

SOMMAIRE

REMERCIEMENTS

RÉSUMÉ

LISTE DES ABRÉVIATIONS

LISTE DES FIGURES

INTRODUCTION 1

Chapitre I GESTION DE FLUX DE DONNÉES DANS UN SYSTÈME

D'INFORMATION 2

I.1. SYSTÈME D'INFORMATION 2

I.1.1. Définitions 2

a. Flux de données

b. Système d'information

I.1.2. Classification des systèmes d'information 3

a. Les systèmes de traitement des transactions

b. Les systèmes d'information et de gestion

c. Les systèmes d'aide à la décision

d. Les systèmes d'information pour dirigeants

I.1.3. Gestion de flux d'information 4

a. Acteurs

b. Identifier les flux de données

c. Contrôler les données

d. Cartographie des bases de données

e. Traçabilité des informations

I.2. SYSTÈME D'INFORMATION GÉOGRAPHIQUE 9

I.2.1. Définitions 9

I.2.2. La composition d'un SIG 10

<i>I.2.3. Les fonctionnalités d'un SIG.....</i>	10
<i>I.2.4. Les avantages d'un SIG</i>	11
<i>I.2.5. Les données géographiques</i>	11
<i>I.2.6. Structure de l'information géographique</i>	12
<i>I.2.7. Modes de représentation de l'information géographique</i>	12
I.3. PROJECTION LABORDE	13
<i>I.3.1. Les projections cartographiques.....</i>	14
<i>I.3.2. Systèmes de coordonnées géographiques</i>	14
<i>I.3.3. Les systèmes de projections.....</i>	16
<i>I.3.4. Les différents systèmes de projection en fonction de la surface de projection utilisée</i>	16
a. Projections coniques	
b. Projections cylindriques	
c. Projection planaire	
Chapitre II MÉTHODOLOGIE ET ENVIRONNEMENT DE CONCEPTION.....	19
II.1. CONCEPT OBJECT	19
<i>II.1.1. Mots-clés</i>	20
<i>II.1.2. Illustration des concepts de classe et d'objet</i>	21
<i>II.1.3. Construction et destruction d'un objet</i>	22
<i>II.1.4. Le principe d'encapsulation.....</i>	23
<i>II.1.5. Le principe de polymorphisme</i>	23
<i>II.1.6. Le principe d'héritage</i>	24
a. Définition	
b. Remarque(s)	
II.2. MODELISATION D'UN SYSTEME.....	25
<i>II.2.1. Analyses d'un système</i>	25
a. Pourquoi	

b. Quoi	
c. Comment	
<i>II.2.2. La méthode d'analyse et de conception d'objet : UML.....</i>	28
a. Les diagrammes	
II.3. ENVIRONNEMENT JAVA ET JAVA 2D.....	35
<i>II.3.1. Introduction</i>	35
<i>II.3.2. Classes et Interfaces de base de l'API Java2D.....</i>	36
<i>II.3.3. Le système de coordonnées en Java2D.....</i>	37
II.4. TRAITEMENTS DE SHAPE	38
<i>II.4.1. Notions de géométrie</i>	38
a. Principaux objets de la géométrie plane	
b. Les systèmes de coordonnées dans le plan	
<i>II.4.2. Généralités sur les transformations du plan</i>	40
a. Les translations	
b. Les rotations	
c. Les symétries	
d. Les homothéties	
<i>II.4.3. Changement de repère par translation</i>	41
<i>II.4.4. Transformation de coordonnées WGS 84 en Laborde Madagascar</i>	42
a. Calcul des paramètres de projection	
b. Transformation de coordonnées géographiques en coordonnées planes dans le système de projection Gauss-Laborde	
c. Transformation de coordonnées planes en projection Gauss-Laborde, en coordonnées géographiques	
Chapitre III OUTIL DE GESTION DE SHAPE DANS LA PROJECTION LABORDE	47
III.1. ANALYSE ET CONCEPTION.....	47
<i>III.1.1. Spécification du problème</i>	47

<i>III.1.2. Analyse des besoins et des contraintes</i>	47
<i>III.1.3. Conception.....</i>	49
a. Décomposition du système	
<i>III.1.4. Modélisation du système</i>	50
a. Diagramme des use case	
b. Diagramme de séquences	
c. diagramme de classe	
III.2. PRESENTATION DE L'OUTIL	65
<i>III.2.1. Gestion des catalogues des données</i>	65
a. Connexion à la base de données	
b. Gestion des listes des tables	
<i>III.2.2. Outil de traitement de carte</i>	69
a. Connexion à la base de données	
b. Création d'un nouveau projet	
c. Création des groupes de tables	
d. Ajout d'une table dans un groupe de table	
e. Habillage d'un shape	
CONCLUSION.....	74
RÉFÉRENCES	75

REMERCIEMENTS

Nous remercions le seigneur Dieu, qui nous a donné la santé, la force, et l'intelligence jusqu'à ce jour. Sans Lui, notre existence et nos efforts ne sont que rien.

*Nous remercions Monsieur **ANDRIANARY Philippe**, Directeur de l'Ecole Supérieure Polytechnique d'Antananarivo (ESPA).*

*Nous adressons notre reconnaissance et notre gratitude à Monsieur **RATSIMBA Mamy Nirina**, chef de Département Electronique à l'ESPA.*

*Surtout, nous tenons encore à exprimer particulièrement notre reconnaissance à Monsieur **RAZAFINDRAIBE Tovohery ANDRIAMPAMONJY Alain**, enseignant à l'ESPA, qui malgré ses préoccupations, a remarquablement eu la volonté de contribuer à notre encadrement.*

Notre remerciement et notre gratitude s'adresse également :

- à Madame **RABEHERIMANANA Lyliane Irène** qui a honoré de sa présence pour présider les membres de jury de ce mémoire
- aux membres de jury constitués par :
 - Monsieur **RASTEFANO Elisée**
 - Monsieur **RATSIMBAZAFY Guy Predon Claude**
 - Monsieur **HERINANTENAINA Edmond Fils**
- à tous les membres du corps enseignant du même Département qui n'ont pas ménagé leurs efforts, en nous partageant leurs connaissances tout au long de ces cinq années d'études.

Aussi, nous portons nos vifs remerciements à toute notre famille pour leur soutien moral et financier, et à toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce mémoire.

RÉSUMÉ

Officiellement à Madagascar, la projection Laborde représente la base de calcul et de représentation de la cartographie. La définition et la méthode de construction de cette projection présentent une singularité qui pose des contraintes majeures quant à son emploi dans les logiciels de Géomatique. Rares sont les logiciels qui intègrent dans son fichier de configuration la projection Laborde de Madagascar.

Le but de notre étude est de concevoir un logiciel SIG flexible à la projection Laborde. La réalisation de la version 1.0 du logiciel TANY SY NY TONTOLONY (TsT 1.0) est assurée par le langage de programmation orienté objet (Java) et un système de gestion et conception de base de données (MySQL).

LISTE DES ABRÉVIATIONS

API	Application Programming Interface
FTM	Foiben-Taosarintanin'i Madagasikara
GPS	Global Positioning System
IGN	Institut Géographique National
JDK	Java Development Kit
JRE	Java Runtime Environment
JVM	Java Virtual Machine
MERISE	Méthode d'Etude et de Réalisation Informatique des Systèmes d'Entreprise
OOD	Object Oriented Design
OMT	Object Modeling Technique
OOSE	Object Oriented Software Engineering
PQC	Pourquoi Quoi Comment
SGBD	Système de Gestion de Base de Données
SGBDR	Système de Gestion de Base de Données Relationnelles
SI	Système d'Information
SIG	Système d'Information Géographique
SQL	Structured Query Language)
UML	Unified Modeling Language
WGS 84	World Geodetic System of 1984
XML	eXtensible Markup Language

LISTE DES FIGURES

<i>Figure I.1:</i> Modèle de flux	2
<i>Figure I.2:</i> Représentation système-flux-acteurs externes	5
<i>Figure I.3:</i> Représentation système-flux-acteurs internes	5
<i>Figure I.4:</i> Flux d'information	6
<i>Figure I.5:</i> Diagramme de flux de facturation	7
<i>Figure I.6:</i> Les constituants d'un SIG	10
<i>Figure I.7:</i> Structure de l'information géographique	12
<i>Figure I.8:</i> Le monde sous forme de globe affichant les valeurs de longitude et de latitude ..	15
<i>Figure I.9:</i> Parallèles et méridiens constituant un graticule	15
<i>Figure I.10:</i> Projection conique	17
<i>Figure I.11:</i> Les différents types de projections cylindriques	17
<i>Figure I.12:</i> Projection planaire	18
<i>Figure II.1:</i> Représentation d'un objet	19
<i>Figure II.2:</i> Représentation d'une classe	21
<i>Figure II.3:</i> Illustration de la construction et la destruction d'un objet	22
<i>Figure II.4:</i> Schématisation	23
<i>Figure II.5:</i> Diagramme des relations de parenté qui existe entre différentes classes	24
<i>Figure II.7:</i> Éléments constitutifs des cas d'utilisation	29
<i>Figure II.6:</i> Les vue et diagrammes d'UML	29
<i>Figure II.8:</i> Diagramme des cas d'utilisation	30
<i>Figure II.9:</i> Exemple des relations d'inclusion entre client et guichet automatique	31
<i>Figure II.10:</i> Représentation d'une relation d'extension entre client et guichet automatique ..	31
<i>Figure II.11:</i> Partie 1 du diagramme de séquence « retrait en espèce »	32
<i>Figure II.12:</i> Partie 2 du diagramme de collaboration (retrait en espèce)	32
<i>Figure II.13:</i> Représentation d'une classe	33
<i>Figure II.14:</i> Association sous forme binaire (seulement entre deux classes)	33
<i>Figure II.15:</i> Exemple d'une classe d'association	34
<i>Figure II.16:</i> Exemple d'une association d'agrégation	34
<i>Figure II.17:</i> Classes et interfaces de base de l'API Java 2D	36
<i>Figure II.18:</i> Le système de coordonnées en Java2D	37
<i>Figure II.19:</i> Exemples d'objets de la géométrie plane	39

<i>Figure II.20:</i> Coordonnées cartésiennes	40
<i>Figure II.21:</i> Coordonnées polaires	40
<i>Figure III.1:</i> Analyse des besoins et des contraintes.....	48
<i>Figure III.2:</i> Système de gestion des shapes.....	49
<i>Figure III.3:</i> Diagramme des use case de l'outil SIG	51
<i>Figure III.4:</i> Diagramme de séquence pour créer des tables	52
<i>Figure III.5:</i> Diagramme de séquence pour modifier les tables.....	53
<i>Figure III.6:</i> Diagramme de séquence pour supprimer les tables	54
<i>Figure III.7:</i> Diagramme de séquence pour créer un projet.....	55
<i>Figure III.8:</i> Diagramme de séquence pour modifier le projet	56
<i>Figure III.9:</i> Diagramme de séquence pour supprimer le projet.....	57
<i>Figure III.10:</i> Diagramme de séquence pour tracer les informations	58
<i>Figure III.11:</i> Diagramme de séquence pour modifier les informations	59
<i>Figure III.12:</i> Diagramme de séquence pour supprimer les informations	60
<i>Figure III.13:</i> Diagramme de séquence pour rechercher les données géographiques	61
<i>Figure III.14:</i> Diagramme de classes de l'outil SIG	64
<i>Figure III.15:</i> Logo de TsT 1.0	65
<i>Figure III.16:</i> Fenêtre principale du catalogue	66
<i>Figure III.17:</i> Fenêtre de connexion à la base	66
<i>Figure III.18:</i> Fenêtre de gestion des listes des tables	67
<i>Figure III.19:</i> Liste des tables	67
<i>Figure III.20:</i> Gestion de table.....	68
<i>Figure III.21:</i> Fenêtre de modification d'un champ	68
<i>Figure III.22:</i> Fenêtre principale de la cartographie	69
<i>Figure III.23:</i> Fenêtre de connexion à la base	70
<i>Figure III.24:</i> Choix de créer ou d'ouvrir un projet.....	70
<i>Figure III.25:</i> Fenêtre de création d'un projet.....	71
<i>Figure III.26:</i> Création d'un nouveau groupe.....	71
<i>Figure III.27:</i> Affichage de groupe de tables.....	72
<i>Figure III.28:</i> Ajout d'une table dans un groupe.....	72
<i>Figure III.29:</i> Affichage d'une table dans un groupe avec traçage des shape.....	73
<i>Figure III.30:</i> Fenêtre d'habillage de shape	73

INTRODUCTION

La connaissance de l'environnement et ses contraintes permet de donner des bonnes gestions. Le système d'information géographique a la vocation de fournir aux hommes cette opportunité sur tout ce qui se passe sur cette planète bleu. De nos jours le SIG est le produit de l'évolution combiné des besoins en information géographique et des progrès de l'informatique. Il existe déjà des logiciels géomatiques qui appliquent des projections les plus connus telles que l'UTM, WGS84, Lambert,...Or, rares ceux qui intègrent dans son fichier de configuration la projection Laborde conçu spécialement pour Madagascar.

C'est dans cette optique que le thème de ce mémoire a été choisi : « **Modélisation et conception des outils SIG compatibles au système de projection Laborde** ». Ce travail a pour objectif de concevoir un logiciel qui intègre seulement la projection Laborde de Madagascar.

Alors que nous avons présenté l'idée générale de notre projet, nous allons présenter la démarche que nous avons envisagée en indiquant le contenu des différentes parties qui constituent cet ouvrage.

Le premier chapitre concerne la gestion de flux de données dans un système d'information ainsi que le système géographique qui conduit à la représentation de la projection Laborde. Dans le deuxième chapitre, nous décris la méthodologie et l'environnement de conception qui englobent toutes les étapes en commençant par le concept objet pour avoir une réflexion menée de la vie et de la qualité du logiciel. Ensuite un environnement de développement Java et pour finir le traitement shape. Enfin, le dernier chapitre est consacré à la conception de l'outil de gestion de shape dans la projection Laborde.

Chapitre I GESTION DE FLUX DE DONNÉES DANS UN SYSTÈME D'INFORMATION

Il y a quelques années la mise en place de SIG paraissait réservée aux grandes structures. Aujourd’hui les progrès informatiques et les possibilités offertes en matière de gestion et d’analyse permettent la généralisation de l’outil à tous les échelons du territoire. Cependant la mise en place du SIG est loin d’être une entreprise facile, car cela nécessite l’élaboration d’un système d’information .En effet, notre analyse sera focalisée sur la connaissance du système d’information, notamment le SIG et la projection LABORDE.

I.1. SYSTÈME D'INFORMATION

Le système d’information est constitué par l’ensemble des flux des données, il a pour but de gérer l’information. La définition du système d’information dépend surtout de la compréhension du flux des données. D'où la nécessité de l'étude de flux des données.

I.1.1. *Définitions*

a. Flux de données

Un flux est un transfert d’informations entre composants du système. Le composant peut être un domaine, une activité ou un acteur externe.

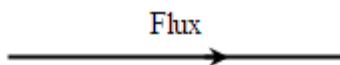


Figure I.1: Modèle de flux

Comme le montre la figure I.1, dans les modèles de flux, un flux de données est représenté graphiquement par une flèche orientée du composant émetteur du flux vers le composant récepteur.

b. Système d’information

De nombreuses définitions ont été proposées pour les systèmes d’information. Voici quelques définitions :

- Le SI est un ensemble organisé de ressource (personnel, données, procédures, matériel, logiciel,...) permettant d’acquérir, de stocker, de structurer et de communiquer des informations sous forme de textes, images, sons, ou de données codées dans des organisations. Le système d’information coordonne grâce à l’information les activités de l’organisation et lui permet ainsi

d'atteindre ses objectifs. Il est le véhicule de la communication dans l'organisation.

- Le SI est aussi un ensemble des éléments participant à la gestion, au stockage, au traitement, au transport et à la diffusion de l'information au sein d'une organisation.
- Le SI représente l'ensemble des ressources d'une organisation (les hommes, le matériel, les logiciels) organisés pour :
 - Collecter les informations
 - Stocker les informations
 - Traiter les informations
 - Diffuser les informations
- C'est une représentation de l'activité du système opérant et/ou du système de pilotage et de ses échanges avec l'environnement.
 - Le système de pilotage décide des actions à conduire sur le système opérant en fonction des objectifs et des politiques de l'entreprise.
 - Le système opérant englobe toutes les fonctions liées à l'activité propre de l'entreprise : facturer les clients, régler les salariés, gérer les stocks.
 - L'entreprise est un système dans lequel transitent de très nombreux flux d'informations.

A fin de gérer efficacement les actifs en technologies de l'information, il faut créer un portrait des applications disponibles. Ce portefeuille d'applications permet de supporter la gouvernance des technologies de l'information. Sans une classification des systèmes d'information, le portefeuille de la technologie de l'information se résume souvent en une liste d'applications triées par départements, ou dans le pire des cas, à une liste alphabétique de solution.

I.1.2. Classification des systèmes d'information

Une typologie qui a été proposée pour réaliser une classification des systèmes d'information en quatre familles. Nous y trouvons donc : les systèmes de traitement des transactions, les systèmes d'information et de gestion, les systèmes d'aide à la décision et les systèmes d'information pour dirigeants.

a. Les systèmes de traitement des transactions

Les systèmes de traitement des transactions, ou systèmes transactionnels sont des systèmes qui servent à collecter l'ensemble des événements important de la vie de l'entreprise. Nous associons donc à cette catégorie des systèmes qui créent de la donnée.

b. Les systèmes d'information et de gestion

Les systèmes d'information et de gestion fournissent aux gestionnaires des rapports de synthèse, ainsi que des accès directs aux données permettant d'obtenir des renseignements. Nous associons donc à cette catégorie des systèmes qui regroupent des données et les structures pour des fins de représentation et d'interrogation.

c. Les systèmes d'aide à la décision

Les systèmes d'aide à la décision supportent des décisions semi-structurées par des modèles descriptifs pour réaliser des portraits de la situation et prescriptif pour essayer de prévoir l'avenir. Nous associons donc à cette catégorie des systèmes qui regroupent des fonctions de simulations et d'analyses.

d. Les systèmes d'information pour dirigeants

Les systèmes d'information pour dirigeants sont utilisés pour prendre de meilleures décisions au niveau stratégique de l'organisation par l'intégration de données externes à l'organisation et par la synthèse des données internes. Nous associons à cette catégorie des systèmes qui fournissent des tableaux de bord et des indicateurs de gestion.

I.1.3. Gestion de flux d'information

Pour gérer le flux d'information, il est nécessaire de maîtriser les cinq points suivants :

- Acteurs
- Les flux de données
- Contrôle des données
- Les outils pour stocker les données
- La traçabilité des données

a. Acteurs

Les acteurs sont représentés par un cercle libellé par le nom de l'acteur [1]. Il représente une unité active intervenant dans le fonctionnement d'un système opérant. Il peut :

- être stimulé par des flux d'information
- transformer et émettre des flux d'information

Un acteur « fait quelque chose », il est actif (exemples : Service comptabilité, guichet,...).

Un acteur est un rôle plutôt qu'une personne physique (« Direction » et pas « Jean-Claude ») Il peut être pertinent de modéliser séparément deux fonctions assumées par une même personne physique. On représente respectivement par la figure I.2 et la figure I.3 ci-dessous les acteurs externes (ou partenaires) et les acteurs internes (ou domaines).

- **acteurs externes** : ce sont les éléments externes avec lesquels le système échange des flux d'information (exemples : clients, fournisseurs, ...).

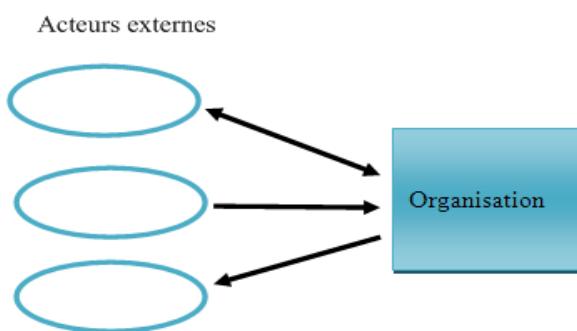


Figure I.2: Représentation système-flux-acteurs externes

- **acteurs internes** : qui font partie du système d'information étudié (exemples : guichet, service informatique, ...) et si le système est complexe, on peut considérer un acteur interne comme un sous-domaine et détailler ce sous-domaine dans un nouveau modèle conceptuel de communication.

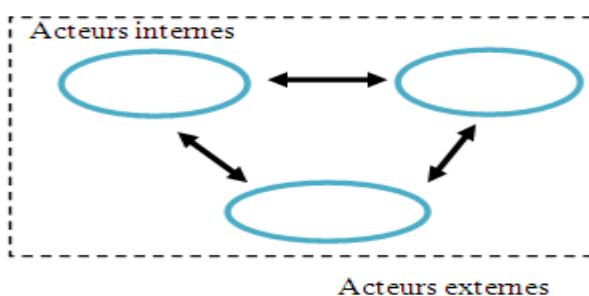


Figure I.3: Représentation système-flux-acteurs internes

Le flux d'information visualisé à la figure I.4 est représenté par une flèche entre deux acteurs, étiquetée par le nom du flux et échange d'informations entre deux acteurs[1].

Exemple : documents, appels téléphoniques, données informatiques

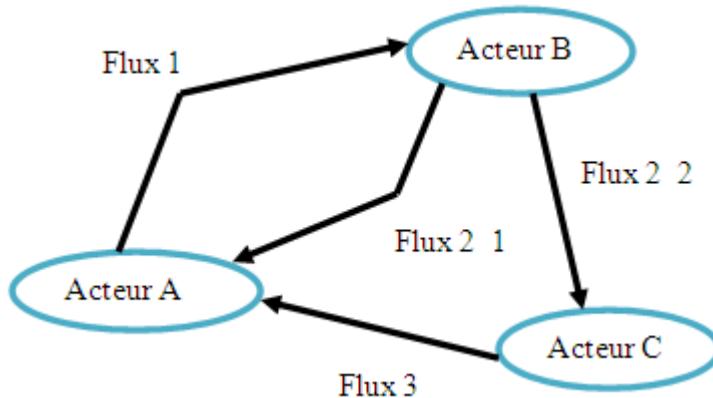


Figure I.4: Flux d'information

b. Identifier les flux de données

Pour déterminer les contrôles à mettre en place, il est nécessaire d'avoir une vue globale de la circulation des données tout au long de la chaîne des traitements, allant de la saisie initiale des données jusqu'à leur archivage final, en passant par leurs modifications et leurs mises à jour, leurs stockages, leurs éditions,

Cette vue globale découle, en grande partie, d'une analyse des processus; mais celle-ci, au lieu de s'attacher à détailler les traitements, se concentre sur les données. Elle a pour objectif d'identifier l'ensemble des bases de données mises en œuvre, et les opérations qui sont faites pour transférer les données d'une base à l'autre.

Pour illustrer le cas d'une application commerciale. Comme le montre la figure I.5, on commence par saisir les commandes des clients, et on constitue une première base de données comprenant les bons de commande. Après contrôle des informations saisies par rapport à la base produits, un premier traitement permet de créer les bons de livraison. Une fois le client livré, un second traitement établit les factures, qui sont ensuite déversées dans la comptabilité clients. Une fois qu'elles sont soldées, ces écritures sont archivées à fin de contrôle. Comme on le voit, cette circulation des données n'est pas simplement un déversement des données d'une base dans l'autre mais un enchaînement de contrôles, de traitements et d'agrégations de données.

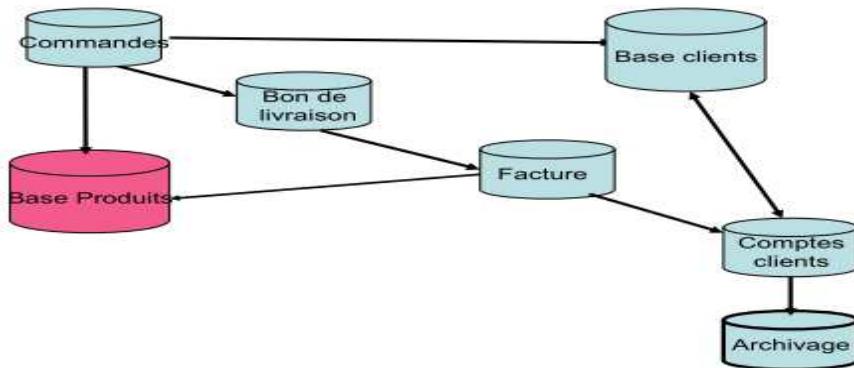


Figure I.5: Diagramme de flux de facturation

c. Contrôler les données

L'ensemble des données doit être contrôlé tout au long de cette circulation. Le niveau des contrôles est déterminé en fonction des niveaux des risques estimés. Il faut prévoir suffisamment de contrôles sans pour autant verser dans une prolifération excessive. Par exemple, dans le cas d'une base de données clients, il faut avant tout veiller à ne pas avoir de doublon et évacuer le plus vite les clients inactifs depuis plusieurs années. Par contre, il ne sert à rien de vérifier les numéros de téléphone ou de télécopie. Les données sont normalement contrôlées au moment de la saisie. Le but est de détecter des anomalies comme des codes erronés, des totaux inexacts, des oubli significatifs, ... Les concepteurs veillent à mettre tous ces contrôles en début de traitement de façon à ne pas avoir ultérieurement de rejets. Par exemple, dans le cas d'une facturation, il ne faut pas rejeter une commande au moment de l'édition. Les contrôles doivent avoir été faits précédemment. Mais ce n'est pas suffisant. Il est ensuite nécessaire d'effectuer des contrôles lors des différents traitements pour s'assurer que les opérations se déroulent normalement. Dans le cas d'une application commerciale, on va, par exemple, s'assurer que :

- toutes les commandes saisies ont donné lieu à l'émission d'un bon de livraison,
- toutes les livraisons faites se sont traduites par le transfert de ces données vers la facturation,
- toutes les factures sont justifiées par un ou plusieurs bons de commandes,
- toutes les factures émises sont déversées en comptabilité clients,
- toutes les écritures soldées ayant plus de X mois d'ancienneté sont archivées .

