

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

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.



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

Sujet :

**Site web de Navigation par
satellite en Algérie basé sur le
système GPS**

Présenté par : Moualek Nassim

Promoteur : M^{lle} Boustia
Encadreur : M^r Moualek

Organisme d'accueil : Catral.

Soutenue le: 16/10/2007

, devant le jury composé de :

M^{me} Wahrani ,
M^r Ferferra ,
M^r Bala ,

Président
Examineur
Examineur

MIG-004-178-A

Résumé

L'organisme d'accueil (la société CATRAL) a décidé de proposer un Système de Navigation par Satellite adapté à la géographie du territoire Algérien. Ce système doit se présenter sous forme d'un site web dynamique.

Ce dernier doit offrir la possibilité pour des abonnés de suivre un ou plusieurs véhicules à la fois en temps réel, tout comme il doit permettre de consulter l'historique du parcours des véhicules choisis sur une période donnée.

Abstract

The organism of reception (the CATRAL society) decided to propose a System of Navigation by Satellite adapted to the geography of the Algerian territory. This system has to appear under shape of a dynamic Web site.

This last one has to offer possibility for subscribers to follow one or several vehicles at the same moment in real time; also it allows to consult the historic of the route of vehicles chosen over a given period.

Table des matières

Introduction générale	01
-----------------------------	----

Chapitre I. Etat de l'art du GPS, des Systèmes d'informations géographiques (SIG), géo référencement et SIG interactifs en ligne (*Web Mapping*)

I. GPS (Système de positionnement global) :

I.1. Historique	03
I.2. Présentation du GPS	03
I.3. Les composants du GPS	04
I.4. Comment obtenir une position d'un GPS ?	05
I.5. Conclusion.....	07

II. SIG (Systèmes d'informations géographiques) :

II.1. Historique	08
II.2. Qu'est ce qu'un système d'information ?	09
II.3. Qu'est ce qu'un système d'information géographique ?	09
II.4. Qu'est ce qu'une information géographique?	09
II.5. Le référentiel cartographique :	
II.4.1. Le Géoïde terrestre	11
II.4.2. L'ellipsoïde de révolution.....	11
II.4.3. Le système de référence terrestre	12
II.4.4. Le système de référence WGS84 (World Géodetic System 1984)	12
II.6. Projection :	
II.6.1. Principes de base	13
II.6.2. Classement des systèmes d'après la surface de projection.....	13
II.6.3. Quelques projections (les plus connues)	14
II.7. Notion d'échelle	16

II.8. Représentation des données spatiales :	
II.8.1. La donnée « raster »	16
II.8.1. La donnée « vecteur ».....	17
II.9. Notion de couches	18
II.10. Conclusion	18

III. Géo-référencement :

III.1. Introduction.....	19
III.2. Points d’amers.....	19
III.3. Calage	19
III.4. Conclusion	22

IV. SIG interactifs en ligne (*Web Mapping*):

IV.1. Qu'est-ce qu'un serveur cartographique (serveur SIG) ?	23
IV.2. Principe générale de fonctionnement d'un serveur cartographique (serveur SIG)	24
IV.3. Les fonctions d'un serveur cartographique (serveur SIG).....	25
IV.4. Conclusion	26

Chapitre II. Démarche et développement du système

I. Démarche de développement	27
---	-----------

II. Analyse :

II.1. Expression initiale des besoins.....	29
II.2. Exigences fonctionnelles du site	29
II.3. Etude de l'existant.....	29
II.4. Redéfinition des besoins.....	31

III. Conception de l'outil :

III.1. Exigence fonctionnelle de l'outil	31
III.2. Identification des acteurs	32
III.3. Cas d'utilisation de l'outil de découpage.....	32
III.4. Diagramme de séquence:	
III.4.1. Configuration de la connexion vers la base de données :	
a. Scénario normal	34
b. Scénario d'exception.....	35
III.4.2. Caler une carte.....	36
III.4.3. Découper une carte déjà calée.....	37
III.5. Diagramme de classe	38
III.6. Diagramme d'état transition	41

IV. Conception du site :

IV.1. Rappel des Exigences fonctionnelles du site	42
IV.2. Identification des acteurs	42
IV.3. Cas d'utilisation :	
IV.3.1. Partie administration	43
IV.3.2. Partie Navigation.....	44
IV.4. Diagramme de séquence :	
IV.4. 1.Partie administration :	
a. Avant l'installation.....	45
b. Après l'installation.....	46
IV.4. 2.Partie navigation :	
a. Poursuite en temps réel.....	47
b. Historique des déplacements.....	49
IV.5. Diagramme de classe	51

V. Conception du Système :

V.1. Diagramme de collaboration.....	54
V.2. Diagramme de composants.....	55
V.3. Diagramme de déploiement.....	56

VI. Implémentation de l'outil :

VI.1. Contexte matériel et logiciel.....	58
VI.2. Concept de base de la bibliothèque « ECWJPEG2000SDK ».....	59
VI.3. Traitement de la contrainte principale de l'outil (taille des images).....	61
VI.4. Calage des cartes.....	62
VI.5. Connexion vers la base de données.....	63
VI.6. Indexation des fragments générés.....	64

VII. Implémentation du site :

VII.1. Contexte matériel et logiciel.....	65
VII.2. MapServer (La solution <i>Open Source</i>) :	
VII.2.1. Présentation de MapServer.....	66
VII.2.2. Origine de MapServer.....	66
VII.3. Présentation du langage PHP.....	67
VII.4. Organisation concrète d'un site internet :	
a. Un site Internet classique (statique).....	68
b. Un site Internet dynamique, avec php et MySQL.....	68
VII.5. Organisation concrète d'un serveur SIG basé sur PHP/MapServer.....	69
VII.6. Fichier de configuration mapfile.....	70
VII.7. Mécanisme de couches et de projections de MapServer.....	70
VII.8. Obtention, Installation et test de MapServer :	
VII.8.1. Obtention de MapServer.....	72
VII.8.2. Installation de MapServer.....	73
VII.8.3. Test de MapServer.....	73

VIII. Vision détaillée du développement de la totalité du système :	
VIII.1. Quelques décisions importantes pour implémenter le site.....	74
VIII.2. Suite de la conception de l’outil de découpage :	
VIII.2.1. Gérer une arborescence.....	79
VIII.2.2. Découper une carte déjà calée.....	81
VIII.3. Conception détaillée de la partie navigation du site :	
VIII.3.1. Poursuite en temps réel	83
VIII.3.2. Historique des déplacements.....	85
VIII.4. Suite des décisions concernant l’implémentation du site.....	87
VIII.5. Sécurité du site	89
IX. Test et validation :	
IX.1. Outil de découpage	90
IX.2. Site de Navigation :	
IX.2.1. Partie administration	
IX.2.1.1. Gérer les kits GPS.....	93
IX.2.1.2. Gérer les clients.....	93
IX.2.1.3. Gérer les abonnements.....	93
IX.2.2. Partie Navigation	94
Conclusion.....	95
ANNEXE.....	98
RÉFÉRENCES.....	114

Tables et figures

Chapitre I. Etat de l'art du GPS, des Systèmes d'informations géographiques (SIG), géo référencement et SIG interactifs en ligne (*Web Mapping*)

Figure I.1 : Les composants du GPS.....	04
Figure I.2 : La constellation des satellites Navstar.....	04
Figure I.3 : Principe de localisation du GPS.....	06
Figure I.4 : l'objet géographique support et/ou référentiel.....	10
Figure I.5 : Le géoïde est la forme théorique de la terre.....	11
Figure I.6 : L'ellipsoïde de révolution.....	11
Figure I.7 : La construction du référentiel géographique.....	12
Figure I.8 : Système de référence WGS84.....	12
Figure I.9 : Le globe terrestre vu comme un ballon dé gonflable, qui peut se ramener à un plan.....	13
Figure I.10 : Surfaces de projections.....	13
Figure I.11 : Projection conforme.....	14
Figure I.12 : Projection équivalente.....	14
Figure I.13 : Projection conique conforme de Lambert.....	14
Figure I.14 : Projection cylindrique de Mercator.....	15
Figure I.15 : Projection UTM divisant la France en 3 fuseaux.....	15
Figure I.16 : Projection UTM divisant l'Algérie en 4 fuseaux.....	15
Figure I.17 : Donnée où l'espace est divisé de manière régulière.....	16
Figure I.18 : Photo aérienne.....	16
Figure I.19 : Carte scannée.....	17
Figure I.20 : Image satellite.....	17
Figure I.21 : Image satellite radar.....	17
Figure I.22 : Objets géométriques.....	18
Figure I.23 : Organisation en couches.....	18
Figure I.24 : Organisation d'un site web classique.....	23
Figure I.25 : Organisation d'un site web orienté SIG.....	23
Figure I.26 : Organisation d'un Serveur cartographique.....	24

Chapitre II. Démarche et développement du système

Figure II.1 : Le modèle développement en cascade	27
Figure II.2 : Différentes parties d'une carte scannée.....	30
Figure II.3 : Ensemble des cas d'utilisation de l'outil de découpage.....	33
Figure II.4 : Diagramme de séquence de la configuration de la connexion vers la base de données (scénario normal)	34
Figure II.5 : Diagramme de séquence de la configuration de la connexion vers la base de données (scénario d'exception).....	35
Figure II.6 : Diagramme de séquence caler une carte	36
Figure II.7 : Diagramme de séquence « découper une carte déjà calée ».....	37
Table II.1 : Dictionnaire de données de l'outil de découpage	38
Figure II.8 : Diagramme de classe de l'outil de découpage	40
Figure II.9 : Diagramme d'état transition des différents objets manipulés par l'outil	41
Figure II.10 : Cas d'utilisation de la partie administration du site	43
Figure II.11 : Cas d'utilisation de la partie navigation du site	44
Figure II.12 : Diagramme de séquence de la partie administration (avant installation).....	45
Figure II.13 : Diagramme de séquence de la partie administration (après installation).....	46
Figure II.14 : Déduction de la vue projetée (cas « Historique des déplacements »)	47
Figure II.15 : Diagramme de séquence de la partie navigation (Poursuit en temps réel).....	48
Figure II.16 : Déduction de la vue projetée (cas « Historique des déplacements »)	49
Figure II.17 : Diagramme de séquence de la partie navigation (Historique des déplacements).....	50
Table II.2 : Dictionnaire de données du site de navigation	51
Figure II.18 : Diagramme de classe du site de navigation	53
Figure II.19 : Diagramme de collaboration de la totalité du système	54
Figure II.20 : Diagramme de composants de la totalité du système.....	55
Figure II.21 : Diagramme de déploiement.....	57
Figure II.22 : Concept « SetFileView ».....	60
Table II.3 : Registres d'un alias ODBC.....	64
Figure II.23 : Organisation concrète d'un serveur SIG basé sur PHP/MapServer.....	69
Figure II.24 : Diagramme de séquence Gestion d'une arborescence	80
Figure II.25 : Diagramme de séquence « découper une carte déjà calée ».....	82
Figure II.26 : Diagramme de séquence de la partie navigation (Poursuite en temps réel).....	84
Figure II.27 : Diagramme de séquence de la partie navigation (Historique des déplacements).....	86
Figure II.28 : Authentification-http	89
Figure II.29 : Interface principale de l'outil de découpage pendant le calage.....	90
Figure II.30 : Interface principale de l'outil de découpage pendant le découpage.....	91
Figure II.31 : Fiche « Gestion d'arborescence »	76
Figure II.32 : Fiche « Configuration de la connexion vers la base de données ».....	92
Figure II.33 : Page « Gérer les kits GPS ».....	93
Figure II.34 : Page « Gérer les clients »	93
Figure II.35 : Page « Gérer les abonnements ».....	93
Figure II.36 : Page navigation	94

Introduction générale

Un système de navigation par satellite est un ensemble d'éléments matériels et logiciels qui, à partir des données transmises par un ou plusieurs satellites de navigation, interagissent afin de fournir en temps réel la position d'un véhicule (et éventuellement aussi sa vitesse, accélération, température... etc.) sur une carte géographique.

Les applications pratiques qui en découlent sont nombreuses : avec ce système on peut par exemple connaître la position d'un véhicule volé, répondre efficacement à un appel d'urgence (penser aux camionneurs qui font des milliers de kilomètres seuls dans le désert Algérien), connaître et sanctionner les excès de vitesse d'un conducteur, ... etc.

Cette technologie est apparue dans les Etats Unis d'Amérique depuis une trentaine d'années dans le domaine militaire et s'est étendue au domaine civil depuis une quinzaine d'années, grâce d'une part au développement d'Internet et des réseaux mobiles de télécommunications GSM (Global System for Mobile communication) et GPRS (General Packet Radio Services) et d'autre part suite à l'intégration des kits GPS (Système de Positionnement Global) dans de nombreuses marques automobiles.

Si dans le passé les ingrédients n'étaient pas réunis pour que cette technologie fasse son apparition en Algérie, il en est autrement à présent. En effet, nous assistons depuis une dizaine d'années à une multiplication des offres de télécommunication mobile avec une réelle couverture nationale. Il est donc possible, maintenant, de recevoir une donnée d'un satellite de navigation n'importe où en Algérie ; cette donnée est ensuite transmise à partir du véhicule en mouvement grâce au kit GPS installé à bord et ceci via une puce d'un opérateur Télécom vers un serveur informatique chargé de stocker cette information dans une base de données.

Cette dernière va constituer le point de départ du projet que nous souhaitons réaliser. Ainsi donc, étant donné une base de données contenant l'historique des positions de chaque véhicule suivi par GPS, nous proposons de concevoir et de réaliser un site web dynamique qui permet à chaque propriétaire d'un véhicule de suivre en temps réel les déplacements de son

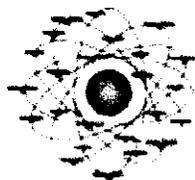
véhicule sur la carte géographique de l'Algérie et ceci à différentes échelles (c.à.d. différents niveaux de détails selon le zoom appliqué). Le site web doit également permettre à l'utilisateur de voir une séquence de déplacements enregistrée dans le passé.

Nous présenterons notre travail selon la structure suivante :

Chapitre I : Nous aborderons quelques domaines essentiels à la réalisation de notre système tel que le GPS, les Systèmes d'informations géographiques (SIG), le géo-référencement et SIG interactifs en ligne (*Web Mapping*).

Chapitre II : Nous présenterons la démarche de développement de notre système, puis nous concevrons ce dernier afin de l'implémenter, et pour finir nous résumerons et critiquerons le résultat final, en vue de suggérer des futures améliorations.

Chapitre I. Etat de l'art



I. GPS

I.1. Historique :

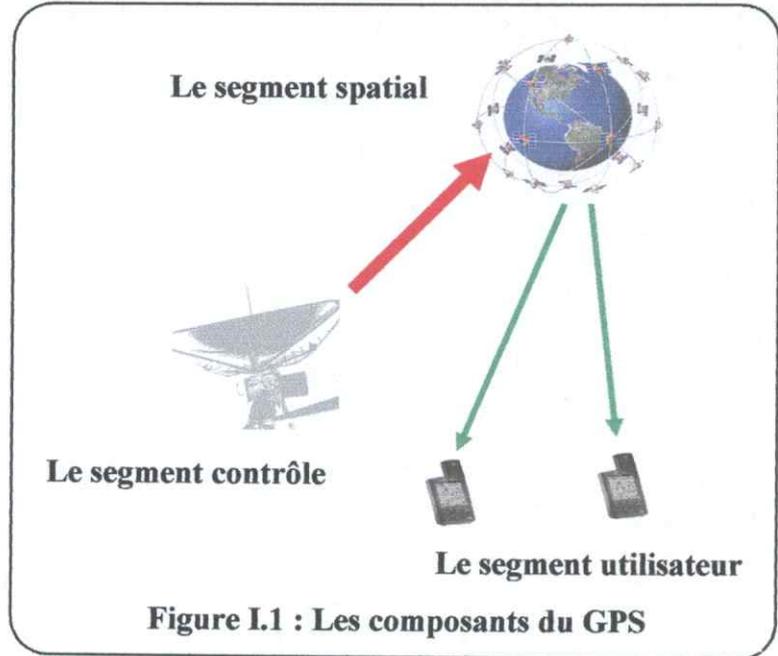
Le GPS ou système de positionnement global a été développé par le département de la défense américaine en 1978. Il a été rendu accessible au public en 1985. Il permet le positionnement précis de n'importe quel lieu quelque soit l'heure et les conditions météorologiques.

Les premiers réseaux GPS permanents ont été installés en Californie en 1991 afin de surveiller la faille de San Andréas. Puis d'autres furent créés. L'ensemble est actuellement fédéré par une organisation mondiale, l'IGS (The International GNSS (Global Navigation Satellite System) Service). [A.CAZ, 1994]

I.2. Présentation du GPS.

Le système de Positionnement Global (GPS) est un système de navigation par satellites destiné à fournir la position, la vitesse et l'heure n'importe où à la surface de la Terre ou dans ses environs. La précision du positionnement par GPS va de 100 mètres à quelques mètres en temps réel. Elle peut atteindre quelques millimètres pour les applications de géodésie en temps différé. Bien qu'il y ait des milliers d'utilisateurs civils du GPS à travers le monde, le GPS a été créé et est contrôlé par le département de la défense américain (U.S. Department of Defense (DoD)). [A.CAZ, 1994]

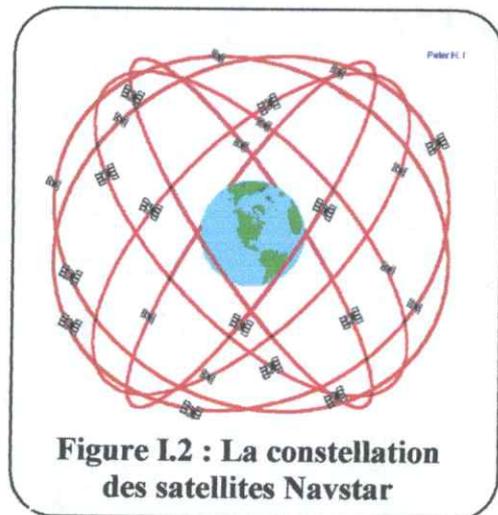
I.3. Les composants du GPS :



Le système GPS peut se décomposer en trois segments:

- **Le segment spatial :** Le segment spatial est composé d'un réseau de 28 satellites en orbite quasi-circulaire autour de la Terre, à une hauteur de +/- 20000 km et dont la période de révolution est de 12 heures sidérales. Ces satellites sont répartis sur 6 plans orbitaux inclinés à 55° par rapport à l'équateur.

La constellation de satellites **Navstar** du système GPS a été conçue de telle manière que, partout sur Terre et à tout moment, un minimum de 4 satellites soient visibles. [A.CAZ, 1994]



Il existe trois catégories de satellites GPS:

- ✓ Bloc I: Satellites lancés entre 1978 et 1985. Tous les satellites du Bloc I sont maintenant hors-service excepté un seul, qui est activé de manière périodique. Leur durée de vie est de 4,5 ans. La principale différence entre ces satellites et les générations suivantes est l'impossibilité de dégrader volontairement le signal transmis.

