des composants graphiques (Fenêtre, Bouton, Forme, champs des textes,.....).

Chapitre VI
MISE EN ŒUVRE

VI.1. Introduction :

Dans ce chapitre nous allons mettre en œuvre l'objectif de notre conception, qui consiste à la mise en place d'un système de gestion commerciale des produits du réseau X25 développer selon l'approche composant et une application IHM qui permet de manipuler les différents données de la base des données, et en fin le module d'impression des factures éditées.

Le langage de programmation choisi est le **JAVA 1.5** et le langage pour générer les états de sortie pour les factures est le Module Quickreport de **Borland DELPHI 5**.

Les traitement sont réalisés avec **ARCHJAVA**.

Le système de gestion de la base de données est **ORACLE 8**.

VI.2. La chaîne des traitements :

On peut lancer directement la chaîne de tous les traitements séquentiels, comme on peut lancer chaque traitement à part.

VI.2.1. Pré – traitement :

```

c:\ Invite de commandes
TT numero 17 ETAT : TICKET UALIDE
TT numero 18 ETAT : TICKET UALIDE
TT numero 19 ETAT : TICKET UALIDE
TT numero 20 ETAT : TICKET UALIDE
-----
<<< OPERATION DE DECODAGE >>>
-----
Decodage de Ticket Numero >>> 0
Decodage de Ticket Numero >>> 1
Decodage de Ticket Numero >>> 2
Decodage de Ticket Numero >>> 3
Decodage de Ticket Numero >>> 4
Decodage de Ticket Numero >>> 5
Decodage de Ticket Numero >>> 6
Decodage de Ticket Numero >>> 7
Decodage de Ticket Numero >>> 8
Decodage de Ticket Numero >>> 9
Decodage de Ticket Numero >>> 10
Decodage de Ticket Numero >>> 11
Decodage de Ticket Numero >>> 12
Decodage de Ticket Numero >>> 13
Decodage de Ticket Numero >>> 14
Decodage de Ticket Numero >>> 15
Decodage de Ticket Numero >>> 16
Decodage de Ticket Numero >>> 17
Decodage de Ticket Numero >>> 18
Decodage de Ticket Numero >>> 19
Decodage de Ticket Numero >>> 20
-----
<<< OPERATION DE HARMONISATION >>>
-----
Harmonisation de ticket numero >>> 0
Harmonisation de ticket numero >>> 1
Harmonisation de ticket numero >>> 2
Harmonisation de ticket numero >>> 3
Harmonisation de ticket numero >>> 4
Harmonisation de ticket numero >>> 5
Harmonisation de ticket numero >>> 6

```

Figure VI. 1 Exécution de sous composant Pré – traitement

VI.2.4. Facturation :

```

C:\ArchJava\bin\Uvalorisation_Produit>cd..
C:\ArchJava\bin>cd facturation
C:\ArchJava\bin\Facturation>java Main
Controle de volume de communication pour un acces
Notion de volume minimale facture pour une communication aboutie

Nombre de communication nationales pour un acces
Patienter...
Traitement numero >>> 0
Traitement numero >>> 1
Traitement numero >>> 2
Traitement numero >>> 3
Traitement numero >>> 4
Traitement numero >>> 5
Traitement numero >>> 6
Traitement numero >>> 7
Traitement numero >>> 8
Traitement numero >>> 9

Nombre de communication internationales pour un acces
Patienter...
Traitement numero >>> 0
Traitement numero >>> 1
    
```

Figure VI. 4 Exécution de l'opération de facturation

VI.3. Application IHM :

L'exécution de l'application commence tout d'abord par le saisie de mot de passe et le choix de type de l'utilisateur (administrateur, utilisateur, opérateur), une fois le mot de passe valider, l'administrateur aura l'accès aux fonctionnalités du système.

VI.3.1. Présentation Générale :

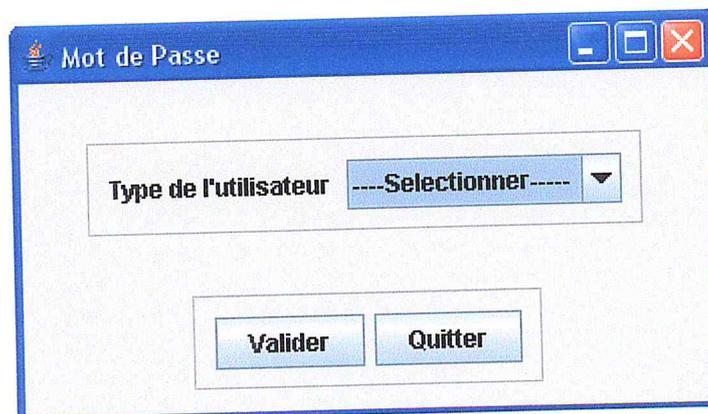


Figure VI. 5 Fenêtre principale de mot de passe

VI.2.2. Calculs des coûts :

```

Connection Base de donnee...
Traitement de ticket de taxation numero >>> 1
Traitement de ticket de taxation numero >>> 2
Traitement de ticket de taxation numero >>> 3
Traitement de ticket de taxation numero >>> 4
Traitement de ticket de taxation numero >>> 5
Traitement de ticket de taxation numero >>> 6
Traitement de ticket de taxation numero >>> 7
Traitement de ticket de taxation numero >>> 8
Traitement de ticket de taxation numero >>> 9
Traitement de ticket de taxation numero >>> 10
Traitement de ticket de taxation numero >>> 11
Traitement de ticket de taxation numero >>> 12
Traitement de ticket de taxation numero >>> 13
Traitement de ticket de taxation numero >>> 14
Traitement de ticket de taxation numero >>> 15
Traitement de ticket de taxation numero >>> 16
Traitement de ticket de taxation numero >>> 17
Traitement de ticket de taxation numero >>> 18
Traitement de ticket de taxation numero >>> 19
Traitement de ticket de taxation numero >>> 20
Traitement de ticket de taxation numero >>> 21
    
```

Figure VI. 2 Exécution de sous composant calculs des coûts

VI.2.3. Coûts des services :

```

C:\ Invite de commandes
C:\ArchJava\bin\Valorisation_Produit>java Main
-----
OPERATION DE VALORISATION DE PRODUIT
-----
Chargement de la base des donnees ...
Traitement de abonnee numero >>> 8
Traitement de abonnee numero >>> 7
Traitement de abonnee numero >>> 9
Traitement de abonnee numero >>> 10
Traitement de abonnee numero >>> 11
Traitement de abonnee numero >>> 2
Traitement de abonnee numero >>> 1
Traitement de abonnee numero >>> 3
Traitement de abonnee numero >>> 4
Traitement de abonnee numero >>> 5
    
```

Figure VI. 3 Exécution de sous composant coûts des services

The screenshot shows a window titled "Mot de Passe" with a standard Windows-style title bar. Inside the window, there are three main input areas: a dropdown menu labeled "Type de l'utilisateur" with "Utilisateur" selected, a text box labeled "Nom de l'utilisateur" containing the text "utilisateur", and a password field labeled "Mot de Passe" filled with asterisks. At the bottom of the window, there are two buttons: "Valider" and "Quitter".

Figure VI. 6 Fenêtre mot de passe pour un utilisateur

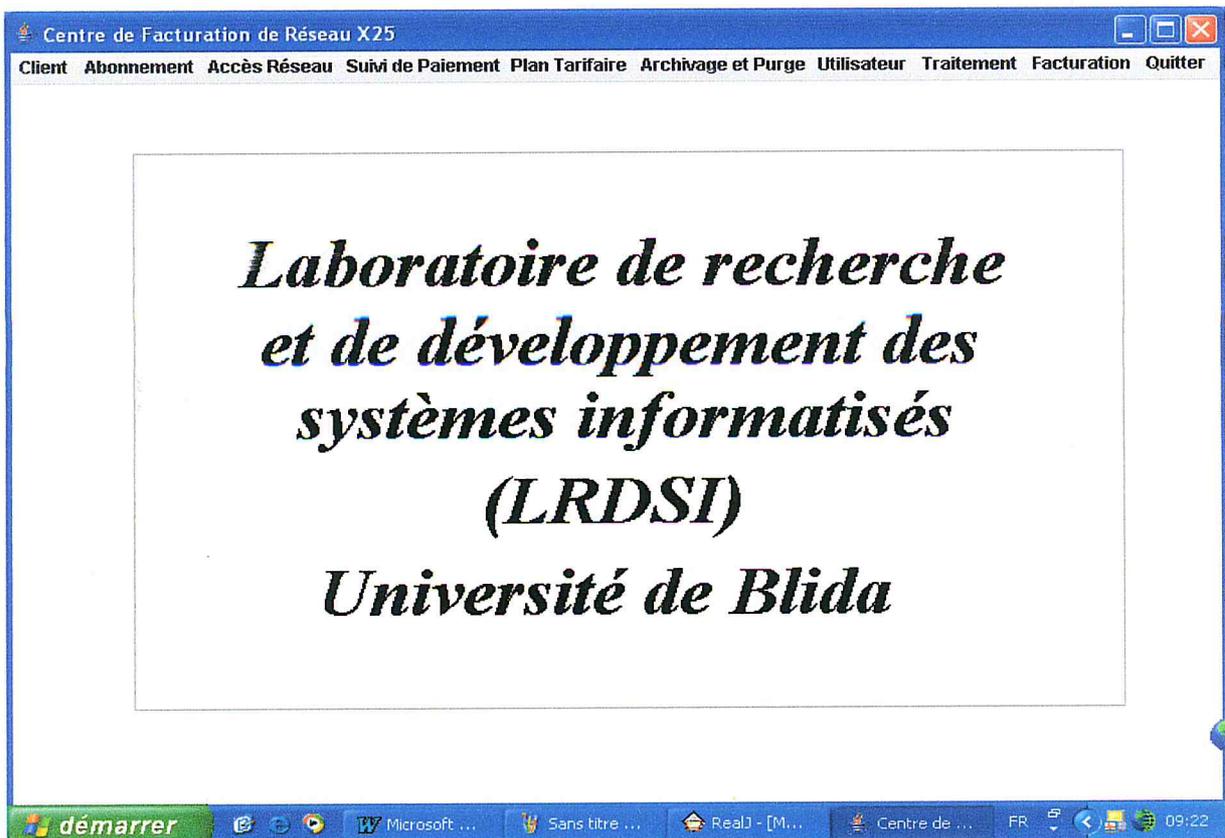


Figure VI. 7 Présentation Générale de l'application

L'interface principale est constituée des onglets suivants :

Client, abonnement, accès Réseau, Suivi de paiement, Plan Tarifaire, Archivage et Purge, Utilisateur, Traitement, Facturation, Quitter.

VI.3.2. Présentation détaillée de quelques onglets :

VI.3.2.1. l'onglet Client :

permet la création, la suppression, la modification et la consultation des clients.

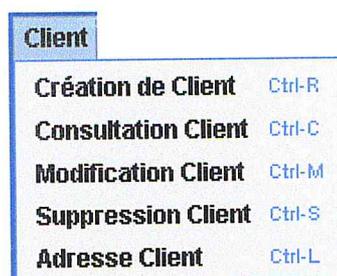


Figure VI. 8 Onglet Client

La fenêtre 'Création d'un Client' est affichée, permettant de saisir les informations d'un nouveau client. Le titre de la fenêtre est 'Création d'un Client' et le contenu principal est 'Création d'un nouveau Client'. Le logo 'ALGERIE TELECOM' est visible en haut à droite.