De même, les principales bases de données, comme les bases clients et produits, doivent être périodiquement contrôlées pour s'assurer que :

- tous les clients et tous les produits existent,
- toutes les mises à jour faites sur ces bases sont tracées : ceci concerne les données saisies, mais aussi les transferts de données, les mises à jour des bases au cours des traitements ,
- les bases de données sont exhaustives : il faut être capable de détecter des disparitions accidentelles ou volontaires de données,
- l'intégrité des données stockées a été protégée, et par exemple certains cumuls n'ont pas pu être faussés à la suite de fausses manœuvres,
- le chaînage des informations entre différentes bases est complet et fonctionne correctement, L'expérience montre ici encore que les contrôles actuels ne sont pas suffisants et qu'il est nécessaire de les renforcer.

d. Cartographie des bases de données

Pour expliquer et analyser les incohérences potentielles, il est nécessaire de disposer d'un document identifiant les principales bases de données de l'entreprise et les relations qu'elles ont entre elles. Il ne s'agit pas du modèle de données logique mais bien du modèle physique. Ce schéma doit faire apparaître un certain nombre d'informations fondamentales pour maîtriser le système :

- la localisation des données : serveur ou système disque où les données sont stockées
- le volume de la base : nombre d'occurrences, taille théorique, taille réelle
- le type de sauvegarde des bases et la périodicité des sauvegardes
- l'éventuelle recopie des données des bases sur le système disque
- la journalisation des mises à jour, en indiquant leur lieu de stockage
- la duplication ou la synchronisation des données.

Le but de la cartographie est de faciliter la connaissance opérationnelle et la localisation des bases de données et de s'assurer que les sauvegardes sont suffisantes compte tenu du niveau de risque.

e. Traçabilité des informations

L'objectif est de retrouver une information ou une donnée à partir d'une autre, issue d'une application informatique logiquement et physiquement éloignée. Il est ainsi possible,

par exemple, de remonter d'un compte du bilan au détail des comptes puis aux écritures et, le cas échéant, aux pièces justificatives. Il faut pour cela mémoriser le parcours suivi par une information. C'est ce que permet le « chemin de révision » aussi appelé « piste d'audit » ou « audit trail ». Ce parcours doit être enregistré à l'aide de pointeurs ou d'index. Il peut aussi être inscrit sur la « carte d'identité » de la donnée. Il est aussi possible de stocker ces données dans un datawarehouse ou entrepôt de données. Pour retrouver les données, il est nécessaire de disposer de logiciels permettant de remonter des cumuls vers les détails justificatifs. Le défaut de chemins de révision doit être considéré comme une fragilité certaine des procédures de contrôle interne.

En conclusion, la maîtrise des données est un point fondamental de la démarche de contrôle interne.

I.2. SYSTÈME D'INFORMATION GÉOGRAPHIQUE

En général le SIG peut acquérir plusieurs acceptations, néanmoins, elle devrait avoir sa propre définition technique, compte tenu de sa spécificité et ses caractéristiques.

I.2.1. *Définitions*

De nombreuses définitions ont été proposées pour les systèmes d'information géographiques, traduisant leurs nombreuses facettes et accentuant l'aspect tantôt architectural tantôt fonctionnel. En fait, les SIG possèdent des dimensions informationnelles, fonctionnelles et humaines [2], [3]. Voici quelques unes de ces définitions :

- Le SIG est un système informatique de matériels, de logiciels et de processus conçus pour permettre la collecte, la gestion, la manipulation, l'analyse, la modélisation et l'affichage des données à référence spatiale (données géoréférencées ou encore données correspondant à une localisation absolue par rapport à un système de référence ou à une localisation relative d'un objet par rapport à un autre) afin de résoudre des problèmes complexes et de gestion.
- Un système d'information géographique est un système d'information capable d'organiser et de présenter des données alphanumériques spatialement référencées, ainsi que de produire des plans et des cartes. Ses usages couvrent les activités géomatiques de traitement et diffusion de l'information géographique. L'usage courant du système d'information géographique est la représentation plus ou moins réaliste de l'environnement spatial en se basant

sur des primitives géométriques : points, des vecteurs (arcs), des polygones ou des maillages (raster).

L'information géographique peut être définie comme l'ensemble de la description d'un objet et de sa position géographique à la surface de la terre. L'IG est composée de :

- La localisation est la position à la surface de la Terre par rapport à un système explicite et définit également la géométrie et la forme.
- Sémantique ou information descriptive : description d'un objet ou d'un phénomène par sa nature, aspect et attribut.

I.2.2. La composition d'un SIG

Un SIG est l'ensemble de matériels, des logiciels, des données, des personnes et des compétences mises en place pour analyser un territoire. La composition d'un SIG est illustrée par la Fig. I.6 ci-dessous [4].

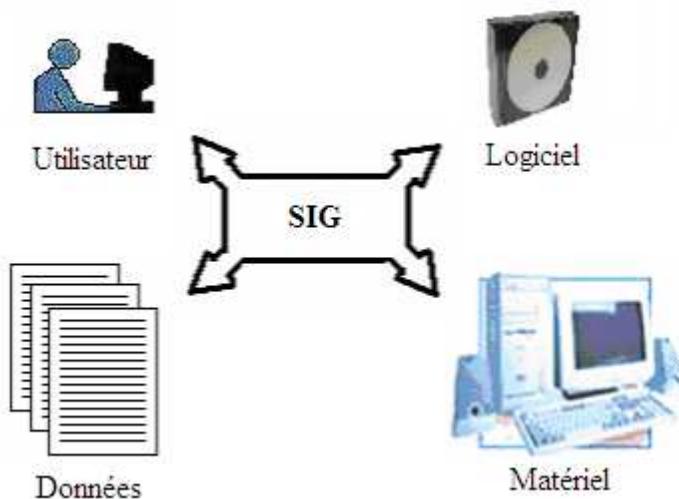


Figure I.6: Les constituants d'un SIG

I.2.3. Les fonctionnalités d'un SIG

Un SIG offre la possibilité de saisir des données géolocalisées, les stocker et les réorganiser, les modéliser, les analyser et aussi les représenter sous forme alphanumérique ou graphique [4].

Entre autres, un SIG assure comme fonctions :

- La conversion d'un système de coordonnées à un autre.
- La disposition des objets dans un système géoréférencé.
- La possibilité de superposition des cartes de sources différentes.

- L'extraction de tous les objets géographiques situés à une distance bien déterminée d'un objet spécifique.
- La fusion des objets ayant une caractéristique commune.
- La détermination du plus court itinéraire le pour se rendre à une destination donnée [5].

I.2.4. Les avantages d'un SIG

Les Systèmes d'Informations Géographiques apportent un ensemble de bien faits, à savoir :

- Le stockage des informations est clair et définitif.
- La compréhension des phénomènes et la prévention des risques.
- L'établissement des cartographies rapides.
- La localisation dans l'espace et dans le temps.
- La réaction rapide face aux catastrophes.
- La fourniture des itinéraires [5].
- L'exploitation d'une grande quantité d'informations.
- Le tri par la superposition sur des couches distinctes.
- La possibilité d'ajouter des étiquettes de texte.
- La lisibilité de l'affichage [4].

I.2.5. Les données géographiques

Les SIG manipulent des données géographiques. On entend par données géographiques une représentation ou une reformulation de la réalité localisée dans l'espace et le temps. C'est en fait la combinaison des données sémantiques en gérant des données alphanumériques spatialement localisées et des données géométriques [4].

L'information géographique est représentée habituellement sous forme cartographique. Elle est composée de deux composantes :

- Une composante graphique : il s'agit de la carte décrivant la forme et les caractéristiques d'un objet en déterminant ses coordonnées cartographiques.
- Une composante attributaire: il s'agit de la légende identifiant les objets représentés en donnant leurs significations.

En se basant sur des primitives graphiques comme les points, les lignes, les arcs et les polygones, les SIG permettent de réaliser une représentation réaliste de l'environnement concret [5].

I.2.6. Structure de l'information géographique

Comme l'indique le paragraphe précédent, l'information géographique est formée par deux types d'informations: les informations spatiales et les informations alphanumériques. Les deux types jouent deux rôles indispensables à la représentation des informations géographiques, mais chacun a sa manière. En effet, les données spatiales sont organisées en couches, par contre les données alphanumériques sont structurées en base de données. L'ensemble forme une base de données géographique qui peut être définie comme étant un ensemble de couches superposées accompagnées par des étiquettes de texte qui y fournissent du sémantique, illustré par la Fig. I.7[5].

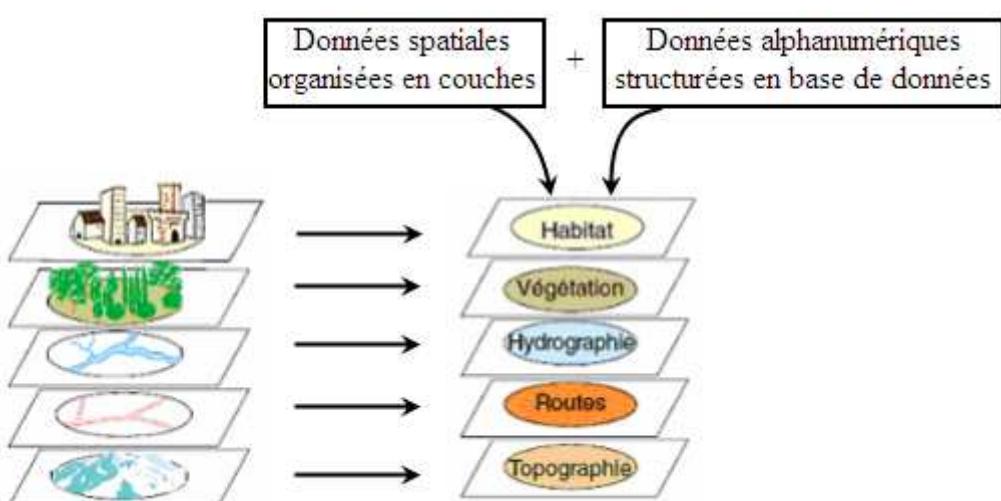


Figure I.7: Structure de l'information géographique

I.2.7. Modes de représentation de l'information géographique

Il existe deux modes possibles pour représenter une information géographique au niveau d'un SIG : le mode vectoriel et le mode raster [4].

- **Le mode vectoriel:** les limites des objets spatiaux sont décrites à travers leurs constituants élémentaires, à savoir les points, les arcs et les polygones. Chaque objet spatial est doté d'un identifiant qui permet de le relier à une table attributaire.
- **Le mode raster** (appelé aussi mode maillé): la réalité est décomposée sur une grille régulière et rectangulaire, organisée en lignes et en colonnes, chaque maille de cette grille ayant une intensité de gris ou une couleur. La juxtaposition des points donne une apparence visuelle. Un objet donné est représenté par un ensemble de pixels de la même intensité.

Le mode vectoriel fait appel à un ensemble de concepts porteurs d'informations comme :

- Les points : ils définissent la localisation d'un élément isolé de taille très petite qui n'a pas une surface réelle pour être représenté par une ligne ou une surface.
- Les lignes : elles représentent la forme d'objet géographique trop étroite pour être décrite par une surface.
Exemples : une route ou une rivière.
- Les polygones : ils représentent la forme et la localisation d'objet homogène.
Exemples : un gouvernorat ou une région.

I.3. PROJECTION LABORDE

Dans chaque pays, il existe une projection officielle fixée par l'Etat et qui devient la projection de référence. A Madagascar, cette projection appartient à la catégorie des projections conformes (conserve les angles) et nomme Laborde Madagascar. Très proche de la projection de type Mercator Oblique, elle est bâtie à partir de l'ellipsoïde Hayford International 1924 dont les caractéristiques sont :

- demi-grand axe : $a = 6378388$ mètres
- demi-petit axe : $b = 6356911,9461$ mètres
- aplatissement : $f = 0,003367003367$
- excentricité : $e = 0,08199188998$
- centre de la projection :
 - latitude : 21 gr Sud
 - longitude : 49 gr Est Paris
 - coefficient de réduction d'échelle : 0,9995
 - coordonnées de l'origine : $E_0 = 400\,000$ mètres
- $N_0 = 800\,000$ mètres
- tolérance de convergence : $\varepsilon = 1.10^{-11}$ d' après l'IGN

Utilisée exclusivement pour Madagascar, elle n'est malheureusement pas prise en compte par les principaux logiciels SIG ni par les GPS. Cette situation pose aujourd'hui aux techniciens en Géomatique et aux professionnels des questions techniques et méthodologiques car les formules de passage vers cette projection demeurent relativement compliquées.

I.3.1. Les projections cartographiques

Une projection est un système de correspondance entre les points de la surface à représenter et les points d'une surface plane, dite surface de projection. A chaque point de la surface à représenter doit correspondre un point et un seul de la surface projetée. En cartographie, la surface à représenter est la terre tandis que la surface de projection peut être un cône, un plan ou un cylindre.

Effectuer une projection va donc consister à établir un système de correspondance entre les points à la surface de la terre (coordonnées géographiques) et les points sur la surface de projection (coordonnées cartographiques).

I.3.2. Systèmes de coordonnées géographiques

Les deux points de référence sur la terre sont les deux pôles. En reliant ces derniers nous obtenons l'axe imaginaire de la planète. Les plans passant par cet axe et la surface de la terre ont la forme d'une ellipse et sont appelés les « méridiens ». Les plans perpendiculaires à l'axe de rotation avec la surface sont appelés les « parallèles », le plus grand est celui de l'équateur. Un point quelconque de la surface terrestre se détermine donc par ses deux coordonnées géographiques: la latitude et la longitude, qui sont représentées par la Fig. I.8. Leur unité de mesure est le grade ou le degré. Les coordonnées sont déterminées par rapport à l'équateur (latitude d'origine) et au méridien d'origine (longitude). Le méridien international est celui de Greenwich, mais il existe aussi le méridien de Paris pour certaines cartes françaises. Un point peut être aussi défini par son altitude calculée à partir du niveau moyen de la mer.

Dans le monde, on utilise comme système de coordonnées de référence le « World Geodetic System of 1984 » abrégé en « WGS 84 », il est bâti à partir de l'ellipsoïde dont les caractéristiques sont :

- demi-grand axe : $a = 6378137$ mètres
- demi-petit axe : $b = 6356752,314$ mètres
- aplatissement : $f = 0,003352810665$
- excentricité : $e = 0,08181919106$

C'est un modèle géocentrique, c'est-à-dire que le centre d'ellipsoïde correspond (environ) au centre de masse de la terre.

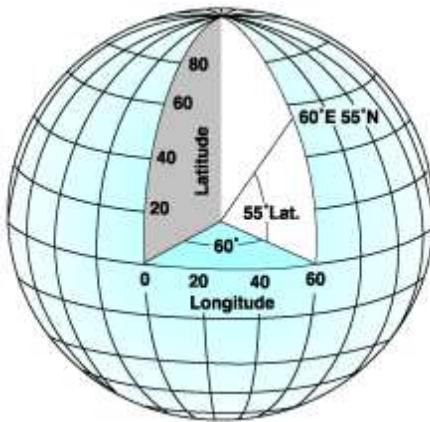


Figure I.8: Le monde sous forme de globe affichant les valeurs de longitude et de latitude

Dans le système sphérique, les « lignes horizontales » ou lignes Est-Ouest, sont des lignes de latitude ou des parallèles. Les « lignes verticales » ou lignes Nord-Sud sont des lignes de longitude ou des méridiens. Ces lignes ceinturent le globe et constituent un réseau quadrillé appelé un graticule, comme le montre la Fig. I.9 ci-dessous.

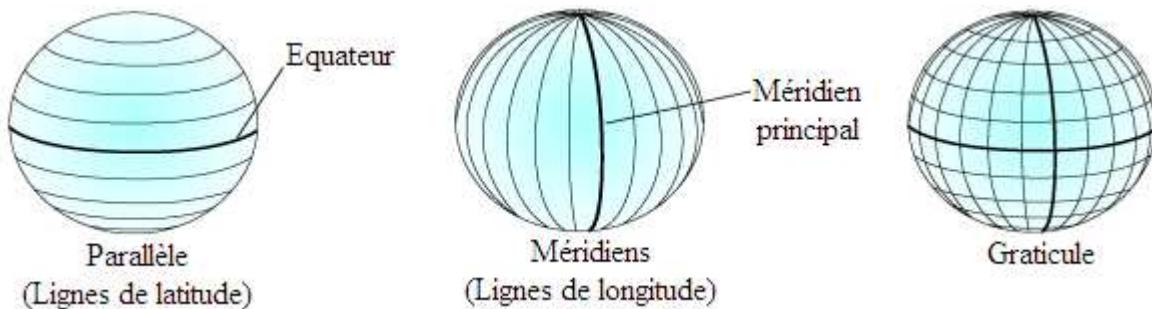


Figure I.9: Parallèles et méridiens constituant un graticule

L'origine du graticule (0,0) est définie d'après le point d'intersection de l'équateur et du méridien principal. Le globe est alors divisé en quatre quadrants géographiques calculés d'après les relèvements au compas effectués à partir de l'origine. Le Nord et le Sud se trouvent au-dessus et au-dessous de l'équateur, l'Est et l'Ouest se situant à gauche et à droite du méridien principal. Les valeurs de latitude sont mesurées par rapport à l'équateur et sont comprises entre -90° au pôle Sud et $+90^\circ$ au pôle Nord. Les valeurs de longitude sont mesurées par rapport au méridien principal. Elles vont de -180° lorsqu'on va vers l'Ouest jusqu'à 180° lorsqu'on va vers l'Est.

Une fois que l'on possède un modèle de globe terrestre, pour la plupart des applications, il est nécessaire de le projeter sur un plan à deux dimensions afin de le visualiser sur une carte ou sur un écran d'ordinateur. Il existe aussi dans cette partie de nombreuses techniques qui ont chacune pour but de respecter certaines propriétés présentes à la surface de la terre [6].

I.3.3. Les systèmes de projections

La planète terre est une sphère légèrement aplatie aux deux pôles du fait de la rotation et représente un renflement équatorial. La transformation du globe terrestre vers une surface plane ne peut s'effectuer sans déformations. Il est dès lors important de définir une projection réduisant au minimum les déformations des longueurs, des angles et des surfaces. La projection est un procédé graphique qui permet de représenter sur un plan de réseau des méridiens et des parallèles afin de pouvoir reporter n'importe quel point terrestre. En général les systèmes de projections permettent de passer de coordonnées géographiques vers les coordonnées planes et vice-versa. Les principaux systèmes existant sont les projections coniques et conformes. On dit qu'une projection conforme si les angles sont préservés, elle est équivalente si elle conserve les surfaces, équidistante si elle conserve les distances sur les méridiens. Si aucune de ces propriétés n'est respectée, il s'agit de projections aphylactiques. Les plus utilisés étant certainement la projection conique de Lambert et la projection de Mercator. D'autres sont tout de même exploitées telles que les projections cylindriques, les projections elliptiques ou les projections azimutales.

I.3.4. Les différents systèmes de projection en fonction de la surface de projection utilisée

Les cartes étant planes, certaines des projections les plus simples sont réalisées sur des formes géométriques qui peuvent être aplatis sans étirer leur surface. Elles sont appelées surfaces développables. De nombreuses projections cartographiques courantes sont classifiées selon la surface de projection utilisée : conique, cylindrique, ou planaire (azimutale) [6].

a. Projections coniques

La Figure I.10 montre la projection conique la plus simple. Elle est tangente au globe le long d'une ligne de latitude. Cette ligne est appelée le parallèle standard. Les méridiens sont projetés sur la surface conique, et se rejoignent au sommet, ou point, du cône. Les lignes parallèles de latitude sont projetées sur le cône en anneaux. Le cône est ensuite « découpé » le long d'un méridien pour obtenir la projection conique finale, qui a des lignes droites convergentes pour les méridiens et des arcs concentriques pour les parallèles. Le méridien se trouvant à l'opposé de la ligne de découpe devient le méridien central.

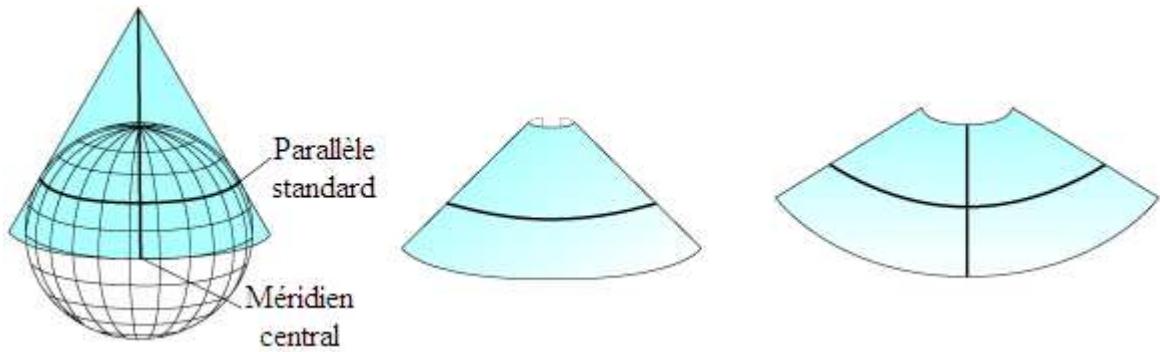


Figure I.10: Projection conique

b. Projections cylindriques

Lorsque la surface de projection est un cylindre tangent à la sphère (la terre est assimilée à une sphère), il s'agit d'une projection cylindrique. Si le cylindre est tangent à la sphère au niveau de l'équateur, on parle de projection cylindrique directe ; si le cylindre est tangent au niveau des pôles, on est alors dans le cas d'une projection cylindrique transverse. Une fois le cylindre développé, le méridien central et l'équateur sont transformés en deux droites orthogonales. Les autres méridiens et parallèles sont transformés en courbes orthogonales entre elles. C'est la projection la plus employée dans le Monde pour l'établissement des cartes topographiques car elle couvre toute la surface terrestre entre les latitudes 80° Nord et 80° Sud. Tous ces différents types de projections cylindriques sont illustrés par la Fig. I.11 ci-dessous.

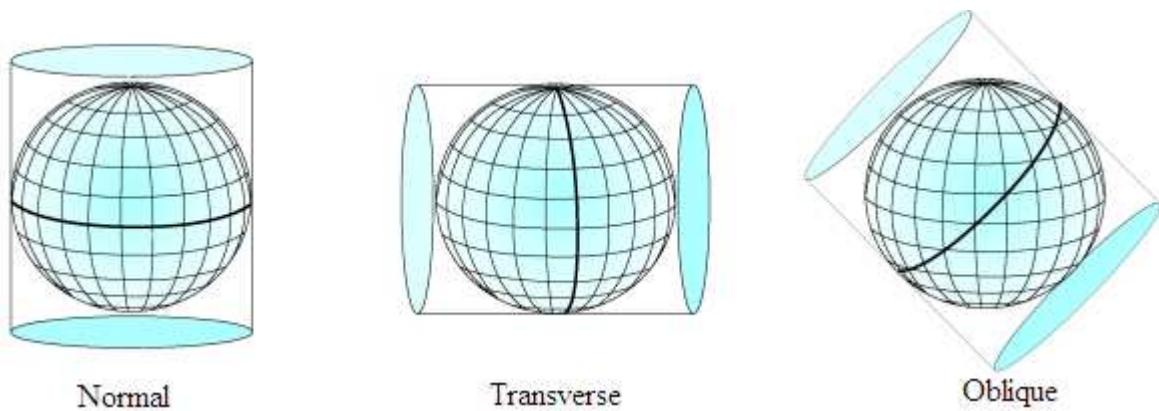


Figure I.11: Les différents types de projections cylindriques

c. Projection planaire

Comme le montre la figure I.12, les projections planaires projettent les données cartographiques sur une surface plane touchant le globe. Une projection planaire est également appelée projection azimutale ou projection zénithale. Ce type de projection est généralement tangent au globe en un point mais peut être également sécant. Le point de contact peut être le pôle Nord, le pôle Sud, un point sur l'équateur ou tout autre point intermédiaire. Ce point définit l'aspect et le point central de la projection. Le point central est défini par une longitude centrale et une latitude centrale. Un aspect peut être polaire, équatorial, ou oblique.