- ✓ Bloc II et IIa: Seconde génération de satellites GPS mis sur orbite à partir de 1985. Ils ont la capacité de dégrader le signal émis. leur durée de vie est de 7,5 ans.
- ✓ Bloc IIr: Ils ont été construits pour avoir une durée de vie de 10 ans. Ils sont capables de communiquer entre eux et ont été mis sur orbite depuis 1996 afin de maintenir une constellation complète.
- **Le segment de contrôle:** Le segment de contrôle comprend cinq stations de poursuite situées à Colorado Springs, Hawaii, Ascension Island, Diego Garcia et Kwajalein. Ces stations sont les yeux et les oreilles du système GPS, elles vérifient l'état des satellites lors de leur passage au dessus d'elles. Ces stations transmettent ensuite leurs données à la station principale de Colorado Springs. C'est là que les paramètres décrivant l'orbite des satellites et la qualité des horloges embarquées sont estimés, la vérification de la santé des satellites et la détermination d'un repositionnement éventuel sont également contrôlés. Cette information est alors renvoyée à trois stations de chargement qui transmettent l'information aux satellites. Grâce à la répartition uniforme des stations de contrôle, tous les satellites GPS sont captés à 92% du temps.
- **Le segment utilisateur:** Il est composé des récepteurs GPS, appareils permettant de calculer la position, la vitesse et l'heure d'un observateur, partout et à tout moment sur Terre et dans ses environs, quelles que soient les conditions climatiques. Un récepteur décode le signal transmis par les satellites pour déterminer la position, la vitesse et le temps de l'utilisateur. [A.CAZ, 1994]

I.4. Comment obtenir une position d'un GPS ?

Les satellites GPS envoient des signaux codés qui peuvent être déchiffrés par des récepteurs GPS. Le récepteur mesure la distance entre la station et plusieurs satellites.

La vitesse de transmission des signaux émis par les satellites est égale à celle de la lumière. Chaque signal intègre une éphéméride avec son heure de départ. On mesure donc la distance entre l'utilisateur et un certain nombre de satellites de positions connues

grâce au temps qu'a mis chaque signal à parvenir jusqu'à votre GPS. On définit ainsi des sphères centrées sur des satellites et dont l'intersection donne la position.

Le récepteur GPS est capable d'identifier le satellite qu'il utilise à l'aide du signal pseudo aléatoire émis par chaque satellite. Il charge, à l'aide de ce signal, les informations sur l'orbite et la position du satellite. Pour mesurer la distance qui sépare le satellite du GPS, on mesure le **temps T** mis par le signal pour aller de l'un vers l'autre.

Distance= Vitesse de la lumière * temps T.

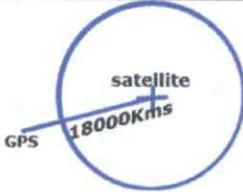
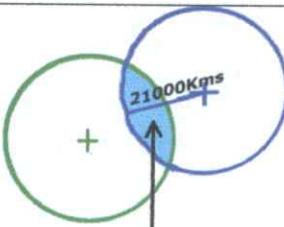
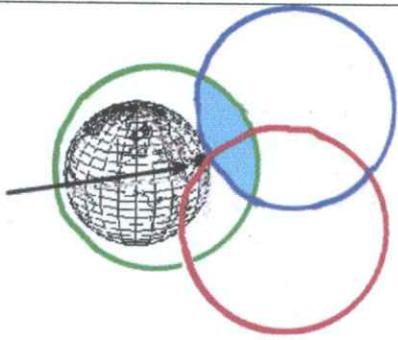
	temps T	Calcul de la distance	Déduction de la position du kit GPS
Premier satellite	0,06 seconde	Distance=300000 *0,06 = 18000Km	
Deuxième satellite	0,07 Seconde	Distance=300000 *0,07 = 21000Km	 Le kit GPS est quelque part ici
Troisième satellite	0,05 Seconde	Distance=300000 *0,05 = 15000Km	 Le kit GPS est ici

Figure I.3 : Principe de localisation du GPS.

Le système GPS permet également de mesurer la vitesse à laquelle se déplace l'utilisateur. Cette évaluation se fait par mesure de Doppler sur le signal provenant d'un satellite GPS. En effet, le signal perçu par le récepteur GPS n'a pas exactement la même fréquence que lorsqu'il est généré par le satellite. Suivant le principe Doppler, le rapport des fréquences est fonction des positions et des vitesses relatives du satellite et de l'utilisateur.

Donc chaque mesure de distance sur un satellite définit une sphère dont le centre est le satellite. Trois satellites sont donc nécessaires pour déterminer la position du récepteur en trois dimensions. Un quatrième satellite est nécessaire pour tenir compte de la non-synchronisation de l'horloge du récepteur et des horloges des satellites. [A.CAZ, 1994]

I.5. Conclusion :

Pour conclure ce chapitre, soulignons que les Européens sont en voie de construire leur système civil de navigation par satellites appelé **Galileo**. Celui-ci devrait être opérationnel autour de 2008-2010. Sans oublier le système russe **GLONASS** qui lui devrait être remis à niveau avec une nouvelle constellation complète de 24 satellites en 2010. L'intégration de ces 3 systèmes de positionnement et de navigation par satellites (aussi nommé GNSS : *Global Navigation Satellite System*) offrira une plus grande couverture, une plus grande intégrité et une meilleure précision des résultats.

La révolution se poursuit donc, et le nombre d'applications dans le domaine du positionnement et de la navigation ne sera limité que par notre imagination...

[R, SAN 1996]

II.2. Qu'est ce qu'un système d'information ?

Un ensemble organisé de ressources (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.

Selon leur finalité principale, on distingue des systèmes d'information supports d'opérations (traitement de transaction, contrôle de processus industriels, supports d'opérations de bureau et de communication) et des systèmes d'information supports de gestion (aide à la production de rapports, aide à la décision...).

II.3. Qu'est ce qu'un système d'information géographique ?

« Un Système d'informations géographiques (SIG) est un système informatique permettant, à partir de diverses sources, de rassembler et organiser, de gérer, d'analyser et de combiner, d'élaborer et de présenter des informations localisées géographiquement contribuant notamment à la gestion de l'espace » [CSCI, 1990]

Il peut être considéré aussi comme 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 de données à référence spatiale afin de résoudre des problèmes complexes d'aménagement et de gestion. [CFCI, 1988]

II.4. Qu'est ce qu'une information géographique (IG) ?

L'information géographique est un ensemble reliant une information de type sémantique (relative à un objet ou à un phénomène du monde terrestre) et une information de type géométrique liée à la localisation de cet objet ou de ce phénomène. C'est donc la représentation d'un objet ou d'un phénomène géographique *localisé dans l'espace*.

[T, CON 2006]

Donc la donnée est géographique dès lors quelle est localisable directement par des coordonnées, ou indirectement par des données littérales de type adresse, numéro de commune, numéro de borne kilométrique, code postal, numéro de parcelle cadastrale, ...



Figure I.4 : l'objet géographique support et/ou référentiel.

Elle peut être dupliquée sans dégradation, elle circule et s'échange à grande vitesse via les réseaux de communication, elle se combine avec d'autres informations en vue d'en créer de nouvelles.

Les informations géographiques sont acquises, stockées, analysées, visualisées et distribuées à l'aide de systèmes d'information géographique. [CIGS, 2001]

Une Information localisée sur quoi ?

II.5. Le référentiel géographique :

Le référentiel géographique est l'ensemble de conventions qui permettent d'associer à tout point d'une partie de la surface terrestre un point unique sur une carte.

La terre est une sphère, mais une sphère imparfaite.

Traditionnellement, pour représenter la surface terrestre on utilise une représentation de la surface qui altère aussi peu que possible les propriétés métriques du terrain (distance entre les points de la surface terrestre, angles, etc.). Le schéma le plus couramment employé est présenté à la **Figure I.7**.

Le référentiel géographique se base sur le **géοïde** terrestre et la définition d'un **ellipsoïde**.

II.5.1. Le géoïde terrestre:

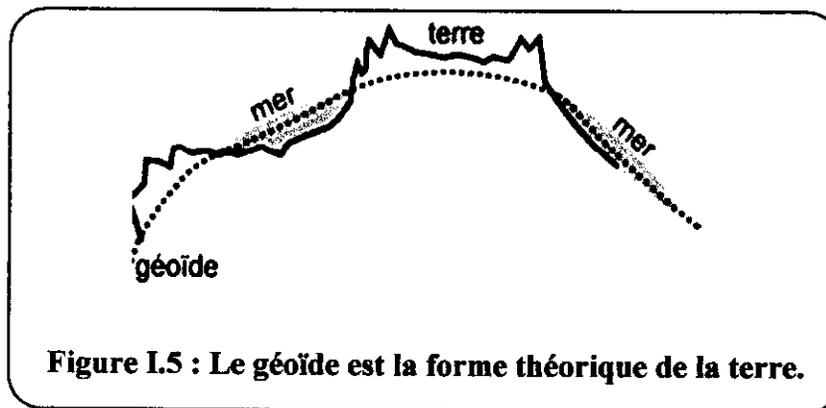


Figure I.5 : Le géoïde est la forme théorique de la terre.

- Le **géoïde** est la forme théorique qui se rapproche le plus de la surface réelle de la terre c'est-à-dire le niveau moyen des mers. Il sert de référence pour déterminer les altitudes.

II.5.2. L'ellipsoïde de révolution:

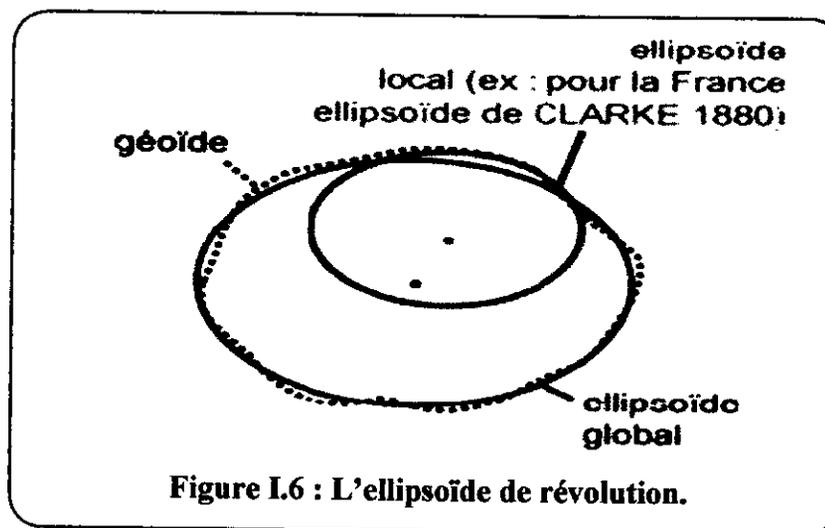


Figure I.6 : L'ellipsoïde de révolution.

- L'**ellipsoïde** est la surface mathématique qui se rapproche le plus de la forme du géoïde ; grâce à ce dernier, on peut calculer les coordonnées géographiques en **LONGITUDE** et en **LATITUDE**.

L'espace géométrique au voisinage de la Terre est parfaitement décrit par trois dimensions. Cette situation conduit naturellement à introduire un système d'axes cartésien centré au centre de la Terre tournant avec la Terre. Ce système d'axe est appelé **système de référence terrestre**. [P, SIL 2000]

II.5.3. Le système de référence terrestre :

Il permet de repérer un point au voisinage de la Terre (donc à sa surface) par trois coordonnées $(x; y; z)$. Sa construction est essentiellement implicite car aucun système de mesure ne permet directement de donner des positions exprimées dans le système terrestre. Les positions résultent toujours d'un calcul effectué à partir d'observations de triangulation. Jusque dans les années 1970, les observations de base étaient faites au sol. Depuis cette époque, les observations sont faites à partir de système spatial (satellites pour l'essentiel). [Les projections et référentiel cartographique_ Patrick SILLARD]

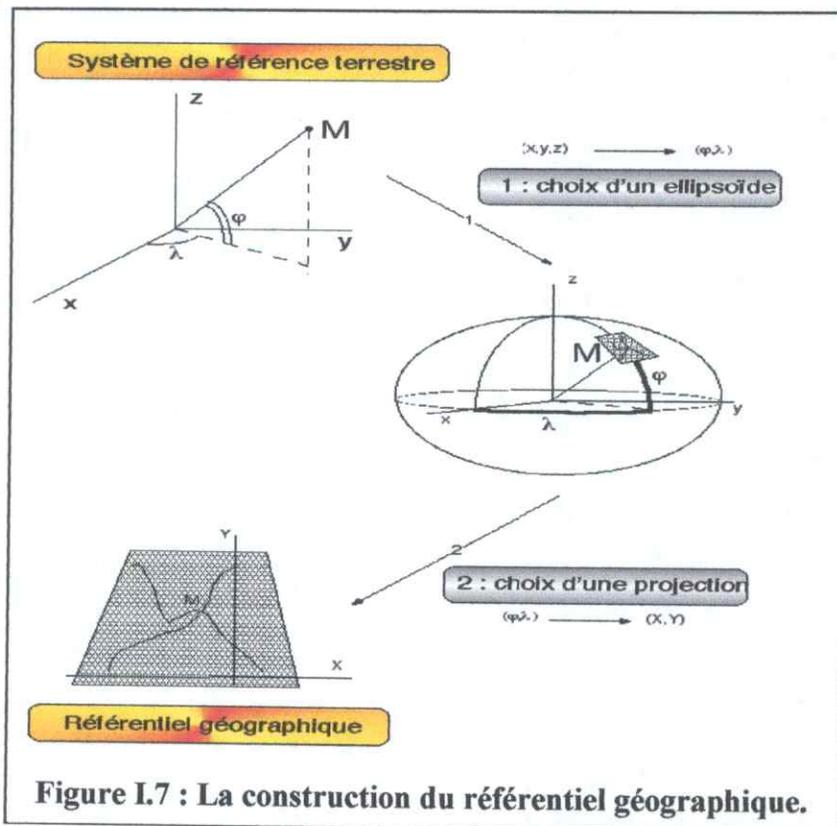


Figure I.7 : La construction du référentiel géographique.

II.5.4. Le système de référence WGS84 (World Geodetic System 1984) le plus utilisé dans le monde:

Système mis au point par le département de Défense US, utilisé par les kits GPS, se base sur des coordonnées en latitude, longitude.

L'exactitude de ce système est de l'ordre du mètre.

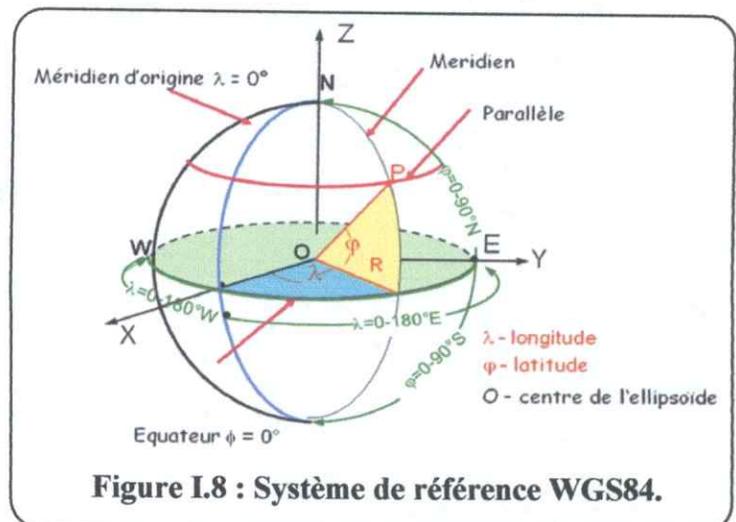


Figure I.8 : Système de référence WGS84.

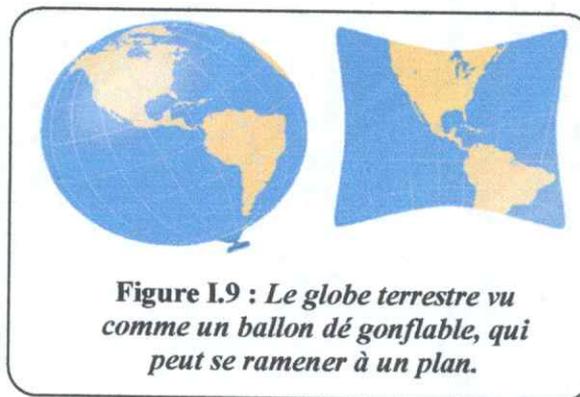
- Le procédé mathématique qui permet le passage de l'ellipsoïde au plan se nomme : **système de représentation plane ou système de projection ou projection.** [P, SIL 2000]

II.6. Projection :

II.6.1. Principe de base :

Pour les besoins cartographiques, on doit représenter sur une surface plane l'image de la terre assimilée à un ellipsoïde, ce qui nécessite l'utilisation d'une représentation plane (ou projection).

Les coordonnées planes ainsi obtenues permettent des mesures directes sur la carte (angles, surfaces) mais toutes représentations planes engendrent des déformations (les distances ne sont jamais conservées).



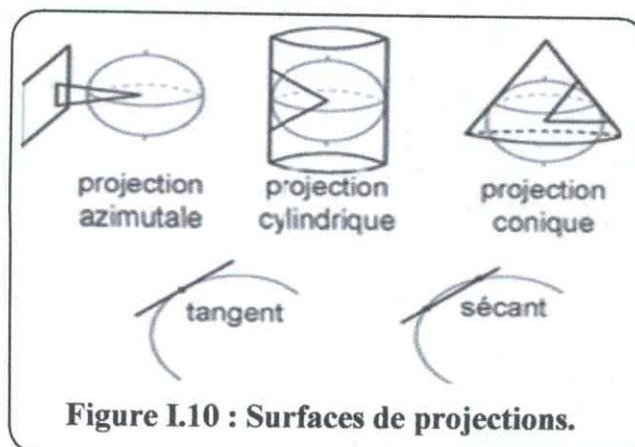
On peut définir une projection comme un moyen de correspondance analytique entre les points (latitude, longitude) de la surface à représenter (l'ellipsoïde terrestre) et les points homologues du plan cartographique (X, Y), telle que cette correspondance soit continue et biunivoque.

$$X = f(\text{latitude}, \text{longitude})$$

$$Y = g(\text{latitude}, \text{longitude})$$

f et g sont deux fonctions continues qui définissent la projection. [J.GIL 2000]

II.6.2. Classement des systèmes d'après la surface de projection :



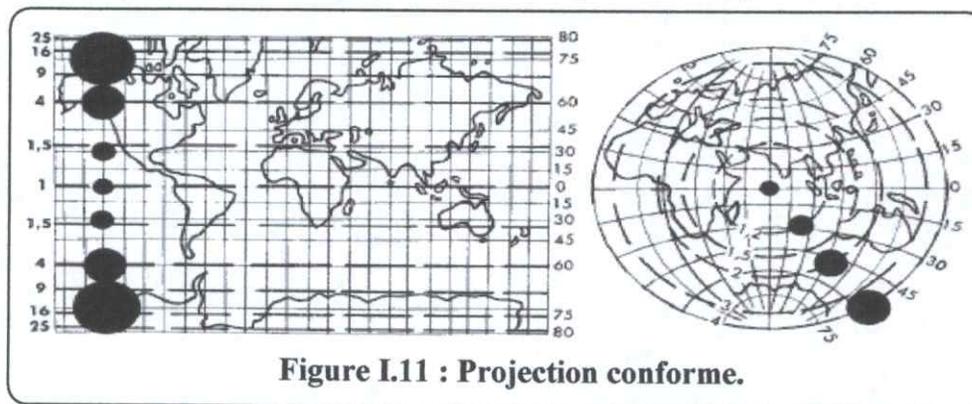
Trois types de Projections : azimutale, conique et cylindrique.