Informations Client

Type de Client : Société (dropdown) Civilité : Monsieur (dropdown) Nom : [champ texte]

Prénom : [champ texte] Complément d'info : [champ texte]

Raison : [champ texte] Numéro Fiscal : [champ texte] Etat : Actif (dropdown)

Adresse

Agence : Agence d'Alger (dropdown) Adresse : [champ texte]

Paramètres du Client

Facturable : Oui (dropdown) Périodicité : Mensuelle (dropdown) Parité : Pair (dropdown) TVA Applicable : Oui (dropdown) Jour de Facturation : [champ texte]

1 (dropdown) Langue : Français (dropdown) Détail : Détaillée (dropdown) Nombre de Création GFA : [champ texte]

Réductions Client

Raccordement National : Tarif Normal (dropdown) Raccordement International : Tarif Normal (dropdown) Communication : [champ texte]

Tarif Normal (dropdown)

Buttons : Sauvegarde, Annuler, Fermer

Barre de tâches Windows : démarrer, Micro..., Sans..., RealD..., Centr..., Créati..., FR, 09:14

Figure VI. 9 Création d'un client

VI.3.2.2. l'onglet abonnement :

Abonnement	
Création Abonnement	Ctrl-E
Consultation Abonnement	Ctrl-N
Résiliation Abonnement	Ctrl-B

Figure VI. 10 Onglet Abonnement

Figure VI. 11 Création d'un abonnement

VI.3.2.3. l'onglet Suivi de paiement

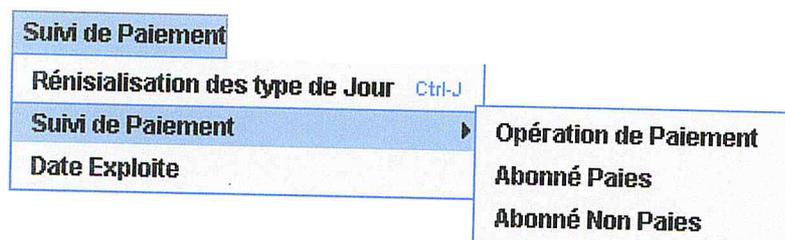


Figure VI. 12 Onglet Suivi de paiement

VI.3.2.4. l'onglet Plan Tarifaire :

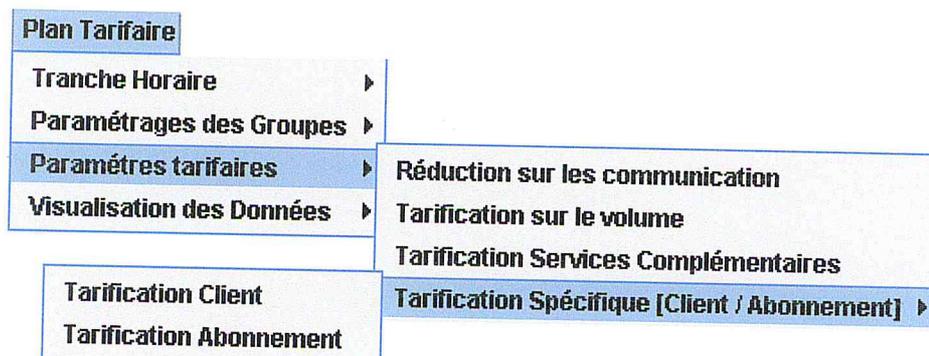


Figure VI. 13 Onglet Plan Tarifaire

Tarifification Services Complémentaires

TARIF POUR LES SERVICES COMPLEMENTAIRES



 اتصالات الجزائر
 ALGERIE TELECOM

Tarif de services complémentaires

Code de Service	Libellé de service	Tarif de service	Type de service
1	Voies GFA	0	1
2	VL Supplémentaires	0	2
3	Service Compléme...	0	3
4	Service Compléme...	0	3
5	Service Compléme...	0	3
6	Service Compléme...	0	3
7	Longueurs de Pack...	0	3
8	Longueurs de Pack...	0	3

Ajouter Tarif

Code de Service :

Libellé de Service Complémentaire :

Tarif de Service Complémentaire :

Type de Service Complémentaire :


 démarrer | Real] - [CFX25 | Centre de Facturati... | Tarification Services ... | FR | 10:05

Figure VI. 14 Tarif des services Complémentaires

VI.3.2.5. l'onglet archivage et Purge :

Archivage et Purge

- Archivage
- Purge de la Base des Données
- Journal
 - Pré - Traitement
 - Valorisation

Figure VI. 15 Onglet Archivage et purge

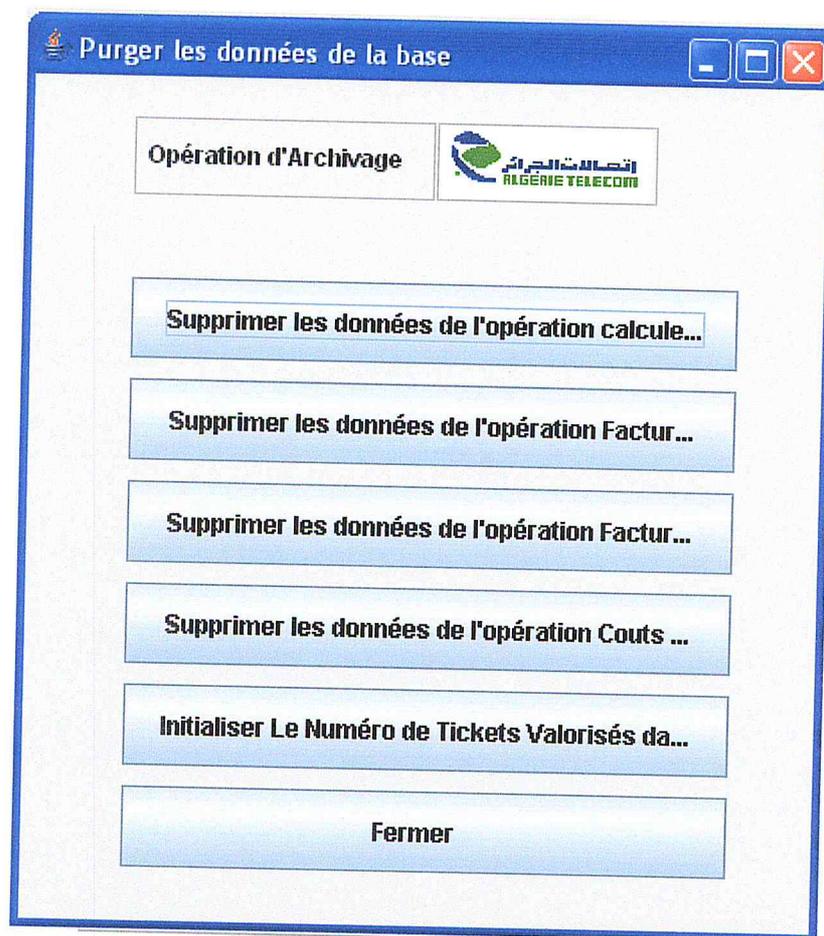


Figure VI. 16 Purge des données

VI.3.2.6. l'onglet Utilisateur :

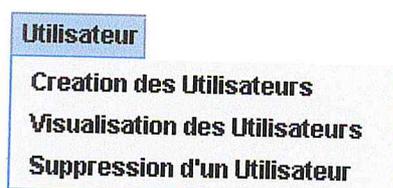


Figure VI. 17 Onglet Utilisateur

The screenshot shows a window titled "Creation des Utilisateurs" with a blue header bar. Below the header, the main content area is titled "Gestion des utilisateurs" and features the logo of "ALGERIE TELECOM" in the top right corner. The form contains three input fields: "Nom Utilisateur" with the value "Adel", "Password" with masked characters "*****", and "Confirmation Password" also with masked characters "*****". Each field has a red asterisk (*) to its right, indicating it is a required field. At the bottom of the form, there are three buttons: "Inscription", "Annuler", and "Quitter". Below the buttons, a legend states "(*) : Information Obligatoire".

Figure VI. 18 Gestion des utilisateurs

VI.3.2.7. l'onglet Traitement :

The screenshot shows a menu titled "Traitement" with a blue header. Below the header, there are two menu items: "Lancer les Traitements" and "Historique des Traitements".

Figure VI. 19 Onglet Traitement

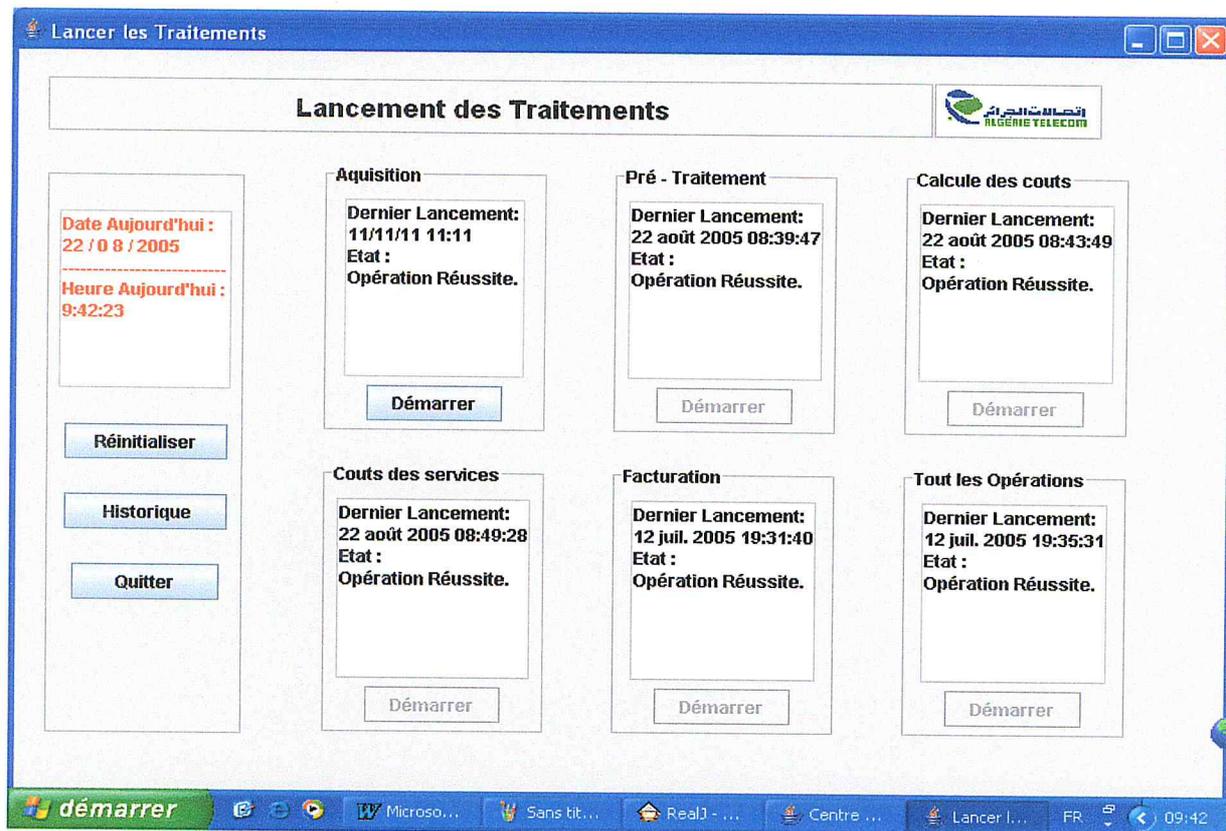


Figure VI. 20 Lancement des traitement



Figure VI. 21 Historique des traitements

VI.3.2.8. l'onglet Facturation :