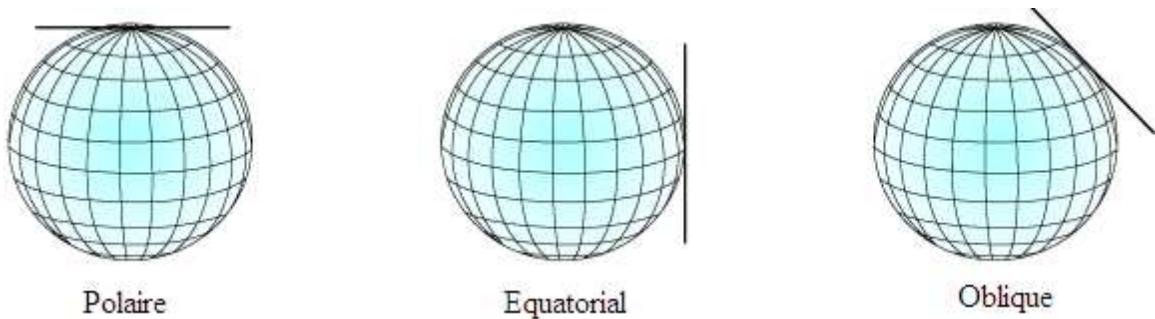


Figure I.12: Projection planaire

Chapitre II MÉTHODOLOGIE ET ENVIRONNEMENT DE CONCEPTION

La conception par objet(se basant sur les données) trouve ses fondements dans une réflexion menée autour de la vie et la qualité du logiciel qui peut être atteinte à travers certains critères : la validité, l'extensibilité, la réutilisable, la robustesse. Vu la nécessité de promouvoir l'abstraction de données, la modularité et la réutilisabilité des composants fabriqués qui conduisent à la nécessité d'une méthode : UML qui a des méthodes objets (orienté objet), ses techniques d'analyse et de conceptions appropriées. Le codage demande un langage de programmation, comme le JAVA, qui suit le concept de base et une approche orientée objet. Un système de gestion de base de données est utile pour ses fonctions qui sont décomposées en trois parties à savoir : le stockage, le calcul et la visualisation des données saisies.

II.1. CONCEPT OBJECT

Un objet représenté par la figure II.1 est une entité autonome, qui regroupe un ensemble de propriétés (données) cohérentes et de traitements associés [7].

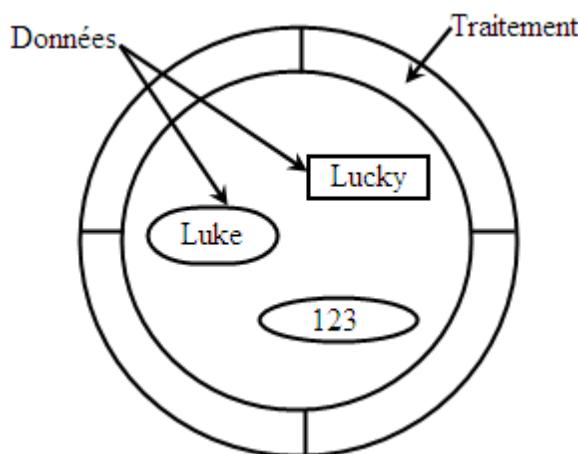


Figure II.1: Représentation d'un objet

Les objets ne sont que des modèles de référence. Les classes sont des descripteurs d'objets, les objets sont les agents effectifs et "vivants" implantant les actions d'un programme. Les objets dans un programme ont une vie propre :

- ils naissent (ils sont créés ou alloués)
- ils agissent (ils s'envoient des messages grâce à leurs méthodes)
- ils meurent (ils sont désalloués, automatiquement en Java)

C'est dans le segment de mémoire de la machine virtuelle Java que s'effectue l'allocation et la désallocation d'objets. Le principe d'allocation et de représentation des objets en Java est une encapsulation de la notion de pointeur.

Il faut entendre par objet, élément de l'univers relatif au problème à traiter. Les objets appartiennent à des catégories appelées classes, qui divisent cet univers.

II.1.1. Mots-clés

- Les structures de données dénies dans l'objet sont appelés ses attributs (propriétés).
- Les procédures et fonctions dénies dans l'objet sont appelés ses méthodes (opérations).
- Les attributs et méthodes d'un objet sont appelés ses membres.
- L'ensemble des valeurs des attributs d'un objet à un instant donné est appelé état interne.
- L'ensemble des méthodes d'un objet accessibles de l'extérieur (depuis un autre objet) est appelé interface. Elle caractérise le comportement de l'objet.
- L'invocation d'une méthode d'interface est appelé appel de méthode. Elle peut être vue comme un envoi de message entre objet.

Les attributs (ou plus exactement leur représentation informatique) et les méthodes sont cachés : c'est le principe d'encapsulation. Ainsi, le programme peut modifier la structure interne des objets ou leurs méthodes associées sans avoir d'impact sur les utilisateurs de l'objet.

Un objet peut appartenir à plus d'**un type** : c'est le **polymorphisme** ; cela permet d'utiliser des objets de types différents là où est attendu un objet d'un certain type. Une façon de réaliser le polymorphisme est le sous-typage (**appelé aussi héritage de type**) : on raffine un type-père en un autre type (« **le sous-type** ») par des restrictions sur les valeurs possibles des attributs. Ainsi, les objets de ce sous-type sont conformes avec le type père.

Un **type** : dans la programmation par objet, chaque objet est typé. Le type définit la syntaxe (appellation) et la sémantique (ce qu'il fait) des messages auxquels peut répondre un objet. Il correspond donc, à peu de chose près, à l'interface de l'objet.

II.1.2. Illustration des concepts de classe et d'objet

Le monde qui nous entoure est constitué de très nombreux objets. Pour comprendre le monde, l'être humain a tendance à regrouper les éléments qui se ressemblent. Regrouper des objets suivant des critères de ressemblance s'appelle classe. Les humains ont classé les animaux les plantes, les champignons, les atomes, ...

Comme le montre la figure II.2, la classe est une description abstraite d'un ensemble d'objets. Elle peut être vue comme la factorisation des éléments communs à un ensemble d'objets [7]. La classe regroupe la définition des membres de classe, c'est-à-dire :

- des méthodes, les opérations que l'on peut effectuer
- des champs, les variables que l'on peut traiter
- des constructeurs, qui permettent de créer des objets
- et encore d'autres choses plus particulières

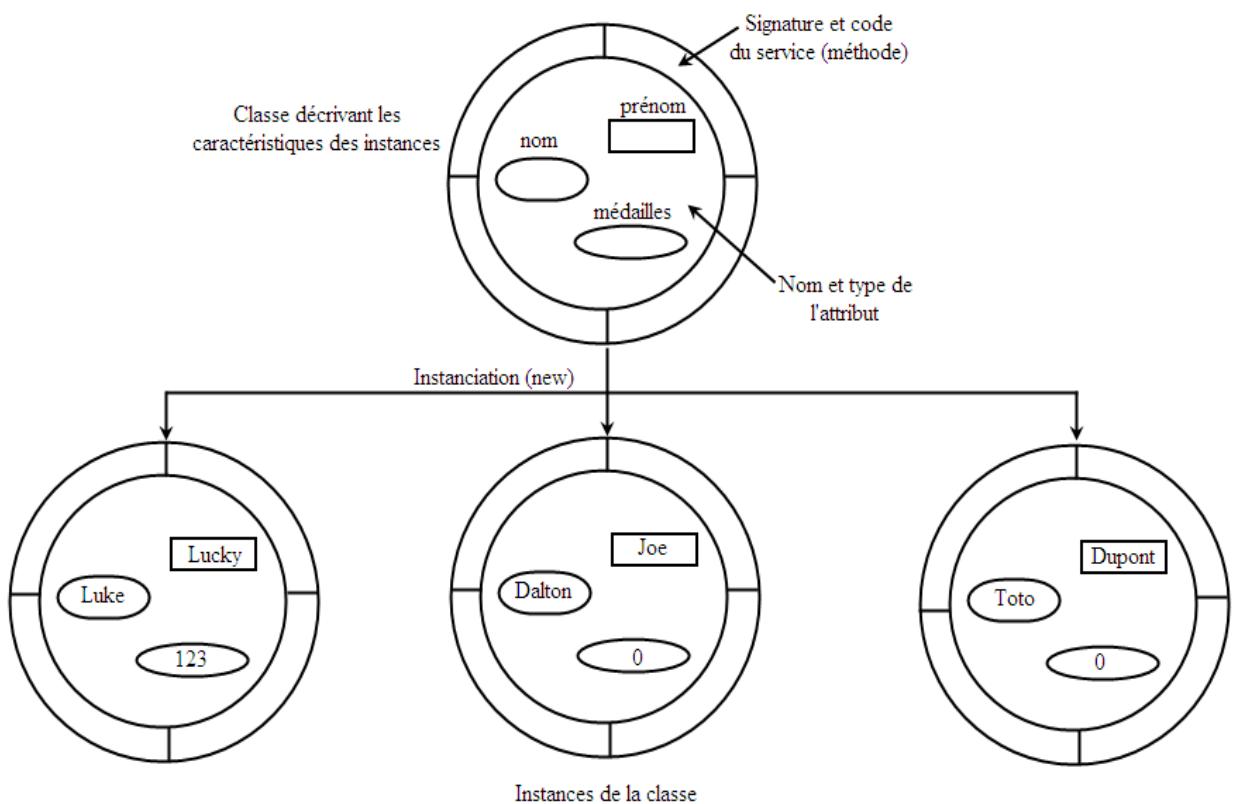


Figure II.2: Représentation d'une classe

II.1.3. Construction et destruction d'un objet

Instants particulièrement importants ; la naissance d' un objet : appelée instanciation (construction), et la mort d'un objet : appelée destruction qui est illustré par la Fig. II.3 ci-dessous [7].

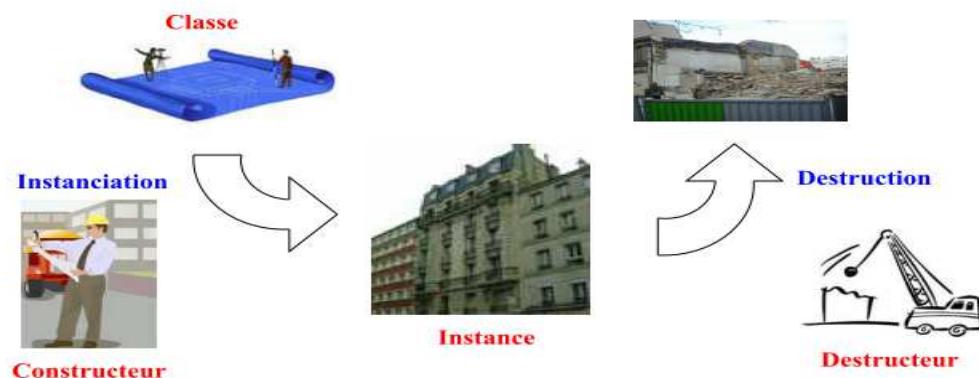


Figure II.3: Illustration de la construction et la destruction d'un objet

Le constructeur est la méthode qui permet de créer des instances dont les traits sont décrits par la classe.

Le destructeur est la méthode qui permet de détruire une instance de la classe.

Il est important de saisir la différence entre les notions de classe et instance de la classe :

classe = attributs + méthodes + mécanismes d'instanciation + mécanismes de destruction

instance de la classe = valeurs des attributs + accès aux méthodes

L'instanciation est le mécanisme qui permet de créer des instances dont les traits sont décrits par la classe.

La destruction est le mécanisme qui permet de détruire une instance de la classe.

L'ensemble des instances d'une classe constitue l'extension de la classe.

II.1.4. Le principe d'encapsulation

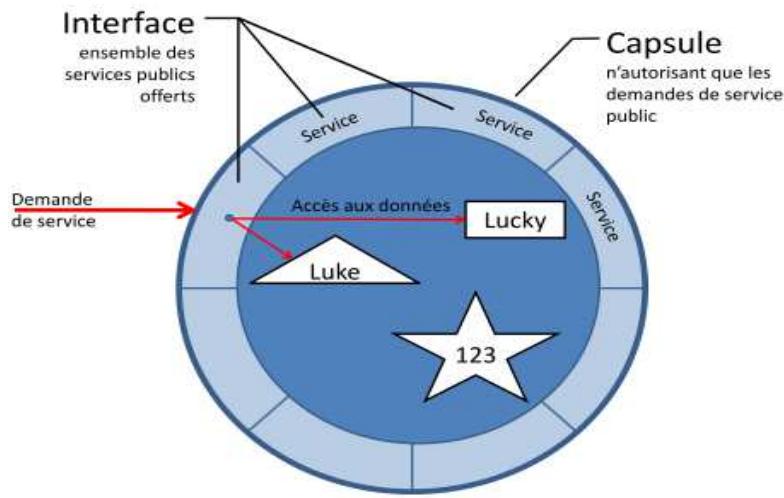


Figure II.4: Schématisation

Le terme encapsulation, que la figure II.4 schématise ci-dessus, désigne le principe consistant à cacher l'information contenue dans un objet et de ne proposer que des méthodes de modification ou accès à ces propriétés (attributs) [7].

- L'objet est vu de l'extérieur comme une boîte noire ayant certaines propriétés et ayant un comportement spécifié.
- La manière dont le comportement a été implémenté est cachée aux utilisateurs de l'objet.

N.B: Protéger la structure interne de l'objet contre toute manipulation non contrôlée, produisant une incohérence.

L'encapsulation nécessite la spécification de parties publiques et privées de l'objet.

- Éléments publics : correspond à la partie visible de l'objet depuis l'extérieur. C'est un ensemble de méthodes utilisables par d'autres objets (environnement).
- Éléments privées : correspond à la partie non visible de l'objet. Il est constitué des éléments de l'objet visibles uniquement de l'intérieur de l'objet et de la définition des méthodes.

II.1.5. Le principe de polymorphisme

Le terme polymorphisme est certainement le terme le plus difficile à définir simplement. Il désigne à la fois :

- la possibilité de spécifier le comportement ad-hoc d'une méthode selon le type d'objets l'invoquant (redéfinition ; en anglais « overriding »).

- la possibilité de définir des comportements différents pour la même méthode selon les arguments passés en paramètres (surcharge ; en anglais « overloading »).

Le polymorphisme peut être vu comme la capacité de choisir dynamiquement la méthode qui correspond au type réel de l'objet.

Exemple : Si l'on considère une classe Rectangle possédant une méthode nommée redimensionner, on peut spécifier un comportement différent selon qu'elle est invoquée avec un argument de type entier ou deux arguments de type flottant :

- redimensionner (5) multiplie la longueur de chacun des côtés par 5.
- redimensionner (1.5, 2.5) ajoute 1.5cm à la largeur et 2.5cm à la longueur.

II.1.6. Le principe d'héritage

a. Définition

L'héritage est un mécanisme de transmission des propriétés d'une classe (ses attributs et méthodes) vers une ou plusieurs sous-classes, illustré par la Fig. II.5 en dessous.

b. Remarque(s)

Un des intérêts de l'héritage est de pouvoir définir de nouveaux attributs et de nouvelles méthodes pour la classe dérivée, qui viennent s'ajouter à ceux et celles héritées. L'héritage entre les classes peut être répété : une classe A ayant hérité d'une classe B peut-elle même transmettre certaines de ses caractéristiques à une classe dérivée C.

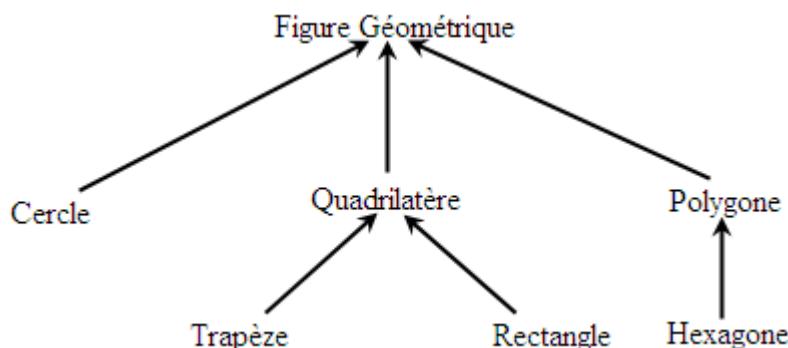


Figure II.5: Diagramme des relations de parenté qui existe entre différentes classes. Une classe peut hériter de caractéristiques de plusieurs classes, on parle alors d'héritage multiple.

N.B: Par opposition, lorsqu'une classe hérite de caractéristiques d'une seule classe, on parle d'héritage simple.

II.2. MODELISATION D'UN SYSTEME

II.2.1. Analyses d'un système

Un système est un ensemble d'éléments en interaction dynamique, dont les éléments sont organisés et coordonnés en vue d'atteindre un objectif, qui évolue dans un environnement. Pour bien comprendre un système , il faut répondre à un certain nombre de questions selon la méthode (PQC).

a. Pourquoi

Pourquoi l'ingénieur doit-il s'intéresser aujourd'hui à cette notion de « systèmes » ? Aujourd'hui, pour de nombreux systèmes, la connaissance que l'on réussit à avoir sur chacun des éléments pris les uns indépendamment des autres ne permet plus de prédire le comportement de l'ensemble c'est-à-dire du système. En d'autres termes, ces systèmes sont devenus de plus en plus complexes, selon une certaine définition de la complexité. Dans ce cas, les approches analytiques, tout particulièrement, ne permettent pas d'acquérir une vision suffisante de leur fonctionnement. Comme nous le verrons un peu plus tard, une des raisons essentielles réside dans le fait que ces approches ne permettent pas de différencier les conditions nécessaires et les conditions déclencheuses du fonctionnement d'un élément d'un système. Il y a donc nécessité de disposer d'autres approches que nous présenterons dans le paragraphe qui se nomme « *Démarches et méthodes aptes à traiter les tâches* ».

b. Quoi

Quelles sont les caractéristiques fondamentales des « systèmes » ? Plusieurs caractéristiques peuvent être considérées. Nous en évoquerons deux. La première porte sur les caractéristiques des éléments d'un tel système et la seconde concerne les caractéristiques d'ensemble de ce système.

i. Caractéristiques des éléments

Celles-ci correspondent pour un système en général à leurs caractéristiques intrinsèques, c'est-à-dire à leur composition au sens large.

Concernant les systèmes de traitement de l'information, avec l'augmentation du nombre de puces, de logiciels et d'ordinateurs dans ce type de système, il est, en effet, devenu de plus en plus difficile de tester, en pratique, l'ensemble des configurations possibles dans lesquelles peut se trouver un ordinateur particulier, compte tenu du nombre de possibilités de configurations devenu très grand de cet ordinateur et du nombre d'ordinateurs présents dans le système.

Concernant les groupes d'êtres humains, les individus, par essence même, ne sont jamais totalement prévisibles. L'homme possède en effet au moins une dimension supplémentaire par rapport à la machine : l'affectivité. Cette dimension fait qu'il possède la volonté de faire ou de ne pas faire ce qu'on lui demande et qu'il éprouve des émotions (du stress, de la satisfaction ou non suite à une activité...). L'homme n'a donc pas que des objectifs et des moyens à la différence des machines même les plus perfectionnées. Il a aussi une volonté qui résulte de ses valeurs.

ii. Caractéristiques de l'ensemble

Cela concerne le nombre et le positionnement des éléments en interaction les uns avec les autres. Les caractéristiques d'ensemble correspondent ainsi au deuxième facteur qui conditionne la manière dont les éléments interagissent entre eux.

- **Nombre d'acteurs :** plus ce nombre est grand, plus augmente la probabilité que l'on soit incapable de prédire le comportement d'un acteur quelconque à un instant donné compte tenu du grand nombre d'interactions qui existe dans le système. Il serait en effet nécessaire d'examiner alors un nombre extrêmement conséquent, et variable dans le temps, d'acteurs pour examiner toutes les interactions. Cela est souvent impossible en un temps donné, toujours limité en pratique. En d'autres termes, connaître de moins en moins le comportement d'un élément résulte d'une méconnaissance de l'ensemble des éléments qui influence elle-même le comportement d'un acteur particulier.
- **Positionnement des éléments les uns par rapport aux autres :** cet aspect peut être déterminant dans l'obtention d'un système ayant un comportement chaotique ou non c'est-à-dire difficilement prévisible en pratique.

c. Comment

i. Tâches

Les tâches s'inscrivent toutes dans une ou plusieurs phases du cycle de vie d'un produit ou d'un système industriel. La tâche a, en fait, comme finalité d'améliorer le passage du produit (au sens large) notamment d'une étape à l'autre du cycle de vie et, plus généralement, d'améliorer la circulation du produit dans le cycle en terme de sécurité, d'efficacité et/ou de confort de réalisation du travail. Concernant la sécurité, il s'agit par

exemple de diminuer les incidents et les accidents du travail effectifs ou potentiels, c'est-à-dire en gestation, dans la structure des organisations..

Concernant l'efficacité, il s'agit par exemple de détecter et de traiter préventivement les dysfonctionnements organisationnels qui sont autant de coûts cachés. de s'organiser pour agrandir une équipe ou un service pour faire face à une augmentation de la production...

Concernant le confort de réalisation, il s'agit par exemple de promouvoir des interfaces hommes-machines plus conviviales, d'améliorer l'ambiance au travail...

Cette approche par les tâches semble a priori préférable à une approche se centrant sur les problèmes. Les bibliothèques sont en effet remplies d'ouvrages qui traitent d'un problème dont la résolution n'est envisagée ni par l'auteur ni par quiconque d'ailleurs. En d'autres termes, il n'y a pas de demandes sous-jacentes, car la mise en œuvre de la résolution n'intéresse personne ou elle est impossible en l'état actuel des connaissances ou des manières d'agir. Par exemple, jusqu'à récemment, les matériels étaient conçus sans que rien ne soit demandé à l'ingénieur concepteur sur la manière dont on pourrait les réparer ou les maintenir plus généralement.

Aujourd'hui, les objets et les grands systèmes techniques sont conçus en intégrant dès le départ non seulement leur maintenance mais également leur déconstruction et leur recyclage.

ii. Démarches et méthodes aptes à traiter les tâches

Les démarches et méthodes représentent des solutions pour traiter les problèmes devant être résolus dans le cadre des tâches précédentes. Ces démarches et méthodes entrent dans l'une des deux modalités suivantes :

- Approche à référence interne, c'est-à-dire s'appuyant fondamentalement sur le propre savoir ou expertise de l'entité étudiée. Dans ce cas, on parlera le plus souvent de méthodes. Celles-ci constituent en fait des supports à une meilleure observation du fonctionnement du système. Elles constituent des outils d'analyse et de représentation ou en d'autres termes des « grammaires » de représentation. Lorsqu'elles sont utilisées dans un deuxième temps pour traiter les problèmes préalablement analysés, le traitement découle directement des aspects analyse et représentation. En effet, les actions de modification du système vont consister à rétablir ou optimiser la cohérence, par exemple de l'entreprise ou du service, par rapport à des modes de fonctionnement considérés comme étant « naturels » ou « normaux ».

- Approche à référence externe, c'est-à-dire privilégiant la cohérence externe de l'entité étudiée vis-à-vis des autres acteurs de son environnement. Dans ce cas, on parlera le plus souvent de démarches. Celles-ci proposeront d'amener l'entité étudiée vers une position estimée adéquate. Ces démarches orientent l'entité par rapport à des référentiels externes. Ceux-ci expriment ce qu'il faut faire dans tel ou tel cas. Ces référentiels externes proviennent de représentations théoriques ou de pratiques.

II.2.2. La méthode d'analyse et de conception d'objet : UML

UML (Unified Modeling Language, "langage de modélisation objet unifié") est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : OMT, OOD et OOSE [8].

Dans le cadre d'analyse, l'UML permet de représenter un système selon différentes vues qui sont indiquées sur la Fig. II.6. Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle ; c'est une perspective du modèle.

Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis) et véhicule une sémantique précise. Combinés, les différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système. Les diagrammes permettent donc d'inspecter un modèle selon différentes perspectives et guident l'utilisation des éléments de modélisation (les concepts objet), car ils possèdent une structure. Une caractéristique importante des diagrammes UML, est qu'ils supportent l'abstraction. Cela permet de mieux contrôler la complexité dans l'expression et l'élaboration des solutions objet. Pour la modélisation, l'abstraction est un des piliers de l'approche objet. Il s'agit d'un processus qui consiste à identifier les caractéristiques intéressantes d'une entité en vue d'une utilisation précise. L'abstraction désigne aussi le résultat de ce processus, plus clairement, l'ensemble des caractéristiques essentielles d'une entité, retenues par un observateur.