Elle peut être tangente ou sécante, directe, transverse ou oblique. [J.GIL 2000]

II.6.3. Quelques projections (les plus connues) :

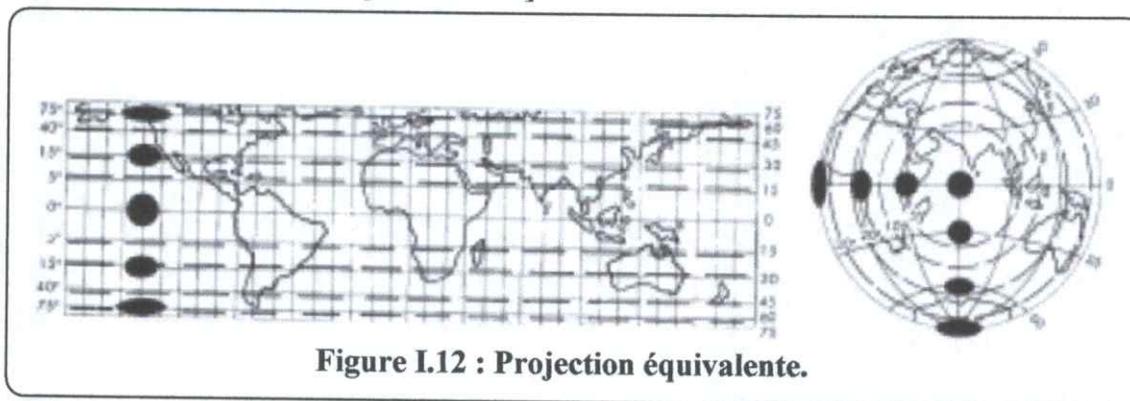
A. Projections conformes :

On choisit de préserver plutôt les angles donc les formes, en particulier les méridiens et les parallèles se coupent à angle droit. [J.GIL 2000]



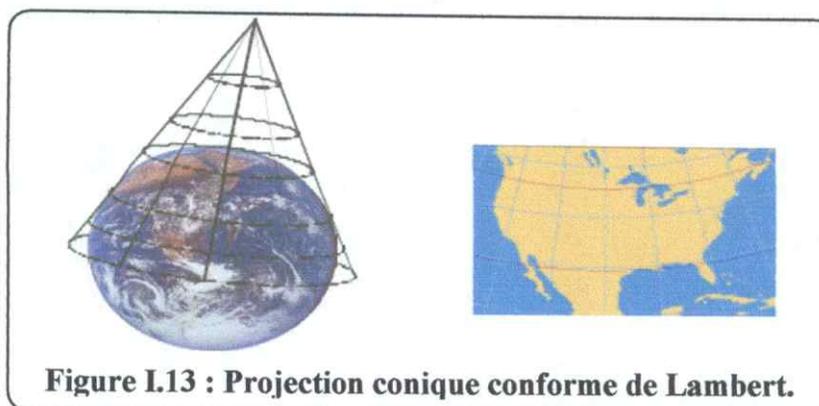
B. Projections équivalentes :

On choisit de préserver les surfaces, l'indicatrice est une ellipse, l'échelle est variable selon les directions. [J.GIL 2000]



C. Projection conique de Lambert :

Le système Lambert est largement utilisé en France. C'est une projection conforme sur un cône tangent à un parallèle. [J.GIL 2000]



D. Projection cylindrique de Mercator (utiliser en ALGERIE):

Appeler aussi UTM (Universel Transverse Mercator).

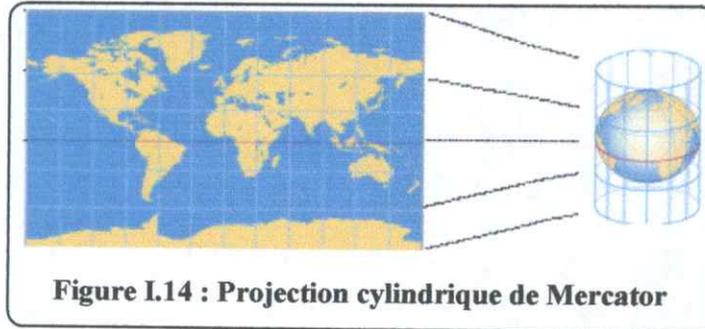


Figure I.14 : Projection cylindrique de Mercator

La projection se fait sur un cylindre tangent à l'équateur. [J.GIL 2000]

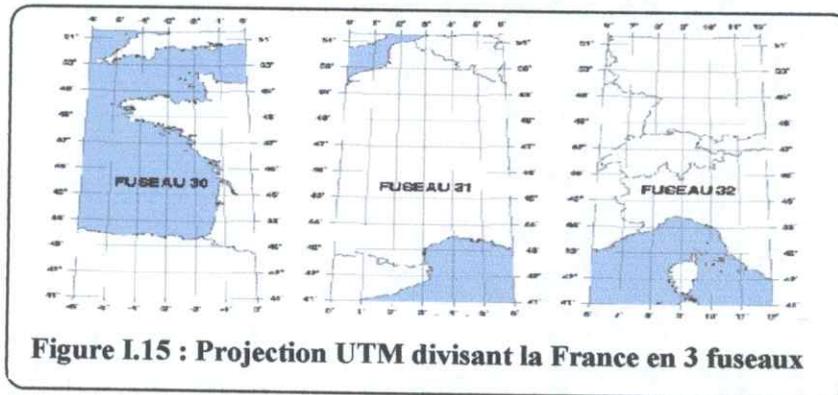


Figure I.15 : Projection UTM divisant la France en 3 fuseaux

C'est le système qui est largement utilisé en Amérique du nord et en Algérie.

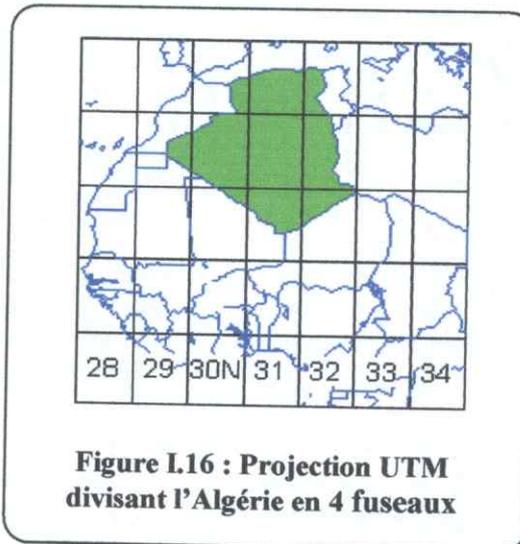


Figure I.16 : Projection UTM divisant l'Algérie en 4 fuseaux

L'Algérie se retrouve sur les fuseaux UTM Nord 29, 30, 31 et 32. [J.GIL 2000]

II.7. Notion d'échelle :

L'échelle est le « rapport entre les dimensions ou distances marquées sur un plan avec les dimensions ou distances réelles » (Petit Robert, édition 1995, 2551p).

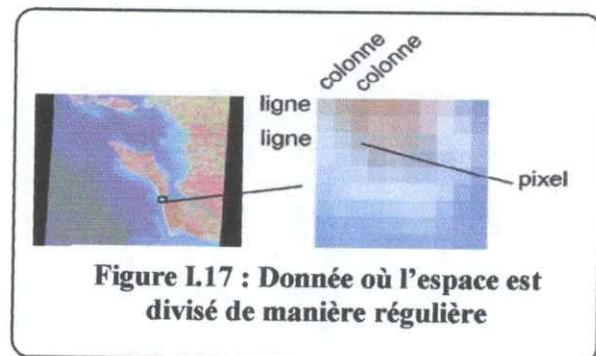
Avec les SIG, on parle plutôt d'échelle d'utilisation, c'est -à -dire le ratio entre l'échelle à laquelle la donnée a été numérisée et les limites de son exploitation.

Une carte est toujours présentée à une échelle précise, qui varie selon le détail ou la portion de territoire à représenter. Par exemple, une carte à **petite échelle** présente une grande portion de territoire avec peu de détails, tandis qu'une carte à **grande échelle** présente une petite portion de territoire avec beaucoup de détails.

II.8. Représentation des données spatiales :

II.8.1. La donnée « raster » :

Mode de représentation et de stockage des données spatiales par une matrice régulière de cellules ou pixels couvrant un espace déterminé. [J.GIL 2000]



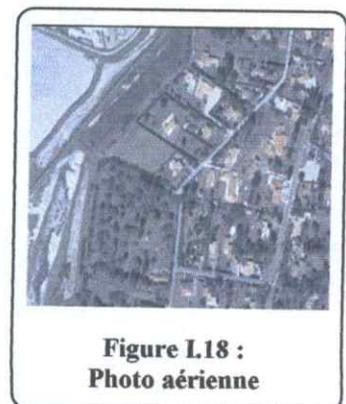
La donnée raster est sous forme de :

a- Photo aérienne :

La photo est la base de nombreuses données géographiques.

A partir d'appareil photo ou de caméra aéroportée (avion, ballon, ...) il est possible d'avoir de nombreux détails de la surface de la terre, elle peut être :

- scannée,
- numérique (directement intégrable sur un disque dur),
- orthorectifiée (corrigé des déformations d'échelle dues aux différentes altitudes, à l'assiette de l'avion, ... on obtient une orthophotographie). [J.GIL 2000]



b- Plan scanné ou carte scannée :

C'est la représentation d'une information déjà interprétée. Ceci montre ses limites.

Par contre, la carte scannée est un bon référentiel visuel car elle est souvent issue de carte papier destinée au grand public. [J.GIL 2000]



Figure I.19 :
Carte scannée

c- Image satellite :

Image issue de capteurs embarqués dans des satellites d'observation placés sur des orbites de 500 à 36000 km d'altitude. L'image représente le rayonnement solaire réfléchi par les objets au sol dans le domaine visible ou proche infrarouge. Elle doit subir plusieurs traitements radiométriques et géométriques avant d'être utilisables dans un SIG. [J.GIL 2000]

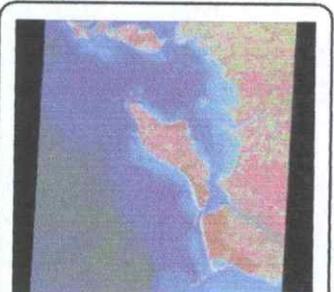


Figure I.20 :
Image satellite

d- Image satellite radar :

Image enregistrée par des capteurs embarqués dans des satellites d'observation Elle représente la réponse du sol à l'onde envoyée par le capteur (principe du flash ou du sonar). [J.GIL 2000]



Figure I.21 :
Image satellite radar

II.8.2. La donnée « vecteur » :

Ce mode de représentation de l'information géographique permet d'associer à chaque objet géographique une information sémantique, sous forme d'attributs ou de relations.

[J.GIL 2000]

Dans le format vectoriel, les objets sont représentés par les *trois primitives géométriques* suivantes:

Le point :

L'objet le plus simple, il peut représenter à grande échelle des arbres, des bornes d'incendie, des collecteurs d'ordures, Mais à des échelles plus petites de type carte routière au 1/1 000 000ème, il représente une capitale régionale. [J.GIL 2000]

La ligne :

La ligne représente les réseaux de communication, d'énergie, hydrographiques, d'assainissement, etc.

Elle peut être fictive, en représentant l'axe d'une route, ou virtuelle en modélisant des flux d'information, d'argent, ... [J.GIL 2000]

La surface :

Elle peut matérialiser une entité abstraite comme la surface d'une commune ou des entités ayant une existence géographique comme une forêt, un lac, une zone bâtie, ...

[J.GIL 2000]

II.9. Notion de couche :

Un SIG contient généralement plusieurs sortes d'objets géographiques qui sont organisés en thèmes que l'on affiche souvent sous forme de couches. Chaque couche contient des objets de même type (routes, bâtiments, cours d'eau, limites de communes, entreprises,...).

Chaque objet est constitué d'une forme (géométrie de l'objet) et d'une description, appelée aussi sémantique.

II.10. Conclusion:

Nous avons vu qu'une information géographique doit être localisée par des coordonnées par rapport à une certaine projection, cette information peut être représentée sous forme raster ou vecteur et ceci à différentes échelles, aussi elle peut faire partie d'une couche afin de représenter un thème particulier.

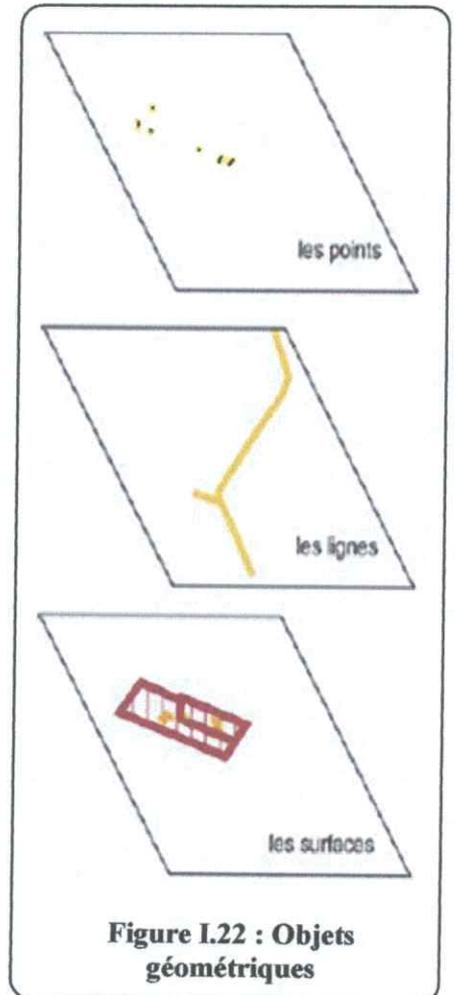


Figure I.22 : Objets géométriques

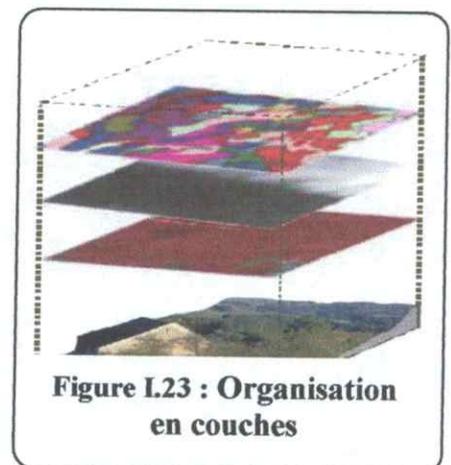


Figure I.23 : Organisation en couches

III. Géo-référencement

III.1. Introduction:

Géo référencer, c'est mettre une image ou des objets graphiques, dessinés dans un plan, en conformité géométrique avec la réalité, dans un système de projection donné.

L'opération a donc pour objet de placer chaque pixel de l'image d'origine à sa position géographique exacte.

Comme il n'est pas possible d'indiquer manuellement la position exacte de chaque pixel de l'image, on utilise un principe d'échantillonnage de points.

Ce principe se nomme **Calage** de carte.

Ces points se nomment **points d'Amers**. [IRD]

III.2. Points d'amers:

Il s'agit de repères dont on connaît les coordonnées géographiques et qui permettent de caler une couche donnée. [T, CON 2006], [IRD]

III.3. Calage:

Le calage des cartes consiste à passer du référentiel image au référentiel géographique choisi au départ. [IRD]

Généralement la première étape d'une procédure de digitalisation ou lors de l'ouverture d'une image raster pour la première fois dans un des logiciels cartographiques tels que (MapInfo, ArcGis). Avant de pouvoir digitaliser une carte papier ou travailler sur une image raster, il faut pointer sur différents points de calage de la carte et entrer leurs coordonnées (par ex. longitude et latitude).

Après avoir calé la carte, le logiciel peut associer une position en longitude et latitude à chaque point se trouvant à la surface de la carte. [T, CON 2006]

Pour caler une carte il faut définir au moins trois points d'amers, sur la carte scannée et dont les coordonnées sont facilement identifiables dans le référentiel géographique choisi. Les points doivent être facilement repérables, pris les plus éloignés possibles et non alignés. [IRD]

Pour le calage, plusieurs modèles de transformation existent :

a. Modèles nécessitant deux amers :

- **Une translation** : dans ce cas, l'image est déjà conforme à la projection géographique, la taille du pixel est connue, la translation n'est utilisée que pour localiser l'image. Elle n'effectue aucune transformation sur les pixels. Un seul point de référence est nécessaire pour effectuer une translation.

- **Une translation et une rotation** : c'est la transformation la plus simple à utiliser dans le cas où l'image serait géographiquement correcte mais doit subir une translation et une rotation pour être en conformité avec repère de la projection. Les distances dans l'image ne sont pas modifiées.

- **Une similitude** : c'est une translation et une rotation suivie d'une homothétie (mise à l'échelle). La similitude est à utiliser lorsque, pour être mise en conformité, avec une projection géographique, l'image doit subir une translation, une rotation, puis une mise à l'échelle. La mise à l'échelle est identique quelle que soit la direction (isométrie).

b. Modèles nécessitant trois amers :

- **Un ajustement polynomial** :

- Du 1^{er} Degré : **linéaire**

Cette déformation est identique à la similitude, sauf que la mise à l'échelle n'est pas identique quelle que soit la direction, en plus claire cet ajustement permet des :

- ✓ Rotations.
- ✓ Translations.
- ✓ Changements d'échelle.

Exemple :

$$x' = a_0 + a_1x + a_2y$$

$$y' = b_0 + b_1x + b_2y$$

Où :

- x' et y' sont les coordonnées d'un point dans le référentiel du monde réel (en UTM);
- x et y sont les coordonnées d'un point dans le référentiel de l'image (en pixel);
- a et b des coefficients.

Lorsqu'il y a :

- Des **distorsions géométriques irrégulières** dans l'image originelle (voussure, étirements locaux, ...);
- Un **changement de projection cartographique**.

Alors, **il faut une fonction polynomiale du 2° ou 3° degré.**

- Du 2° degré : **quadratique**

$$x' = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2$$

$$y' = b_0 + b_1x + b_2y + b_3xy + b_4x^2 + b_5y^2$$

- Du 3° degré : **cubique**.

Cependant, **l'accroissement du degré du polynôme entraîne l'accroissement du nombre de points amers.**

En théorie, il faut autant de points amers que de coefficients.

En pratique, en raison de **l'ajustement par les moindres carrés** on double les points amers :

- transformation **linéaire** => **6** points amers ;
- transformation **quadratique** => **12** points amers ;
- transformation **cubique** => **20** points amers ;

[IRD], [V, GOD 2006]

- **Un ajustement projective (ou homographie) :**

C'est une projection classique d'un plan sur un plan à partir d'un point. C'est la déformation naturelle obtenue sur une photographie, lorsque le terrain photographié correspond à un plan.

Cette transformation doit être combinée à une déformation prenant en compte l'altitude, sinon l'échelle obtenue n'est pas correcte. Quatre points sont nécessaires pour fixer les coefficients de cette transformation, ce qui revient à calibrer la position de la caméra par rapport au plan de projection.

- **Un ajustement par triangulation :**

Cette déformation combine une première transformation globale (rotation, similitude, polynomiale de degré 1, homographie), avec une déformation locale de degré 1 dans chaque triangle résultant d'une triangulation à partir des points d'amers saisis.

Cette transformation est la plus efficace lorsque l'on ne dispose pas d'un modèle numérique de terrain permettant de connaître l'altitude en chaque point. En effet, elle établit un modèle de déformation polynomiale de degré 1 dans chaque triangle.

La déformation correspond à un découpage de l'espace en facettes planes, et si le réseau de points d'amer est dense et homogène (et d'autant plus dense que les différences d'altitudes sont grandes), le redressement permet d'obtenir directement une image en conformité avec la projection géographique choisie.

Ce type de déformation permet également d'assurer une jointure parfaite entre différentes images : une fois une image redressée, il suffit de saisir des points d'amers entre l'image redressée et l'image à caler pour faire coïncider les deux images.