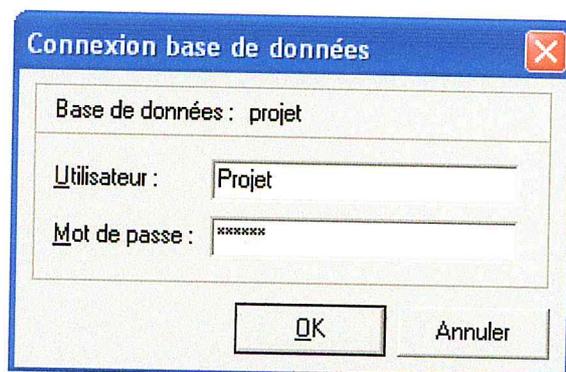


Figure VI. 22 Mot de passe de facturation

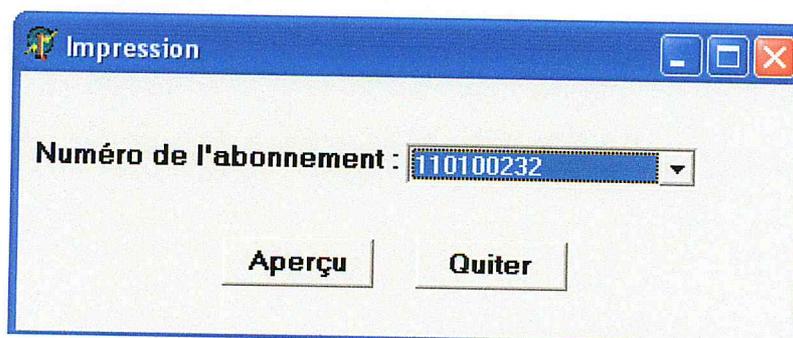


Figure VI. 23 Choix de numéro d'abonné

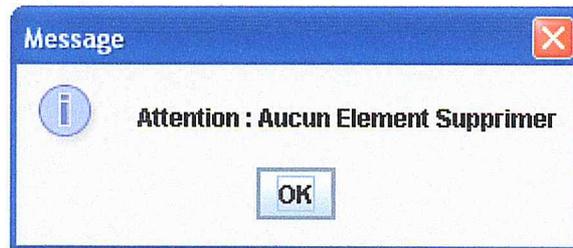


Figure VI. 26 Message d'avertissement

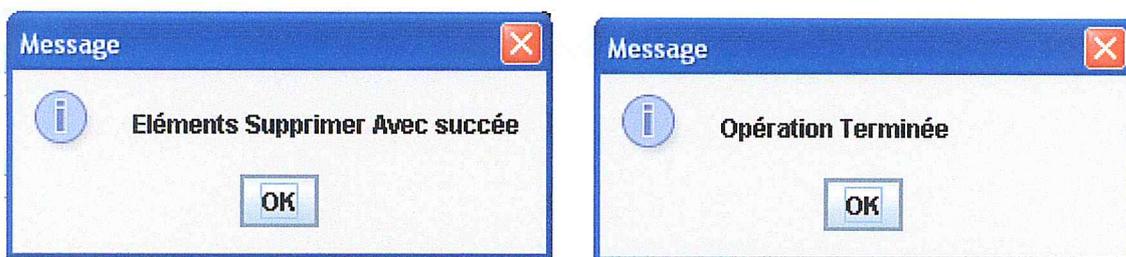


Figure VI. 27 Message de confirmation

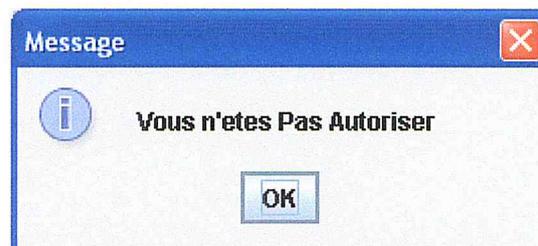


Figure VI. 28 Message d'erreur

VI.3.4. les icônes d'opérations :

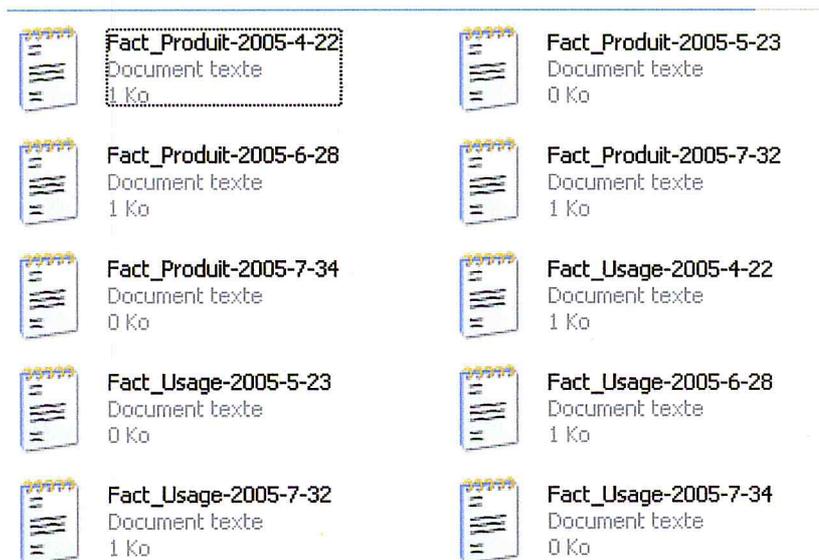


Figure VI. 29 Icône des données archiver

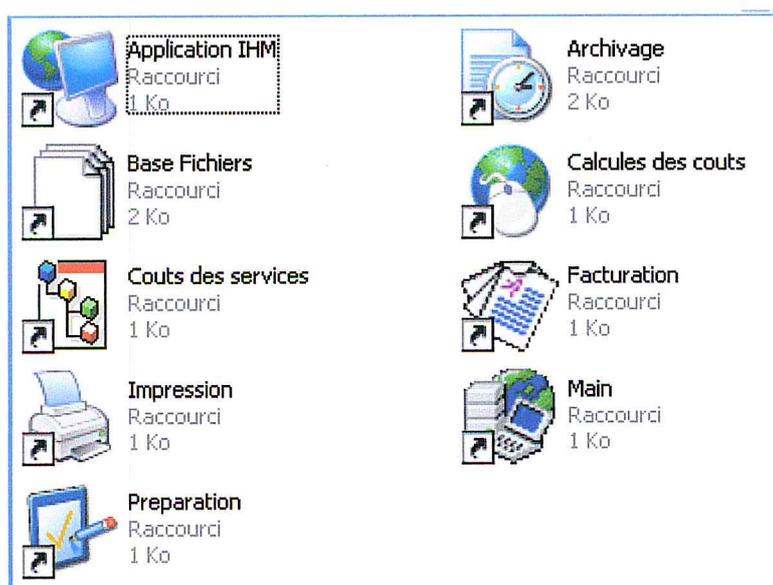


Figure VI. 30 Icône des différents traitements

Conclusion et perspectives

Conclusion et perspectives :

L'approche de description architecturale a permis notamment de prendre en compte les descriptions de haut niveau de systèmes complexes, et de raisonner sur leur propriétés a un haut niveau d'abstraction. cette approche décrit un système logiciel comme un ensemble de composants qui interagissent entre eux par le biais de connecteurs, elle décrit aussi explicitement les composants et définit des mécanismes opérationnels permettant leur réutilisation.

Les perspectives se résument comme suit :

- Les ADLs permet la définition de l'architecture globale d'un système avant même la fin de la construction de ses composants (implémentation).
- Elle fournit un très haut niveau de granulaire (niveau de composants).
- Elle fournit une représentation hiérarchique plus forte que la relation d'héritage et elle supporte des configurations plus détaillées pour chaque partie de l'architecture [RFC02].
- La complexité croissante des système et leur évolution de plus en plus rapide ont motivé un intérêt accru pour les ADLs.
- L'architecture logicielle permet d'améliorer le coût de production et la maintenance d'un logiciel grâce à l'étude des structures de haut niveau du logiciel.
- L'orienté objet ne permet que des interaction simple de type appel de procédure ou des invocations de méthodes qui ne sont certainement pas suffisantes pour décrire des interaction complexes. Par contre dans les ADL Un connecteur peut décrire des interactions simples de type appel de procédure ou accès a un variable partagée, mais aussi des interactions complexes.

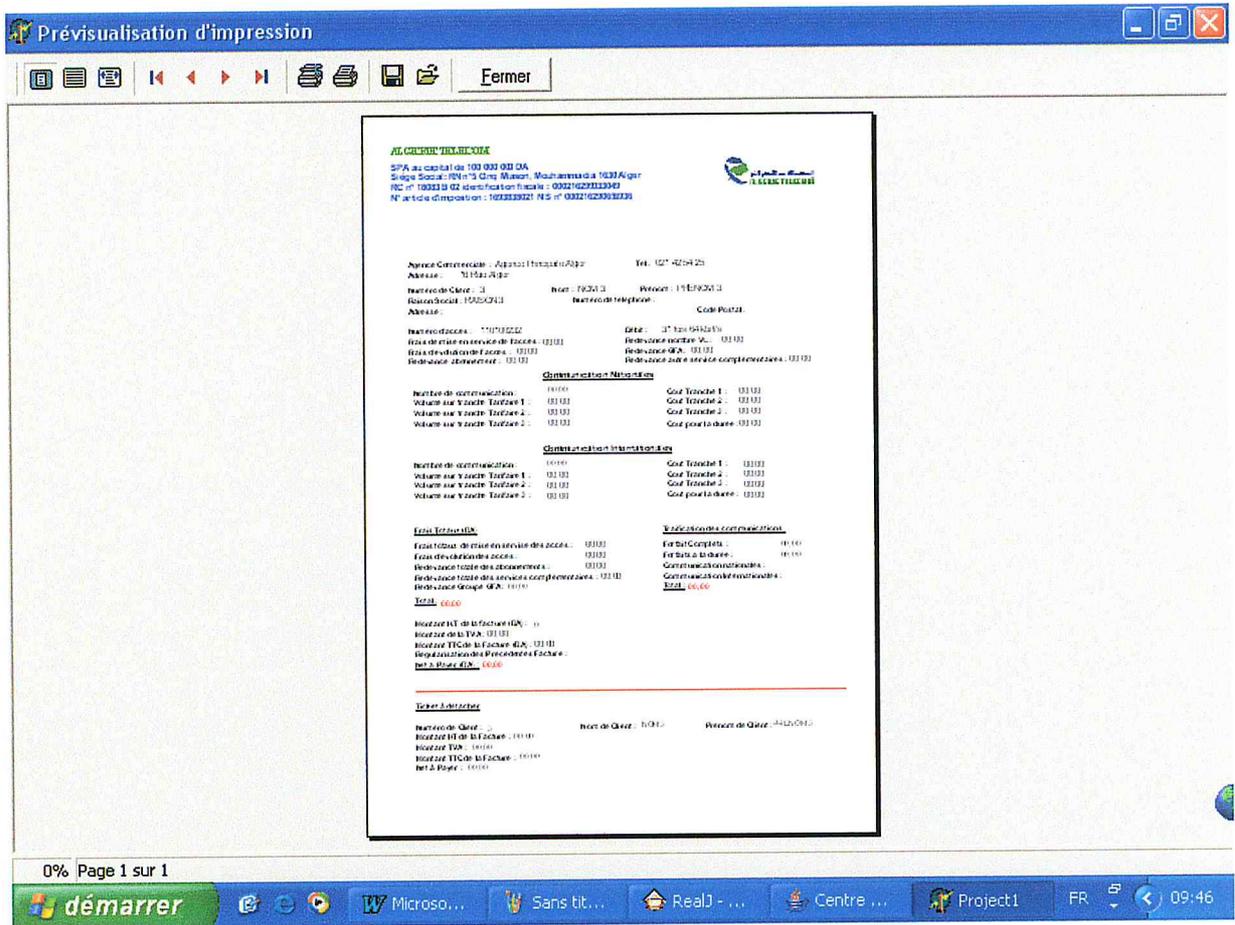


Figure VI. 24 La facture

VI.3.3. les messages :

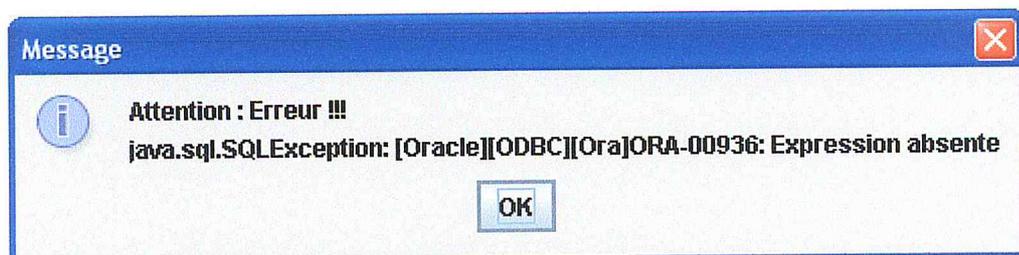


Figure VI. 25 message d'erreur

Annexes

Annexe 1

Le modèle Objet – relationnel

Introduction :

(Extrait de Livre Objet – relationnel de Christian Soutou)

Est une extension de modèle relationnel et prendre en charge :

- ◆ Les types abstraits des données :
 - ✓ Les structures des données complexes.
 - ✓ L'encapsulation.
 - ✓ L'héritage.
- ◆ L'identité des objets (OID : Object IDentifier) via les pointeurs.

Ces concepts peuvent se combiner entre eux, en effet, une TAD (type abstrait des données) peut contenir une structure de données complexe qui comporte des pointeurs.

Les concepts de structures de données complexes et d'identité d'objets sont en contradiction avec les principes de base du modèle relationnel (le modèle relationnel repose sur la première forme normale) par contre le concept d'identité d'objet est mis en œuvre par l'utilisation des pointeurs physiques, ce qui est en opposition avec le modèle relationnel, basé sur les valeurs.

Les associations entre tables sont assurées par les attributs de type clé étrangère qui jouent le rôle de pointeur logique.