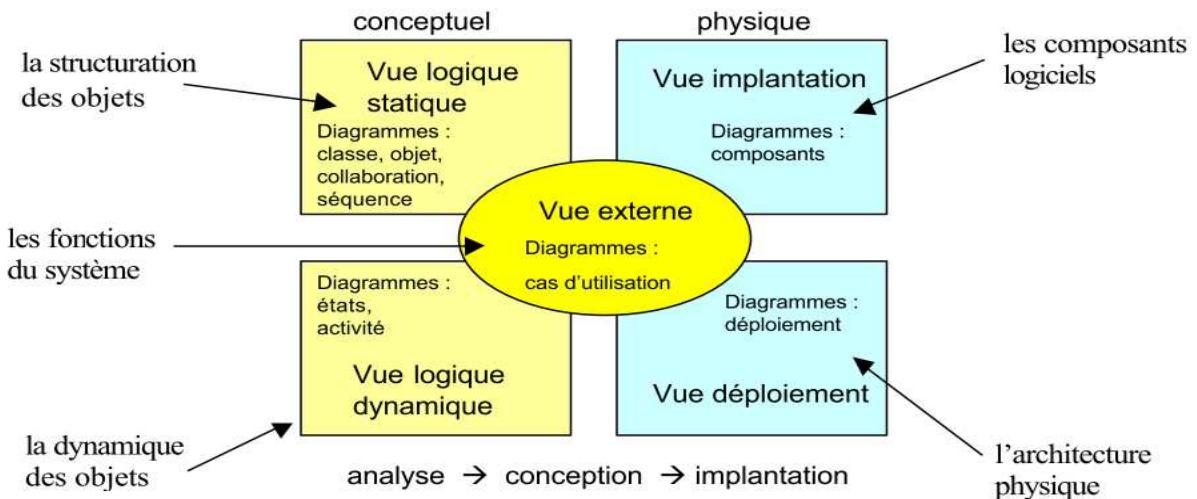


Figure II.6: Les vue et diagrammes d'UML

a. Les diagrammes

i. *Les cas d'utilisation (use case)*

Ils permettent d'effectuer une délimitation du système et de décrire son comportement.

Ils constituent aussi une représentation orientée fonctionnalités du système.

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

Dans la modélisation par les cas d'utilisation, deux concepts fondamentaux interviennent : les cas d'utilisation (utilisation du système) et les acteurs (utilisateurs du système) qui sont indiqués par la Fig. II.7 [9].

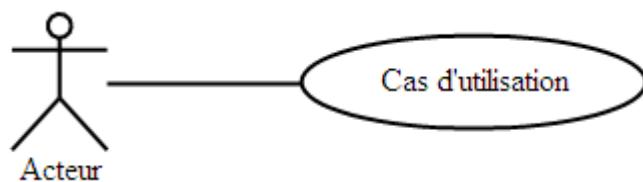


Figure II.7: Éléments constitutifs des cas d'utilisation

- Diagramme des cas d'utilisation

La Figure II.8 ci-dessous montre le diagramme de cas d'utilisation d'une « retrait en espèce

Le scénario :

- Le guichetier saisit le n° de compte du client.
- L'application valide le compte auprès du système central.
- L'application demande le type d'opération au guichetier.
- Le guichetier sélectionne un retrait d'espèces de 200F.
- Le système “ guichetier ” interroge le système central pour s'assurer que le compte est suffisamment approvisionné.
- Le système central effectue le débit du compte.
- Le système notifie au guichetier qu'il peut délivrer le montant demandé.

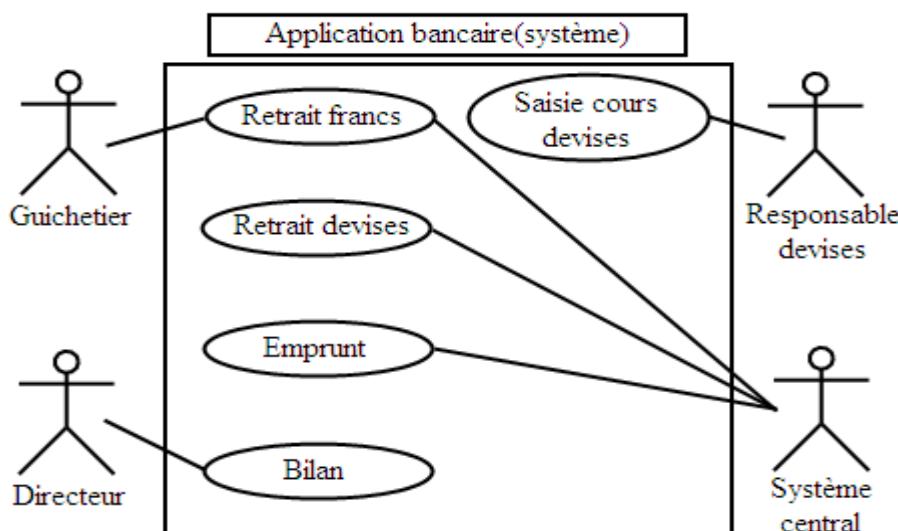


Figure II.8: Diagramme des cas d'utilisation

- Structurations des cas d'utilisation

- La relation d'inclusion

Un cas A inclut un cas B si le comportement décrit par le cas A inclut le comportement du cas B : le cas A dépend de B. Lorsque A est sollicité, B l'est obligatoirement, comme une partie de A. Cette dépendance est symbolisée par le stéréotype << include >>. La Figure II.9 montre un exemple des relations d'inclusion entre client et guichet automatique.

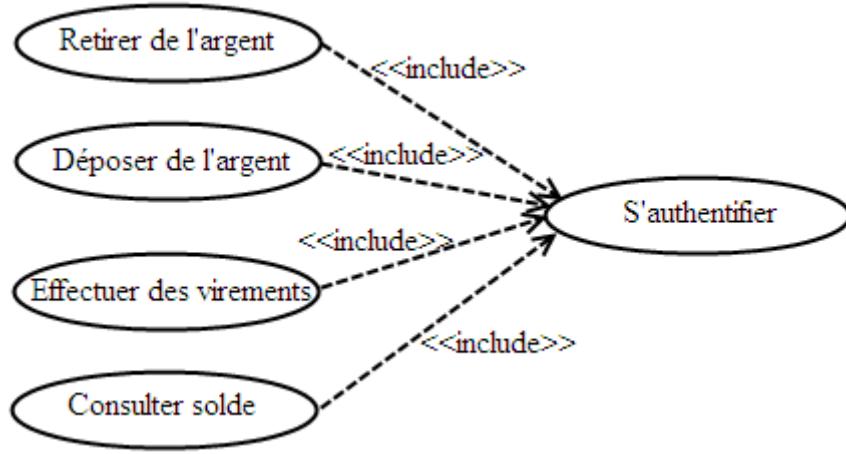


Figure II.9: Exemple des relations d'inclusion entre client et guichet automatique

Cette relation indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. En d'autre terme, il faut s'authentifier avant de retirer de l'argent.

Remarque : Dans une relation «include», le cas d'utilisation de base utilise systématiquement les enchaînements provenant du cas inclus.

- La relation d'extension

On dit qu'un cas d'utilisation A étend un cas d'utilisation B lorsque le cas d'utilisation A peut être appelé au cours de l'exécution du cas d'utilisation B. Exécuter B peut éventuellement entraîner l'exécution de A : contrairement à l'inclusion, l'extension est optionnelle. Cette dépendance est symbolisée par le stéréotype *<<extend>>*.

La Figure II.10 suivante montre une représentation d'une relation d'extension entre client et guichet automatique.

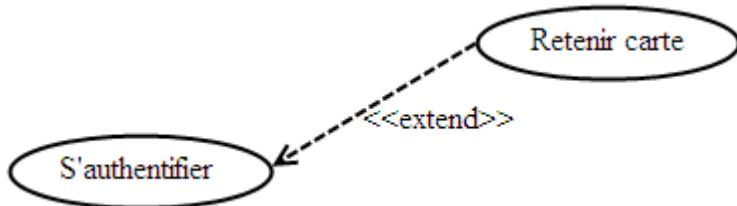


Figure II.10: Représentation d'une relation d'extension entre client et guichet automatique

La relation indique que les cas d'utilisation source étend (précise) les objectifs (comportement) du cas d'utilisation destination.

ii. Séquence

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages [9]. Pour bien comprendre cet diagramme, illustrons par un exemple de scénario de retrait en espèce sur un guichet automatique qui est représenté par la Fig. II.11 ci-dessous.

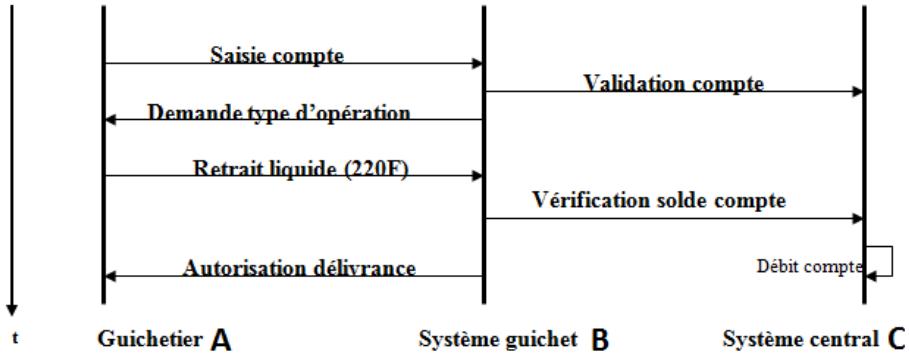


Figure II.11: Partie 1 du diagramme de séquence « retrait en espèce »

iii. Collaboration

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

Suit à l'exemple précédent(retrait en espèce), construisons le diagramme de collaboration, représenté par la Fig. II.12.

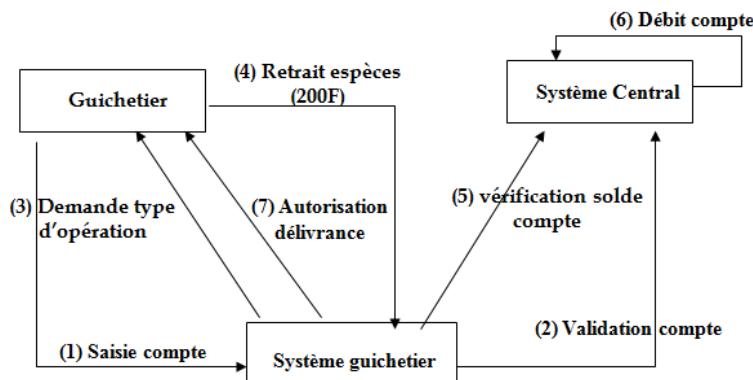


Figure II.12: Partie 2 du diagramme de collaboration (retrait en espèce)

iv. Classe et objet

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes [10].

Une classe, représentée par la figure II.13 ci-dessous, correspond à un concept global d'information et se compose d'un ensemble d'informations élémentaires, appelées attributs de la classe qui servent à la décrire.

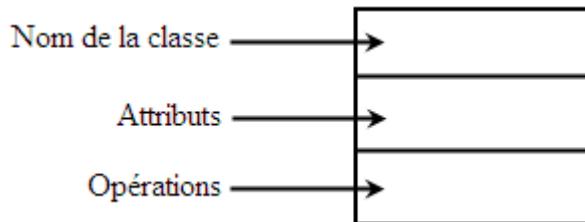


Figure II.13: Représentation d'une classe

UML définit trois niveaux de visibilité pour les attributs et les opérations :

- Public qui rend l'élément visible à tous les clients de la classe
- Protégé qui rend l'élément visible aux sous classes de la classe
- Privé qui rend l'élément visible à la classe seule
 - Les associations

Une association représente une relation structurelle entre deux ou plusieurs classes d'objets qui sont représentées par la Fig. II.14 ci-dessous.



Figure II.14: Association sous forme binaire (seulement entre deux classes)

Il arrive en effet que l'information que l'on veut représenter nécessite la mise en relation de plus de deux classes.

- La multiplicité d'une association

Chaque rôle peut porter une multiplicité montrant combien d'objets de la classe considérée (celle qui joue ce rôle) peuvent être liés à une instance de l'autre classe par l'association, à savoir :

- 1..1 noté 1 c'est-à-dire un et seul
- 0..1 c'est-à-dire zéro ou un

- 0..* noté * c'est-à-dire de zéro à n
- 1..* c'est-à-dire de un à n
- n..m c'est-à-dire de n à m

La multiplicité sous la forme d'un couple de cardinalité ci-dessous est illustré dans un exemple de la Fig. II.15 ci-dessous.

Cardinalité : précise le nombre d'instances qui participent à une relation.

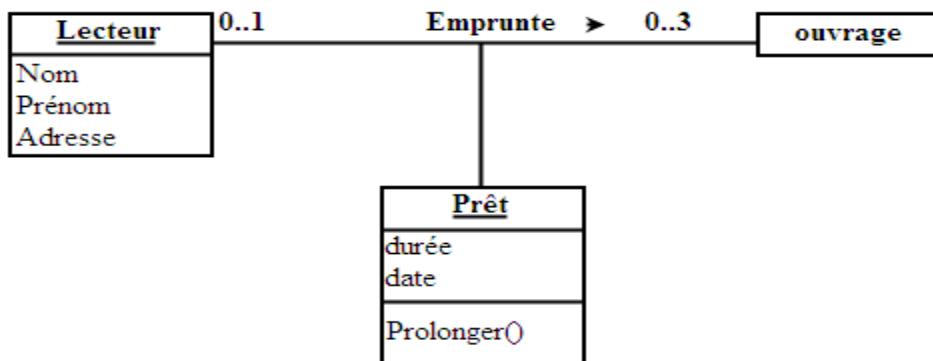


Figure II.15: Exemple d'une classe d'association

– L'agrégation

Lorsqu'une association entre deux instances d'une classe, en plus d'une particularité dont le sens est du style : une instance est composée d'une ou plusieurs autres instances. On peut dire également qu'une agrégation est une association de type “ Composé–Composant ”. Où, l'instance composé est l'agrégat et les composants sont les instances agrégées, qui est illustré dans un exemple de la Fig. II.16 ci-dessous.

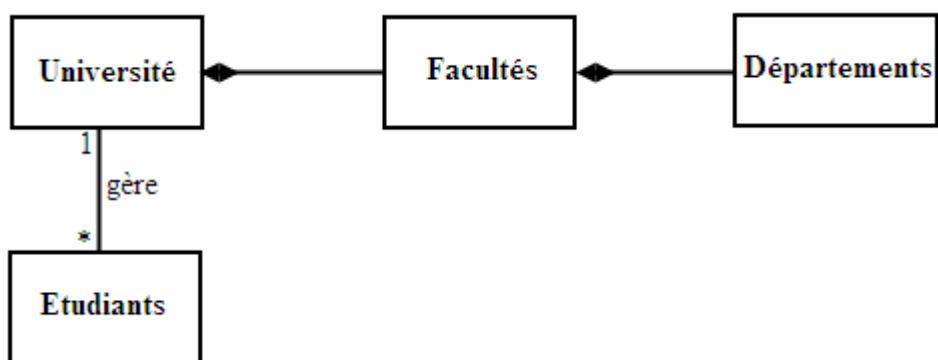


Figure II.16: Exemple d'une association d'agrégation

II.3. ENVIRONNEMENT JAVA ET JAVA 2D

Plusieurs environnements existent pour le développement Java. Voici quelques exemples : de Sun (JRE, JDK), de Microsoft (Visual Java), etc. Tous ces environnements, installent la machine virtuelle de Java (JVM : Java Virtual machine).

Le JRE (Java Runtime Environment) est le conteneur de l'environnement d'exécution du Java qui doit être installé sur toutes les machines susceptibles de lancer des applications de type Java.

Le JDK (Java Development Kit) contient le JRE. C'est l'environnement indispensable à la création de programmes Java. Il inclut un compilateur, un interpréteur, ainsi qu'un environnement de développement comprenant des outils et des utilitaires favorisant la programmation. Le JDK contient un grand nombre d'outils tels que :

- **javac** : Java Compiler, le compilateur Java ; il compile un fichier source .java en un fichier exécutable .class contenant du bytecode.
- **java** : Exécute le ou les fichiers compilés par javac. C'est l'interpréteur Java, c'est à dire une implémentation de la machine virtuelle Java (JVM).
- **javadoc** : C'est un utilitaire très puissant qui permet de construire, à partir des commentaires insérés dans des sources Java, des fichiers HTML à propos des classes, méthodes, données membres....définies dans ces sources.
- **apt** : Annotation processing tool, nouveauté de Java 2 Standard Edition 5.0 (1.5 « Tiger »), permet l'exploitation des annotations (à ne pas confondre avec les commentaires).
- **applet viewer** : Ce programme permet d'exécuter une Applet Java sans nécessiter d'utiliser un navigateur web. Il dispose d'une interface graphique.
- **jdb** : Débogueur Java qui permet de détecter les erreurs de programmation.
- **jar** : Permet la création et la gestion des fichiers JAR (Java Archive).

II.3.1. *Introduction*

En Java 2, Java2D remplace les classe Graphics de Java 1. Java2D est une API composée par un ensemble de classes destinées à l'imagerie et à création de dessin 2D [11]. Elle permet de :

- dessiner des lignes, des rectangles et toute autre forme géométrique ;
- replisser les formes par des couleurs unies ou en dégradés et lui ajouter des textures ;
- ajouter du texte, lui attribuer différentes polices et contrôler son rendu ;

- dessiner des images, éventuellement en effectuant des opérations de filtrage.

II.3.2. Classes et Interfaces de base de l'API Java2D

Les classes et les interfaces de base de l'API Java 2D sont représentées par la Fig. II.17 suivante :

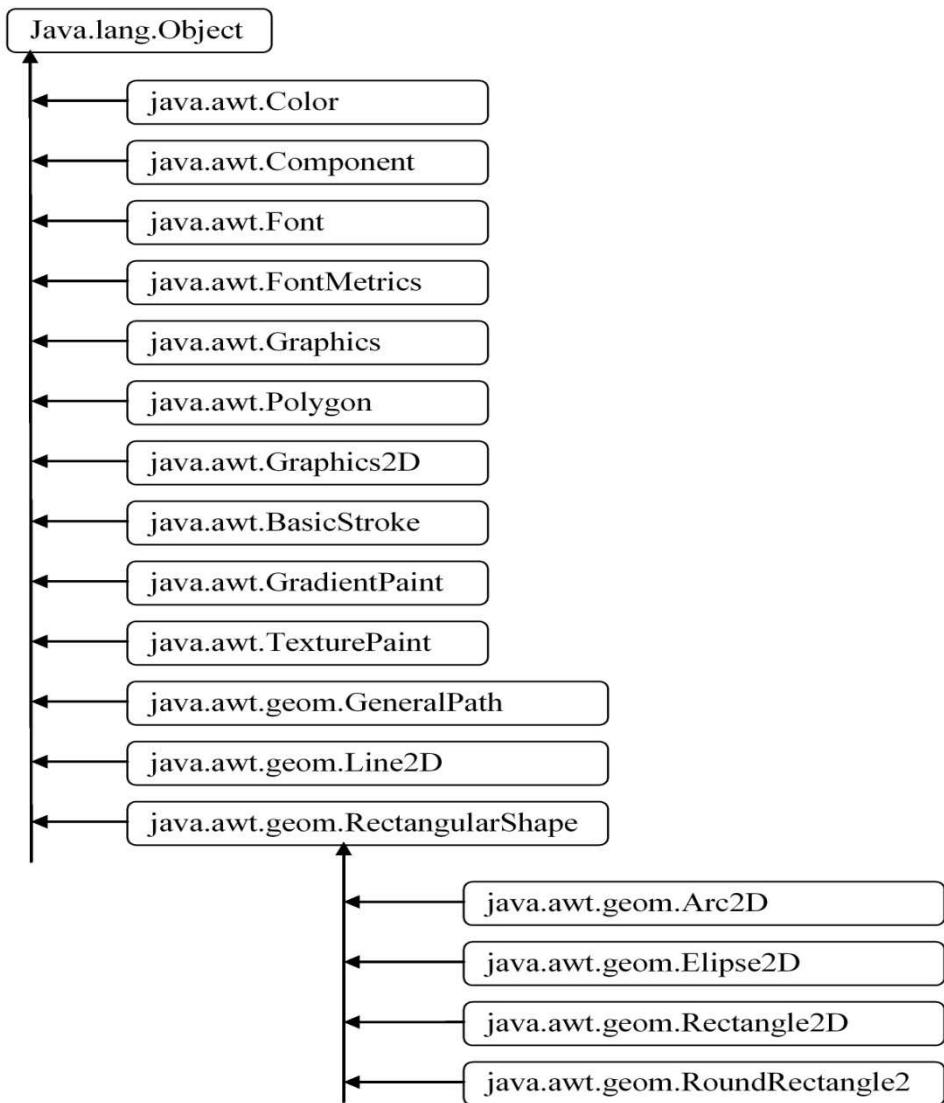


Figure II.17: Classes et interfaces de base de l'API Java 2D

- La classe **Color** permet de manipuler les couleurs.
- La classe **Component** représente un objet graphique qui peut être affiché à l'écran et interagir avec l'utilisateur.
- La classe **Font** est utilisé pour le rendu de polices de caractères.
- La classe **FontMetrics** permet de récupérer des informations sur les polices de caractères.

- La classe **Polygon** permet de créer des polygones et d'effectuer des opérations dessus.
- La classe **Graphics2D** fournit un contrôle sophistiqué de la géométrie, des transformations de coordonnées, de la gestion de couleur, et de la disposition des textes.
- La classe **BasicStroke** définit un ensemble de caractéristiques pour le contour de primitives graphiques.
- Les classes **GradientPaint** et **TexturePaint** sont utilisées pour définir de formes avec des couleurs ou des textures.
- La classe **GeneralPath** représente un chemin géométrique construit de lignes droites.
- La classe **Line2D** permet de représenter des segments et d'effectuer de nombreuses opérations sur celles-ci.
- La classe **RectangularShape** est une classe de base pour de nombreuses formes dont la géométrie peut être définie par un cadre rectangulaire fournit de nombreuses méthodes de manipulation de formes.
- La classe **Arc2D** permet de manipuler des arcs.
- La classe **Ellipse2D** permet de manipuler des ellipses.
- La classe **Rectangle2D** permet de manipuler des rectangles.
- La classe **RoundRectangle2D** permet de manipuler des rectangles avec des coins ronds.

II.3.3. Le système de coordonnées en Java2D

Il est présenté par la figure II.18 suivante :

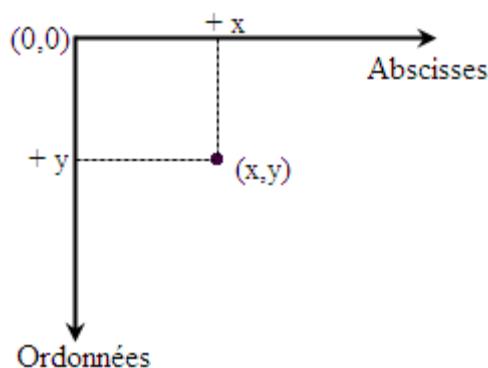


Figure II.18: Le système de coordonnées en Java2D

Le point de coordonnées (0, 0) est donc en haut à gauche.

L'abscisse est calculée de gauche à droite.

L'ordonnée est calculée de haut en bas.

II.4. TRAITEMENTS DE SHAPE

En géométrie les traitements de Shape englobent les moyens pour tracer, pour modifier, pour créer et de stocker des différentes formes géométriques. Mais son application au SIG se limitera aux quelques formes tels que les points, les polylignes et les polygones. Cela nous permet de voir quelques notions de géométrie, les généralités sur les transformations plan, le changement de repère par translation, et enfin l'utilité de la transformation de coordonnées géographiques WGS 84 en coordonnées géographiques Laborde Madagascar.

II.4.1. *Notions de géométrie*

a. Principaux objets de la géométrie plane

La géométrie plane définit des objets élémentaires, des abstractions idéalisées qui approchent des objets réels. Les objets suivants sont les principaux objets de la géométrie plane :

- le point est l'objet géométrique le plus élémentaire. Il n'a aucune dimension ; on dit qu'il est infiniment petit (on peut l'imaginer comme la pointe d'une aiguille). Il peut également être défini comme le point d'intersection de deux droites;
- la droite est représentée par un trait droit. L'objet idéal « droite » est une ligne continue dans une direction fixée, sans saut ni interruption, sans début ni fin, formée par la succession d'une infinité de points alignés. Elle est ainsi le plus court chemin entre deux de ses points ;
- les polygones sont des figures fermées, composées de plusieurs segments de droite.

Ces principaux objets de la géométrie plane sont illustrés par la Fig. II.19 ci-dessous.