Plus il y a de points d'amers, plus la précision est importante. [IRD]

III.4. Conclusion :

Le calage est important, car il permet de renseigner les pixels en leur donnant les dimensions réelles du terrain tel que chaque objet dans l'image puisse être superposable à chaque objet réel leur correspondant, leur différence donnant le moins d'erreurs possibles. [V, GOD 2006]

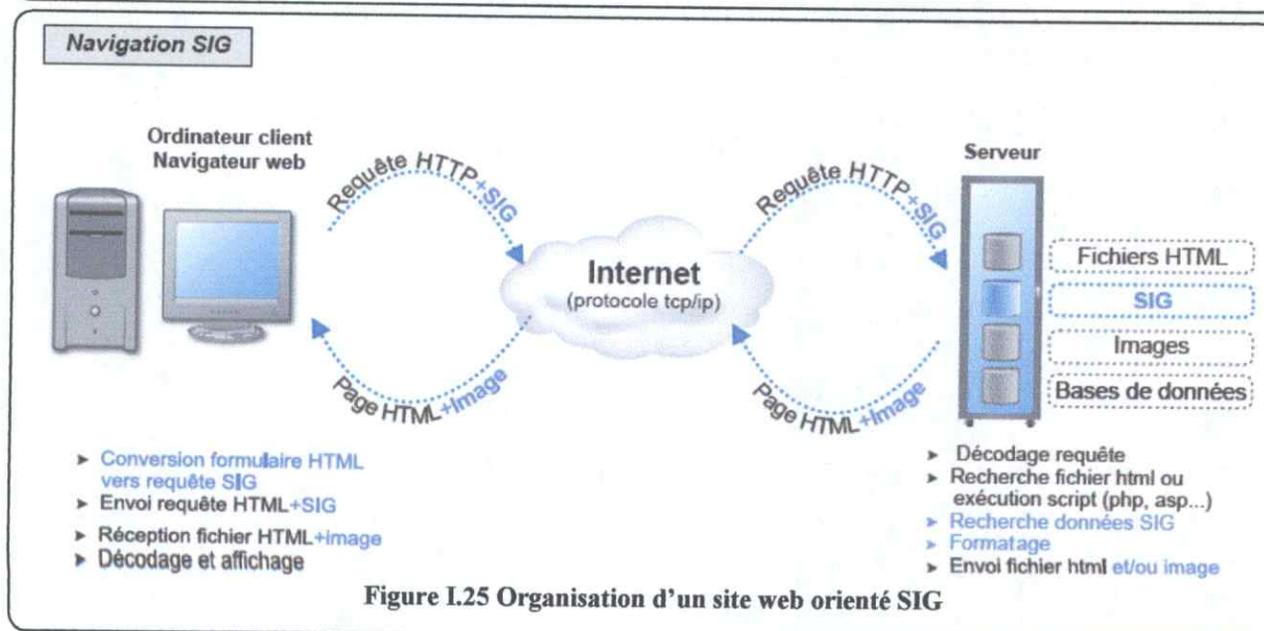
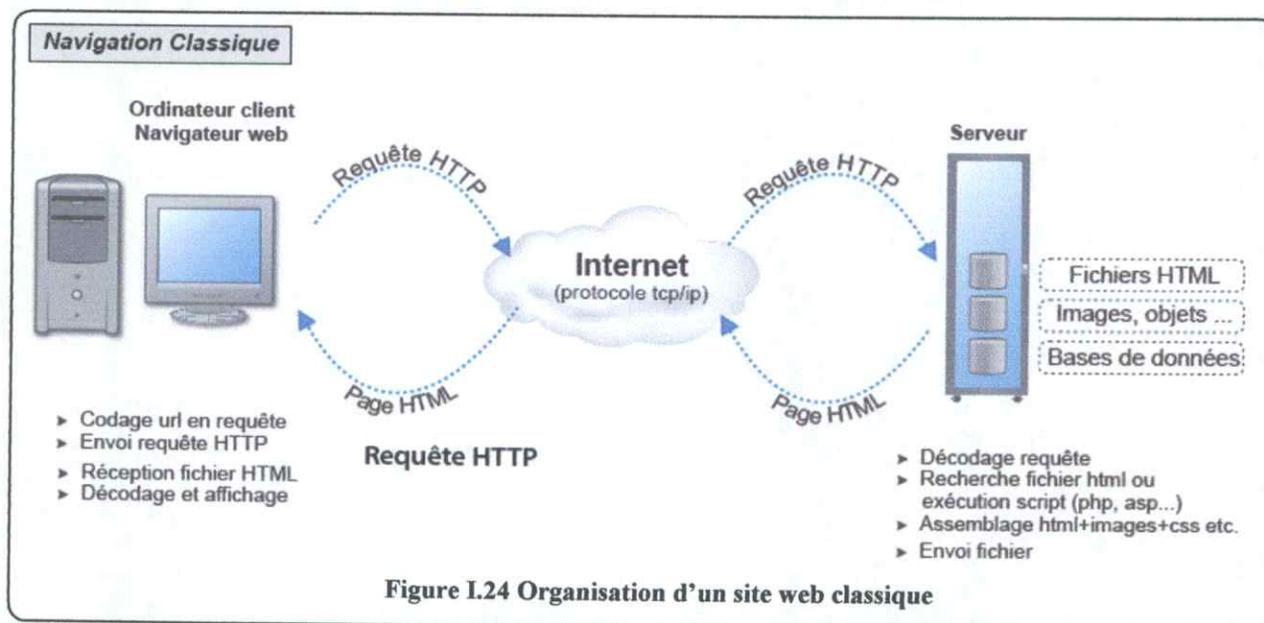
IV. SIG interactifs en ligne (Web Mapping)

IV.1. Qu'est-ce qu'un serveur cartographique (serveur SIG)?

Un serveur SIG est un serveur Internet classique (HTTP) doté de fonctionnalités SIG, c'est-à-dire qu'il est capable de :

- Gérer des données géo référencées.
- Exécuter des requêtes numériques, textuelles et spatiales.
- Générer des documents cartographiques.
- Retourner ces documents à l'ordinateur client qui les a demandés.

Ce serveur SIG prend généralement la forme d'une couche logicielle installée sur la machine serveur, qui va intercepter les requêtes de type SIG et compléter les réponses du logiciel serveur par des éléments cartographiques.

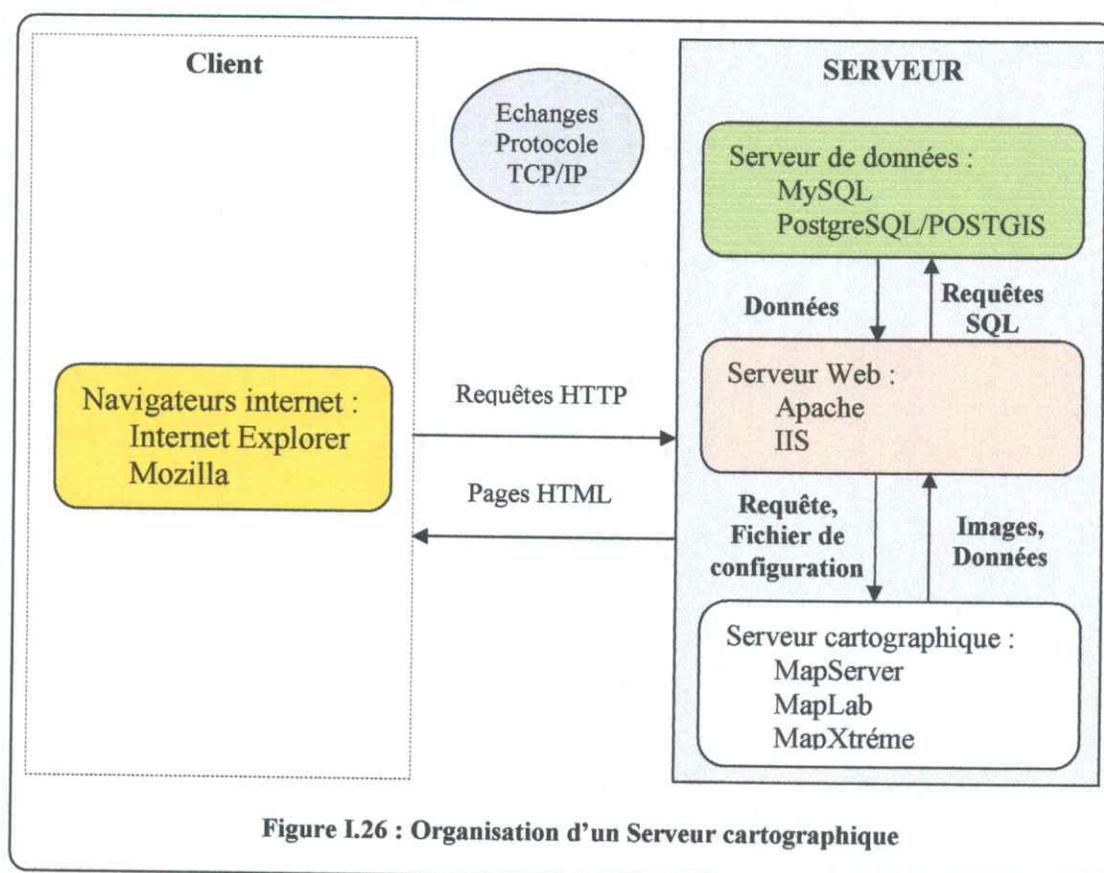


Par « serveur » on entend la machine qui sert des données Internet en réponse à des requêtes, mais aussi les logiciels qui effectuent concrètement cette opération sur la machine. [L, JEG 2006]

IV.2. Principe générale de fonctionnement d'un serveur cartographique (serveur SIG):

Le serveur cartographique est le guichet automatique auquel l'utilisateur fait appel pour afficher des cartes sur son poste informatique. Par le protocole de communication Internet, TCP/IP, des ordinateurs branchés en réseau peuvent échanger de l'information via un navigateur Web. L'architecture est de type client/serveur, c'est-à-dire qu'un ordinateur dit serveur répond aux requêtes d'une série d'ordinateurs dits clients.

L'utilisateur, à partir de son terminal effectue des requêtes pour demander l'affichage d'une zone géographique spécifique; le serveur cartographique interprète cette requête et renvoie la zone géographique sous la forme d'une image matricielle (png, jpg,...) ou vectorielle (svg, swf,...). [L, JEG 2006]



Le moteur cartographique peut être contrôlé par des langages de script tels que PHP, javascript, Python ou Perl qui lui permettent de générer dynamiquement une carte en réponse à une **requête** préparée par une interface utilisateur, ces requêtes sont le plus souvent liées à un **fichier de configuration** exécuté par le serveur cartographique.

Ce dernier peut chercher l'information nécessaire à la réalisation de la carte dans ses propres ressources, mais aussi sur des serveurs de données distants.

La diffusion de l'information en ligne requiert une installation côté serveur avec des logiciels tels qu'Apache (projet *Open Source*) ou IIS (Internet Information Services, de Microsoft) qui tournent en tâche de fond et donnent accès aux serveurs de cartes à l'Intranet et à l'Internet.

Ces logiciels serveurs voient souvent leurs fonctions étendues par des interpréteurs de scripts comme PHP ou ASP. Le serveur cartographique s'appuie sur ces éléments pour recevoir des requêtes et renvoyer des images et des données. Côté client, un navigateur web suffit, accompagné éventuellement par un *viewer* (visionneuse en français), pour afficher la carte.

Les données peuvent être gérées par des logiciels spécifiques, les SGBDR, tels PostgreSQL, MySQL, Oracle, entre autres, qui peuvent être installés directement sur le serveur contenant le serveur cartographique ou sur un autre serveur, distant. Qu'importe le lieu, l'important est de pouvoir consulter et éditer des données à distance. [Mapp, 2007]

IV.3. Les fonctions d'un serveur cartographique (serveur SIG):

Le serveur SIG va donc ajouter aux fonctions habituelles d'un serveur Internet des fonctions en relation avec la gestion et le traitement de données graphiques géo référencées. [L, JEG 2006]

➤ Stockage et gestion des données SIG :

Cette fonction centrale du serveur se base le plus souvent sur un stockage des données SIG sous la forme de fichiers, qui sont lus et traités au moment des requêtes.

Plus intéressant, le stockage peut se faire par une solution SGBD classique étendue de possibilités de stockages d'éléments graphiques géoréférencés. Par exemple :

- ✓ PostgreSQL possède une extension nommée PostGIS
- ✓ Oracle possède une capacité OracleSpatial
- ✓ ArcGIS peut se baser sur une BDD Access, sur une GéoDatabase...

✓ MySQL est étendu par MyGIS

De plus, beaucoup de solutions logicielles de SGBD sont aujourd'hui compatibles entre elles par le biais de passerelles de conversion comme ODBC. L'ajout de données SIG à une SGBD classique s'effectue le plus souvent par conversion des informations graphiques en format texte ou binaire, lié à un dictionnaire de données spécifiques (topologique). Cette conversion s'effectue soit automatiquement en interne au logiciel (solutions commerciales), soit par traduction des fichiers (PostGIS, MyGIS, FME...).

➤ **Traitement des requêtes spatiales :**

Le serveur SIG doit être capable de rendre les services de base d'un véritable moteur SIG, au delà du stockage des données, c'est-à-dire la possibilité d'effectuer des requêtes à composante spatiale :

- ✓ Inclusion / juxtaposition / croisement
- ✓ Calculs de longueurs et superficies
- ✓ Mesure de distances, zones tampons
- ✓ Mise à jour des données graphiques et attributaires [L, JEG 2006]

➤ **Formatage et diffusion :**

Enfin le serveur SIG doit être capable de retourner l'information sous une forme adaptée à l'interface Internet, c'est-à-dire au site affiché dans le navigateur de la machine cliente. Cela suppose en particulier la capacité de transformer les données graphiques et alphanumériques issues d'une requête dans le SGDB dans un format compatible avec les navigateurs :

- ✓ gif, jpeg ou png pour les bitmaps.
- ✓ svg, swf (flash), pour les vecteurs (plugin nécessaire)... [L, JEG 2006]

IV.4. Conclusion:

Les serveurs cartographiques offrent des services orientés SIG, car ils permettent à des informaticiens sans connaissances approfondis du domaine de réaliser des applications SIG ou d'entendre des systèmes classiques avec des fonctionnalités géographiques.

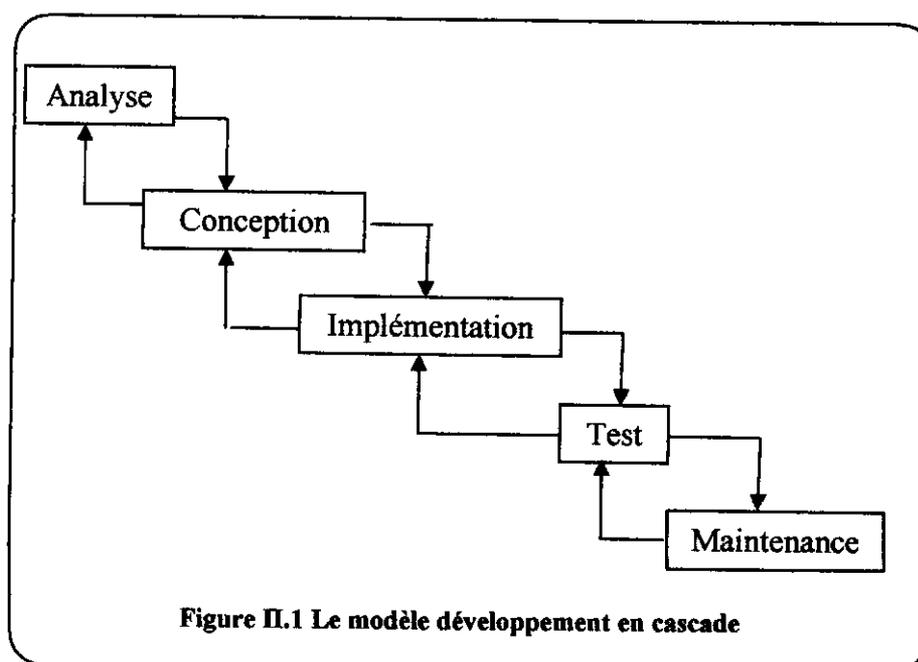
Chapitre II. Démarche et développement du système

I. Démarche de développement :

Les systèmes informatiques devenant de plus en plus complexes, le recours à la modélisation est devenu une pratique indispensable au développement des systèmes informatiques, en effet cette modélisation d'unifier les notations nécessaires aux différentes activités d'un processus de développement. [P, ROQ 2003]

Dans notre projet nous utiliserons la notation UML qui se définit comme un langage de modélisation graphique et textuelle destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, concevoir des solutions et communiquer les points de vue.

Nous avons souhaité suivre le modèle de développement « en cascade » qui présente un cycle de vie d'un logiciel par une suite de phases (analyse, conception, implémentation, test, et maintenance), qui s'enchaînent dans un déroulement linéaire depuis l'analyse des besoins jusqu'à la maintenance, les résultats de chaque étape sont testés et on ne passe à l'étape suivante que s'ils sont validés. [MULL, 1997]



En effet vu la complexité de notre projet et notre inexpérience dans le domaine des SIG, le modèle en cascade est le candidat idéal, car il permet d'alterner les différentes étapes du développement, puisqu'on ne peut terminer la partie conception du système sans débiter l'implémentation de ce dernier.

Il est de nature traditionnelle d'associer à UML un processus de développement, nous avons adopté le processus UP.

UP (Unified Process) : est une méthode générique de développement de logiciel, générique signifie qu'il est nécessaire d'adapter UP au contexte du projet, de l'équipe, du domaine et/ou de l'organisation.

Le processus unifié est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel.

Caractéristiques essentielles du processus unifié :

- **Le processus unifié est à base de composants,**
- **Le processus unifié utilise le langage UML**
- **Le processus unifié est piloté par les cas d'utilisation,**
- **Centré sur l'architecture,**
- **Itératif et incrémental.**

Dans notre thèse nous nous limiterons aux diagrammes suivants :

1. Cas d'utilisation.
2. Scénario et diagramme de séquence.
3. Diagramme de classe.
4. Diagramme d'état/transition.
5. Diagramme de collaboration.
6. Diagramme déploiement.

II. Analyse:

II.1. Expression initiale des besoins :

L'organisme d'accueil (la société CATRAL) a décidé de proposer un Système de Navigation par Satellite adapté à la géographie du territoire Algérien. Ce système doit se présenter sous forme d'un site web dynamique.

Ce dernier doit offrir la possibilité pour des abonnés de suivre un ou plusieurs véhicules à la fois en temps réel, tout comme il doit permettre de consulter l'historique du parcours des véhicules choisis sur une période donnée.