Le concept de table existe toujours mais il est étendu, en effet, la structure d'une table objet – relationnelle est définie à l'aide d'un TAD.

En conséquence, les attributs d'une table sont, au choix, de type traditionnel (nombre, chaîne de caractères, date,), une collection (ordonnée ou non), un pointeur (référence physique) d'un type défini par le concepteur, Ou une combinaison autorisée des types précédents.

Les enregistrements d'une table objet – relationnelle peuvent être assimilés à des objets, en effet, ils sont définis à l'aide d'une structure qui dispose de méthodes et doté d'un OID distinct.

1) Type abstraits de données (TAD) :

La première extension de modèle relationnel concerne les TAD.

Un TAD peut être perçu comme :

- **Un nouveau type d'attribut** défini par l'utilisateur qui enrichit la collection existante de type disponibles par défaut (nombre, chaîne de caractères,).
- **Une structure de donnée partagée**, dans ce cas, un type peut être utilisé par une ou plusieurs tables et entrer dans la composition d'un ou plusieurs autres types.

Un TAD inclus des méthodes qui sont des procédures ou des fonctions, elles permettent de manipuler les objets du type abstrait.

1.1) Nouveau type d'attribut :

Un utilisateur désire se référer au calendrier républicaine (les mois sont de 30 jours). Une solution consiste à définir le TAD *DATEREPUBL* à partir de type date proposé par le SGBD.

Le nouveau type permet d'effectuer des opérations sur les dates conformément au calendrier républicaine.

Syntaxe SQL

```
CREATE TABLE evenements_historiques
    (num_event number primary key, titre char(50),
    moment DATEREPUBL) ;
```

1.2) Structure de donnée partagée :

Un TAD considéré comme une structure de donnée partagé permet de définir une partie ou la totalité d'une ou plusieurs tables, il peut également entrer dans la composition de différents TAD.

Exemples :

1.2.1) Type de base :

Type abstrait :

Type : Personne_type

Personne_type			
Nom	Prénom	Date de naissance	Nationalité

Table objet – relationnelle utilise ce TAD :

Table : Personne of Personne_type

Nom	Prénom	Date de naissance	Nationalité

1.2.2) Type utilise un autre type :

Type abstrait définit à partir d'un autre type :

Type : Employé_type

Employé_type					
Numéro_SS	Unité	Personne_type			
		Nom	Prénom	Date de naissance	Nationalité

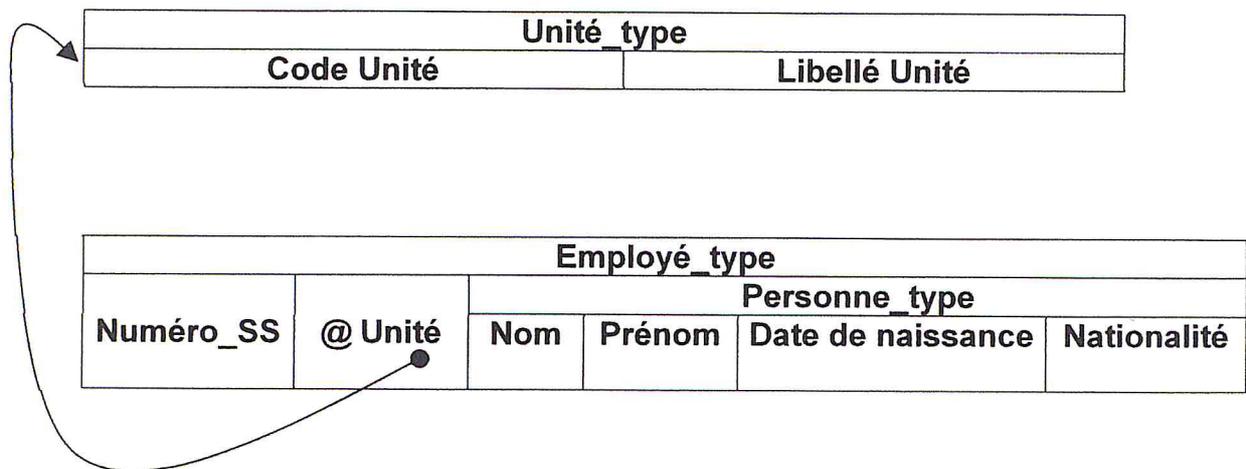
Table objet – relationnelle utilise ce TAD :

Table : Employé of Employé_type

Numéro_SS	Unité	Personne			
		Nom	Prénom	Date de naissance	Nationalité

1.2.3) Pointeurs

Type utilise un pointeur :



Tables objet – relationnelles contenant des objets :

Code Unité	Libellé Unité
01	Montage et assemblage
02	Peinture
.....	

Numéro_SS	@Unité	Personne			
		Nom	Prénom	Date de naissance	Nationalité
0901010989	●	x	y	12/12/1976	Algérienne
.....					
.....					

1.2.4) Structure de type collection :

Enseignant :

Numéro	Nom	Adresse E-Mail
0102	Amin	Amin@yahoo.fr
		Amin_info@caramail.com
		Amin@hotmail.com

1.2.5) Encapsulation :

Dans le modèle objet – relationnel l'Encapsulation se traduit par la possibilité de définir des méthodes (procédures ou fonctions) dans la déclaration de TAD.

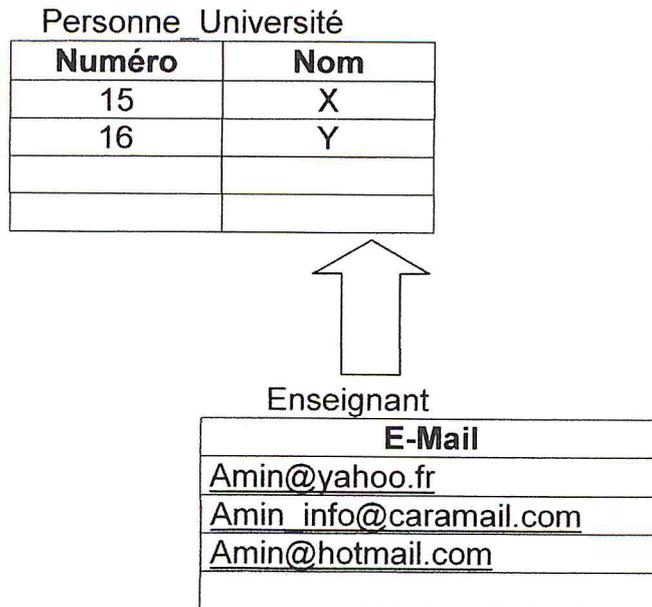
Un TAD donc décrit la structure des objets mais également le comportement.

Le concept d'Encapsulation a pour but d'améliorer l'extensibilité et la maintenance des applications.

1.2.6) Héritage :

Dans le modèle objet – relationnel l'héritage se traduit par la possibilité de définir des TAD génériques.

Exemple : tables objet – relationnelles avec héritage



2) Identité des objets :

La deuxième extension du modèle relationnel porte sur la manipulation des OID. Dans le modèle objet – relationnel, l'identité des objets est prise en compte par l'utilisation de pointeurs dans les tables et a pour but d'améliorer la fonction de navigation dans le langage de requêtes.

Certaines requêtes SQL d'interrogation d'Oracle sont simplifiées en raison de la présence de pointeurs dans les tables objet – relationnelles.

3) Exemples (livre Objet – relationnel sous oracle8) :

3.1) Requêtes monotables:

Avec le modèle relationnel :

Poste_travail

Nserie	typeposte	Serveur#
P1	Win95	P3
P2	Win95	P3
P3	WinNT	Null

Table relationnelle.

Requête : numéro et type de poste du serveur du poste client P1 ?

```
SELECT nserie,typeposte
FROM Poste_travail
WHERE nserie = (SELECT serveur FROM poste_travail WHERE nserie = 'P1') ;
```

↑
JOINTURE

Avec le modèle objet – relationnel :

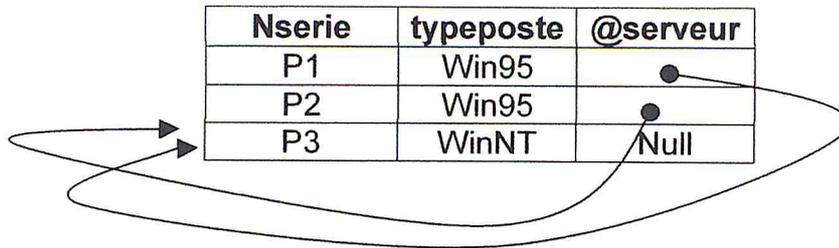


Table Objet – relationnelle

Requête : numéro et type de poste du serveur du poste client P1 ?

```
SELECT p.serveur.nserie,p.serveur.typeposte
FROM poste_travail p
WHERE p.nserie = 'P1' ;
```

3.2) Requêtes multitables:

Avec le modèle relationnel :

Segment :

Ind IP	nomseg	longueur
130.40.30	ICARE	25
130.40.31	DEDALE	75
130.40.31	MINOS	40

Poste travail :

nserie	adrIP	typeposte	Nsegment#
P1	01	Win95	130.40.30
P2	02	Win95	130.40.30
P3	03	WinNT	130.40.30
P4	01	TX	130.40.31

Requête : indicatif IP et longueur des segments sur lesquels sont connectés des poste de type WinNT ?

```
SELECT ind_IP,longueur
FROM segment WHERE ind_IP IN
(SELECT nsegment FROM poste_travail WHERE typeposte ='WinNT') ;
```

Avec le modèle objet – relationnel :Poste travail :

nserie	adrIP	typeposte	@seg
P1	01	Win95	●
P2	02	Win95	●
P3	03	WinNT	●
P4	01	TX	●
P5	02	UnixHP	

Ind_IP	nomseg	longueur
130.40.30	ICARE	25
130.40.31	DEDALE	75
130.40.32	MINOS	40

Requête : indicatif IP et longueur des segments sur lesquels sont connectés des poste de type WinNT ?

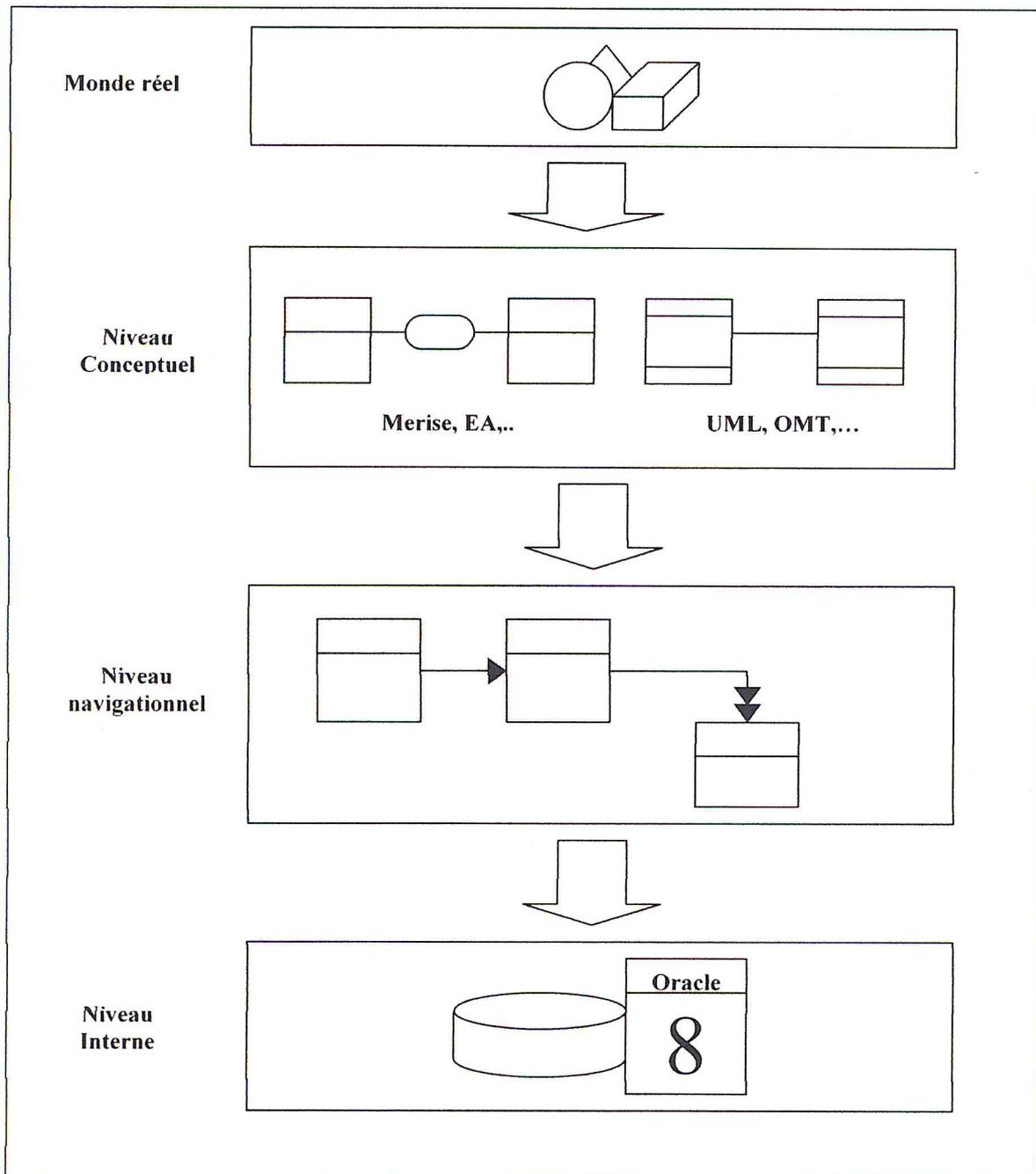
```
SELECT p.seg.ind_IP,p.seg.longueur
FROM poste_travail p
WHERE p.typeposte = 'WinNT' ;
```

Annexe 2

Le modèle Navigationnel

1) Introduction :

Le terme navigationnel est utilisé car ce modèle inclut des pointeurs. Ces pointeurs permettent d'améliorer la navigation entre objet au niveau de la base de données, il propose en outre d'autres notions qui ne sont pas prises en compte par le modèle CODASYL (type ensemble, structures...).