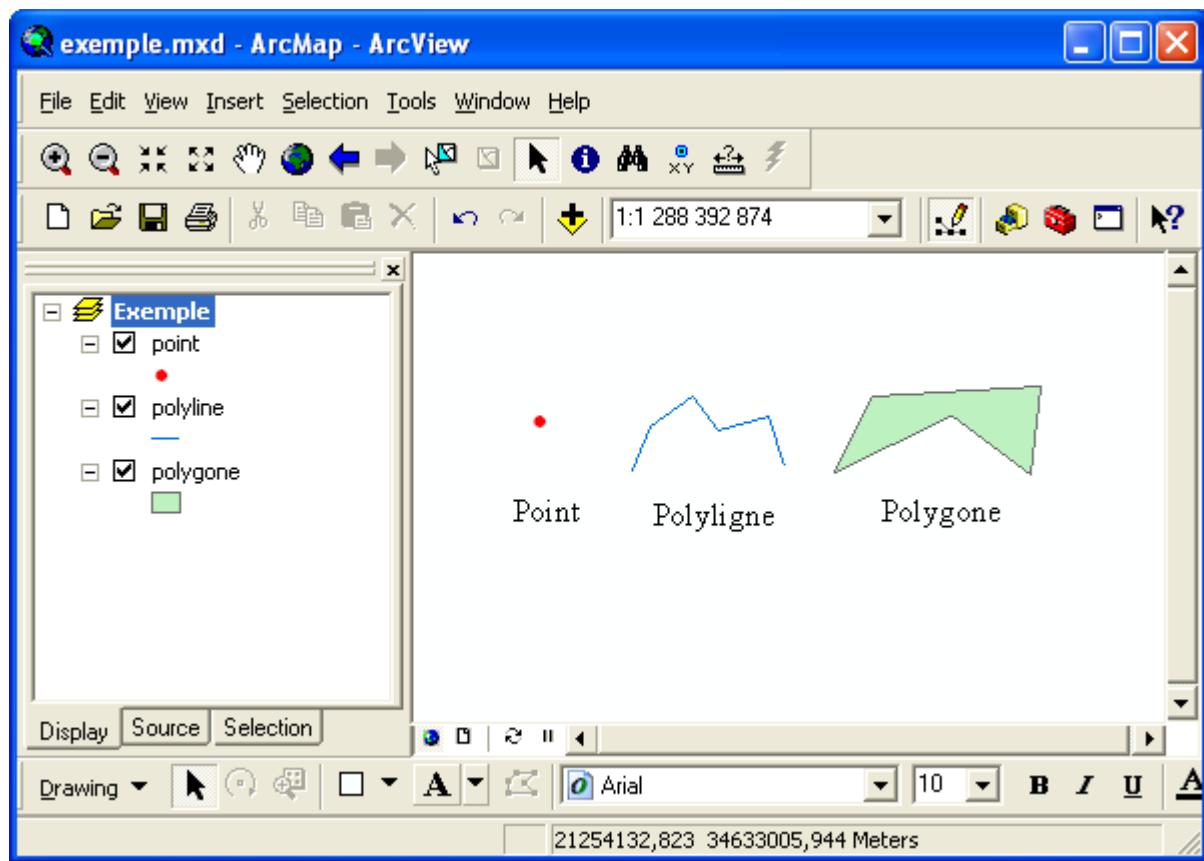


Figure II.19: Exemples d'objets de la géométrie plane

b. Les systèmes de coordonnées dans le plan

i. *Coordonnées cartésiennes*

Le plan étant muni d'un repère orthonormé (O, \vec{i}, \vec{j}) , tout point peut être repéré par deux nombres réels appelés abscisse et ordonnée [12]. Ils sont représentés sur la Fig. II.20.

Ecrire $M(x, y)$ dans le repère (O, \vec{i}, \vec{j}) c'est dire que: $\overrightarrow{OM} = x\vec{i} + y\vec{j}$. La distance OM est alors telle que, d'après le théorème de Pythagore :

$$OM = \sqrt{x^2 + y^2} \quad (\text{II.1})$$

Si deux points A et B sont tels que $A(x_A, y_A)$ et $B(x_B, y_B)$ alors :

$$\begin{cases} \overrightarrow{AB} = (x_B - x_A)\vec{i} + (y_B - y_A)\vec{j} \\ AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \end{cases} \quad (\text{II.2})$$

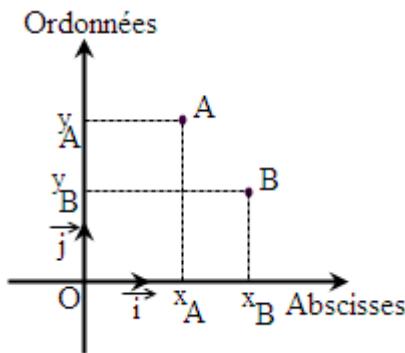


Figure II.20: Coordonnées cartésiennes

ii. Coordonnées polaires

Le plan étant muni d'un repère orthonormé (O, \vec{i}, \vec{j}) , tout point M peut être repéré par deux nombres réels l'un étant la distance de l'origine O à M , l'autre étant une mesure de l'angle orienté du vecteur \vec{i} au vecteur \overrightarrow{OM} . Cet angle du vecteur \vec{i} au vecteur \overrightarrow{OM} est appelé angle polaire du point M . Les deux nombres qui décrivent ainsi la position du point M sont souvent notés ρ et θ et sont appelés coordonnées polaires du point M . Ils sont représentés sur la Fig. II.21 ci-dessous.

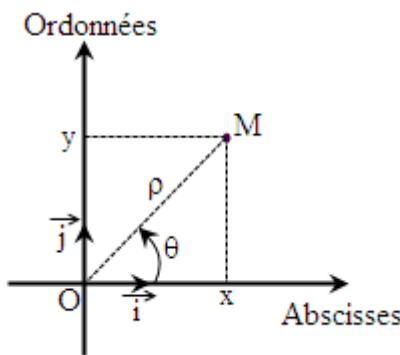


Figure II.21: Coordonnées polaires

A partir des coordonnées polaires ρ et θ du point M , il est facile de retrouver les coordonnées cartésiennes du même point... il suffit de projeter pour obtenir :

$$\begin{cases} x = \rho \cos \theta \\ y = \rho \sin \theta \end{cases} \quad (\text{II.3})$$

II.4.2. Généralités sur les transformations du plan

On appelle transformation une application bijective du plan sur lui-même, associant deux objets géométriques, points ou figures. Les transformations les plus répandues sont les translations, les rotations, les symétries et les homothéties.

a. Les translations

Soit un vecteur \vec{u} du plan. On appelle translation de vecteur \vec{u} la transformation qui à tout point M du plan associe le point M' tel que $\overrightarrow{MM'} = \vec{u}$.

b. Les rotations

Soit un point O du plan et d'un angle α . On appelle rotation de centre O et d'angle α la transformation qui à tout point M du plan associe le point M' tel que $\overrightarrow{OM'} = \overrightarrow{OM}$ et tel que α soit l'angle orienté formé par les vecteurs \overrightarrow{OM} et $\overrightarrow{OM'}$. Autrement dit les points M et M' sont les extrémités d'un arc de cercle centré en O et de mesure α . Le seul point invariant par cette rotation est le point O .

c. Les symétries

Soient deux droites D et D' non parallèles. On appelle symétrie par rapport à D et parallèlement à D' la transformation qui à tout point M du plan associe le point M' tel que $\overrightarrow{MM'} = 2\overrightarrow{MP}$, où P est l'intersection de D avec la droite passant par M et parallèle à D' .

d. Les homothéties

Soit un point O du plan et k un réel. On appelle homothétie de centre O et de rapport k la transformation qui à tout point M du plan associe le point M' tel que $\overrightarrow{OM'} = k\overrightarrow{OM}$. Si $k = 1$, l'homothétie est l'identité I . Si $k = -1$, l'homothétie est la rotation de centre O et d'angle π , encore appelée symétrie centrale de centre O .

II.4.3. Changement de repère par translation

Soit O' le point de coordonnée (x_0, y_0) dans le repère (O, \vec{i}, \vec{j}) . Dans la translation de vecteur $\overrightarrow{OO'}$, le repère (O, \vec{i}, \vec{j}) a pour image le repère (O', \vec{i}, \vec{j}) . Soit M un point du plan, (x, y) son coordonnée dans le repère (O, \vec{i}, \vec{j}) et (X, Y) dans (O', \vec{i}, \vec{j}) . On a les égalités suivantes :

$$\begin{cases} \overrightarrow{OO'} = x_0\vec{i} + y_0\vec{j} \\ \overrightarrow{OM} = x\vec{i} + y\vec{j} \\ \overrightarrow{O'M} = X\vec{i} + Y\vec{j} \end{cases}$$

Alors :

$$\begin{cases} x = x_0 + X \\ y = y_0 + Y \end{cases}$$

Or ce résultat n'est pas utilisable en Java2D car en Java2D l'axe des ordonnées est inversé, d'où la formule suivante :

$$\begin{cases} x = x_0 + X \\ y = y_0 - Y \end{cases} \quad (\text{II.4})$$

II.4.4. Transformation de coordonnées WGS 84 en Laborde Madagascar

Les coordonnées géographiques WGS 84 sont des coordonnées reconnus internationaux et qui sont les plus utilisés dans la plupart des logiciels SIG et du GPS. Ce qui nécessite la transformation de coordonnées géographiques WGS 84 en coordonnées géographiques Laborde Madagascar pour que les coordonnées soient utilisables à Madagascar.

Les deux centres de projections sont différents et les axes de deux ellipsoïdes ne sont pas parallèles et les différences de centres ΔX , ΔY , ΔZ plus la différence de deux rayons Δa , Δb et la différence de l'aplatissement Δf déterminent ce qu'on appelle datums. Les datums en usage au FTM sont les suivants :

$$\begin{cases} \Delta X = -189 \text{ m} \\ \Delta Y = -242 \text{ m} \\ \Delta Z = -91 \text{ m} \\ \Delta a = -251 \text{ m} \\ \Delta b = -160 \text{ m} \\ \Delta f = -0,00001419 \end{cases} \quad (\text{II.5})$$

Il y a plusieurs méthodes pour faire cette transformation, on va utiliser la méthode de Molodensky qui permet d'effectuer une conversion directe entre deux systèmes de coordonnées géographiques [6].

D'où les formules suivantes :

$$(M + h)\Delta\phi = -\sin\varphi\cos\lambda\Delta X - \sin\varphi\sin\lambda\Delta Y + \cos\varphi\Delta Z + \frac{e^2\sin\varphi\cos\varphi}{(1-e^2\sin^2\varphi)^{1/2}} \Delta a + \sin\varphi\cos\varphi \left(M \frac{a}{b} + N \frac{b}{a} \right) \Delta f \quad (\text{II.6})$$

$$\Delta h = \cos\varphi\cos\lambda\Delta X + \cos\varphi\sin\lambda\Delta Y + \sin\varphi\Delta Z - (1 - e^2\sin^2\varphi)^{1/2}\Delta a + \frac{a(1-f)}{(1-e^2\sin^2\varphi)^{1/2}} \sin^2\varphi\Delta f \quad (\text{II.7})$$

Avec :

- h : hauteur de l'ellipsoïde (en mètres)
- φ : latitude
- λ : longitude
- a : demi-grand axe de l'ellipsoïde (en mètres)
- b : demi-petit axe de l'ellipsoïde (en mètres)
- f : aplatissement de l'ellipsoïde
- e : excentricité de l'ellipsoïde

M et N représentent, respectivement, le rayon méridien et la grande normale pour une latitude donnée. Les équations correspondant à M et N sont les suivantes :

$$M = \frac{a(1-e^2)}{(1-e^2\sin^2\varphi)^{3/2}} \quad (\text{II.8})$$

$$N = \frac{a}{(1-e^2\sin^2\varphi)^{1/2}} \quad (\text{II.9})$$

La méthode Molodensky abrégée est une version simplifiée de la méthode Molodensky. Les équations sont les suivantes :

$$M\Delta\varphi = -\sin\varphi\cos\lambda\Delta X - \sin\varphi\sin\lambda\Delta Y + \cos\varphi\Delta Z + (a\Delta f + f\Delta a) \cdot 2\sin\varphi\cos\varphi \quad (\text{II.10})$$

$$(N + h)\cos\varphi\Delta\lambda = -\sin\lambda\Delta X + \cos\lambda\Delta Y \quad (\text{II.11})$$

$$N\cos\varphi\Delta\lambda = -\sin\lambda\Delta X + \cos\lambda\Delta Y \quad (\text{II.12})$$

$$\Delta h = \cos\varphi\cos\lambda\Delta X + \cos\varphi\sin\lambda\Delta Y + \sin\varphi\Delta Z + (a\Delta f + f\Delta a)\sin^2\varphi - \Delta a \quad (\text{II.13})$$

Pour trouver les coordonnées cartésiennes, on doit concourir aux équations suivantes [13]:

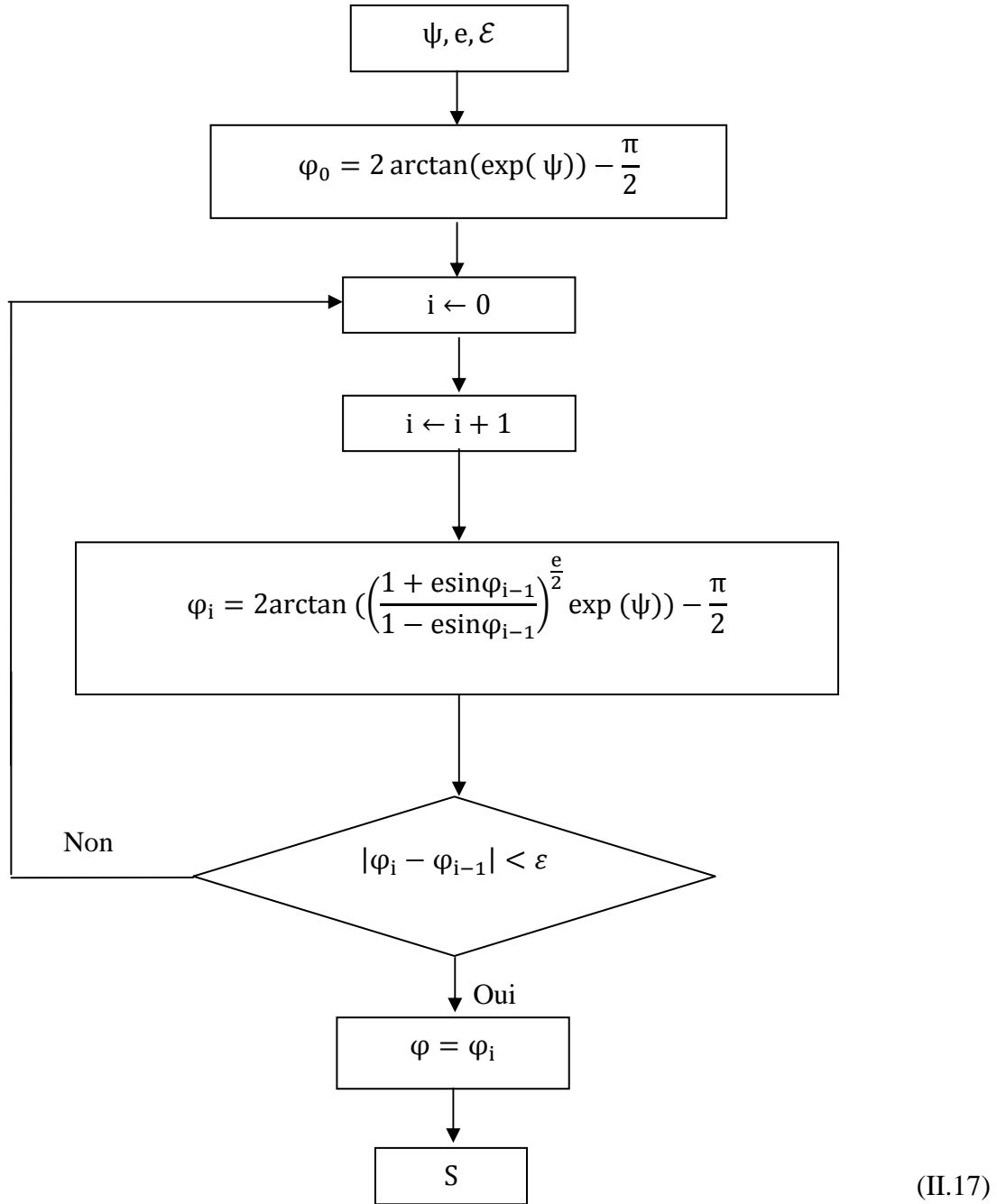
$$\begin{cases} X = (N + h)\cos\varphi\cos\lambda \\ Y = (N + h)\cos\varphi\sin\lambda \\ Z = (N(1 + e^2) + h)\sin\varphi \end{cases} \quad (\text{II.15})$$

De plus il y a une autre possibilité pour faire la transformation de coordonnées géographiques (λ, φ) en coordonnées planes (X, Y) en appliquant les algorithmes à la projection cartographique GAUSS – LABORDE [14].

- **Algorithme 01 :** calcul de la latitude isométrique sur un ellipsoïde de première excentricité e au point de latitude φ .

$$\psi(\varphi, e) = \ln \left(\tan \left(\frac{\pi}{4} + \frac{\varphi}{2} \right) * \left(\frac{1-e\sin\varphi}{1+e\sin\varphi} \right)^{\frac{e}{2}} \right) \quad (\text{II.16})$$

- **Algorithme 02 :** calcul de la latitude φ à partir de la latitude isométrique ψ .



On notera $\psi^{-1}(\psi, e)$ la valeur de la latitude à partir de la latitude isométrique ψ pour un ellipsoïde de première excentricité e .

a. Calcul des paramètres de projection

Ils sont calculés à partir de l'algorithme 01 et 02.

Les paramètres en entrée sont :

- a : demi-grand axe
- e : première excentricité de l'ellipsoïde
- λ_0 : longitude origine par rapport au méridien d'origine
- φ_0 : latitude du point d'origine
- k_0 : facteur d'échelle au point d'origine
- X_0, Y_0 : coordonnées planes du point d'origine
- ε : tolérance de convergence = 1.10^{-11} d' après l'IGN

Les paramètres en sortie sont :

- λ_c : longitude origine par rapport au méridien d'origine
- φ_c : latitude du point d'origine (sphère)
- c : constante de la projection
- n_1 : exposant de la projection ellipsoïde-sphère
- n_2 : rayon de la sphère intermédiaire
- X_s, Y_s : constantes sur X, Y

L'IGN donne les résultats suivants :

$$\left\{ \begin{array}{l} \lambda_c = \lambda_0 \\ n_1 = \sqrt{1 + \frac{e^2}{1-e^2} \cos^4 \varphi_0} \\ \varphi_c = \arcsin \left(\frac{\sin \varphi_0}{n_1} \right) \\ c = \psi(\varphi_c, 0) - n_1 \psi(\varphi_0, 0) \\ n_2 = k_0 a \frac{\sqrt{1-e^2}}{1-e^2 \sin^2 \varphi_0} \\ X_s = X_0 \\ Y_s = Y_0 - n_2 \varphi_c \end{array} \right. \quad (\text{II.18})$$

b. Transformation de coordonnées géographiques en coordonnées planes dans le système de projection Gauss-Laborde

Les paramètres en entrée sont :

- φ : latitude
- λ : longitude
- e : première excentricité de l'ellipsoïde
- λ_c : longitude origine par rapport au méridien d'origine

- c : constante de la projection
- n_1 : exposant de la projection ellipsoïde-sphère
- n_2 : rayon de la sphère intermédiaire
- X_s, Y_s : constantes sur X, Y

Soient :

$$\Lambda = n_1(\lambda - \lambda_c) \text{ et } \psi_s = c + n_1\psi(\varphi, e)$$

Alors :

$$\begin{cases} X = X_s + n_2\psi(\arcsin\left(\frac{\sin\Lambda}{\operatorname{ch}\psi_s}\right), 0) \\ Y = Y_s + n_2 \arctan\left(\frac{\operatorname{sh}\psi_s}{\cos\Lambda}\right) \end{cases} \quad (\text{II.19})$$

c. Transformation de coordonnées planes en projection Gauss-Laborde, en coordonnées géographiques

Les paramètres en entrée sont :

- X, Y : coordonnées planes du point en projection Gauss-Laborde
- e : première excentricité de l'ellipsoïde
- λ_c : longitude origine par rapport au méridien d'origine
- c : constante de la projection
- n_1 : exposant de la projection ellipsoïde-sphère
- n_2 : rayon de la sphère intermédiaire
- X_s, Y_s : constantes sur X, Y

Soient :

$$\Lambda = \arctan\left(\frac{\operatorname{sh}\left(\frac{X-X_s}{n_2}\right)}{\cos\left(\frac{Y-Y_s}{n_2}\right)}\right) \text{ et } \psi_s = \psi\left(\arcsin\left(\frac{\sin\left(\frac{Y-Y_s}{n_2}\right)}{\operatorname{ch}\left(\frac{X-X_s}{n_2}\right)}\right), 0\right)$$

Alors :

$$\begin{cases} \lambda = \lambda_c + \frac{\Lambda}{n_1} \\ \varphi = \psi^{-1}\left(\frac{\psi_s - c}{n_1}, e\right) \end{cases} \quad (2.20)$$

NB : Toutes ces formules sont des résultats d'après les calculs de l'IGN.

Chapitre III OUTIL DE GESTION DE SHAPE DANS LA PROJECTION LABORDE

Cet outil de gestion de shape permet de présenter et d'analyser les informations géographiques. L'objectif principal est de fournir un moyen pour manipuler ces informations dans l'environnement informatique et de projection Laborde.

L'élaboration d'un outil SIG suppose la disposition de certains outils de base tels que : le langage de programmation, langage de modélisation et un outil de stockage. Dans notre cas, les outils de base utilisés sont les suivants : UML (comme langage de modélisation objet), JAVA (comme langage de programmation) et MySQL (comme SGBD).

Pour arriver à la réalisation de cet outil, nous avons fait des analyses de besoins fonctionnelles et des contraintes en spécifiant d'abord le problème existant qui nous poussent, ensuite la phase de conception et en fin la réalisation proprement dite du logiciel.

III.1. ANALYSE ET CONCEPTION

Cette étape nous permet de définir l'environnement et les contraintes du système et d'en sortir les différents modèles de conception.

III.1.1. Spécification du problème

La plupart des logiciels SIG, que nous avons utilisé jusqu'aujourd'hui, n'intègrent pas correctement la projection Laborde qui est spécifique pour Madagascar. Ils ont une projection Mercator Oblique mais elle n'est pas précise car les écarts de calculs de la valeur des coordonnées sont importants (plus de 100m dans certain endroit). Pourtant, le coût d'acquisition et de maintenance de ces logicielles sont énormes (de l'ordre de 10 millions Ar par licence). Il s'avère nécessaire donc de trouver des outils simples, adaptables à la projection Laborde avec un coût moins cher.

III.1.2. Analyse des besoins et des contraintes

La cartographie désigne la réalisation et l'étude des cartes géographiques. Le principe majeur de la cartographie est la représentation de données sur un support réduit représentant un espace généralement tenu pour réel. L'objectif de la carte, c'est une représentation concise et efficace, la simplification de phénomènes complexes (politiques, économiques, sociaux, etc.) à l'œuvre sur l'espace représenté afin de permettre au public une compréhension rapide et pertinente.

La Figure III.1ci-dessous nous montre les listes des besoins fonctionnels et contraintes, pour le système de gestion de Shape.