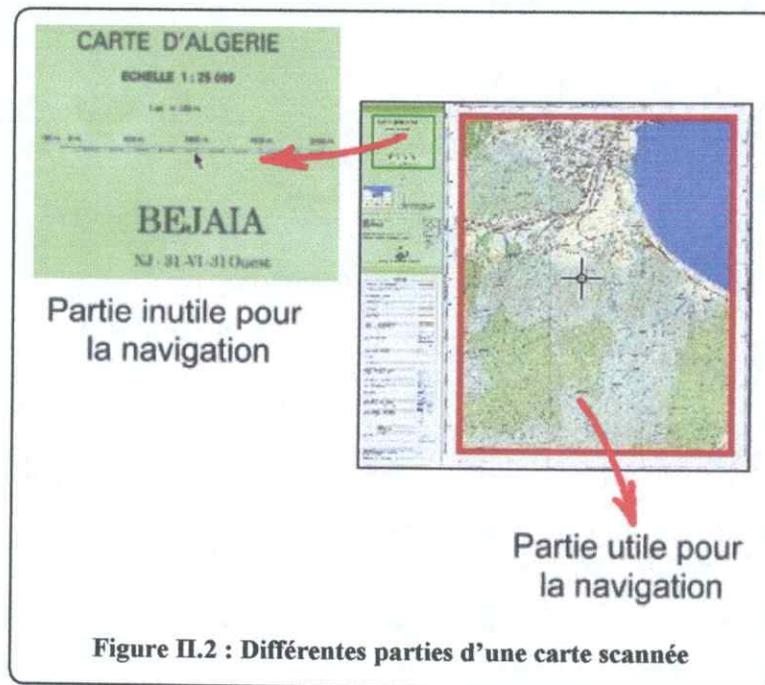
II.2. Exigences fonctionnelles du site :

- Le site doit permettre de gérer les abonnements des clients.
- Le site doit répondre à deux qualifications majeures :
 - chargement rapide.
 - Localisation précise des véhicules (une erreur d'un pixel sur la carte est une erreur de dizaines voir de centaines de mètres dans la réalité).
- Suivi en temps réel de un ou plusieurs véhicules à la fois.
- Accès à l'historique des positions antérieures de un ou plusieurs véhicules à la fois.
- Le site doit choisir soigneusement la vue à projeter afin de garantir un encadrement total des différentes positions des véhicules sélectionnées.

II.3. Etude de l'existant :

- Les coordonnées émises par le kit GPS obéissent au système de référence terrestre WGS84.
- Les données géographiques se traduisent en plusieurs types : photos aériennes, images satellites, cartes scannées..., ces dernières constitueront la matière première de notre site.
- Ces cartes scannées ont des points communs qu'il convient de noter :
 - Ceux sont des fichiers de grandes tailles (plusieurs mégaoctets chacun en format JPEG compressé).
 - Elles subissent une inclinaison due à la projection UTM.
 - Elles se pressentent avec les systèmes de projections UTM ou WGS84.
 - Elles existent sur plusieurs échelles différentes.

- Elles se composent de deux parties :



- **Une partie inutile pour la navigation**, sur les quatre bords de la carte, qui contient des informations liées à cette dernière (échelle, type de repère, année de réalisation, région de la carte...).
- **Une partie utile pour la navigation**, puisque elle contient une représentation à une échelle donnée d'une certaine région du monde réel.

Les deux premiers points constituent de lourdes contraintes par rapport à nos deux exigences fonctionnelles (rapidité et précision) et ceci pour les raisons suivantes :

- Un site web qui charge, coté utilisateur, des images de plusieurs mégaoctets ne peut pas offrir des temps de réponse satisfaisants même s'il n'y a qu'un seul utilisateur connecté.
- D'autre part, si nous négligeons l'inclinaison subite par ces cartes, cela engendrera une mauvaise intercalation entre les cartes adjacentes, et également des erreurs de positionnement des véhicules.

II.4. Redéfinition des besoins :

L'idéal pour un site web qui crée puis envoie à l'utilisateur une vision d'une zone géographique précise, est que les images servant à la synthèse de cette zone géographique soient de tailles aussi réduites que possible, afin de réduire au maximum les temps de lecture sur disque des fichiers images scannées.

Le pire des cas est celui d'un véhicule qui se trouverait à l'intersection de quatre cartes, si l'on suppose que chaque carte scannée fait 6 mégaoctets cela implique au total qu'il y'a 24 mégaoctets à lire du disque pour construire une vision au tour du point d'intersection.

La solution au problème consiste à fragmenter chaque image scannée en petits morceaux (n'excédant pas quelques kilooctets) et de ne lire à chaque requête de l'utilisateur que les fragments nécessaires à la construction de la zone géographique demandée.

Dans l'exemple cité précédemment du point d'intersection, les fragments nécessaires vont être les quatre fragments les plus proches de ce point.

Cela nous amène donc à la nécessité de concevoir un outil de découpage des cartes scannées avant de pouvoir entamer la conception du site web proprement parlé.

III. Conception de l'outil :

III.1. Exigence fonctionnelle de l'outil:

- L'outil doit être capable de manipuler des images de grande taille sans mettre en péril le matériel utilisé, tout en étant précis et si possible rapide.
- L'outil doit supporter en entrée deux types de projection UTM (Universel Transverse Mercator) et WGS84.
- L'outil doit prendre en compte l'inclinaison de la carte.
- L'outil doit découper la partie utilisable de la carte en fragments (petits morceaux) avec la correction appropriée.
- L'outil doit générer un fichier de calage pour chaque fragment, ce fichier portera le même nom du fragment avec une extension (.WLD).

- L'outil doit gérer une arborescence pour classer les fragments selon leurs échelles et leurs localisations sur le territoire Algérien.
- L'outil doit indexer ses fragments pour permettre une recherche rapide d'un fragment.

III.2. Identification des acteurs :

Nous considérons que la société aura désigné un responsable qui aura la tâche de caler, découper et enfin indexer les fragments, nous l'appellerons « Responsable découpage ».

III.3. Cas d'utilisation de l'outil de découpage :

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier, il modélise un service rendu par le système, et exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné. [P, ROQ 2003]

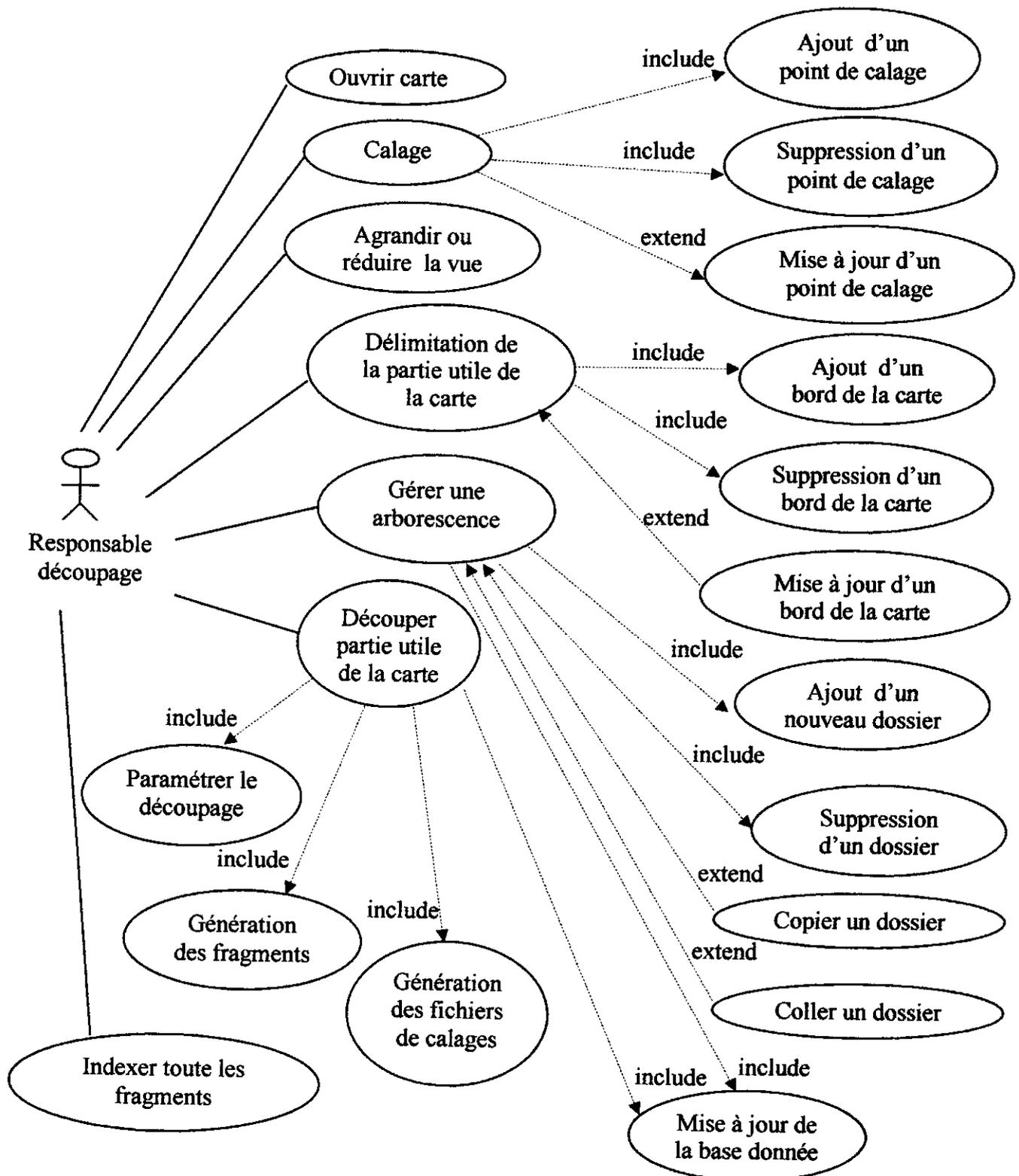


Figure II.3 : Ensemble des cas d'utilisation de l'outil de découpage.

N.B : La nécessité de gérer une arborescence sera démontré dans la partie implémentation du site de navigation.

III.4. Diagramme de séquence :

Ils décrivent le comportement du système vu de l'extérieur, sans préjuger de comment il sera réalisé, ils montrent non seulement les interactions acteurs/système mais également les événements du système déclenchés par les acteurs. [P, ROQ 2003]

III.4.1. Configuration de la connexion vers la base de données :

Scénario :

a. Scénario normal :

- L'utilisateur sélectionne dans le menu principal le bouton « configuration » puis « connexion vers base de données » ;
- Lecture de la base des registres pour vérifier si l'utilisateur a déjà configuré une connexion.
- Si l'utilisateur a déjà configuré, lu et affiché la dernière configuration.
- La fiche « connexion vers base de données » s'affiche.
- L'utilisateur saisit les informations concernant la base de données.
- L'utilisateur clique sur « Tester la connexion ».
- La connexion réussie.
- Un message affirmant la réussite de cette dernière s'affiche.
- L'utilisateur clique sur « OK ».
- Mise à jour de la base des registres.
- La fiche « connexion vers base de données » se ferme.

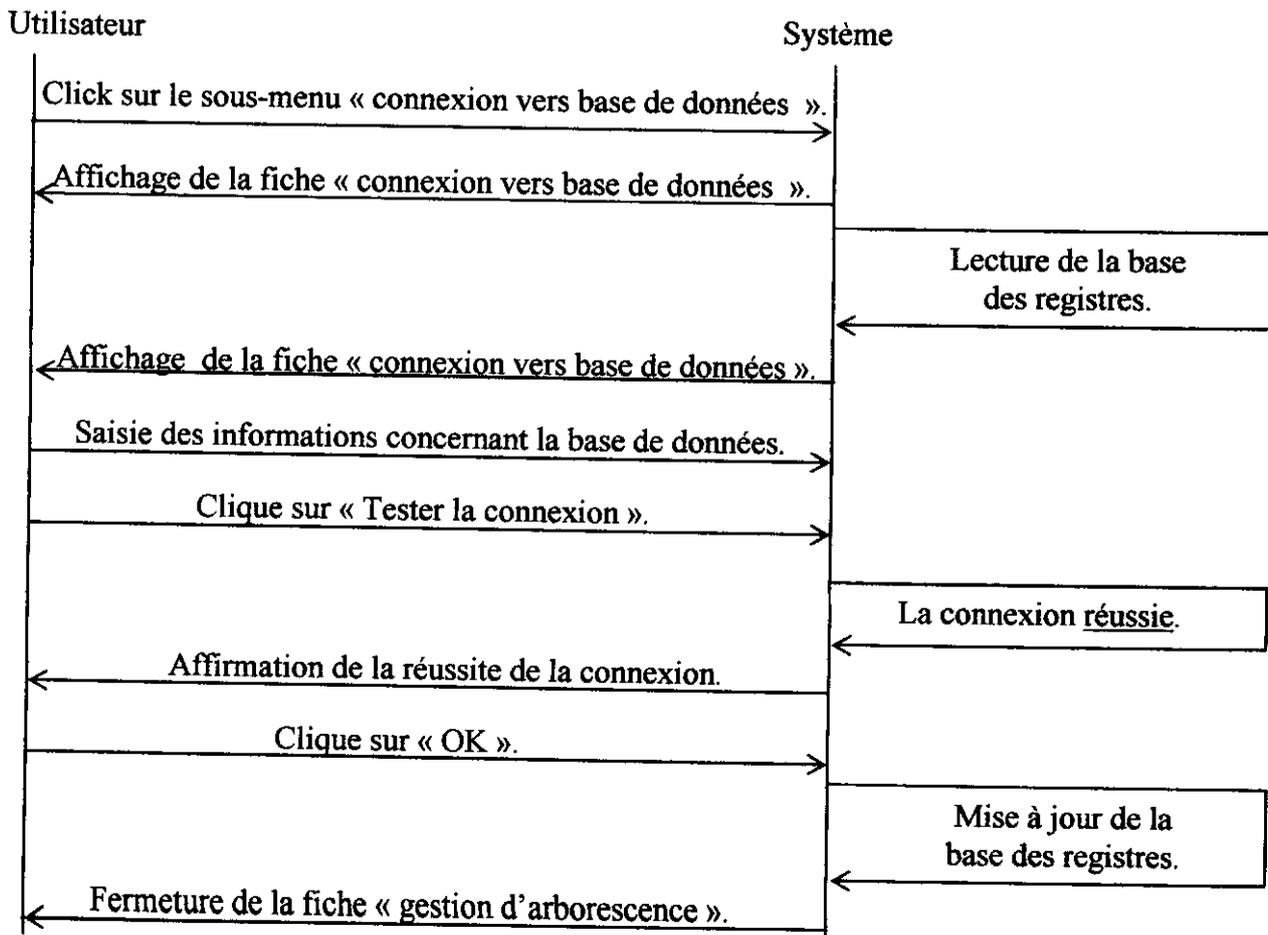


Figure II.4 : Diagramme de séquence de la configuration de la connexion vers la base de données (scénario normal).

b. Scénario d'exception :

- L'utilisateur sélectionne dans le menu principal le bouton « configuration » puis « connexion vers base de données » ;
- Lecture de la base des registres pour vérifier l'existence d'une connexion déjà configurée.
- La fiche « connexion vers base de données » s'affiche avec si possible la dernière configuration de la connexion.
- L'utilisateur saisit les informations concernant la base de données.
- L'utilisateur clique sur « Tester la connexion ».
- La connexion échoue.
- Un message affirmant l'échec de cette dernière.
- L'utilisateur clique sur « abandon ».
- La fiche « connexion vers base de données » se ferme.

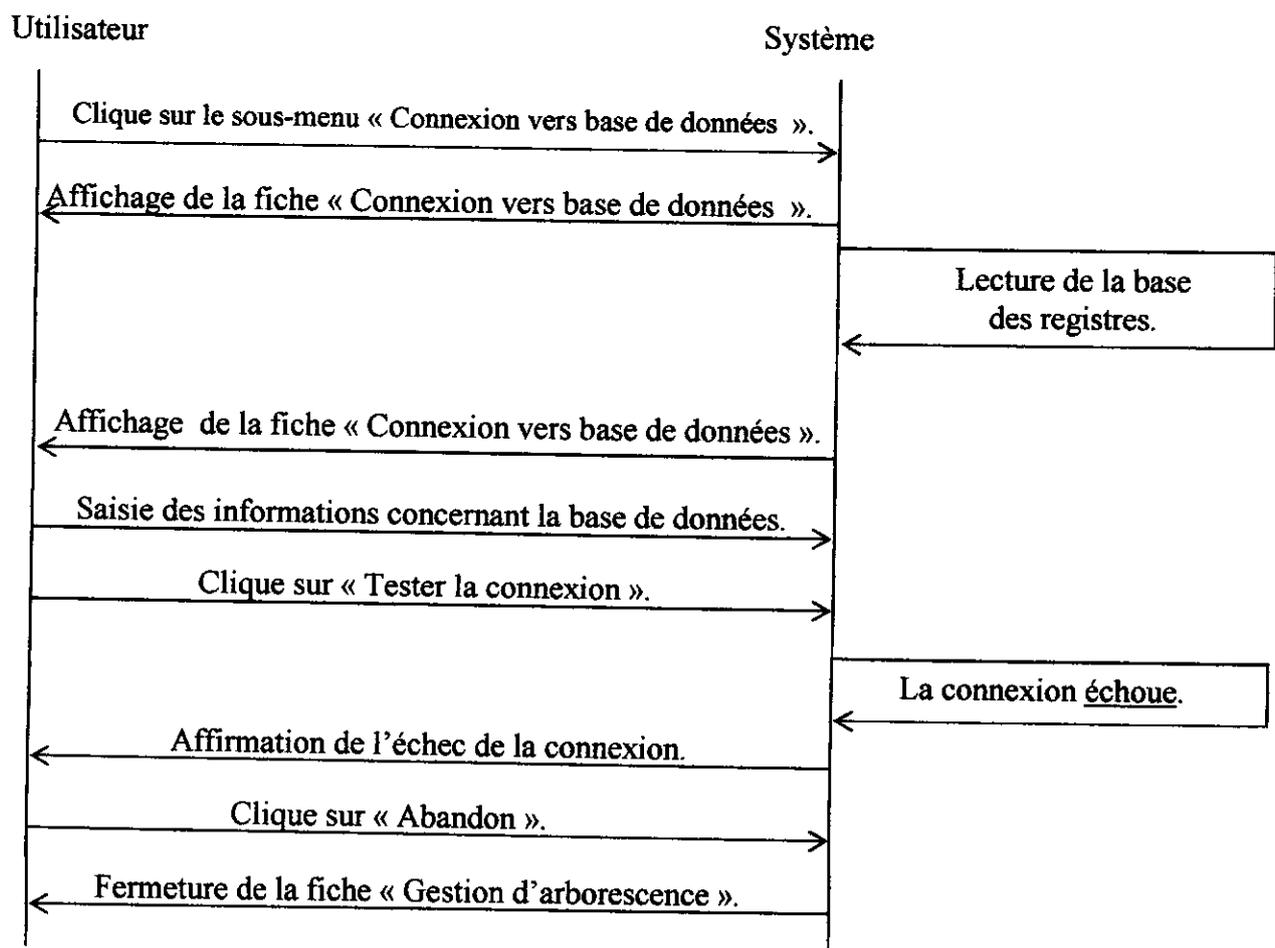


Figure II.5 : Diagramme de séquence de la configuration de la connexion vers la base de données (scénario d'exception).

III.4.2. Caler une carte :

- L'utilisateur lance l'outil de découpage.
- L'outil s'affiche avec la partie découpage désactivé.
- L'utilisateur ouvre une carte.
- L'utilisateur choisit l'échelle.
- L'utilisateur sélectionne l'outil « point de calage ».
- Clique sur la carte.
- Sauvegarde de la vue en cours.
- La zone autour du clique est zoomée.
- L'utilisateur augmente la précision du clique grâce aux touches déplacements.
- L'utilisateur confirme le point en appuyant sur la touche « Entrer ».
- La fiche « point de calage » s'affiche.
- L'utilisateur saisit les coordonnées du point.
- L'utilisateur confirme les coordonnées en cliquant sur « ok ».
- Fermeture de la fiche « point de calage ».
- Restauration de la vue sauvegardée.
- L'utilisateur ajoute de la même façon deux points de calages.
- L'utilisateur sélectionne l'outil « point de bordure ».
- Clique sur les 4 bords de la carte.
- Calcul des coefficients de calage.
- Activation de la partie découpage.