Les approches objet permettent de représenter directement les données complexes. Les principaux avantages d'un niveau pivot placé entre un schéma conceptuel et un schéma de base de données objet – relationnelle sont les suivants :

- ❖ Les liens intra – classe et les structures de données complexes sont mieux représentés.
- ❖ La traduction des associations n – aires au niveau physique est clairement décrit.
- ❖ L'évolution de la base de données est facile : ajout, modification, suppression d'une propriété ou d'une table.
- ❖ Il est plus aisé de transposer des classe extraites du schéma conceptuel en type et en tables objet – relationnelle.
- ❖ La visualisation des chemins entre classe facilite la définition de requêtes d'interrogations et l'organisation des méthodes (fonctions ou procédures).

Le formalisme du modèle navigationnel permet de représenter :

- ◆ Les aspects statique et dynamique des classes.
- ◆ Les liens intra et inter – classes.
- ◆ Les hiérarchies d'héritage.

2) Les classes et les liens :

Les classes contiennent des attributs (propriétés) et des signatures de méthodes (fonction ou procédures).

Les attributs peuvent être :

- Simple (chaîne de caractères, nombre,...).
- De type structure (composition d'attributs).
- De type référence (Pointeur).
- De type ensemble (collection de structures ou de références).
- Des combinaisons des types précédents.

Les liens peuvent exister :

- A l'intérieur d'une même classe (intra – classes).
- Ou entre deux classes (inter – classes).

La représentation graphique des liens :

Dans un schéma navigationnel, les liens monovalués sont représentés par des flèches simples et les liens multivalués par des flèches a double pointe.

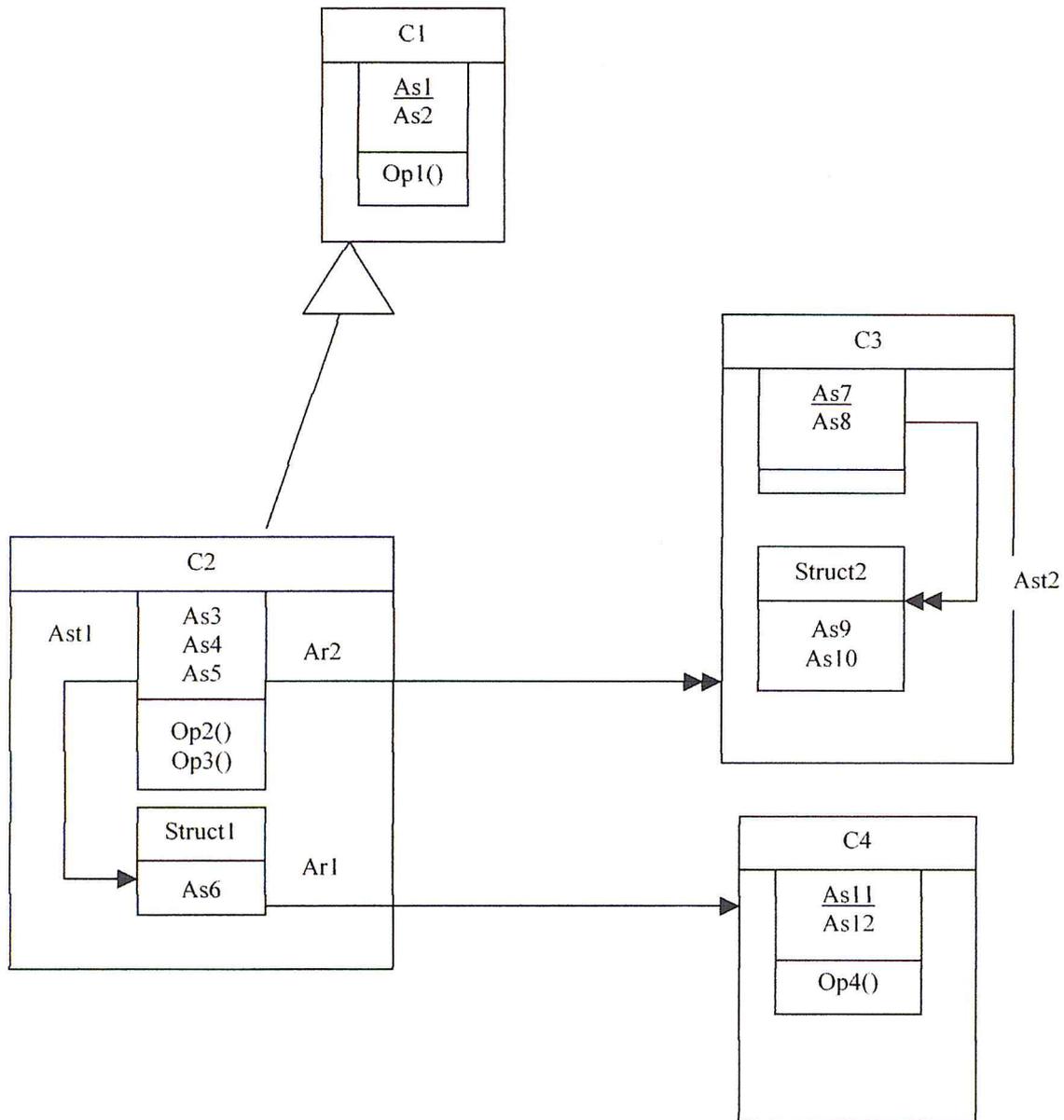


Lien monovalués
Référence ou structure



lien multivalués
ensemble de références ou de structures

Exemple général :



La classe C2 est une sous classe de la classe C1 (C2 hérite de C1), les classes C1, C2 et C4 possèdent des opérations (méthodes), les classes C1, C3 et C4 ont un identifiant (attribut souligné), les types des attributs sont les suivants :

- ❑ As1, As2, ..., As12 sont simple,
- ❑ Ast1 est de type structure struct1.
- ❑ Ar1 est de type référence vers une classe.
- ❑ Struct1 est une structure composée d'un attribut simple et d'une référence vers une classe.
- ❑ Ast2 est de type ensemble de structures struct2.
- ❑ Struct2 est une structure composée de deux attributs simples.
- ❑ Ar2 est un ensemble de références vers une classe.

3) Du conceptuel au navigationnel :

Ce paragraphe décrit le processus de passage d'un schéma conceptuel (de type entité – association ou UML) au modèle navigationnel.

Transformation des entités / classes :

Classes du schéma navigationnel :

Chaque **entité** du schéma conceptuel entité – association devient une classe du schéma navigationnel, les propriétés de l'entité sont les propriétés de la classe.

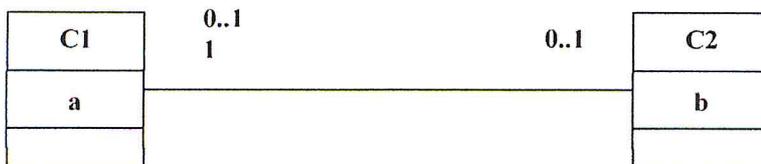
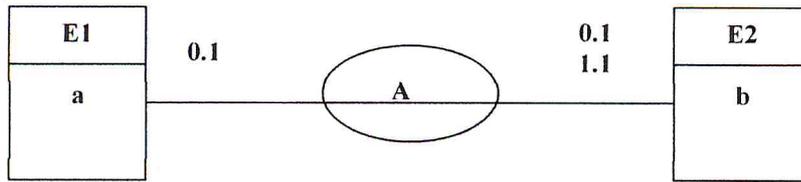
Chaque **classe** du diagramme de classe UML, exceptées les classes – associations, devient une classe du schéma navigationnel.

Identifiants des classes du schéma navigationnel :

Chaque classe, exceptées les sous – classes du schéma navigationnel, doit contenir *une propriété qui joue le rôle d'identifiant*, si aucune propriété existante ne convient, il est nécessaire d'ajouter une propriété permettant d'identifier chaque objet de la classe.

3.1) Transformation des associations 1 – 1 :

Il y a quatre possibilités de transformation d'une association 1 – 1 du schéma conceptuel dans le modèle navigationnel.



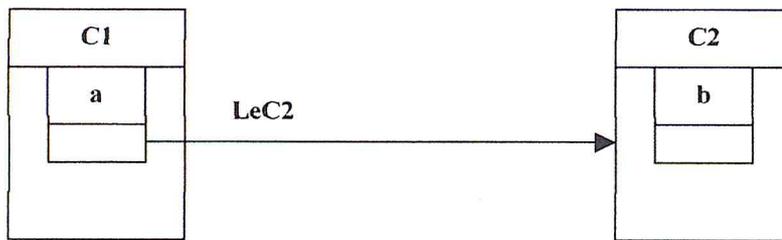
3.1.1) Première solution :

La première solution de transformation met en œuvre un lien monovalué entre les deux classes, ce lien est «LeC1», la cardinalité minimale 0 figure à la source du lien.

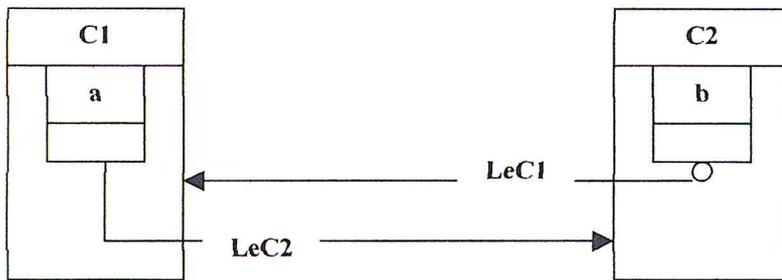


3.1.2) Deuxième solution :

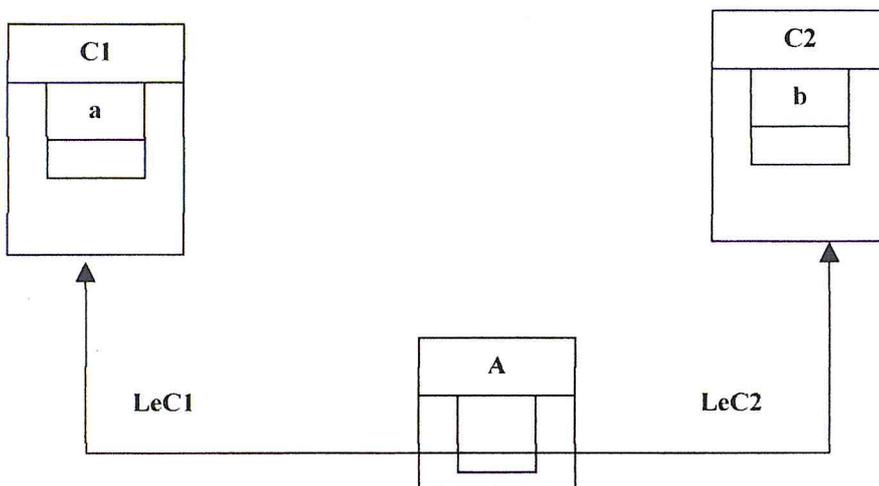
La deuxième solution de transformation est symétrique de la précédente, elle met en œuvre un lien monovalué 'LeC2' entre les deux classes.