Figure III.1: Analyse des besoins et des contraintes

- **Intégration des données géographiques externes :** des données prises sur terrain, des shapes et des données géographique issue des autres outils
- **Production carte :** établissement des cartes et des plans géographiques structuré par des présentations par couche
- **Modélisation :** paramétrier les conteneurs des données (les champs et les tables)
- **Edition :** gérer les informations dans chaque conteneur et leur relation
- **Restructuration des données :** réorganiser les données géographiques traitées pour un traitement spécifiques
- **Archivage des données :** donner un moyen pour stocker et restituer les données en dehors de l'application
- **Multiplateforme :** fonctionnant sur des divers systèmes (Windows, Linux)

III.1.3. Conception

a. Décomposition du système

On peut définir 5 sous systèmes :

- Système d'acquisition
- Système de modélisation
- Système de traitement
- Système de représentation
- Système de stockage

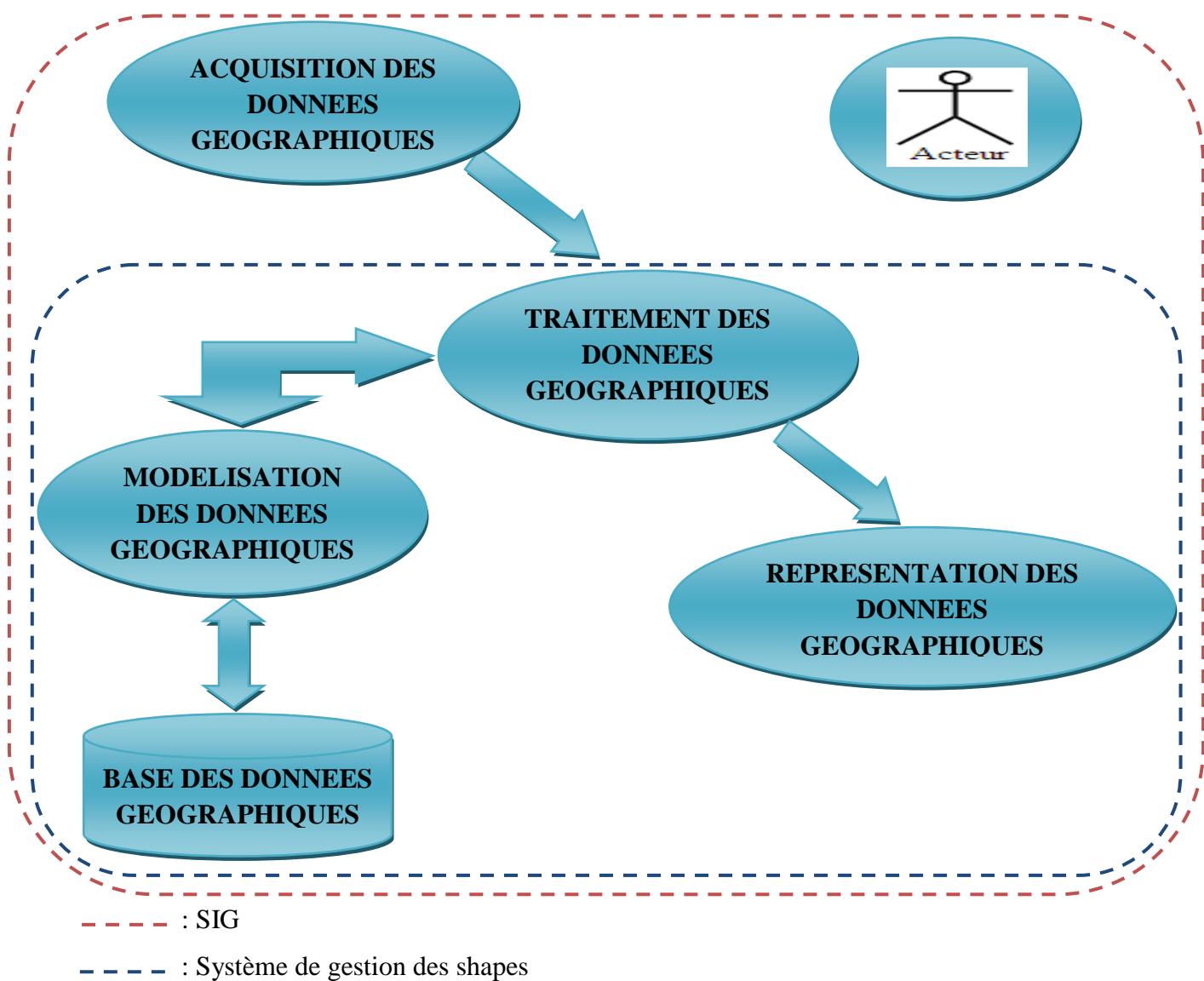


Figure III.2: Système de gestion des shapes

i. Système d'acquisition

Ce système permet de se communiquer avec les outils externes. Il offre la possibilité de :

- Traiter les images de fond
- Convertir des formats de données
- Exporter des données

ii. Système de modélisation

La modélisation consiste à fournir un moyen de construire de shape et des tables à partir des informations saisies par l'utilisateur. Ce système fourni aussi un catalogue de données après la modélisation.

iii. Système de traitement

Le traitement de informations géographique consiste à :

- Tracer des géométries
- Saisir des données
- Analyser les contraintes topographiques
- Rechercher des données
- Modifier ou supprimer des informations

iv. Système de représentation

La visibilité d'une carte nécessite des informations organisées, et des représentations bien distinctes. Pour établir les légendes et d'autre caractéristique de la cartouche de la carte, on doit définir des symbologies pour chaque donnée et métadonnées pour les shapes.

v. Système de stockage

Pour faciliter les échanges de données avec les outils externes, on adopte le stockage de données dans un serveur de base de données.

III.1.4. Modélisation du système

a. Diagramme des use case

Le diagramme use case qui correspond à notre projet est représenté par la figure III.4. Dans ce diagramme nous constatons la présence de trois acteurs principaux, à savoir l'éditeur de structure, l'éditeur d'information et le simple utilisateur. Et chaque acteur a chacun leurs propres rôles qui sont : d'abord l'éditeur de structure est le gestionnaire du projet et pour cela

il a pour tâche la création de table, de modifier les tables et de supprimer les tables. Ensuite, l'éditeur d'information est le gestionnaire des données et ses tâches sont notamment la création de projet, de modifier le projet, de supprimer le projet, de tracer les informations et de modifier les informations. Et enfin le simple utilisateur a pour principal fonction de rechercher ou de visualiser les données alphanumérique ou spatial.

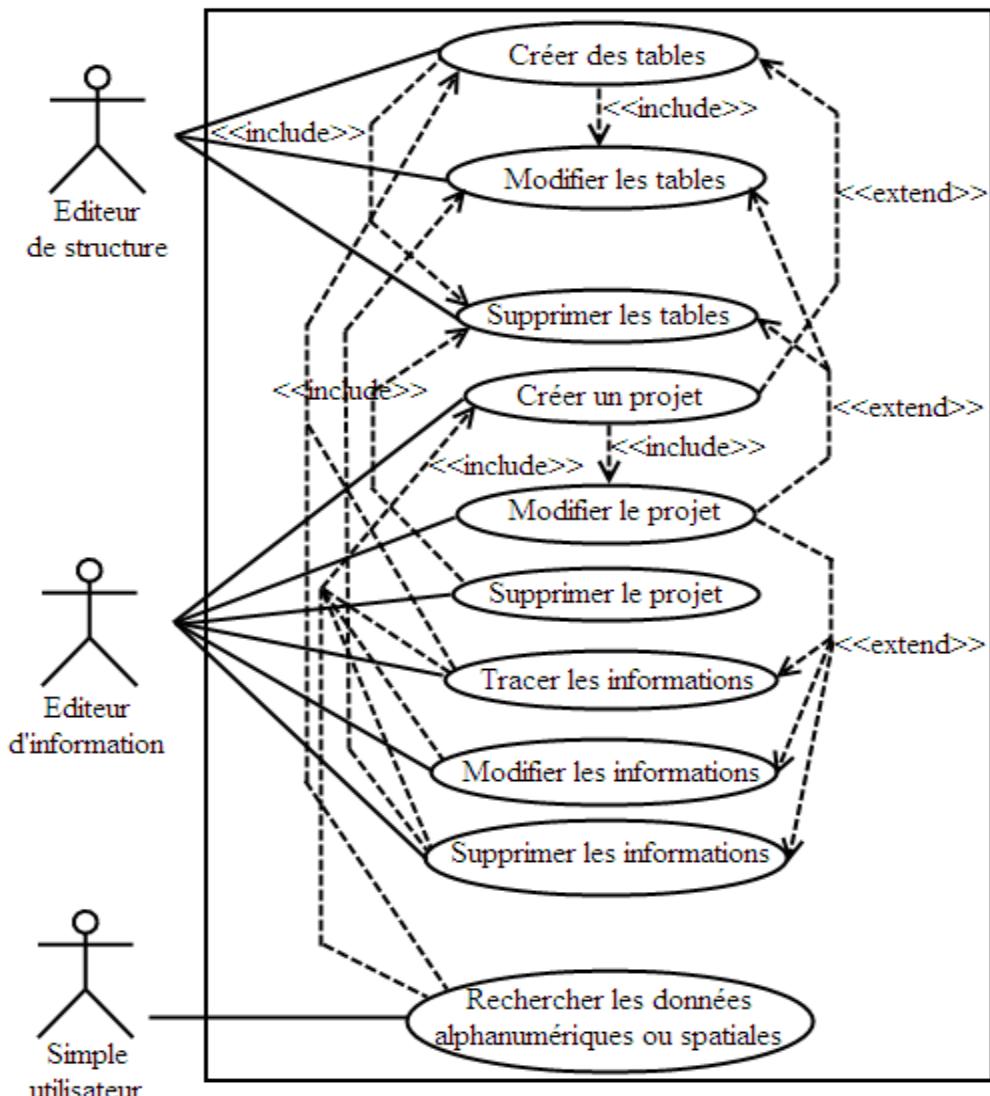


Figure III.3: Diagramme des use case de l'outil SIG

b. Diagramme de séquences

Le diagramme de séquence est servi pour illustrer un cas d'utilisation et chaque cas d'utilisation a sa propre figure la représentant. Pour chaque figure comporte, l'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme ; le temps s'écoule "de haut en bas" de cet axe. La disposition des objets sur l'axe horizontal n'a pas de conséquence pour la sémantique du diagramme.

Et ci-dessous sont énumérées les descriptions du cas d'utilisation, les tâches, contenu dans le diagramme des uses case de l'outil SIG.

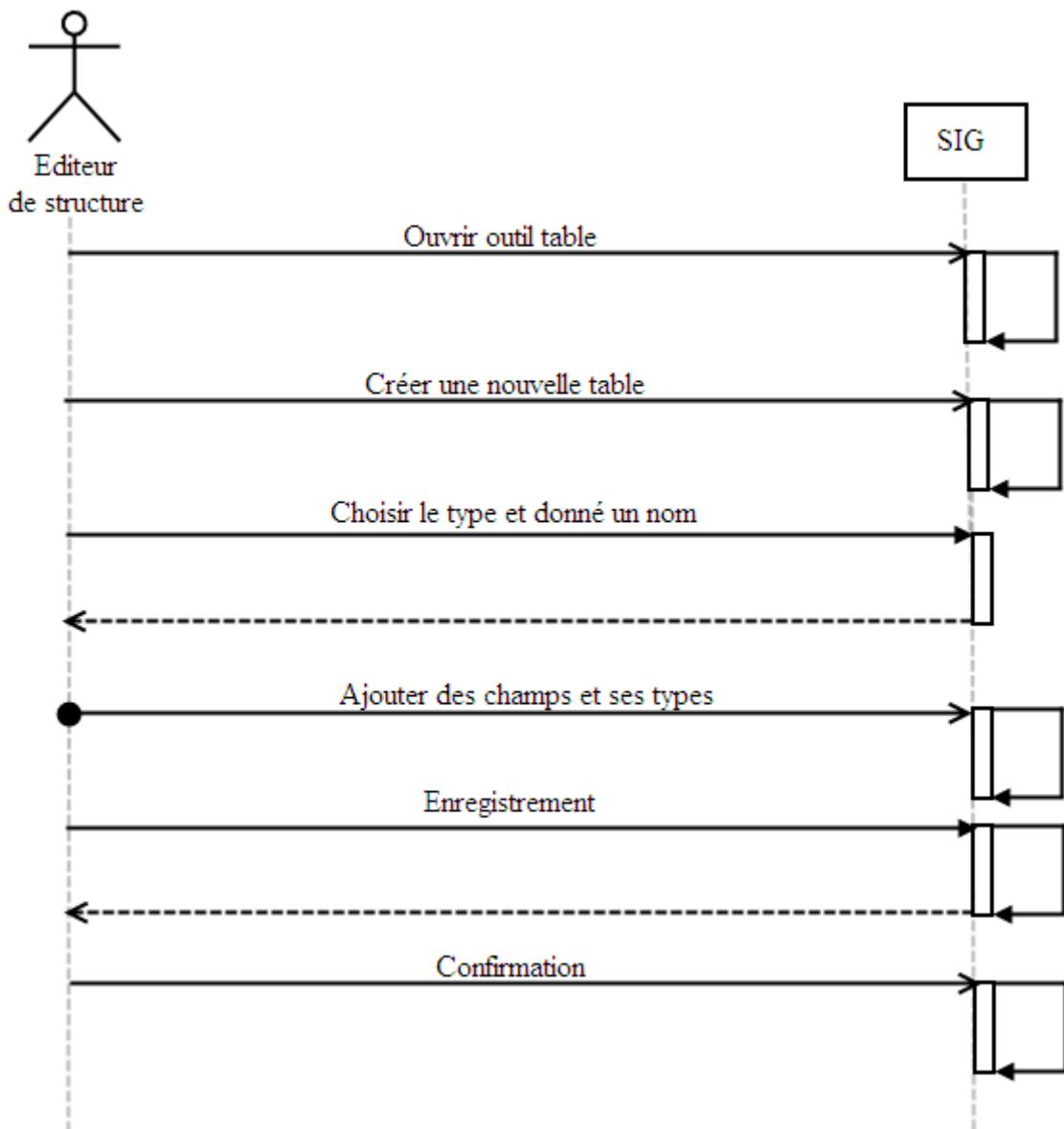


Figure III.4: Diagramme de séquence pour créer des tables

La Figure III.4 ci-dessus montre le diagramme de séquence pour créer des tables, lequel obéit au processus de manipulation spécifique selon l'éditeur de structure. Pour créer une table, on procède par :

- Ouvrir l'outil table
- Créer une nouvelle table
- Choisir le type et donné un nom
- Ajouter des champs et ses types
- Enregistrement

- Confirmation

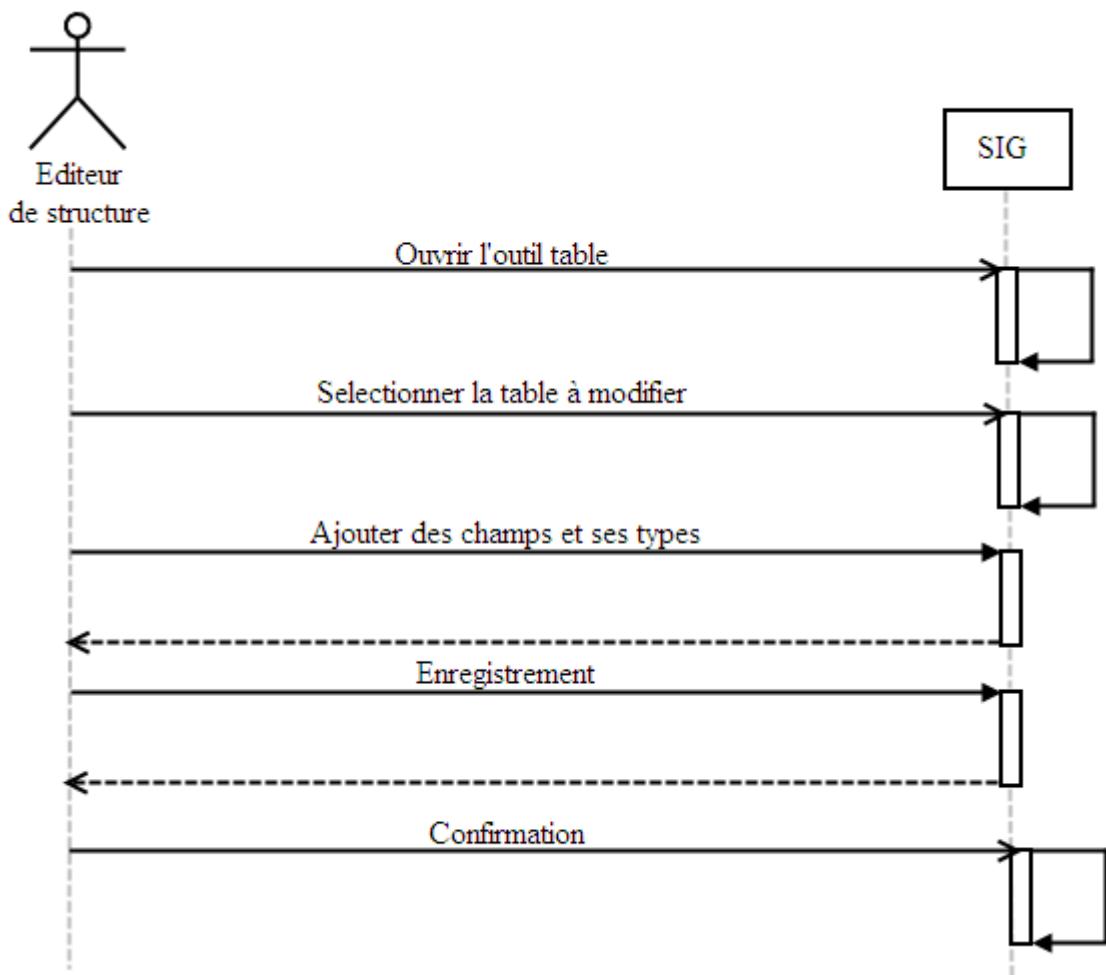


Figure III.5: Diagramme de séquence pour modifier les tables

La Figure III.5 ci-dessus montre le diagramme de séquence pour modifier les tables, lequel obéit au processus de manipulation spécifique selon l'éditeur de structure. Pour modifier une table, on procède par :

- Ouvrir l'outil table
- Sélectionner la table à modifier
- Ajouter des champs et ses types
- Enregistrement
- Confirmation

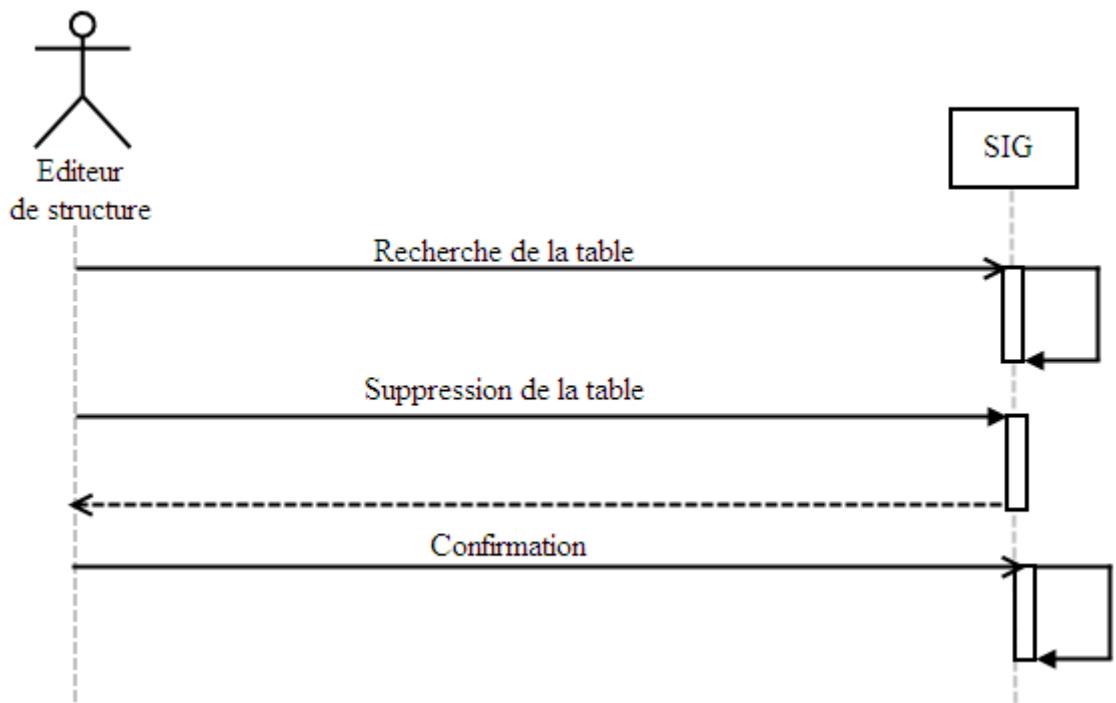


Figure III.6: Diagramme de séquence pour supprimer les tables

La Figure III.6 ci-dessus montre le diagramme de séquence pour supprimer les tables, lequel obéit au processus de manipulation spécifique selon l'éditeur de structure. Pour supprimer une table, on procède par :

- Recherche de la table
- Suppression de la table
- Confirmation

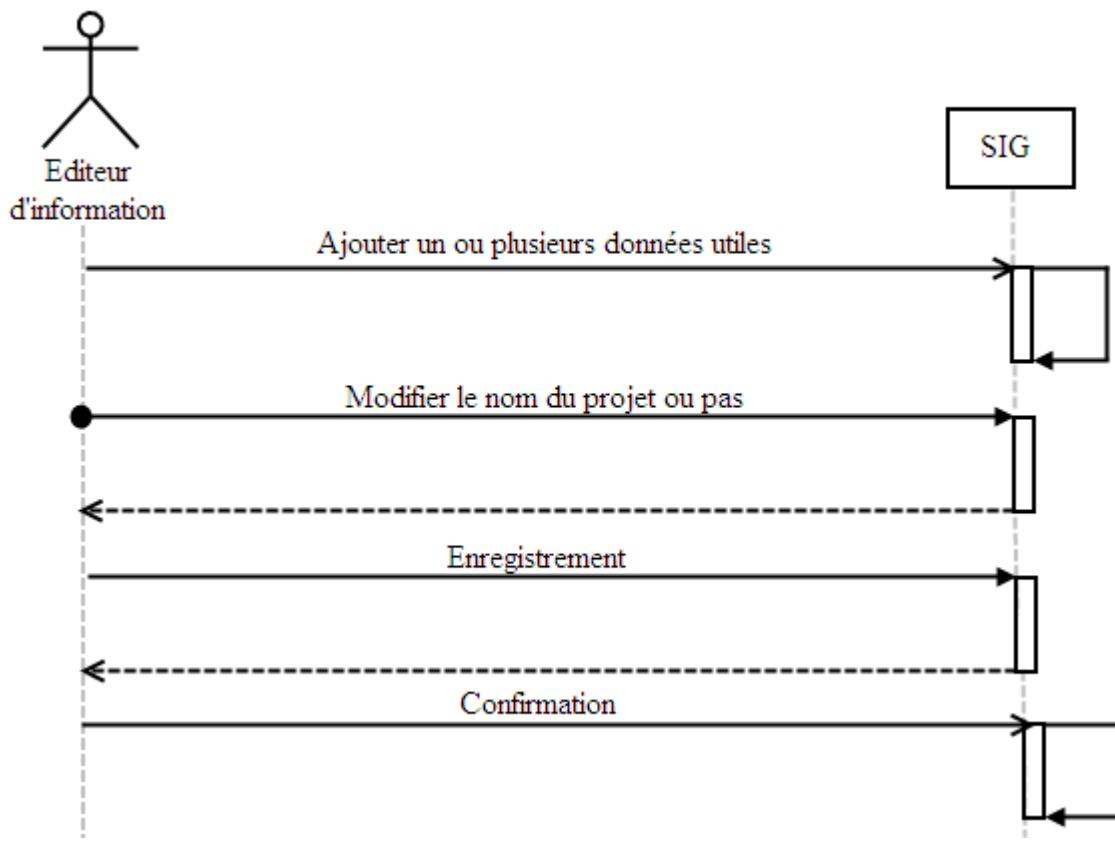


Figure III.7: Diagramme de séquence pour créer un projet

La Figure III.7 ci-dessus montre le diagramme de séquence pour créer un projet, lequel obéit au processus de manipulation spécifique suivant l'éditeur d'information. Pour créer un projet, on procède par :

- Ajouter un ou plusieurs données utiles
- Modifier le nom du projet ou pas
- Enregistrement
- Confirmation

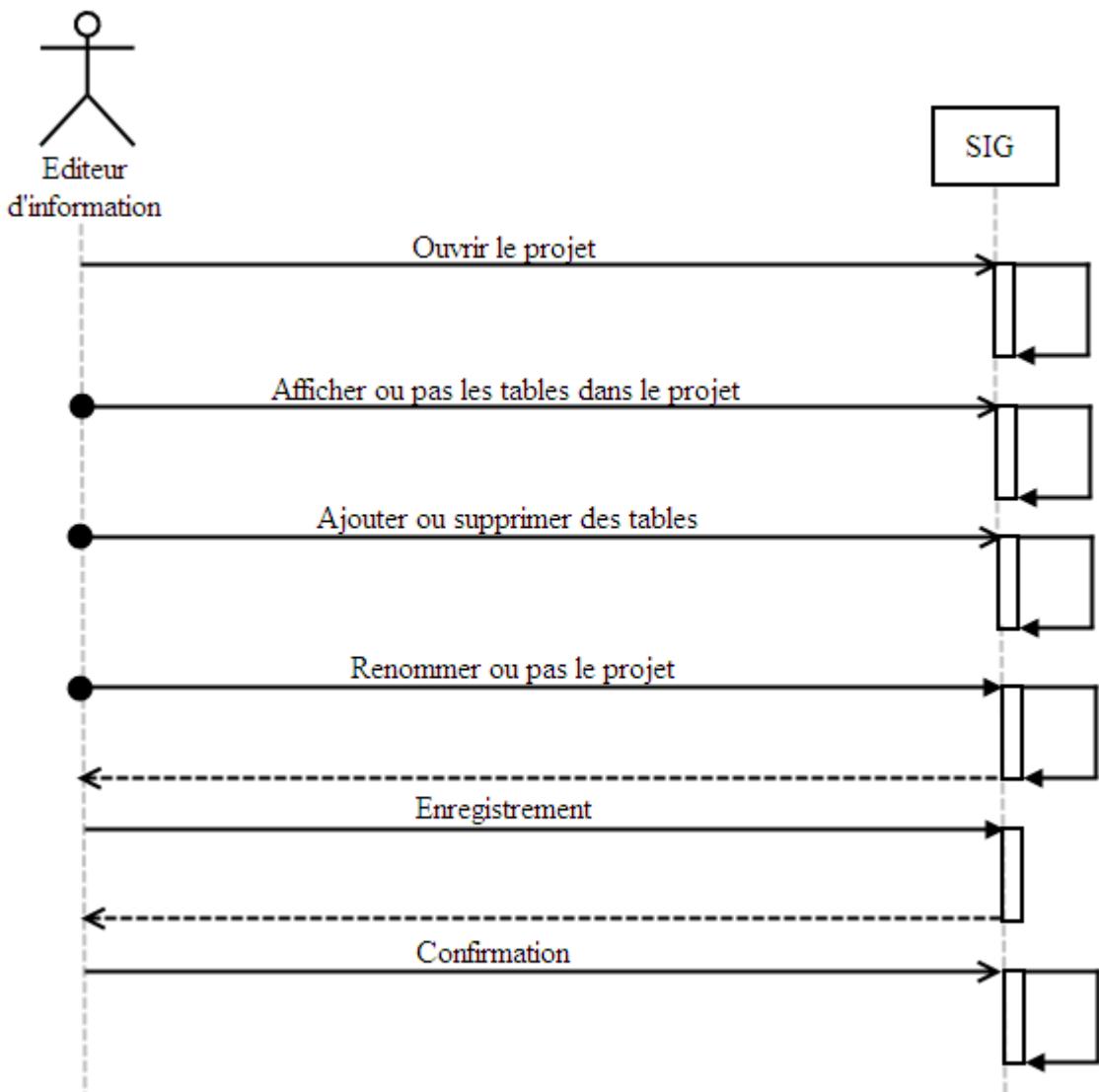


Figure III.8: Diagramme de séquence pour modifier le projet

La Figure III.8 ci-dessus montre le diagramme de séquence pour modifier le projet, lequel obéit au processus de manipulation spécifique suivant l'éditeur d'information. Pour modifier le projet, on procède par :