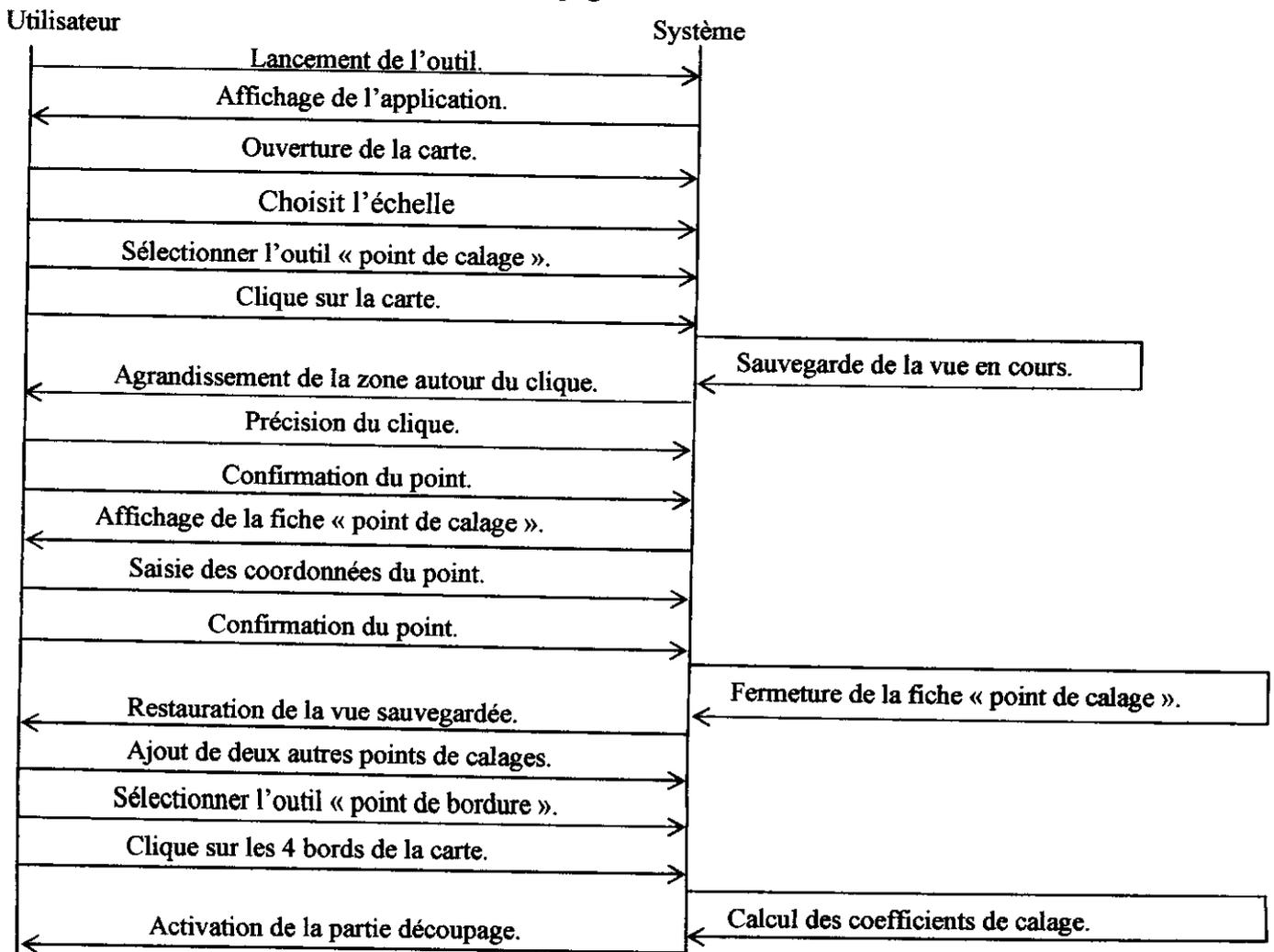


Figure II.6 : Diagramme de séquence caler une carte.

III.4.3. Découper une carte déjà calée:

- L'utilisateur sélectionne un dossier sur le disque dur.
- L'utilisateur saisit les paramètres de découpage:
 - o Taille de fragments (les fragments seront carrés).
 - o Taux de compression des fragments (sortie en format JPEG).
 - o Nom de la carte.
- L'utilisateur clique sur « découper ».
- La progression du découpage est affichée.
- Le programme génère les fragments.
- Le programme génère un fichier de calage d'extension « .WLD » pour chaque fragment.
- Mise à jour automatique de la base de données.
- Un message « fin du découpage » est affiché.

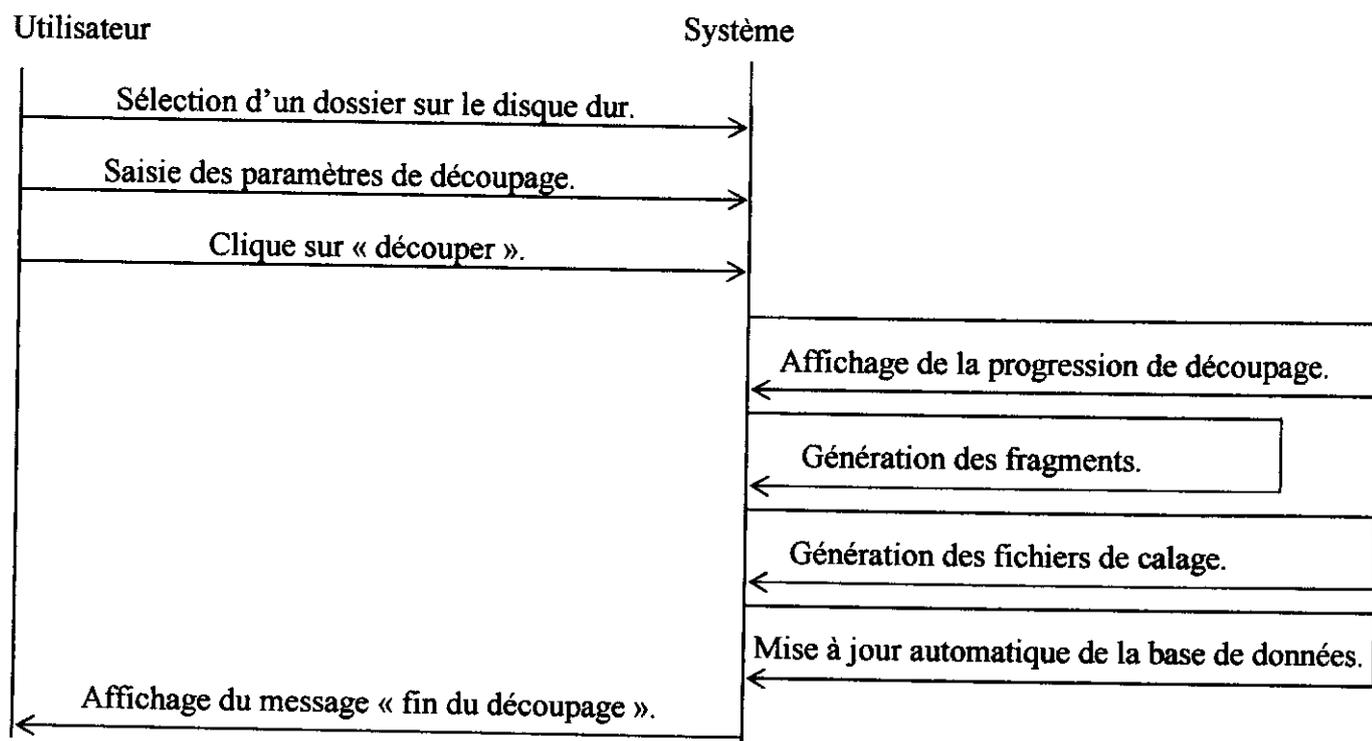


Figure II.7 : Diagramme de séquence « découper une carte déjà calée ».

III.5. Digramme de classe : Notre application sera basée sur les classes suivantes :

- Classe carte.
- Classe calage_carte.
- Classe point de calage.
- Classe bordure.
- Classe vue.
- Classe fragment.
- Classe calage_fragment.

Dictionnaire de données :

Code	Désignation	TYPE
nom_carte	Nom de la carte.	Chaîne de caractères
taille_carte	Largeur et hauteur de la carte.	Entiers.
echelle	Echelle de la carte.	Réel.
path_fragments	Chemin sur disque dur des fragments découpés	Chaîne de caractères
x_coefficient	Vecteur de coefficient du polynôme sur le plan des x.	Vecteur.
y_coefficient	Vecteur de coefficient du polynôme sur le plan des y.	Vecteur.
point_image	Coordonnées du point sur l'image en pixel.	Point (réel, réel).
point_carte	Coordonnées du point dans le repère UTM	Point (réel, réel).
UTM_zone	Zone UTM de la carte.	Chaîne de caractères
top_Left	Coordonnées du coin supérieur gauche.	Point (réel, réel).
top_Right	Coordonnées du coin supérieur droit.	Point (réel, réel).
bottom_left	Coordonnées du coin inférieur gauche	Point (réel, réel).
Bottom_right	Coordonnées du coin inférieur droit.	Point (réel, réel).
ok_top_Left	Indique l'existence du point du coin supérieur gauche.	Booléen.
ok_top_right	Indique si le point du coin supérieur droit existe.	Booléen.
ok_bottom_left	Indique si le point du coin inférieur gauche existe ou pas.	Booléen.
ok_bottom_right	Indique si le point du coin inférieur droit existe ou pas.	Booléen.
rec_canvas	Rectangle de la zone d'affichage (fenêtre d'affichage).	Rectangle.
rec_image	Rectangle de la zone projetée dans la zone d'affichage.	Rectangle.
zoom_echel_x	Le rapport entre la largeur de la zone projetée sur la largeur de la zone d'affichage.	Réel.
zoom_echel_y	Le rapport entre la hauteur de la zone projetée sur la hauteur de la zone d'affichage.	Réel.

Left	Coordonnées du coin supérieur gauche du fragment sur le plan des x dans le repère UTM.	Réel.
Top	Coordonnées du coin supérieur gauche du fragment sur le plan des y dans le repère UTM.	Réel.
Width	Largeur du fragment.	Réel.
Height	Hauteur du fragment.	Réel.
pixel_to_wld_x	Equivalent d'un pixel sur la carte dans le repère UTM sur le plan des x.	Réel.
x_angle	Angle d'inclinaison sur le plan des x.	Réel.
y_angle	Angle d'inclinaison sur le plan des y.	Réel.
pixel_to_wld_y	Equivalent d'un pixel sur la carte dans le repère UTM sur le plan des y.	Réel.
left	Coordonnées du coin supérieur gauche du fragment sur le plan des x dans le repère UTM.	Réel.
top	Coordonnées du coin supérieur gauche du fragment sur le plan des y dans le repère UTM.	Réel.

Table II.1 : Dictionnaire de données de l'outil de découpage

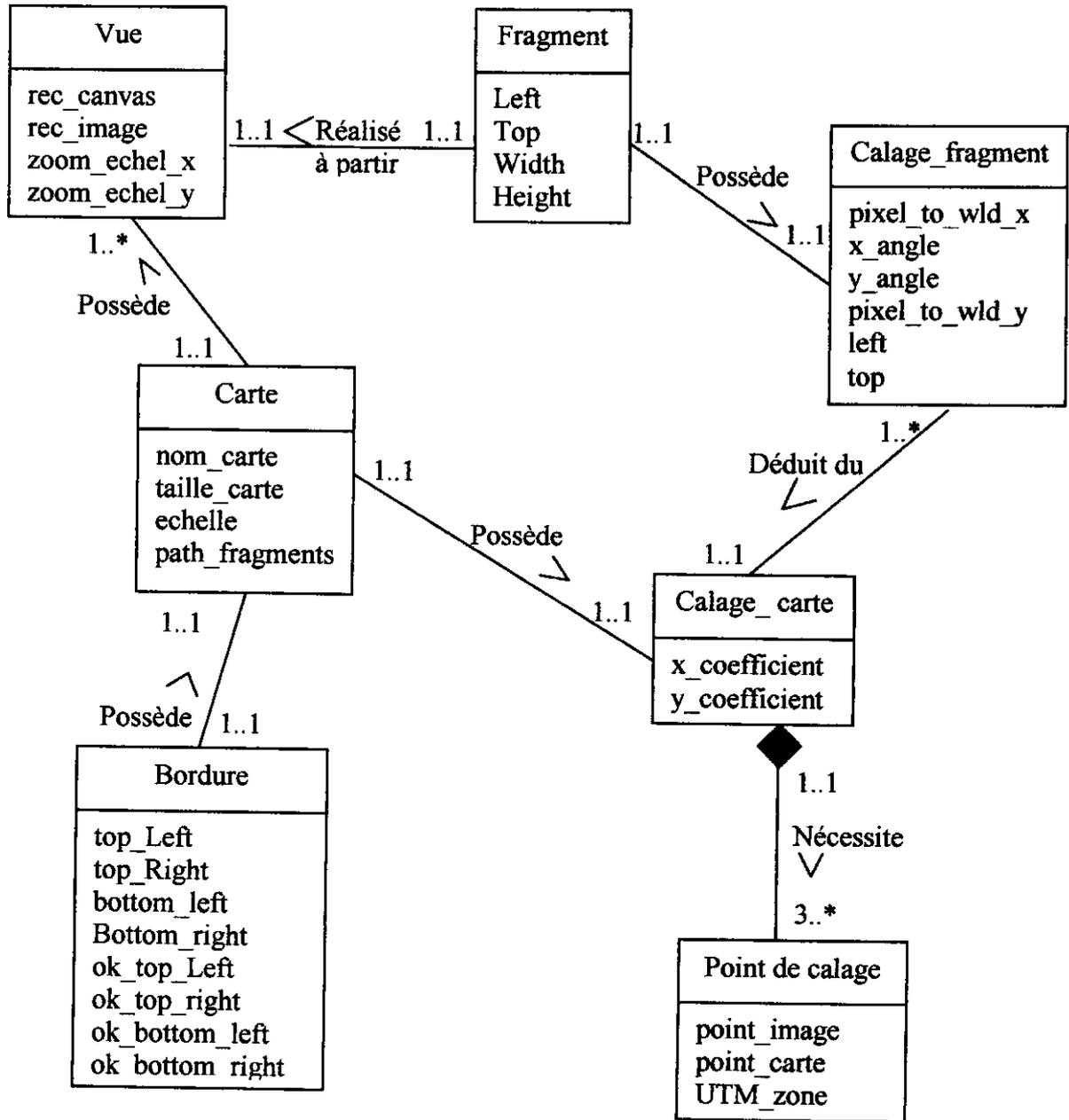


Figure II.8 : Digramme de classe de l'outil de découpage

III.6. Diagramme d'état transition :

Les diagrammes d'états-transitions permettent de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.

Un état se caractérise par sa durée et sa stabilité, il représente une conjonction instantanée des valeurs des attributs d'un objet.

Une transition représente le passage instantané d'un état vers un autre.

Une transition est déclenchée par un événement. En d'autres termes : c'est l'arrivée d'un événement qui conditionne la transition.

Les transitions peuvent aussi être automatiques, lorsqu'on ne spécifie pas l'événement qui la déclenche. [P, ROQ 2003]

Donc notre diagramme d'état transition présentera les changements d'états de l'ensemble des objets lié à une instance carte scannée.

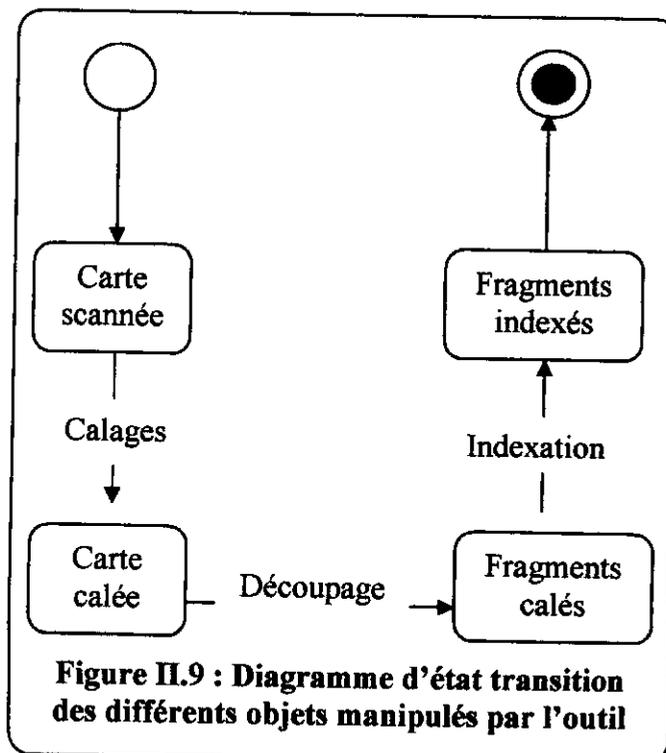


Figure II.9 : Diagramme d'état transition des différents objets manipulés par l'outil

IV. Conception du site :

IV.1. Rappel des Exigences fonctionnelles du site :

- Le site doit permettre de gérer les abonnements des clients.
- Le site doit répondre à deux qualifications majeures :
 - chargement rapide.
 - Localisation précise des véhicules (une erreur d'un pixel sur la carte est une erreur de centaines voir de milliers de mètres dans la réalité).
- Suivi en temps réel de un ou plusieurs véhicules à la fois.
- Accès à l'historique des positions antérieures de un ou plusieurs véhicules à la fois.
- Le site doit choisir soigneusement la vue à projeter afin de garantir un encadrement total des différentes positions des véhicules sélectionnées.

IV.2. Identification des acteurs :

Les acteurs du site sont :

- **Administrateur** : personne qui gère les abonnements des clients.
- **Abonné** : personne qui profite du service navigation pour suivre ces véhicules.