**3.1.3) Troisième solution (Deux liens monovalués) :**

La troisième solution est une combinaison des deux précédentes, elle met en œuvre deux liens monovalués entre les deux classes, ces liens sont 'LeC1' et 'LeC2'.

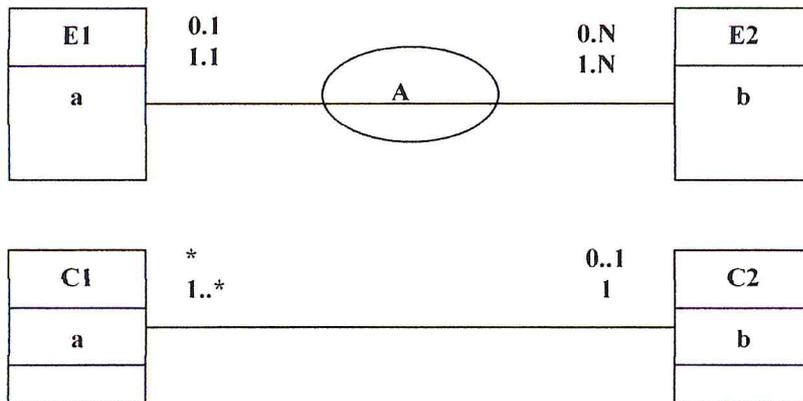
**3.1.4) Quatrième solution (solution universelle) :**

La solution universelle de transformation d'une association 1 – 1 est basée sur l'utilisation d'une troisième classe, cette classe porte le nom de l'association et contient deux liens monovalués qui la relient aux deux autres, ces liens sont 'LeC1' et 'LeC2'.



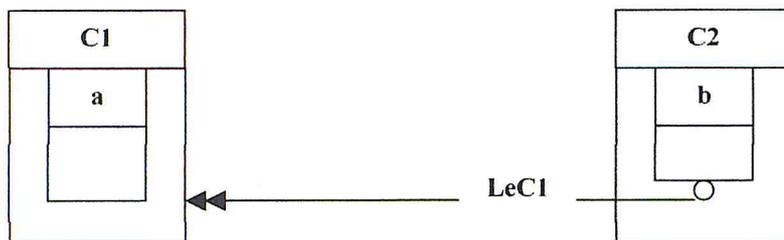
3.2) Transformation des associations 1 – N :

Il y a quatre possibilités de transformation d'une association 1 – N du schéma conceptuel dans le modèle navigational.



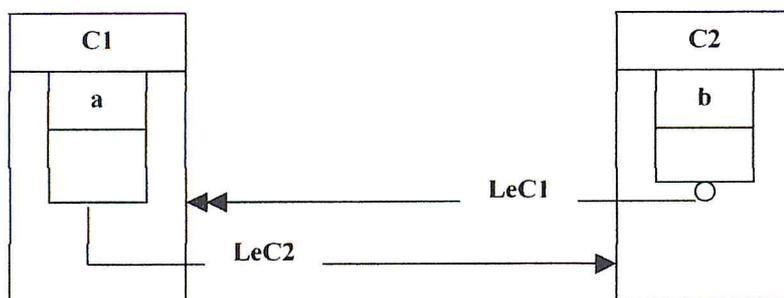
3.2.1) Première solution (un lien multivalué) :

La première solution de transformation met en œuvre un lien multivalué entre les deux classes, ce lien est 'LeC1'.



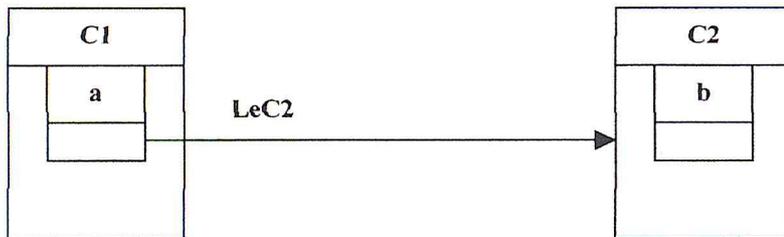
3.2.2) Deuxième solution (un lien multivalué et un lien inverse) :

La deuxième solution met en œuvre un lien multivalué (du père vers ses fils) ainsi qu'un lien monovalué qui joue le rôle d'un référence inverse (du fils vers le père), ici le lien multivalué est le 'LeC1' et le lien monovalués est le 'LeC2'.



3.2.3) Troisième solution (un lien monovalué) :

Il s'agit de recourir uniquement à un lien monovalué du fils vers le père, ce lien est 'LeC2'.

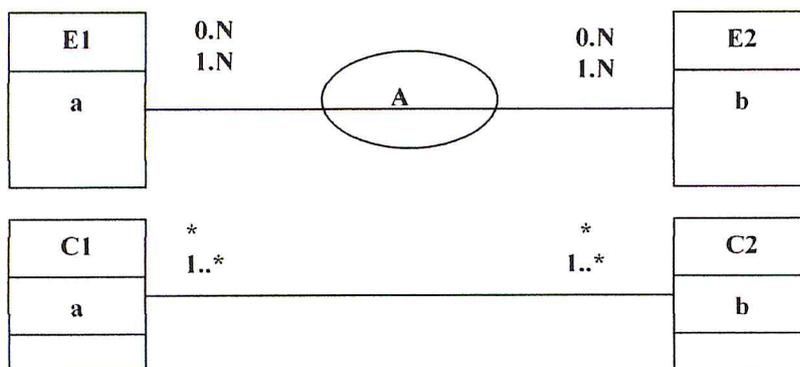


3.2.4) Quatrième solution (solution universelle) :

La solution universelle de transformation est basée sur l'utilisation d'une troisième classe, cette classe porte le nom de l'association et contient deux liens monovalués qui la relient aux deux autres (même schéma que la transformation des associations 1 – 1, la solution universelle).

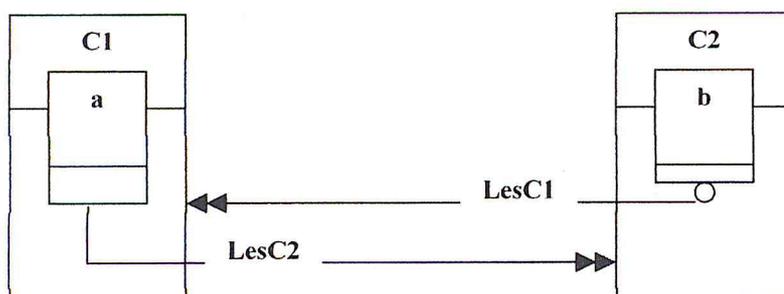
3.3) Transformation des associations N – N sans propriété :

Il existe deux possibilités de transformation d'une association N – N sans propriété du schéma conceptuel dans le modèle navigational.



3.3.1) Première solution (deux liens multivalués) :

La première solution de transformation d'une association N – N sans propriété met en œuvre deux liens multivalués entre les deux classes, ces liens sont 'lesC1' et 'lesC2' avec une cardinalité minimale égale à 0 à la source des deux liens.

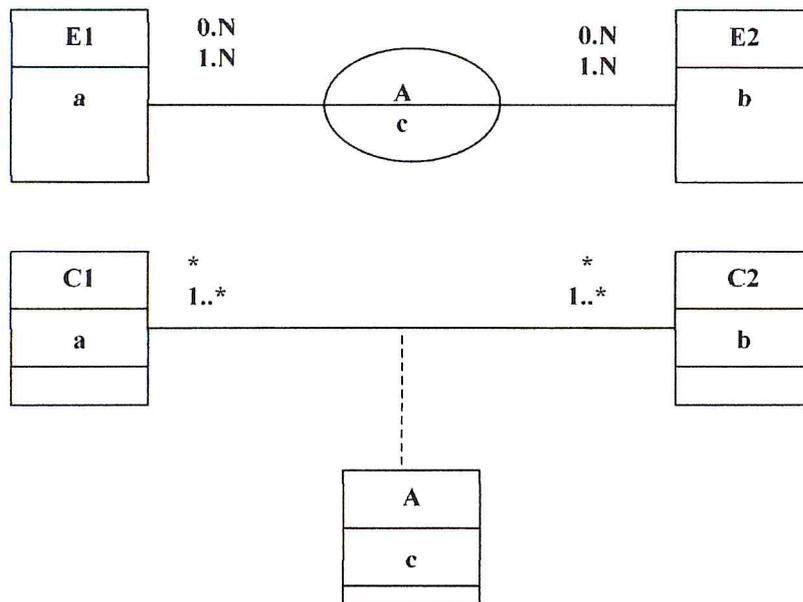


3.3.2) Deuxième solution (solution universelle) :

Comme pour les associations 1 – 1 et 1 – N, la solution universelle de transformation d’une association N – N sans propriété met en œuvre un troisième classe qui porte le nom de l’association et qui contient deux liens monovalués.

3.4) Transformation des associations N – N avec propriétés :

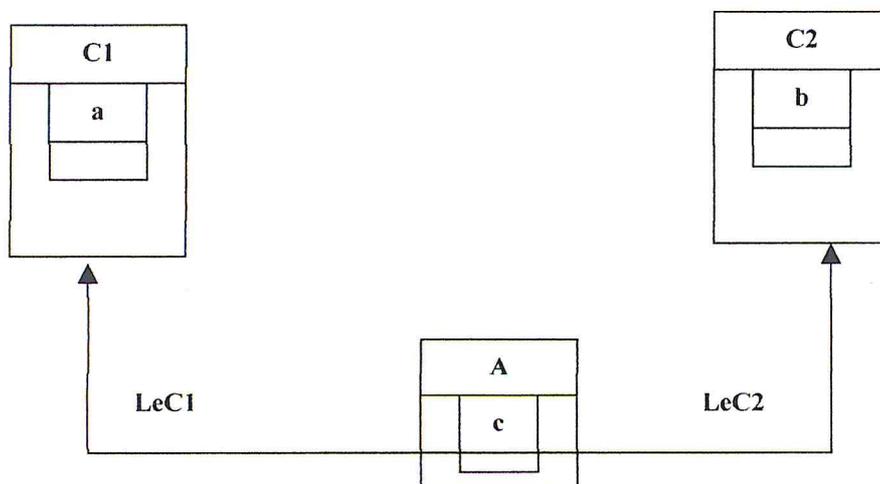
Il existe trois possibilités de transformation d’une association N – N avec propriétés du schéma conceptuel dans le modèle navigationnel.



3.4.1) Première solution (solution universelle) :

La solution universelle de transformation d’une association N – N avec propriétés implique la création d’une troisième classe qui porte le nom de l’association et qui contient :

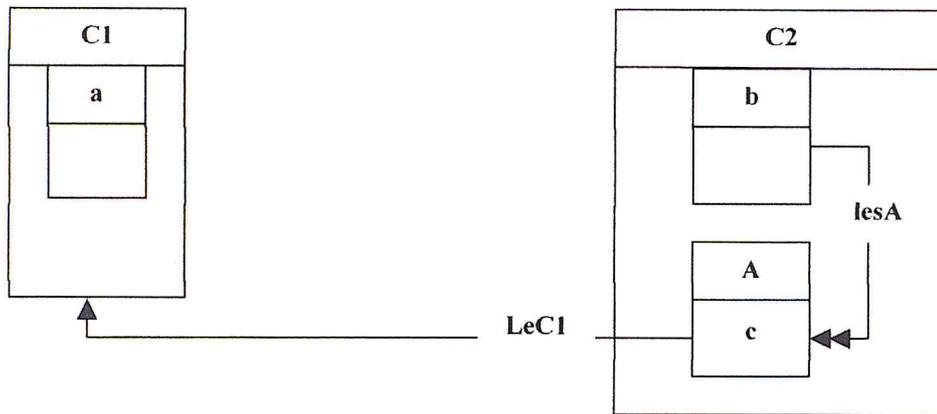
- ❑ Les attributs de l’association.
- ❑ Deux liens monovalués vers chaque classe.