- Ouvrir le projet
- Afficher ou pas les tables dans le projet
- Ajouter ou supprimer des tables
- Renommer ou pas le projet
- Enregistrement
- Confirmation

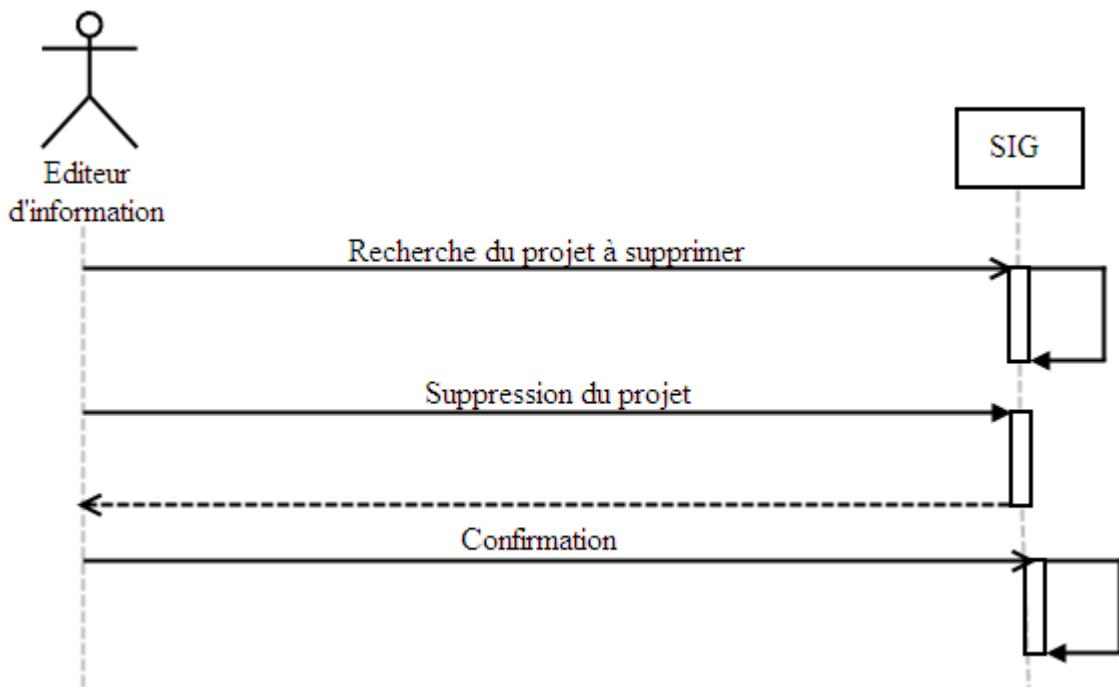


Figure III.9: Diagramme de séquence pour supprimer le projet

La Figure III.9 ci-dessus montre le diagramme de séquence pour supprimer le projet, lequel obéit au processus de manipulation spécifique suivant l’éditeur d’information. Pour supprimer le projet, on procède par :

- Recherche du projet à supprimer
- Suppression du projet
- Confirmation

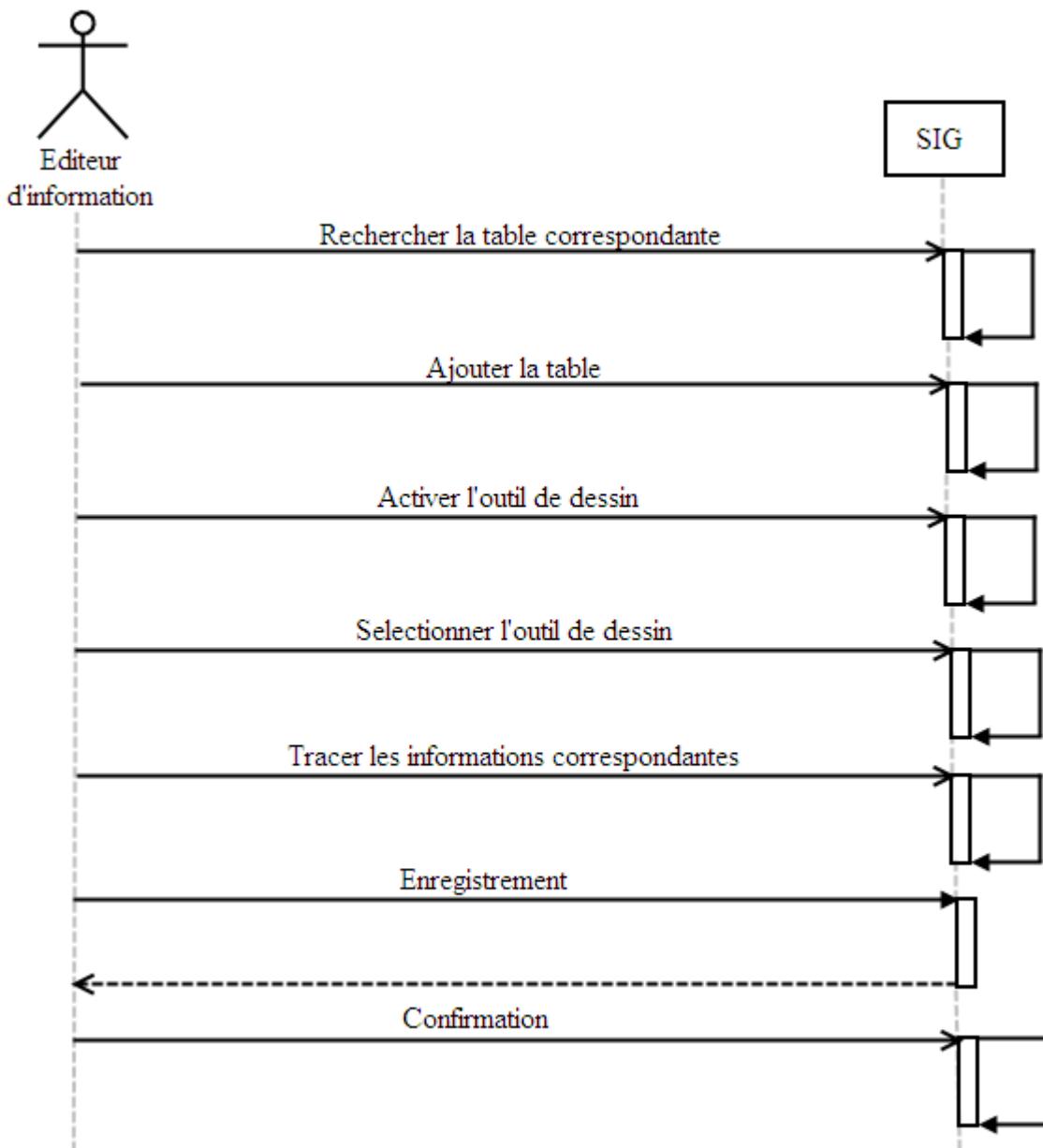


Figure III.10: Diagramme de séquence pour tracer les informations

La Figure III.10 ci-dessus montre le diagramme de séquence pour tracer les informations, lequel obéit au processus de manipulation spécifique selon l'éditeur d'information. Pour tracer les informations, on procède par :

- Rechercher la table correspondante
- Ajouter la table
- Activer l'outil de dessin
- Sélectionner l'outil de dessin
- Tracer les informations correspondantes
- Enregistrement

- Confirmation

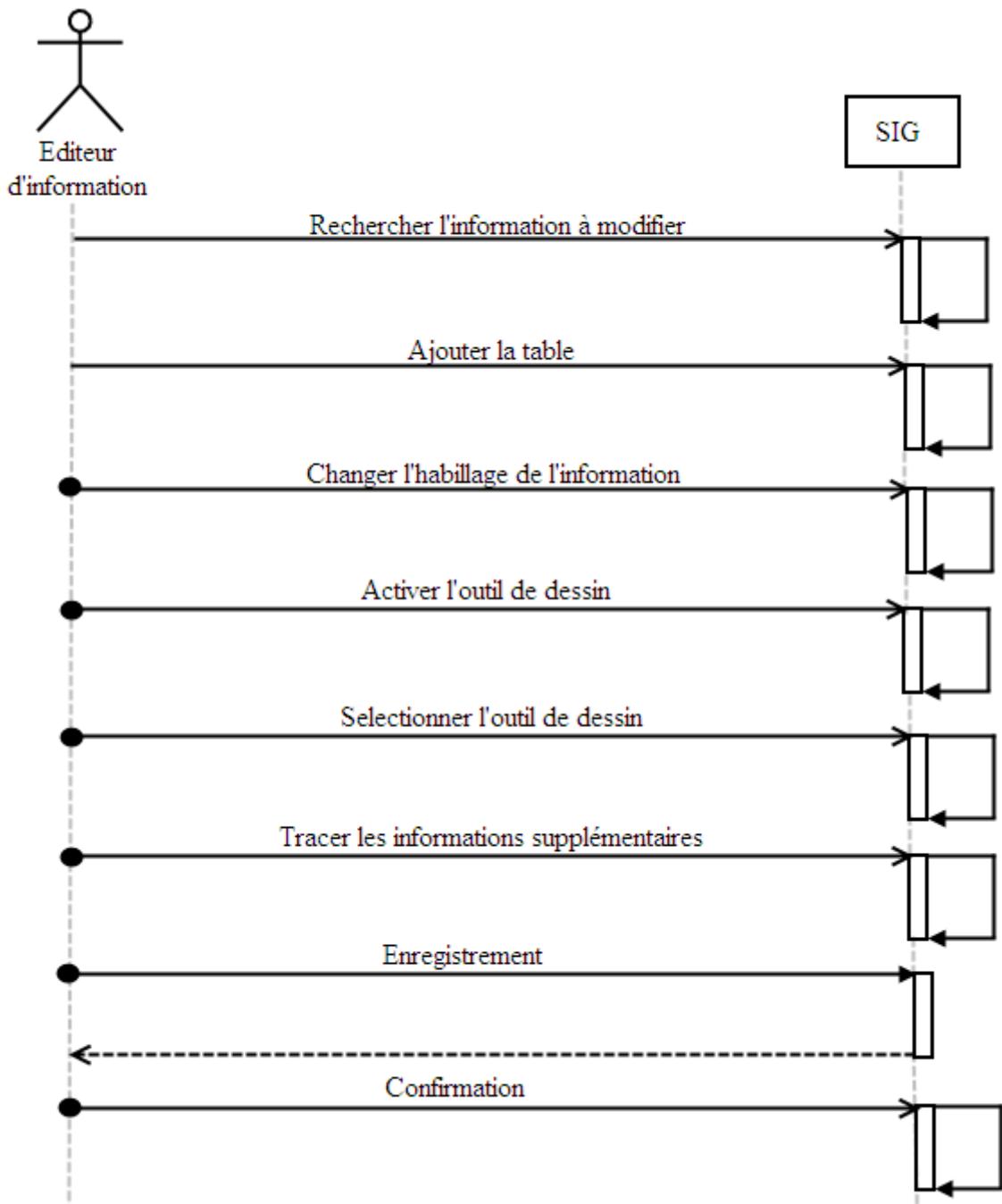


Figure III.11: Diagramme de séquence pour modifier les informations

La Figure III.11 ci-dessus montre le diagramme de séquence pour modifier les informations, lequel obéit au processus de manipulation spécifique selon l'éditeur d'information. Pour modifier les informations, on procède par :

- Rechercher l'information à modifier
- Ajouter la table
- Changer l'habillage de l'information

- Activer l'outil de dessin
- Sélectionner l'outil de dessin
- Tracer les informations supplémentaires
- Enregistrement
- Confirmation

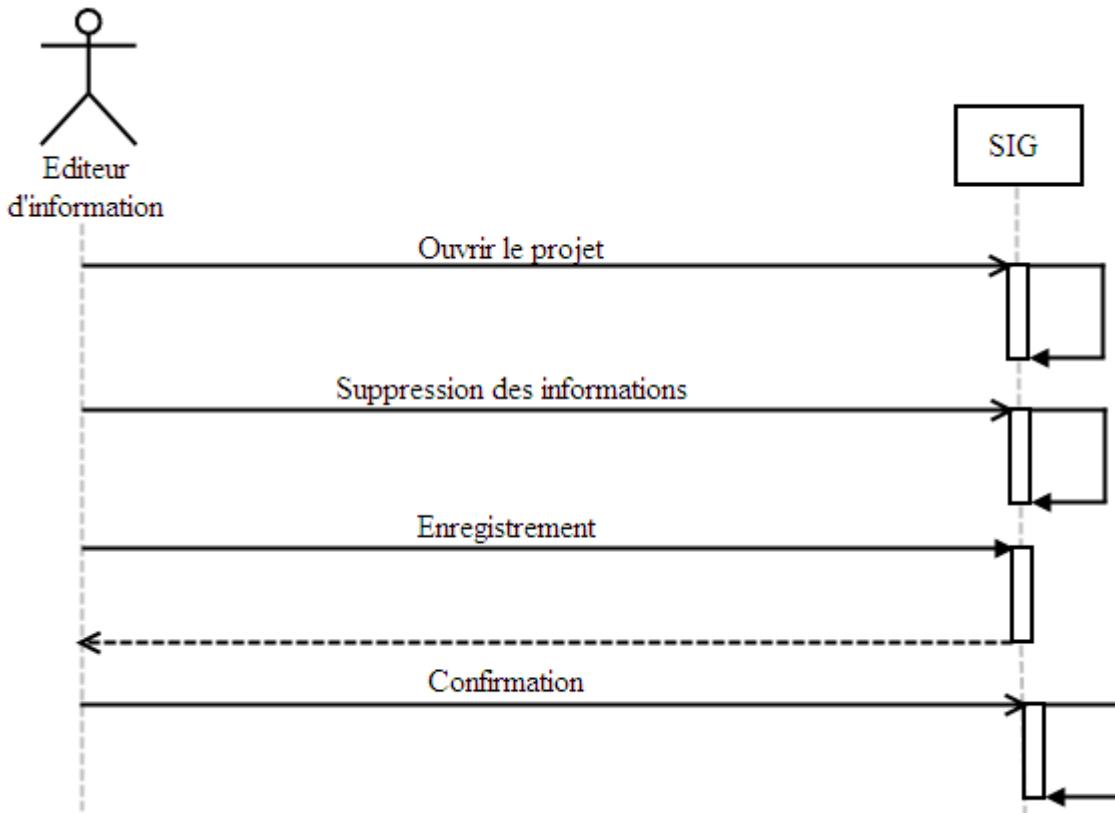


Figure III.12: Diagramme de séquence pour supprimer les informations

La Figure III.12 ci-dessus montre le diagramme de séquence pour supprimer les informations, lequel obéit au processus de manipulation spécifique selon l'éditeur d'information. Pour supprimer les informations, on procède par :

- Ouvrir le projet
- Supprimer les informations
- Enregistrement
- Confirmation

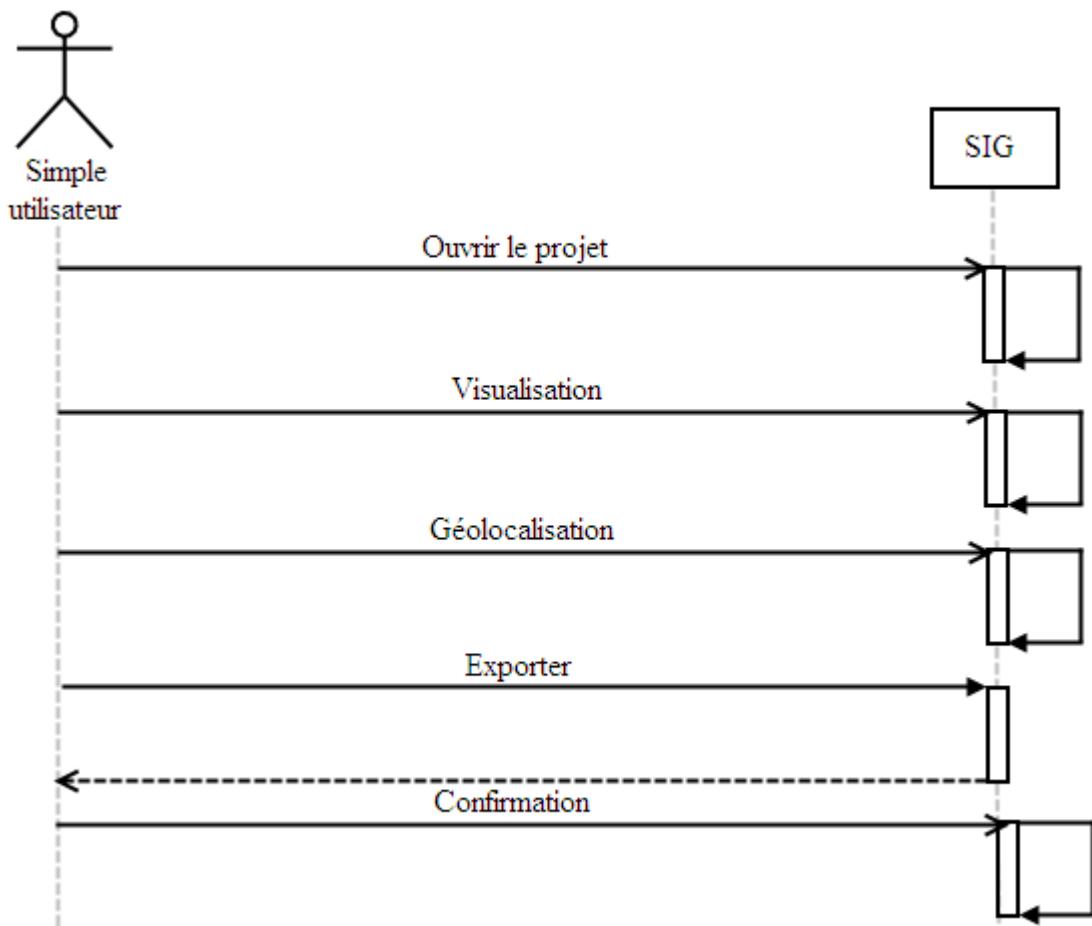


Figure III.13: Diagramme de séquence pour rechercher les données géographiques

La Figure III.13 ci-dessus montre le diagramme de séquence pour rechercher les données alphanumériques ou spatiales, lequel obéit au processus de manipulation spécifique suivant l'éditeur d'information. Pour rechercher les données alphanumériques ou spatiales, on procède par :

- Ouvrir le projet
 - Visualisation
 - Géolocalisation
 - Exporter
 - Confirmation
- c. diagramme de classe

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Les principaux éléments de cette vue statique sont les classes et leurs relations : association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation. La Figure III.14 est le

diagramme de classes de l'outil SIG et chaque classe doit avoir ces responsabilités. D'où les responsabilités de chaque classes :

- **FormMain:** Elle hérite de la classe JFrame, elle représente l'interface graphique de notre outil SIG.
- **Toc :** Elle hérite de la classe JTree, elle a pour rôle d'afficher la structure du projet, d'afficher les informations ou pas.
- **Canvas :** Elle hérite aussi de la classe JPanel, elle représente l'espace pour tracer, de déplacer, de zoomer les différentes formes.
- **Mysql :** elle gère l'ouverture, exécution des requêtes SQL et la fermeture de la base de donnée.
- **Table :** elle a pour rôle de stocker les données alphanumériques ou spatiales et chaque table est formée par un ou plusieurs champs.
- **TraiterTable :** elle prend les responsabilités d'ajouter, de supprimer, de modifier et de compléter la table, le table traiter est stocker temporairement, c'est la responsabilité de la classe enregistrement1.
- **Projet :** elle pour rôle d'ouvrir, de modifier ou de supprimer totalement le projet.
- **Transformation:** elle a pour rôle de faire le calcul de transformation des coordonnées géographiques Wgs84 en coordonnées géographiques qui est mesuré degré décimal et en coordonnées planes Laborde Madagascar en mètre.
- **Point :** elle a pour rôle de stocker les listes des coordonnées des points x et y dans la table.
- **Segment :** elle a pour rôle de stocker les listes de segment pour qu'on puisse tracer les polylignes et les polygones.
- **Polyligne :** elle a comme responsabilité de stocker les listes des polylignes dans la table.
- **Polygone :** elle a pour rôle de stocker les listes des polygones dans la table.
- **Habilage :** elle a pour rôle de prendre en charge la mise en forme des shape qui sont stockés dans la table.

- **Tracer** : elle prend la responsabilité de tracer tous les shape.
- **TraiterShape** : elle a pour rôle de faire le traitement des shape qui sont stocké temporairement, c'est la responsabilité de la classe enregistrement.

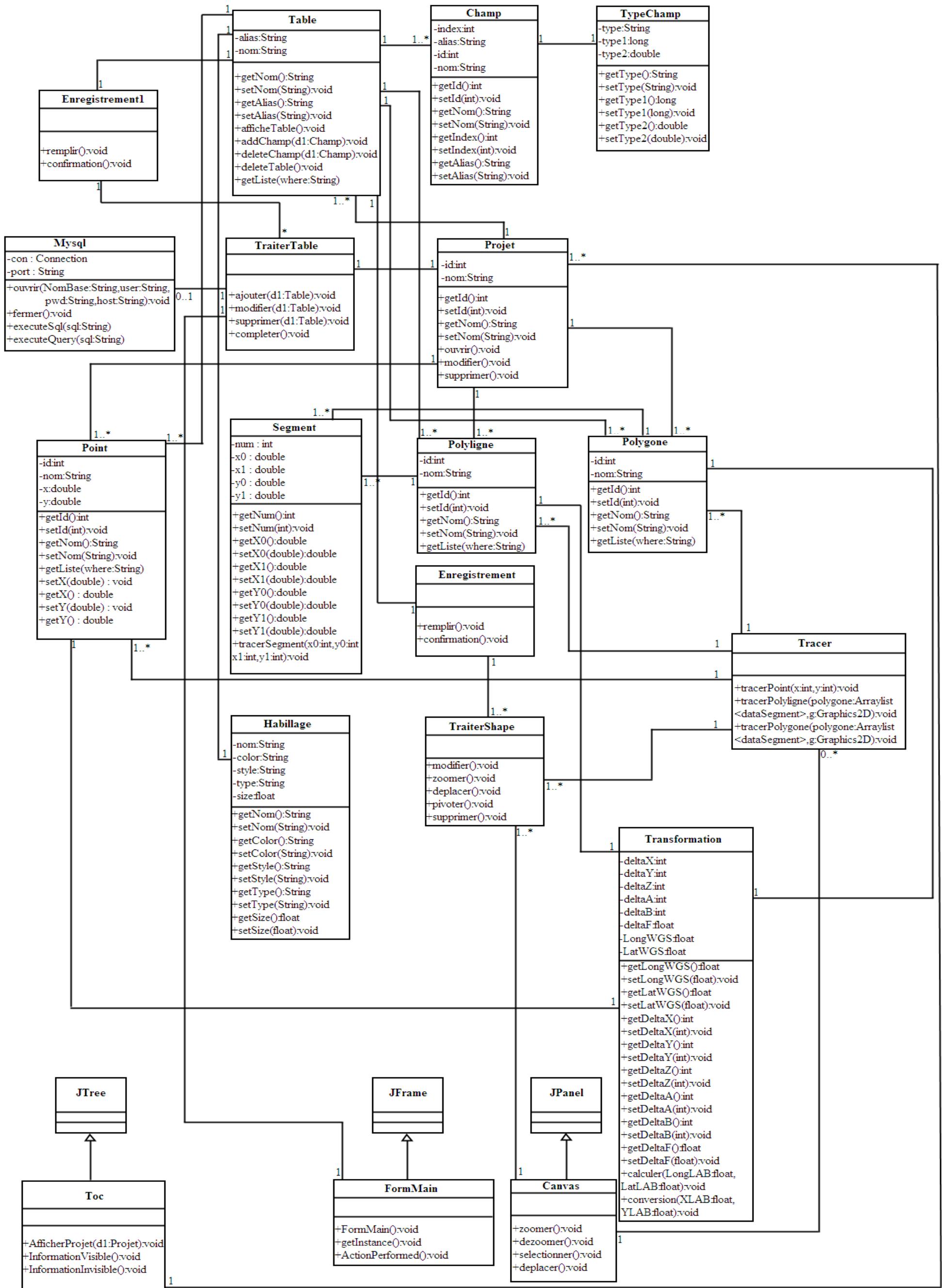


Figure III.14: Diagramme de classes de l'outil SIG

III.2. PRESENTATION DE L'OUTIL

Le logiciel est basé sur le langage Java et sur un système de gestion de base de données MySQL.