IV.3. Cas d'utilisation :

IV.3.1. Partie administration :

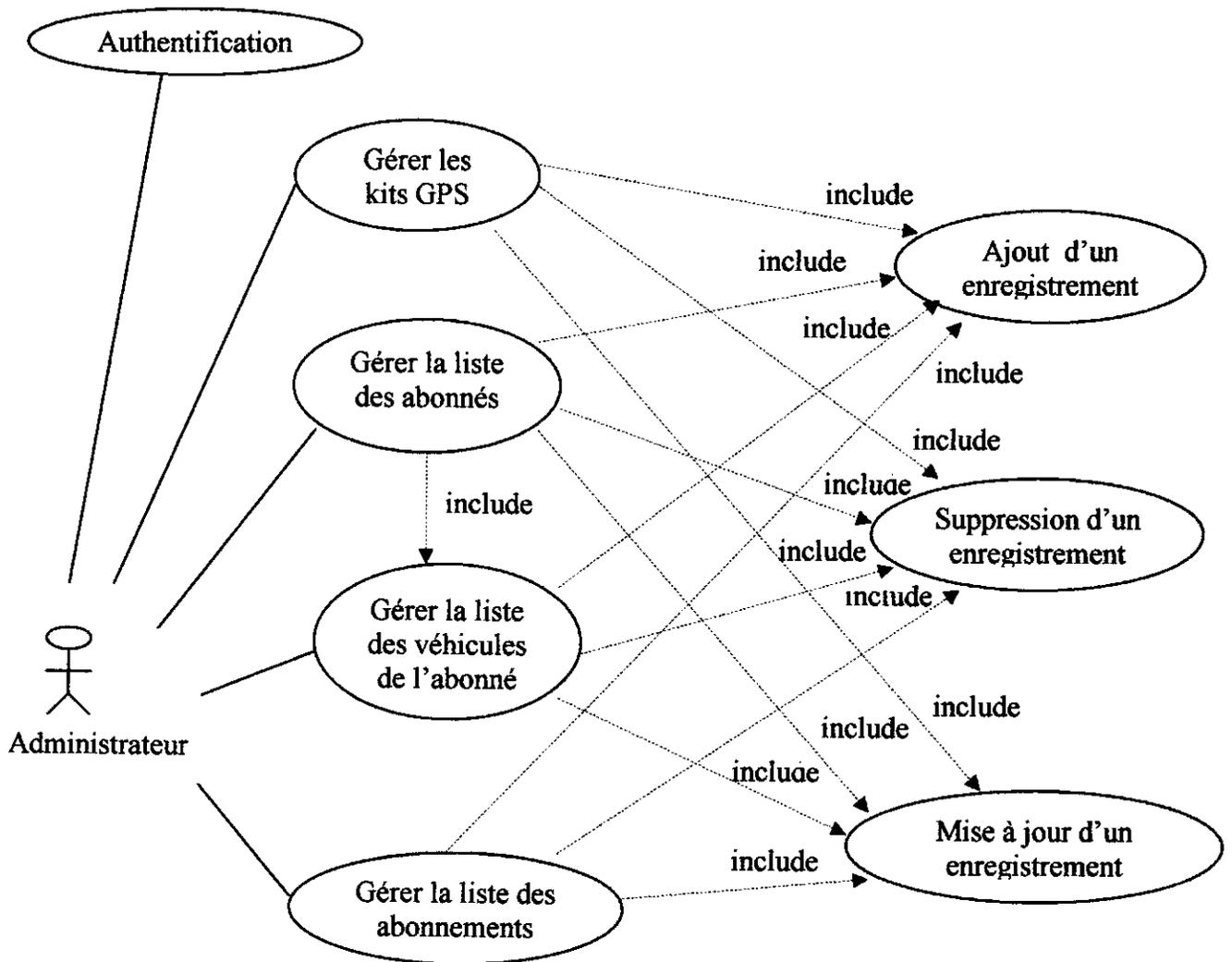


Figure II.10 : Cas d'utilisation de la partie administration du site.

IV.3.2. Partie Navigation :

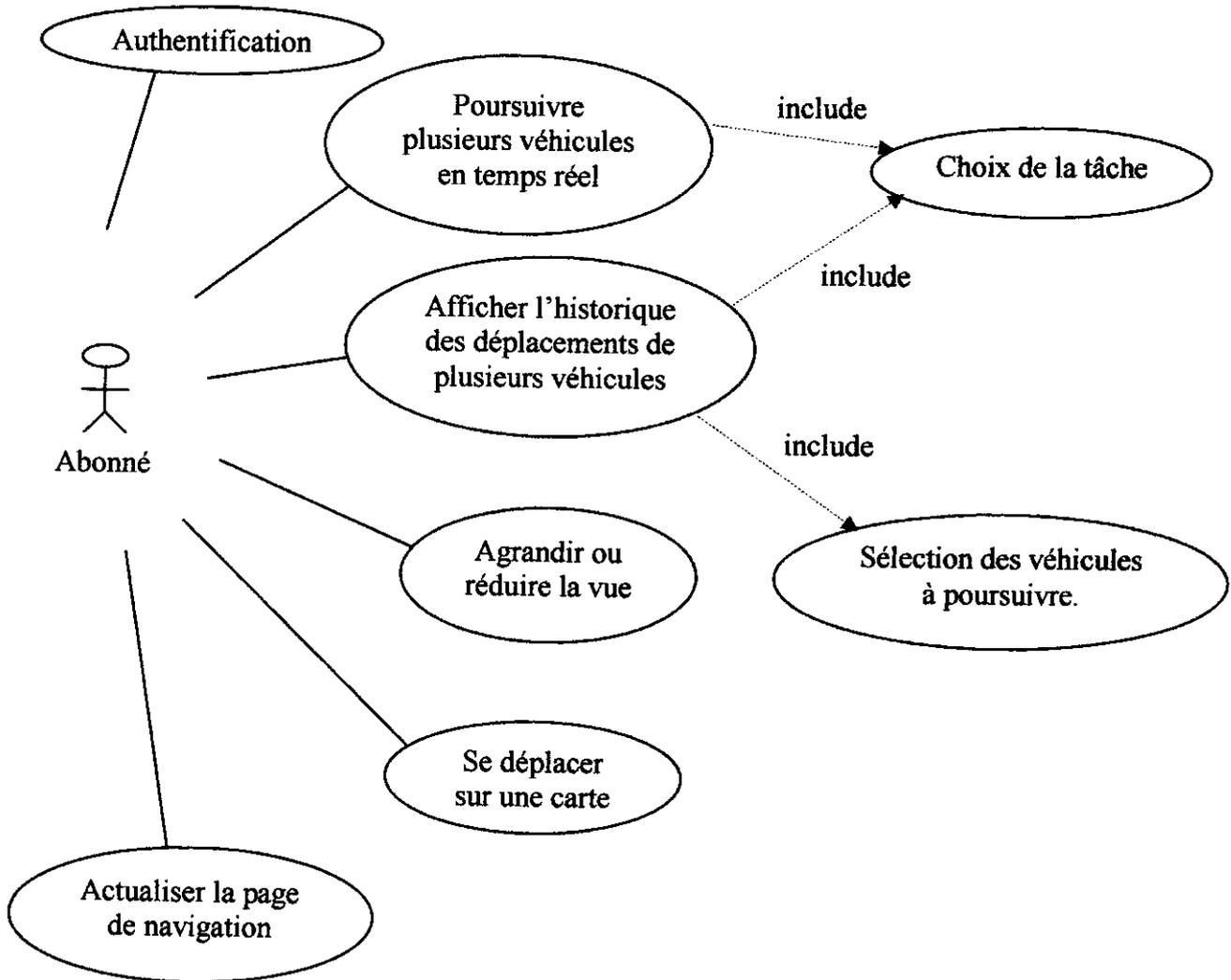


Figure II.11 : Cas d'utilisation de la partie navigation du site.

IV.4. Diagramme de séquence :

IV.4.1. Partie administration :

a. Avant installation:

Scénario :

- L'administrateur tente d'accéder à la page administration du site.
- Une boîte de dialogue « authentification » s'affiche.
- L'administrateur saisit le nom d'utilisateur et mot de passe.
- Le nom d'utilisateur et mot de passe sont justes, alors l'administration accède à la page administration contenant les quatre liens suivants :
 - Kits GPS.
 - Client.
 - Abonnement.
 - Installation
- L'administrateur clique sur le lien Kits GPS.
- Etant donné que le système n'est pas encore installé, l'administrateur est dirigé automatiquement vers la page installation.
- L'administrateur saisit les informations concernant la base de données.
- Clique sur « Terminer ».
- Tenter d'établir une connexion vers le serveur de base de données.
- Connexion réussie.
- La base de données est créée.
- Un fichier de configuration est généré.
- Retour à la page « Administration ».

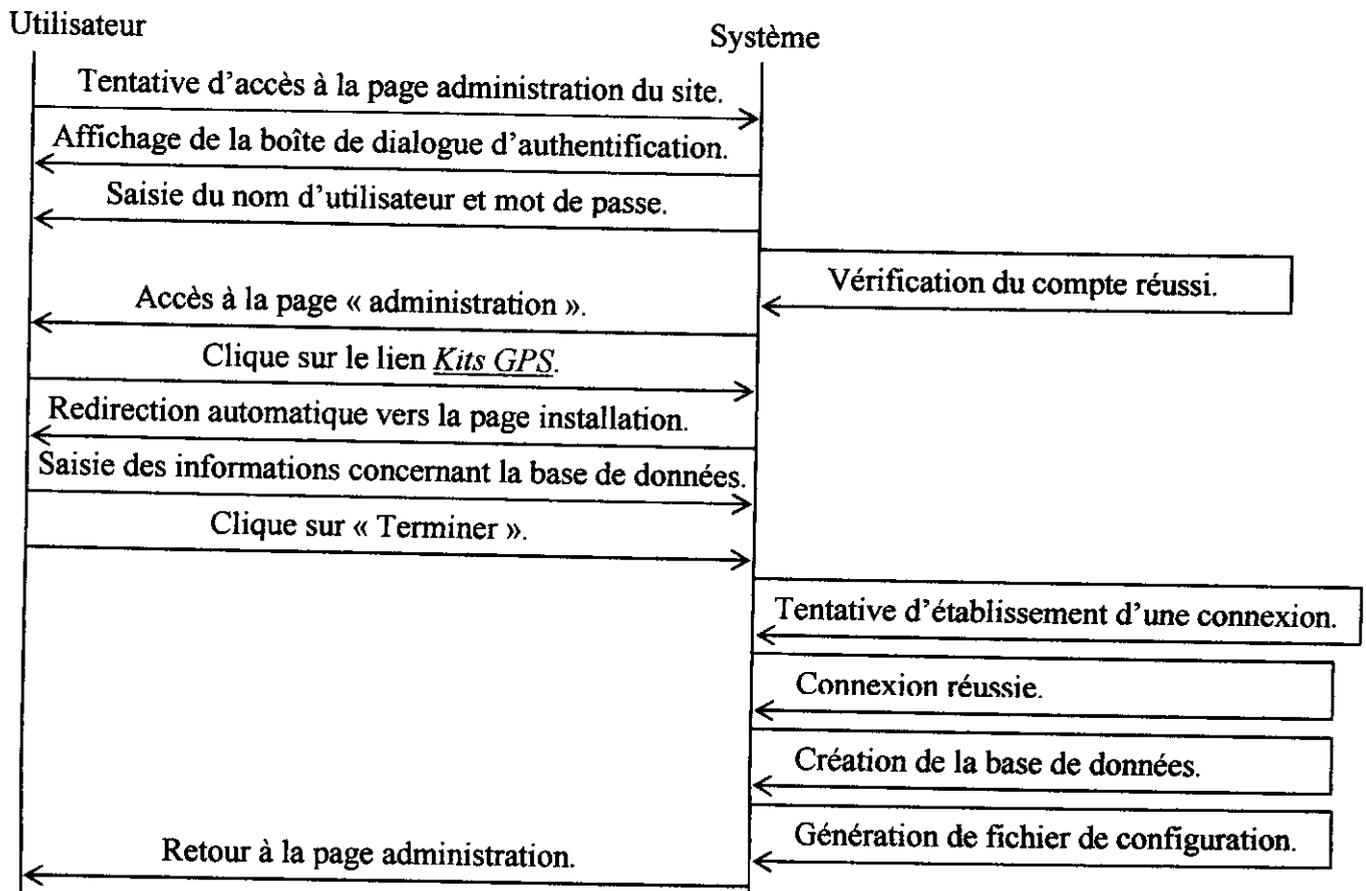


Figure II.12 : Diagramme de séquence de la partie administration (avant installation).

b. Après l'installation :

Scénario :

- L'administrateur tente d'accéder à la page administration du site.
- Une boîte de dialogue authentification s'affiche.
- L'administrateur saisit le nom d'utilisateur et le mot de passe.
- Le nom d'utilisateur et mot de passe sont justes, alors le serveur tente un accès vers la page administration.
- Le système est déjà installé, l'administrateur accède à la page administration contenant les quatre liens suivants :
 - Kits GPS.
 - Client.
 - Abonnement.
 - Installation
- L'administrateur clique sur le lien Kits GPS.
- Chargement des Kits GPS de la base de données.
- L'administrateur accède à la page Kits GPS.
- L'administrateur sélectionne le nombre x de kits à ajouter.
- Clique sur « Ajouter N Kits GPS ».
- X lignes sont ajoutées dans la liste des kits GPS.
- L'administrateur saisit les informations concernant les kits GPS.
- Clique sur « Terminer ».
- Mise à jour de la base de données.

N.B : Cette mise à jour nous la détaillerons dans la partie implémentation du site.

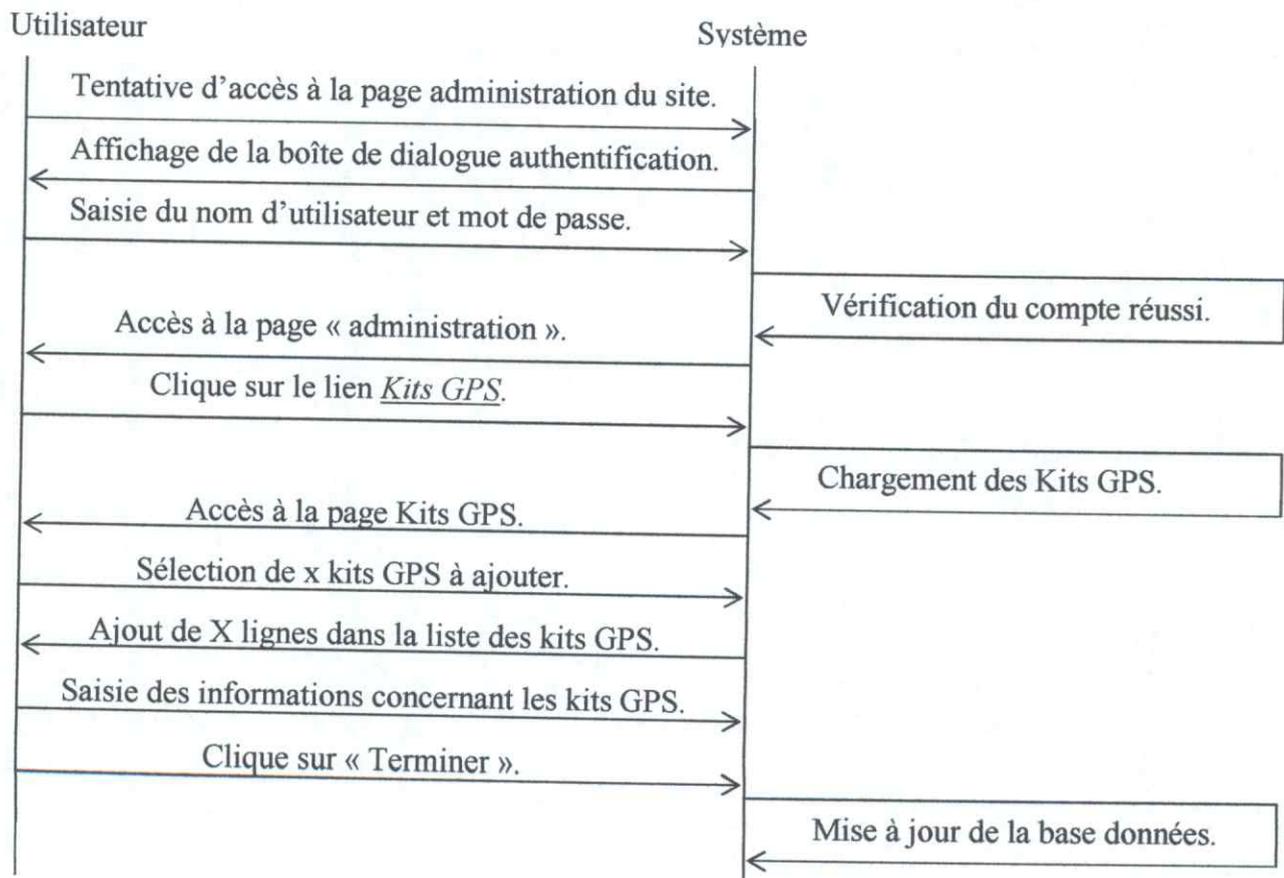


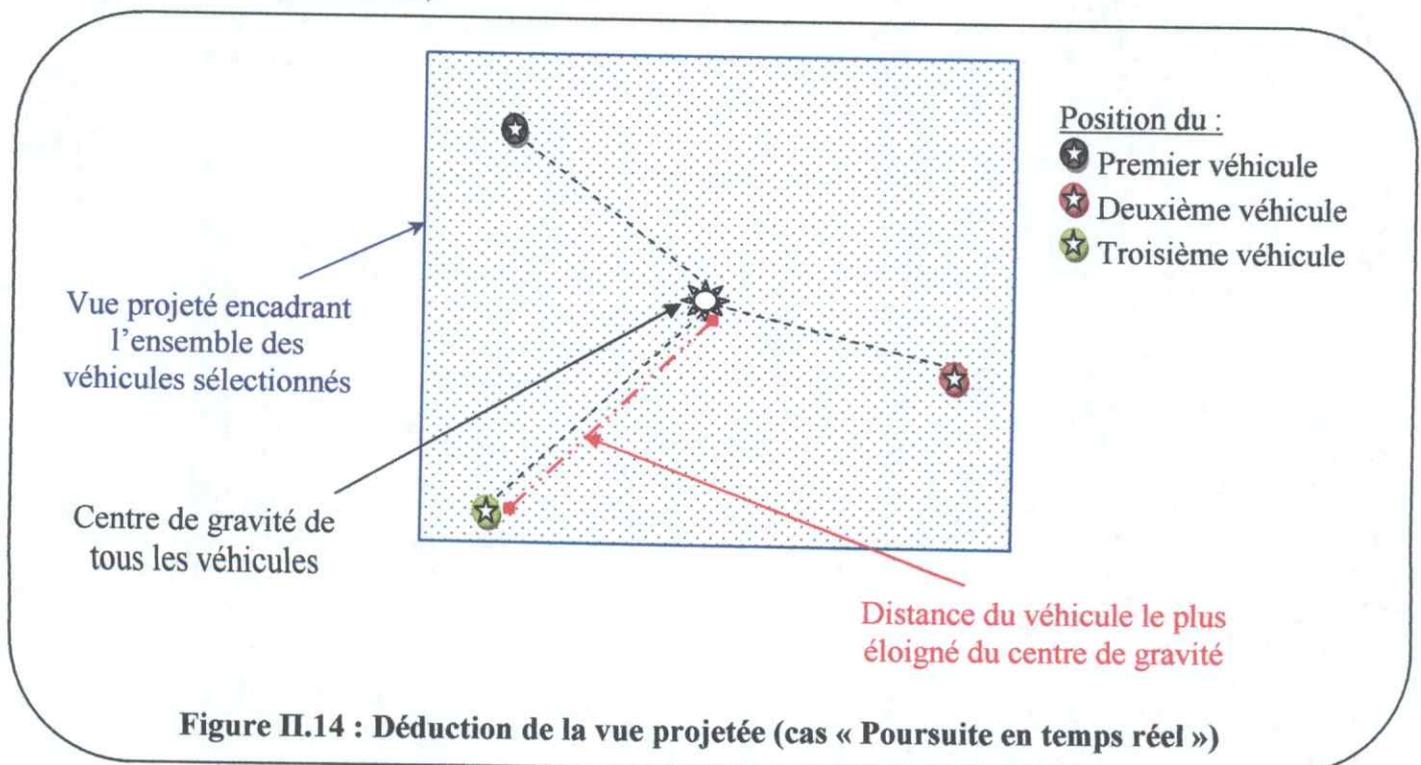
Figure II.13 : Diagramme de séquence de la partie administration (après installation).