3.4.2) Deuxième solution (une structure et deux liens) :

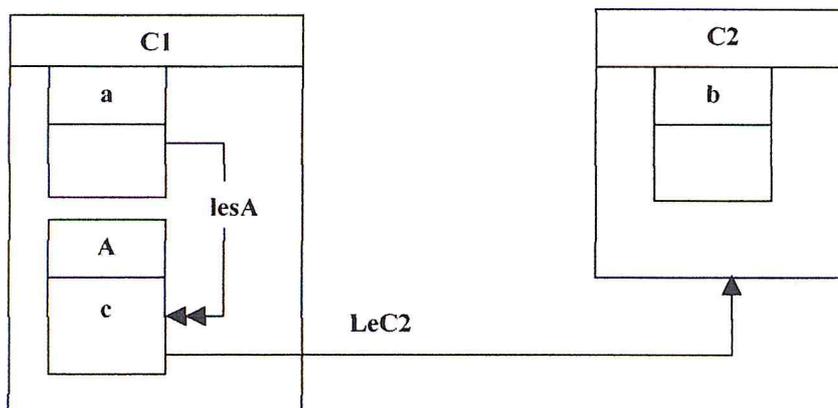
La deuxième solution de transformation d'une association N – N avec propriétés entre deux classes C1 et C2 met en œuvre :

- Une structure dans la classe C2 qui contient :
 - Les attributs de l'association.
 - Un lien monovalués vers la classe C1.
- Un lien multivalué vers la classe C1.



3.4.3) Troisième solution (symétrique de la deuxième solution) :

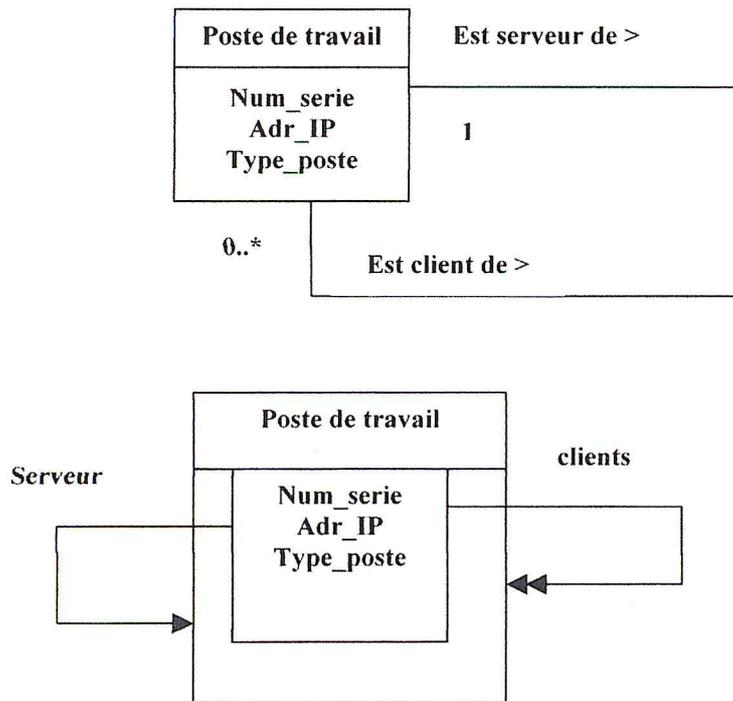
La deuxième et la troisième solution sont symétrique, la classe C1 contient l'attribut 'c' et le lien monovalués 'LeC2' vers la classe C2, la classe C1 et attachée ç cette structure par le lien multivalué 'LesA'.



3.5) Transformation des associations réflexives :

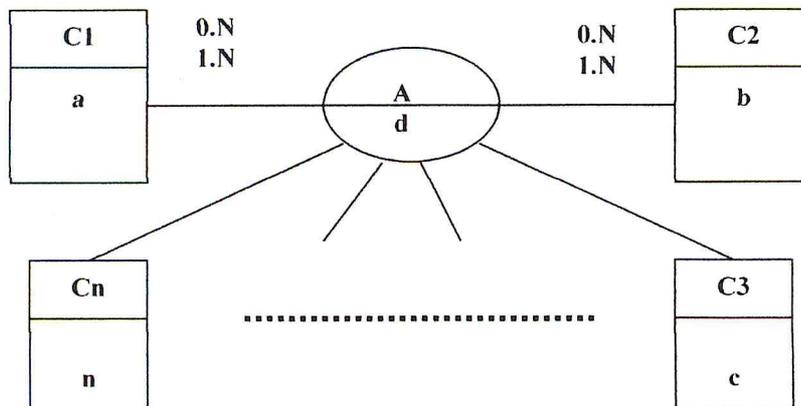
Les associations réflexives peuvent être binaires ou n – aires, en conséquence, il convient d'utiliser une solution de transformation en fonction du type de l'association réflexives (1 – 1, 1 – N, N – N, ou n – aires).

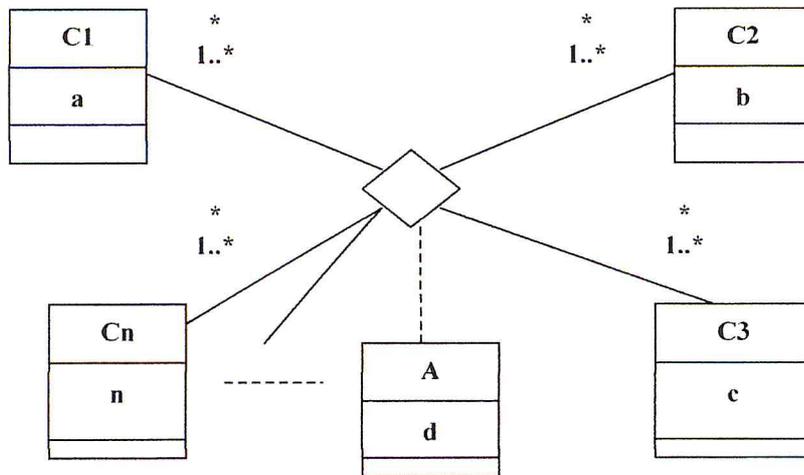
Exemple d'association 1 – N réflexive :



3.6) Transformation des associations n – aires :

Il existe n + 1 possibilités de transformation d'une associations n – aires du schéma conceptuel dans le modèle navigationnel, il est possible de privilégier chaque entité / classe qui compose l'association n – aire ou de n'en privilégier aucune.

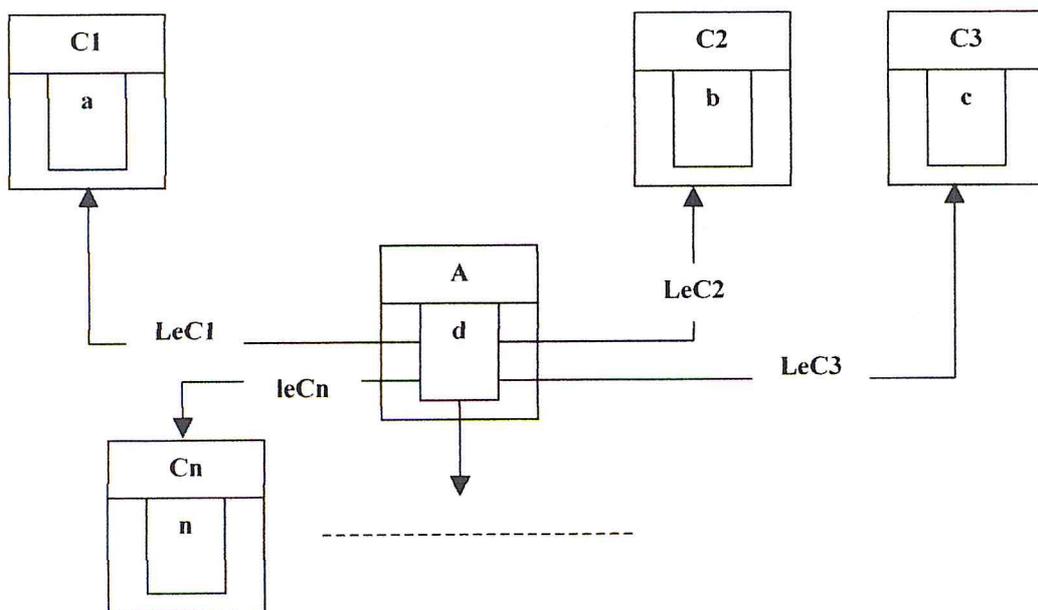




3.6.1) Première solution (solution universelle) :

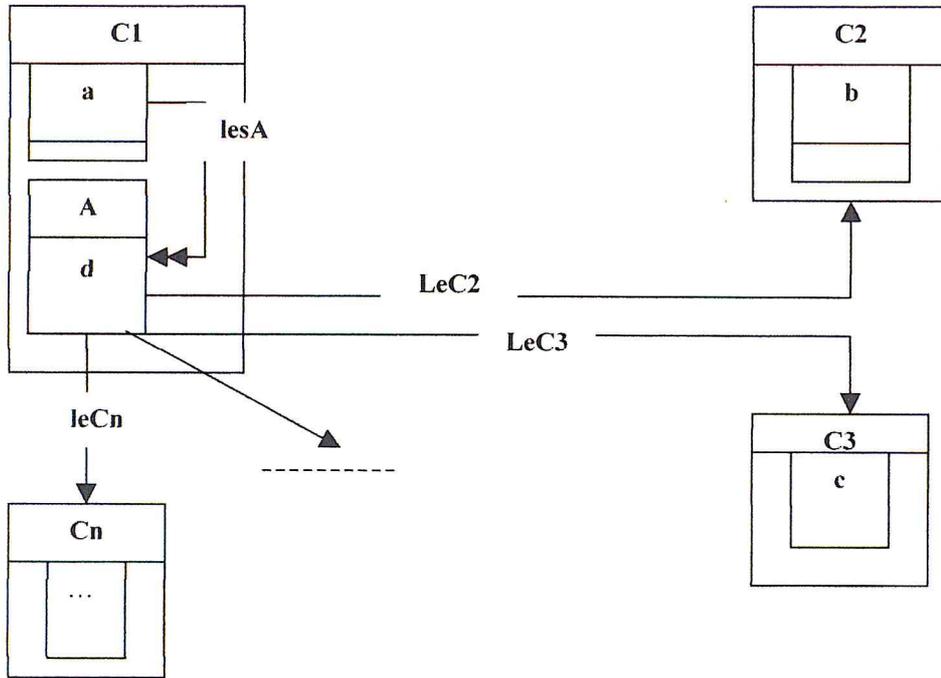
Cette solution consiste à mettre en œuvre une n+1 classe, elle porte le nom de l'association et contient :

- Les attributs de l'association.
- n liens monovalués qui la relient aux autres classes.



3.6.2) Autre solution (une structure et n liens) :

- Une structure contenant :
 - les attributs de l'association.
 - n-1 liens monovalués vers les autres classes qui composent l'association n – aires.
- Un lien multivalué dans la classe privilégiée vers la structure.



4) Bilan des transformations :

Les liens multivalués (qui traduisent par des structures imbriquées dans les tables Objet – relationnelles) privilégient l'accès d'une table par rapport à une autre, certaines requêtes qui sont a priori simples nécessitent d'être programmées (utilisation de curseurs) car elles sont difficiles à exprimer en une seule instruction SQL de type SELECT.

D'autre part, l'utilisation des liens monovalués (qui se traduisent pour certains par des pointeurs dans les tables Objet – relationnelles) simplifie l'expression des requêtes dans le langage d'interrogation, en ce sens, il est intéressant d'utiliser la solution universelle.

5) Les solutions les plus 'Relationnelles' :

Le tableau suivant indique la solution du modèle navigational la plus proche d'une solution relationnelle classique, par exemple, dans le cadre d'une association 1 – N, le modèle relationnel impose l'utilisation d'un attribut de type clé étrangère dans la table fils référençant la table père.

La solution la plus proche au niveau navigational est la troisième, c'est à dire celle qui met en œuvre un lien monovalués.

Association	Modèle navigational
1 – 1	Solution 2
1 - N	Solution 3
N – N sans propriétés	Solution 2
N – N avec propriétés	Solution 1
n-aires sans propriétés	Solution 2
n-aires avec propriétés	Solution 1

6) Mise en œuvre avec ORACLE 8:

Oracle 8 permet de définir des objets à structure complexe principalement par l'utilisation des TAD, d'identifiant OID et des tables imbriquées.