- Le choix de Java réside donc sur :
 - Licence gratuit
 - Langage orienté objet multi-plateforme
 - Connexion à des bases de données via JDBC (exemple : mysql-connector-java-5.1)
- Le choix de MySQL réside donc sur :
 - Licence gratuit
 - Adaptable au code Java
 - Stable et optimale
 - Possède des types champ Blob qui permet de stocker des géométries

Dans cet outil, on a deux applications indépendantes à savoir l'outil de gestion de catalogue des données (catalogue) qui gère les listes des tables et l'outil de traitement de carte (cartographie) qui a pour fonction de gérer des projets. Notre logiciel se nomme **TsT 1.0** (TANY SY NY TONTOLONY) et ayant comme logos représentés par les figures suivante :



Figure III.15: Logo de TsT 1.0

NB : Le 1.0 indique que ce logiciel est la première version

III.2.1. Gestion des catalogues des données

Pour lancer le programme :

- Double cliquer sur le fichier « catalogue.jar »

- la fenêtre principale de la gestion de catalogue de données apparaît, illustrée par la figure III.16 ci-dessous.

Cette fenêtre comporte des menus qui permet de :

- se connecter à la base de données
- gérer la liste des tables
- quitter l'application



Figure III.16: Fenêtre principale du catalogue

- a. Connexion à la base de données
- Cliquer sur le menu « Configurer source »
 - Dans le formulaire ci-dessous, remplir le nom du serveur, le mot de passe de l'utilisateur root du Mysql et le chemin de stockage des fichiers de configurations

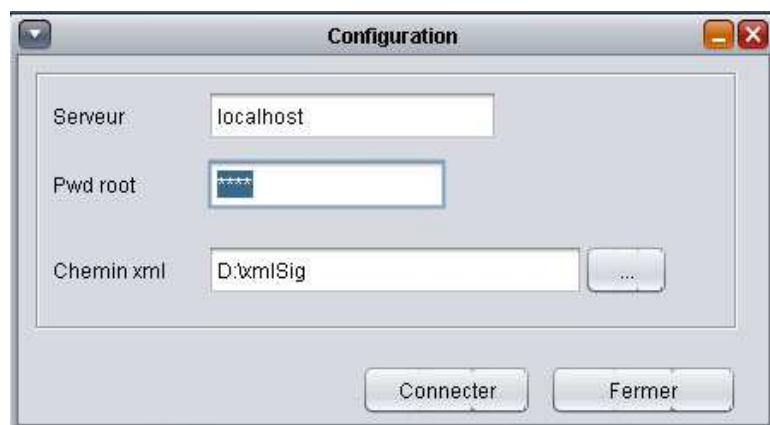


Figure III.17: Fenêtre de connexion à la base

- Cliquer sur le bouton « Connecter » pour appliquer les configurations et pour connecter au serveur de base de données.

b. Gestion des listes des tables



Figure III.18: Fenêtre de gestion des listes des tables

- Cliquer sur le menu « Tables » et puis « Liste »
- Dans le formulaire ci-dessus, on peut rechercher une table existante, ajouter une nouvelle table, modifier, consulter ou supprimer une table sélectionnée dans la liste



Figure III.19: Liste des tables

Pour ajouter une table :

- Cliquer sur le bouton « Ajouter »

- Dans le formulaire de gestion de table représenté par la figure ci-dessous, compléter les informations générales sur la table (nom, alias, type, description) ainsi que la liste des champs



Figure III.20: Gestion de table

Pour ajouter un champ dans l'onglet « Champs » :

- Remplir les informations sur le champ : nom, alias, type, valeur par défaut
- Cliquer sur le bouton « Ajouter »

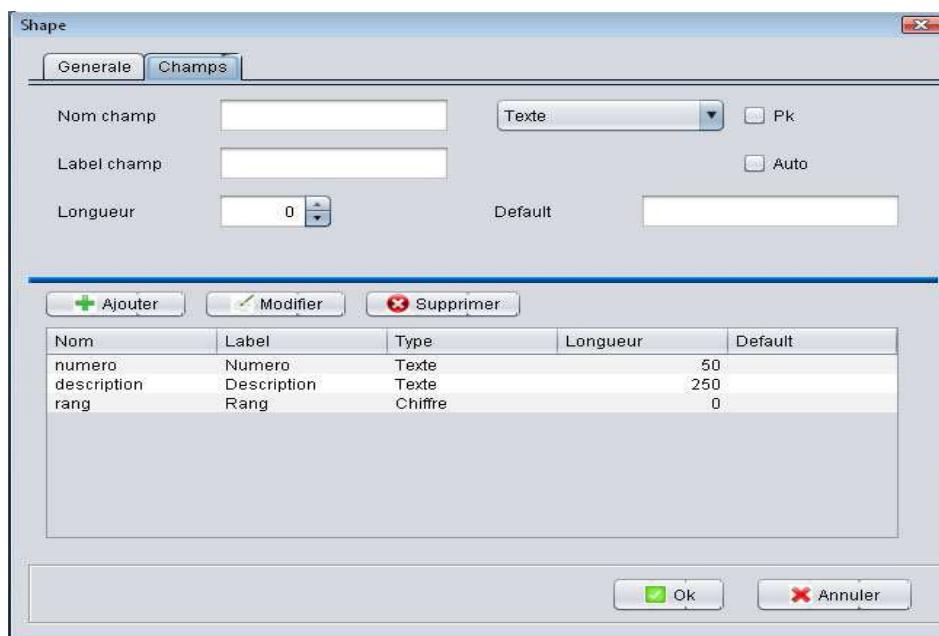


Figure III.21: Fenêtre de modification d'un champ

Pour modifier un champ :

- Cliquer sur une ligne dans la liste des champs
- Modifier les informations

- Cliquer sur le bouton « Modifier » pour appliquer les modifications

Remarque :

Le bouton « Ok » permet de recréer la table dans la base de données. Les anciennes données seront donc perdues au cours de la modification de la table.

III.2.2. Outil de traitement de carte

Pour lancer le programme :

- Double cliquer sur le fichier « Cartographie.jar »
- la fenêtre principale de la Carte apparaît

Cette fenêtre comporte des menus qui permet de :

- se connecter à la base de données
- créer nouveau projet
- ouvrir projet
- fermer projet
- connaître les informations et les données dans la table
- sélectionner ou rafraîchir ou agrandir ou rétrécir la visualisation des shapes sélectionnés
- éditer des shapes sélectionnés
- créer ou modifier ou supprimer des groupes de tables
- ajouter ou supprimer la table dans des groupes de table, insérer une image de fond et définir les caractéristiques du shape sélectionné
- donner les informations sur la géométrie du shape sélectionné
- quitter l'application

et deux zones d'affichage :

- la zone d'affichage à gauche permet de contenir les groupes de table
- la zone d'affichage à droite permet de dessiner les informations shapes



Figure III.22: Fenêtre principale de la cartographie

- a. Connexion à la base de données
- Cliquer sur le menu « Configurer source »
 - Dans le formulaire ci-dessous remplir le nom du serveur, le mot de passe de l'utilisateur root du Mysql et le chemin de stockage des fichiers de configurations.



Figure III.23: Fenêtre de connexion à la base

- Cliquer sur le bouton « Connecter » pour appliquer les configurations et pour connecter au serveur de base de données.

- b. Création d'un nouveau projet
- Cliquer sur le menu « Fichier » puis sur le sous menu « Nouveau projet » qui est représenté par la figure ci-dessous.

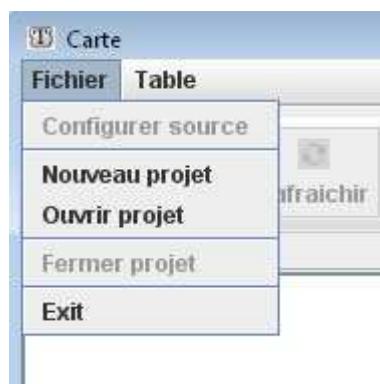


Figure III.24: Choix de créer ou d'ouvrir un projet

- Dans le formulaire ci-dessous « Gestion de projet » : tapez le nom, l'auteur, la date de création et la description du projet

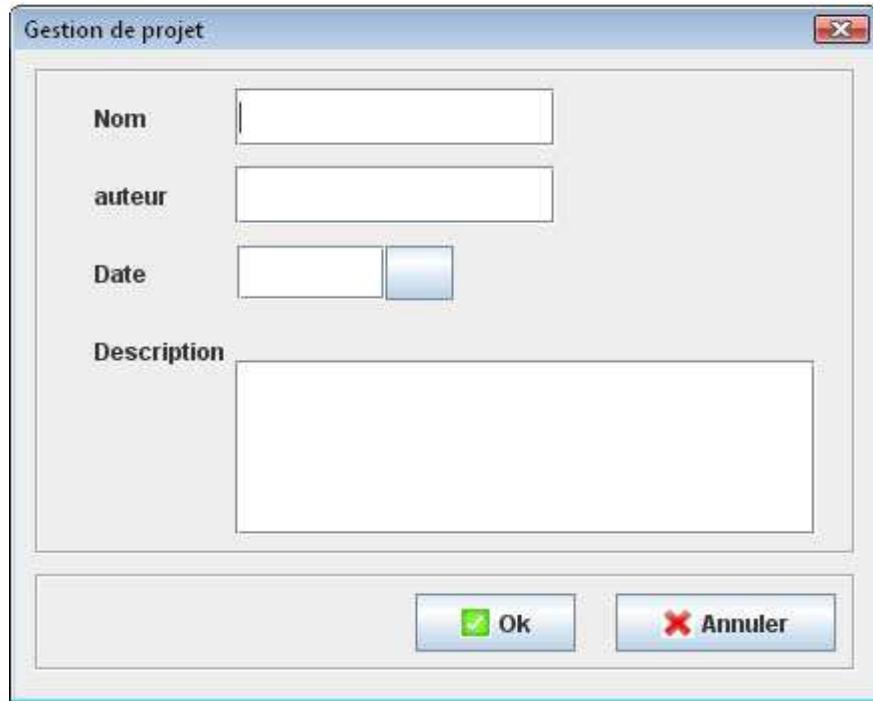


Figure III.25: Fenêtre de création d'un projet

- Appuyez sur le bouton « Ok » pour confirmer la création ou sur le bouton « Annuler » pour annuler l'opération
- c. Création des groupes de tables
- Cliquer sur le bouton « + Groupe » et saisir le nom du nouveau groupe dans le formulaire ci-dessous



Figure III.26: Crédit d'un nouveau groupe

- Appuyez sur « Ok » pour confirmer et un nouveau groupe apparaît sur la zone d'affichage à gauche illustré par la figure ci-dessous.

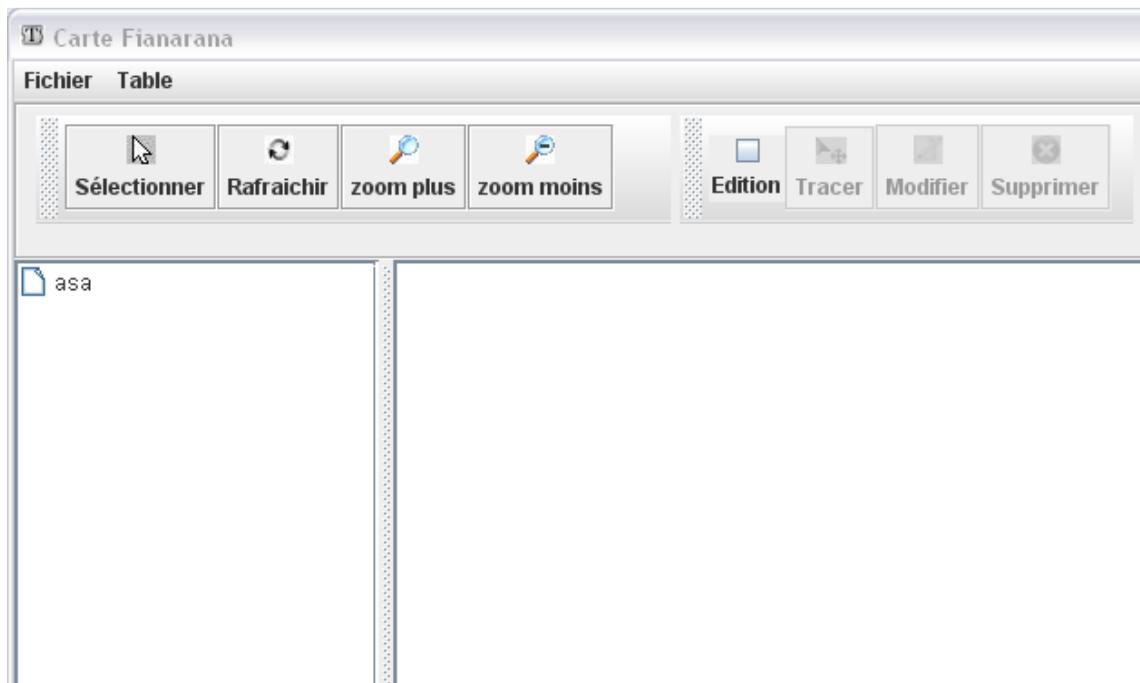


Figure III.27: Affichage de groupe de tables

- Ajout d'une table dans un groupe de table

Par le formulaire ci-dessous :

- Cliquer sur le bouton « + Table » et sélectionner la table

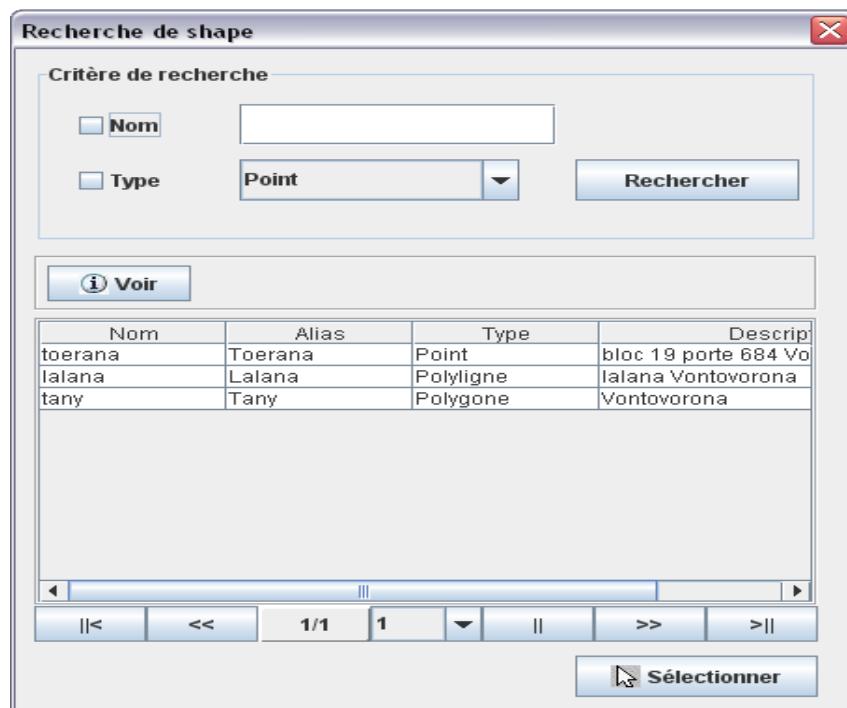


Figure III.28: Ajout d'une table dans un groupe

- Appuyer sur le bouton « Sélectionner » pour appliquer la sélection, puis la table sélectionnée s'affiche vers le groupe correspondant, illustré par la figure ci-dessous.

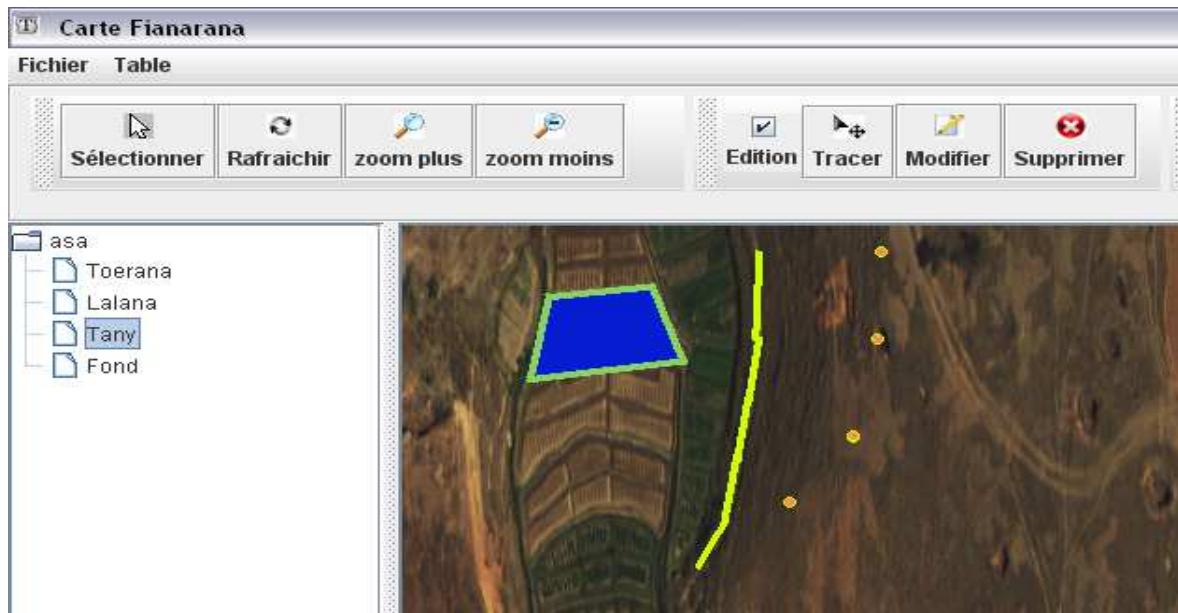


Figure III.29: Affichage d'une table dans un groupe avec traçage des shape

- Habillage d'un shape
- Sélectionner la table dans un groupe de tables, puis appuyer sur le bouton « Symbologie» et après choisir la couleur et la taille de la bordure, la couleur de remplissage et en fin la compacité du shape. Toutes ces étapes sont illustrées par la figure ci-dessous.

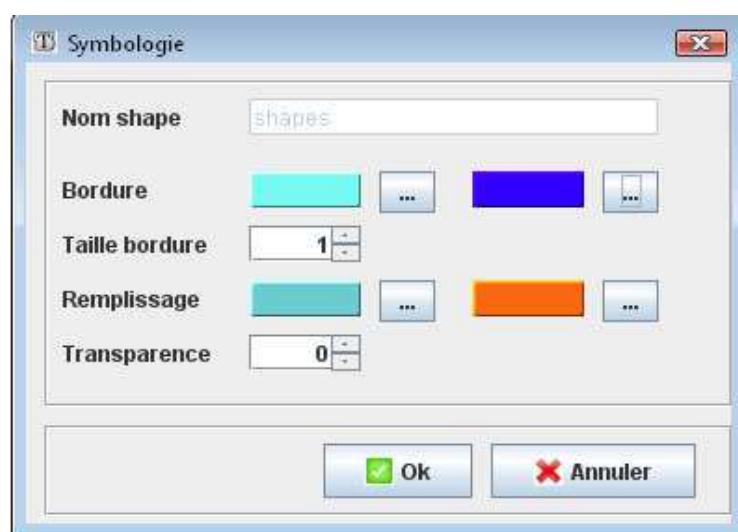


Figure III.30: Fenêtre d'habillage de shape

- En fin appuyer sur le bouton « Ok » pour appliquer les choix.

CONCLUSION

L’insuffisance du système de projection qui est indispensable pour Madagascar dans le logiciel géomatique existant nous a permis de modéliser et de concevoir des outils SIG compatible au système de projection Laborde.

La conception d’un outil de système d’information géographique commence par la maîtrise de gestion de flux de données ainsi que le système géographique qui conduit à la représentation de la projection Laborde. Deuxièmement, pour pouvoir réaliser des outils SIG, une méthodologie et environnement de conception a été proposé au deuxième chapitre. Cette méthodologie englobe toutes les étapes en commençant par le concept objet pour avoir une réflexion menée de la vie et de la qualité du logiciel. Ensuite un environnement de développement Java et pour finir le traitement shape. Enfin, la conception et la réalisation d’un outil de gestion de shape dans la projection Laborde, qui se termine par l’estimation du coût de la réalisation.

L’élaboration de ce travail de mémoire de fin d’étude nous a permis de savoir manipuler le logiciel SIG ArcGIS, d’implanter un outil de gestion et de décision qui répond au besoin de l’Etat Malgache ou d’un organisme.

Ce projet est loin d’être terminer puisque plusieurs améliorations sont prévues, par exemple implémentation d’un outil statistique et de mettre un architecture client serveur dans cet outil.

RÉFÉRENCES

- [1] «*Modélisation de Système d'information*», Pierre Gérard, <http://www-lipn.univ-paris13.fr/~gerard/docs/cours/.pdf>, Décembre 2010.
- [2] «*Système d'information géographique*», <http://fr.wikipedia.org/wiki/.htm>, Décembre 2010.
- [3] «*Formation de base en S.I.G. et Base de Données*», PRISMM (décembre 2004)/.pdf, Décembre 2010.
- [4] «*Réalisation d'un Système d'Informations Géographiques pour l'enseignement supérieur*», Aymen ABEDA, Ecole Supérieur des Communications de Tunis, 2006/2007/.pdf, Décembre 2010.
- [5] «*Qu'est ce qu'un système d'information géographique ?*», Élisabeth HABERT, Institut de recherche pour le développement, France, 2000, /.pdf, Décembre 2010.
- [6] «*Comprendre les projections*», <http://www.esrifrance.fr/.pdf>, Décembre 2010.
- [7] «*La programmation orientée objets*», SouheibBaarir/.pdf, Janvier 2011.
- [8] «*Génie logiciel : SIXIEME PARTIE : La modélisation objet.UML*», /.pdf, Jacques Lonchamp, CNAM - CRA Nancy, 2003, Février 2011.
- [9] «*Méthodologie UML* », DI GALLO Frédéric, CNAM ANGOULEME, /.pdf, 2000-2001, Février 2011.
- [10] «*UML 2- Laurent AUDIBERT*», <http://www.developpez.com/Cours-UML/html/.html>, Février 2011.
- [11] «*Conception de la visualisation 2D, Version1.0*», CALATRAVA Jérôme, ESTIENNE Sébastien, 2005, /.doc, Mars 2011.
- [12] «*Géométrie Analytique* »,deuxième année, tronc commun, ESPA, 2007.
- [13] «*Cours Métrologie* »,troisième année, option IGF, ESPA, 2010.
- [14] «*Projection cartographique Gauss-Laborde*», service de géodésie et nivellation, IGN, /.pdf, Janvier 1995, Décembre 2011.

AUTEUR : Monsieur ANDRIANANDRASANA Eric Benjamin

Contact : 032 44 843 60/034 66 577 55

E-mail : benjarakoto@gmail.com

TITRE : CONCEPTION DE LOGICIEL DE CARTOGRAPHIE SOUS LA PROJECTION LABORDE

Nombre de page : 75

Nombre de figures : 63

RESUME

Officiellement à Madagascar, la projection Laborde représente la base de calcul et de représentation de la cartographie. La définition et la méthode de construction de cette projection présentent une singularité qui pose des contraintes majeures quant à son emploi dans les logiciels de Géomatique. Rares sont les logiciels qui intègrent dans son fichier de configuration la projection Laborde de Madagascar.

Le but de notre étude est de concevoir un logiciel SIG flexible à la projection Laborde. La réalisation de la version 1.0 du logiciel TANY SY NY TONTOLONY (TsT 1.0) est assurée par le langage de programmation orienté objet (Java) et un système de gestion et conception de base de données (MySQL).

Mots clés : SIG, Projection LABORDE, Modélisation UML, Java

Rapporteur : RAZAFINDRAIBE Tovohery Andriampampony Alain

Adresse de l'auteur : Lot 61 BIS parcelle 13/81 Tanambao v TOAMASINA