Un TAD oracle 8 peut être :

- Une ou plusieurs tables.
- Un ou plusieurs types.
- Un ou plusieurs attributs dans une ou plusieurs tables.

Type utilisé par une table :

```
CREATE TYPE adresse_type AS OBJECT  
(norue NUMBER, rue VARCHAR2(20),ville VARCHAR2(30),code_postal  
VARCHAR2(10)) ;
```

le résultats est une structure de données disponible pour la déclaration d'autres types ou de tables.

```
CREATE TABLE adresse OF adresse_type ;
```

Type utilisé par un autre type :

```
CREATE OR REPLACE TYPE adresse_type AS OBJECT  
(norue NUMBER, rue VARCHAR2(20),ville VARCHAR2(30),code_postal  
VARCHAR2(10)) ;
```

```
CREATE OR REPLACE TYPE etudiant_type AS OBJECT  
(numero NUMBER, bac VARCHAR2(5),adr adresse_type) ;
```

```
CREATE TABLE etudiant OF etudiant_type  
(CONSTRAINT PK_ETUD PRIMARY KEY (numero)) ;
```

Type utilisé par un attribut d'une table relationnelle :

```
CREATE TABLE employe  
(numero NUMBER PRIMARY KEY,  
dept VARCHAR2(5),  
salaire NUMBER,  
adr adresse_type) ;
```

Type utilisé par un attribut d'une table objet – relationnelle :

```
CREATE OR REPLACE TYPE adresse_type AS OBJECT  
(norue NUMBER, rue VARCHAR2(20),ville VARCHAR2(30),code_postal  
VARCHAR2(10)) ;
```

```
CREATE TYPE employe_type AS OBJECT  
(numero NUMBER, dept VARCHAR2(5),salaire NUMBER, adr adresse_type) ;
```

```
CREATE TABLE employe OF employe_type
(PRIMARY KEY (numero)) ;
```

Pointeur (attributs de type REF) :

Référence simple :

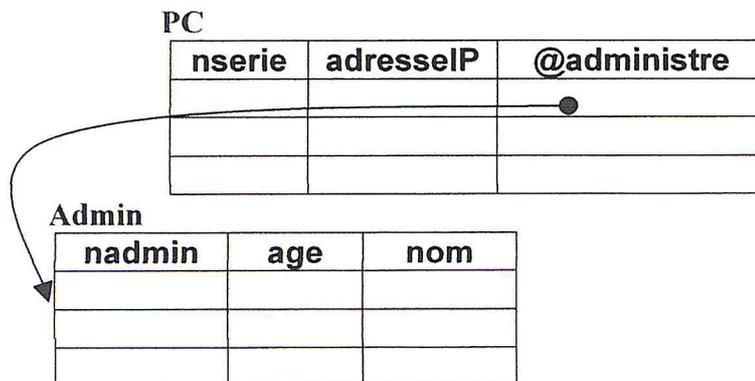
Chaque PC est géré par un administrateur (numéro, âge, nom), on choisit de faire pointer chaque PC vers son administrateur.

```
CREATE TYPE administrateur_type AS OBJECT
(nadmin VARCHAR2(5), age NUMBER, nom VARCHAR2(30)) ;
```

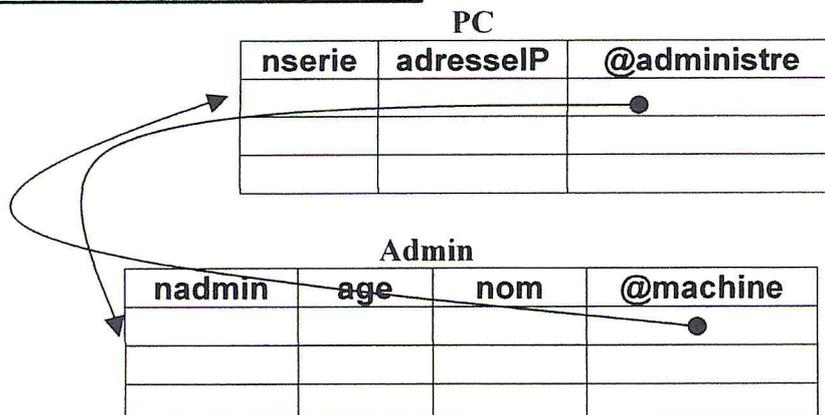
```
CREATE TYPE PC_type AS OBJECT
(nserie VARCHAR2(5), adresselP VARCHAR2(15),
administre REF administrateur_type) ;
```

déclaration des tables :

```
CREATE TABLE admin OF administrateur_type
(CONSTRAINT PK_admin PRIMARY KEY (nadmin)) ;
CREATE TABLE PC of PC_type (PRIMARY KEY (nserie)) ;
```



Référence inverse physique :



Tables imbriquées :

```
CREATE OR REPLACE TYPE emp_type AS OBJECT
(ninsee VARCHAR2(13), âge NUMBER, nom VARCHAR2(30)) ;
```

```
CREATE OR REPLACE TYPE emps_type AS TABLE OF emp_type
```

```
CREATE OR REPLACE TYPE departement_type AS OBJECT
(numdep VARCHAR2(11), budget NUMBER , employes emps_type) ;
```

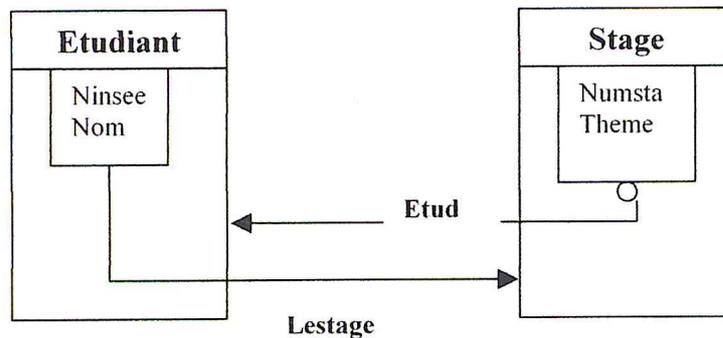
```
CREATE TABLE Département OF departement_type
(PRIMARY KEY (numdep)) ;
NESTED TABLE employes STORE AS tabemp ;
```

Département :

numdep	budget	{employes}		
		ninsee	age	nom

Du Navigationnel à ORACLE 8 :

Exemple Générale (relation 1 – 1) :



```
CREATE TYPE etudiant_type :
```

```
CREATE OR REPLACE TYPE stage_type AS OBJETC ;
(numstage NUMBER, theme VARCHAR2(15), etud REF etudiant_type) ;
```

```
CREATE OR REPLACE TYPE etudiant_type AS OBJECT
(ninsee VARCHAR2(13), nom VARCHAR2(30), lestage REF stage_type) ;
```

```
CREATE TABLE Etudiant OF etudiant_type
(PRIMARY KEY(ninsee),
CONSTRAINT lestage_nn Lestage NOT NULL) ;
```

```
CREATE TABLE stage OF stage_type
(PRIMARY KEY (numstage),
Constraint ref_etud etud SCOPE IS etudiant) ;
```

Annexe 3

Les commandes de base de FTP

Introduction :

FTP: File Transfert Protocol (Protocole de transfert de fichiers)

Le FTP sert à faire transiter des fichiers. Il est utile pour prendre des données, mais aussi pour en mettre sur un serveur. C'est le protocole qui est utilisé pour déposer des pages HTML sur un serveur web (l'hébergeur). C'est un protocole très ancien quasiment universel, et les clients sont disponibles pour toutes les plateformes.

Pour utiliser le FTP, il faut que la machine distante fasse tourner un serveur ftp. En principe, ce programme ouvre le port 21 et attends les connections.

Les commandes de bases de FTP :

Cd : Change de répertoire distant

Close : Termine la session ftp

Dir : Affiche le contenu du répertoire distant

Get : Télécharger un fichier distant

Hash : active l'affichage progressif de l'état du transfert

Help : Obtenir de l'aide

Lcd : Changer de répertoire local

Lls : Afficher les fichiers du répertoire local

ls : Afficher les fichiers du répertoire distant

mkdir : Créer un répertoire sur le serveur distant

mget : Télécharger plusieurs fichiers en une seule commande

mput : Envoyer plusieurs fichiers en une seule commande

open : Connection ftp distante

put : Envoyer un fichier

pwd : Obtenir de chemin actuel sur la machine distante

rename : Renommer un fichier

rm : Effacer un fichier distant

rmdir : Supprimer un répertoire distant



Glossaire

GLOSSAIRE

ASCII

Acronyme de : American Standard Code for Information Interchange.

CG

Champ Générique de ticket de taxation.

CUS

Champ Utile de ticket de taxation.

CV

Circuits Virtuels.

CVC

Circuits Virtuels Commutés.

CVP

Circuits Virtuels Permanents.

DBA

En anglais : Data base Administrator. Administrateur de la base de données.

FTP

En anglais : File Transfert Protocole. Protocole de transfert de fichiers entre station de travail.

RFC

En anglais : Request For Comments. Rédigés par des experts, ces documents à caractère technique font autorité sur l'Internet, car dûment motivés et rendus publics seulement après discussion et accord entre diverses parties.

SVR

Serveur de supervision.

TT

Ticket de taxation.

UML

Langage destinée aux phases amont de la réalisation d'un logiciel, UML est une technique de modélisation unifiée issue de méthodes plus anciennes comme OMT et OOD.

URL

En anglais : Uniform Resource Locator, méthode d'identification d'une ressource Internet (serveur, fichier, groupe de discussion ...) et le moyen d'y accéder Ex : <http://www.ajeep.com//index.php> indique d'aller rechercher la page indexe.html du serveur ajeep.com à l'aide de protocole de transfert http.

Bibliographie

Bibliographie :

[01] : Christian Soutou, " Objet – relationnel sous Oracle 8 Modélisation avec UML", Mars 1999.

[02] : Cay S.Horstmann, Gary Cornell, " Au cœur de JAVA 2 Volume 1", CampusPress 2000.

[03] : W.David Lockman, "Le Programmeur ORACLE 8 DBA ", campuspress Juin 2001.

[04] : Antoine Mirecourt, Pierre – Yves Saumont, " JAVA 2 SDK 1.3" , Basic micro Edition 2001.

[05] : N.Kanaoui. "Architecture logicielle : les langages de description d'architecture", Université de Nantes, Septembre 2002.

[06] : D.Garlan et M.shaw, "An introduction to Software Architecture, Software Engineering & knowledge Engineering Vol II, 1998.

RFC :

[RFC01] : Nadia Kanaoui, Tahar Khammaci et Mourad Oussaleh, IRIN, *Université de Nantes* " Les ADLs : une voie prometteuse pour les architectures logicielles".

[RFC02] : Tahar Khammaci, Adel Smeda et Mourad Oussalah, *Université de Nantes*, " Une architecture logicielle à base de composants et d'objets pour la description de systèmes complexes ".

[RFC03] : Michel Nolard, Facultés Universitaires Notre-Dame de la Paix, Institut d'Informatique, "Intégrer l'Architecture d'une Application avec son Implémentation grâce à ArchJava"

[RFC04] : Projet ACCORD, "Etat de l'art sur les Langages de Description d'Architecture (ADLs)", Juin 2002.

[RFC05] : Ke Zheng, Lin Liao, Tool Evaluation Report, "ArchJava Evaluation Report", février 2002.

[RFC06] : Jonathan Aldrich, Craig Chambers, David Notkin, Department of Computer Science and Engineering, University of Washington, "Component-Oriented Programming in ArchJava".

[RFC07] : Andrei Alexandrescu, Konrad Lorincz, "ArchJava: An Evaluation".

[RFC08] : Jonathan Aldrich, Craig Chambers, David Notkin, Department of Computer Science and Engineering University of Washington, " Architectural Reasoning in ArchJava".

[RFC09] : Michel Gardie, "La couche réseau : Le protocole X.25" , 20 décembre 2004.

[RFC10] : Frédéric Peschanski, Thomas Meurisse, Jean-Pierre Briot, Laboratoire d'Informatique de Paris 6, " Les Composants Logiciels : Evolution technologique ou nouveau paradigme ?".

Webographie :

[URL01] : <http://www.archjava.org>

[URL02] : <http://www.developpez.com>

[URL03] : <http://www.uml.free.fr>

[URL04] : <http://vlemiere.nerim.net/X25.htm>