IV.4.2. Partie navigation :

a. Poursuite en temps réel:

Scénario :

- L'abonné accède à la page d'accueil du site.
- L'abonné saisit son nom d'utilisateur et mot de passe.
- Vérification du compte réussi.
- L'abonné accède à la page navigation du site.
- Chargement des véhicules de l'abonné.
- Envoi d'une requête SIG au serveur cartographique.
- Le serveur cartographique exécute la requête et renvoie le résultat.
- Affichage du résultat (vue d'ensemble de la carte d'Algérie).
- L'abonné sélectionne les véhicules à poursuivre.
- L'abonné choisit l'option « Poursuite en temps réel ».
- L'abonné clique sur « Actualiser » (du site non pas du navigateur WEB).
- Chargement de la base de données de la dernière position de chaque véhicule sélectionné.
- Calcul du centre de gravité des véhicules sélectionnés.
- Calcul de la distance du véhicule le plus éloigné du centre de gravité, parmi les véhicules sélectionnés.
- Déduire les coordonnées de la zone à projeter à partir des deux résultats précédents.
- Envoi d'une requête SIG au serveur cartographique.
- Le serveur cartographique exécute la requête et renvoie le résultat.
- Affichage du résultat (Vue la plus proche incluant tous les véhicules positionnés).



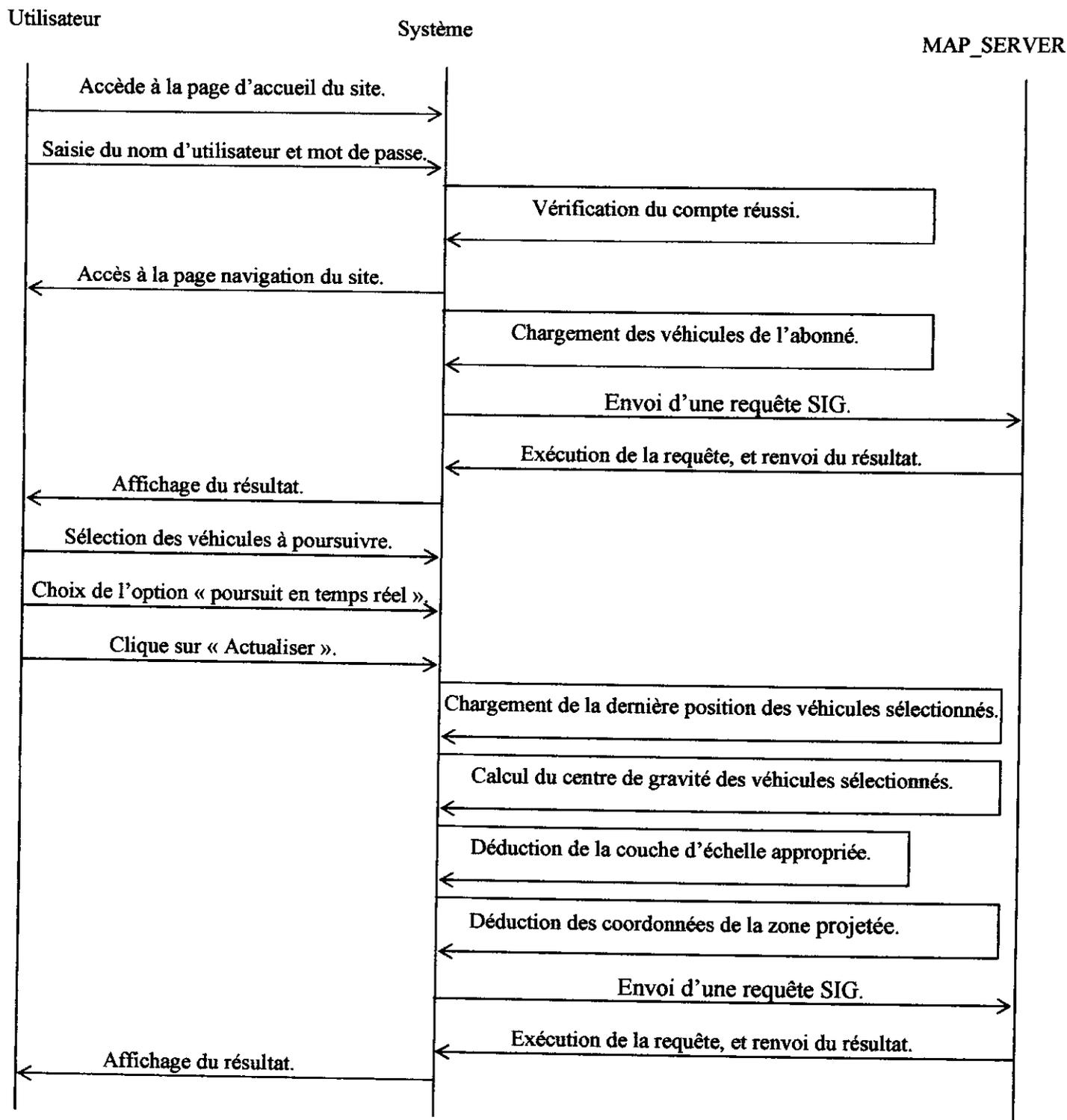
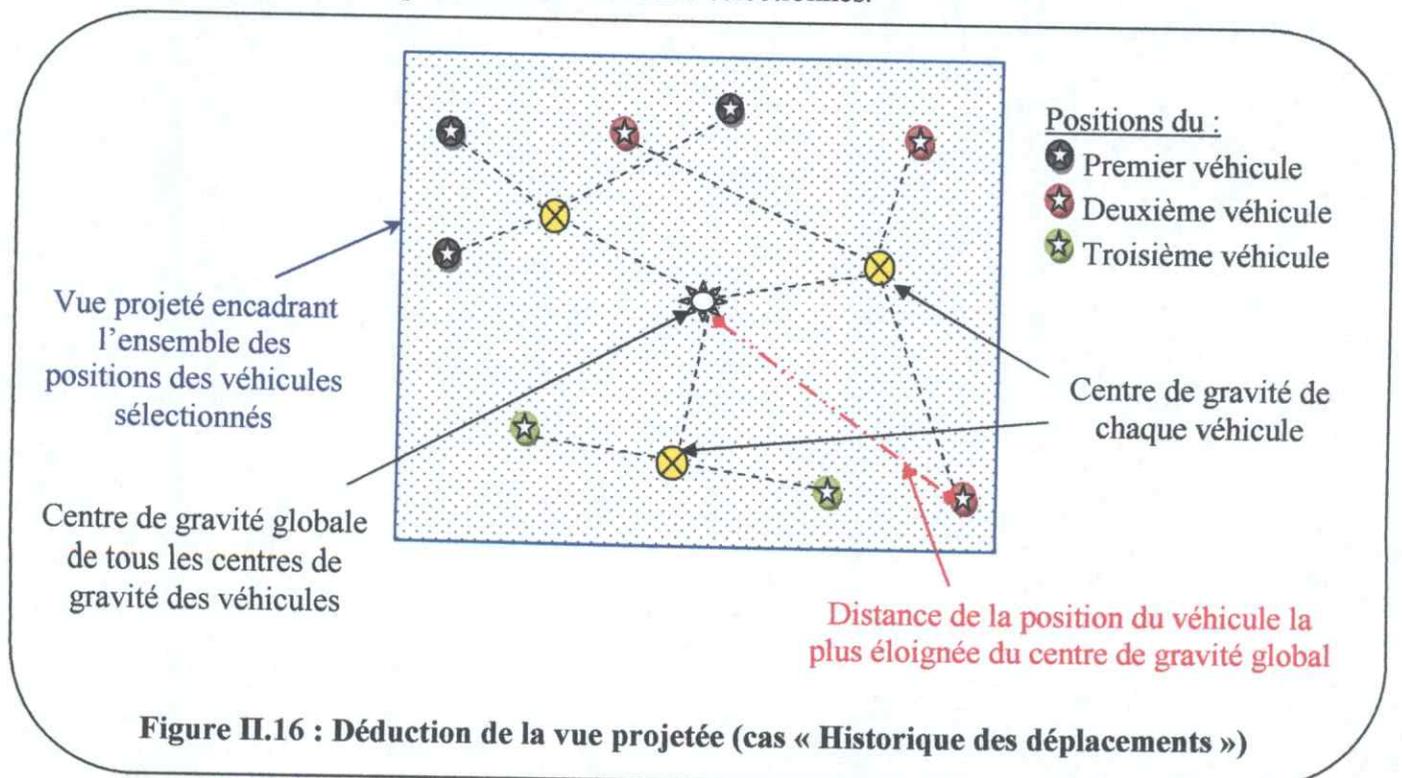


Figure II.15 : Diagramme de séquence de la partie navigation (Poursuit en temps réel).

b. Historique des déplacements:**Scénario :**

- L'abonné accède à la page d'accueil du site.
- L'abonné saisit son nom d'utilisateur et mot de passe.
- Vérification du compte réussi.
- L'abonné accède à la page de navigation du site.
- Chargement des véhicules de l'abonné.
- Envoi d'une requête au serveur cartographique.
- Le serveur cartographique exécute la requête et renvoie le résultat.
- L'abonné sélectionne les véhicules à poursuivre.
- L'abonné choisit l'option « Archives des déplacements ».
- L'abonné choisit la date de début et le nombre de jours à consulter à partir de la date de début.
- L'abonné clique sur « Actualiser » (du site non pas du navigateur WEB).
- Chargement de la base de données des positions des véhicules sélectionnés prises pendant cette période.
- Pour chaque véhicule on calcule le centre de gravité de ses positions.
- Calcul du centre de gravité global des centres de gravité des véhicules.
- Calcul de la distance de la position du véhicule la plus éloignée du centre de gravité global, parmi les toutes les positions des véhicules sélectionnés.
- Déduire les coordonnées de la zone à projeter à partir des deux résultats précédents.
- Déduire la couche d'échelle appropriée.
- Envoi d'une requête SIG au serveur cartographique.
- Le serveur cartographique exécute la requête et renvoie le résultat.
- Affichage du résultat (Vue la plus proche incluant tous les véhicules positionnés).

N.B : Après réflexion nous nous sommes rendu compte que le centre de gravité global des centres de gravité des véhicules est équivalent au centre de gravité de l'ensemble des positions des véhicules sélectionnés.



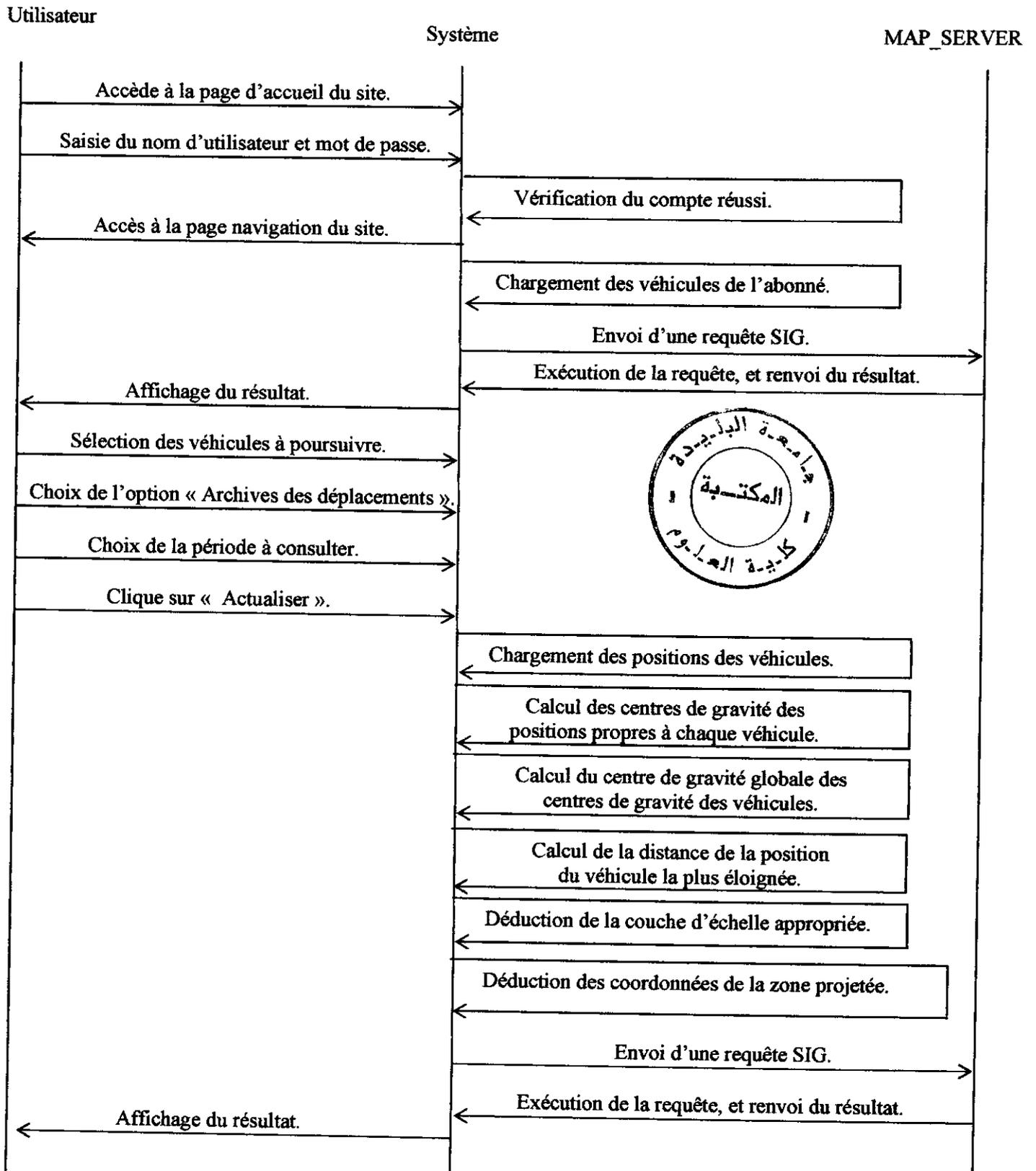


Figure II.17 : Diagramme de séquence de la partie navigation (Historique des déplacements).

IV.5. Digramme de classe :

Notre base de données se limitera aux entités suivantes :

Véhicules, Client, Abonnement, GPS_Kit, Vehicule_Position.

Dictionnaire de données :

Code	Désignation	TYPE
Num_Vehicule	Numéro du véhicule.	INT(20) AUTO_INCREMENT.
Mat	Matricule du véhicule.	CHAR(20).
Libellé	Nom du véhicule apparaissant sur le site de navigation.	CHAR(30).
Marque	Marque du véhicule.	CHAR(30).
id_Client	Identifiant du propriétaire du véhicule.	CHAR(30).
id_Kit	Identifiant du kit GPS lié au véhicule.	INT(30). INT(30).
Num_Client	Numéro du client.	INT(30) AUTO_INCREMENT.
Type	Type du client particulier ou société.	CHAR(20).
Intitulé	Intitulé de la société.	CHAR(50).
Nom	Nom d'un particulier.	CHAR(30).
Prénom	Prénom d'un particulier.	CHAR(30).
Num_Tel	Numéro de téléphone.	CHAR(20).
User	Nom du compte utilisateur permettant l'accès au site.	CHAR(30).
Pass	Mot de passe du compte utilisateur permettant l'accès au site.	CHAR(30).
Cpt_active	Permet d'activer ou de désactiver un compte sans le supprimer	BOOLEAN default '1'.
Num_Abonnement	Numéro d'abonnement	INT(30) AUTO_INCREMENT
Date_Debut	Date début abonnement.	Date default '1900-01-01'
Date_fin	Date fin abonnement.	Date default '1900-01-01'
id_Client	Identifiant du client.	INT(30)
Num_Kit	Identifiant du kit GPS.	INT(30) AUTO_INCREMENT.
IMEI	Numéro de série du kit GPS.	CHAR(50).
Date_disp	Date de disponibilité du kit_GPS	Date default '1900-01-01'.

id_vehicule	Numéro du véhicule.	INT(20).
Date_position	Date à laquelle la position a été prise.	Date time.
Lat	Latitude du véhicule.	Double.
Lon	Longitude du véhicule.	Double.
Vitesse	Vitesse de déplacement du véhicule.	Double.
Altitude	Altitude du véhicule.	Double.

Table II.2 : Dictionnaire de données du site de navigation

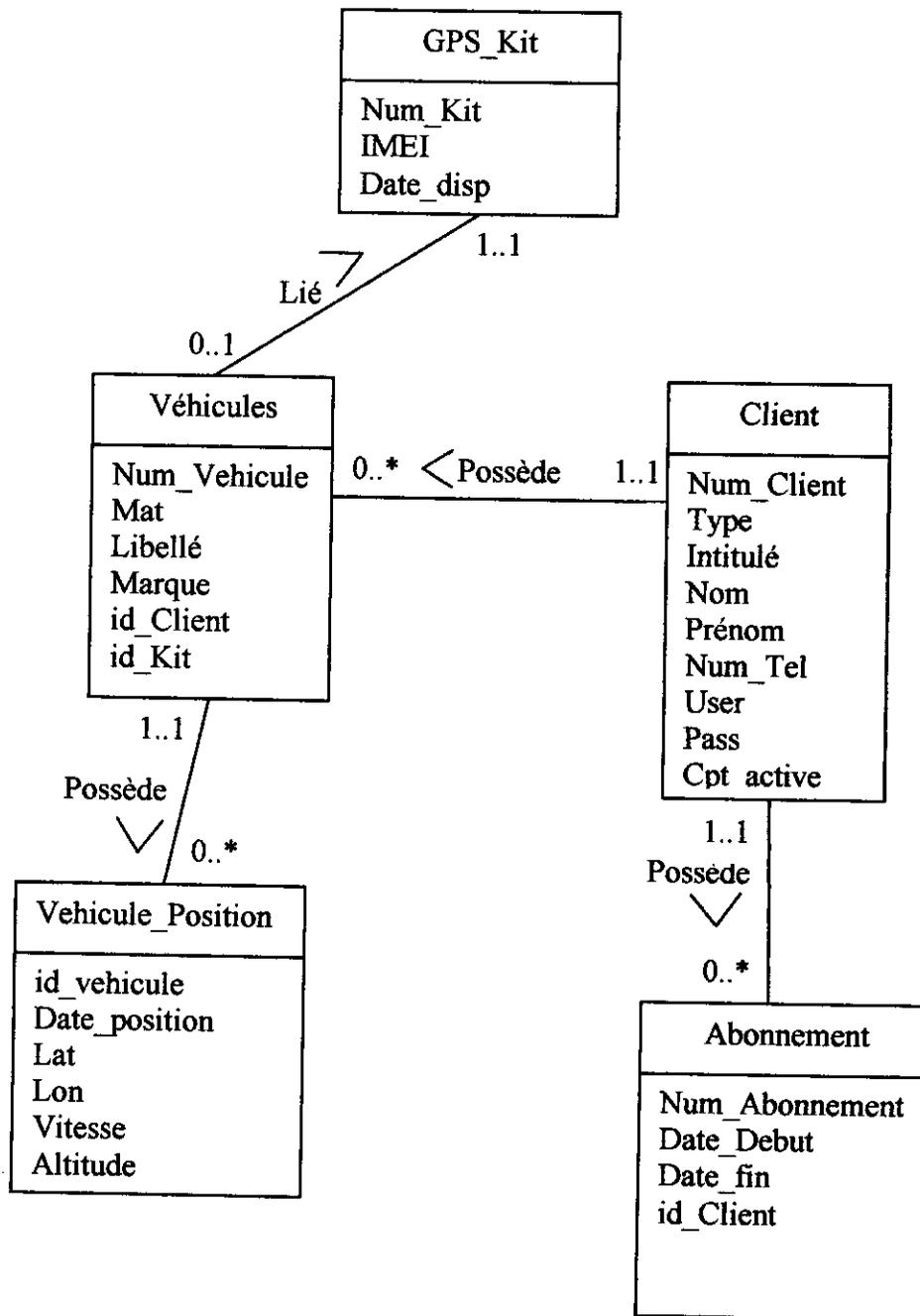


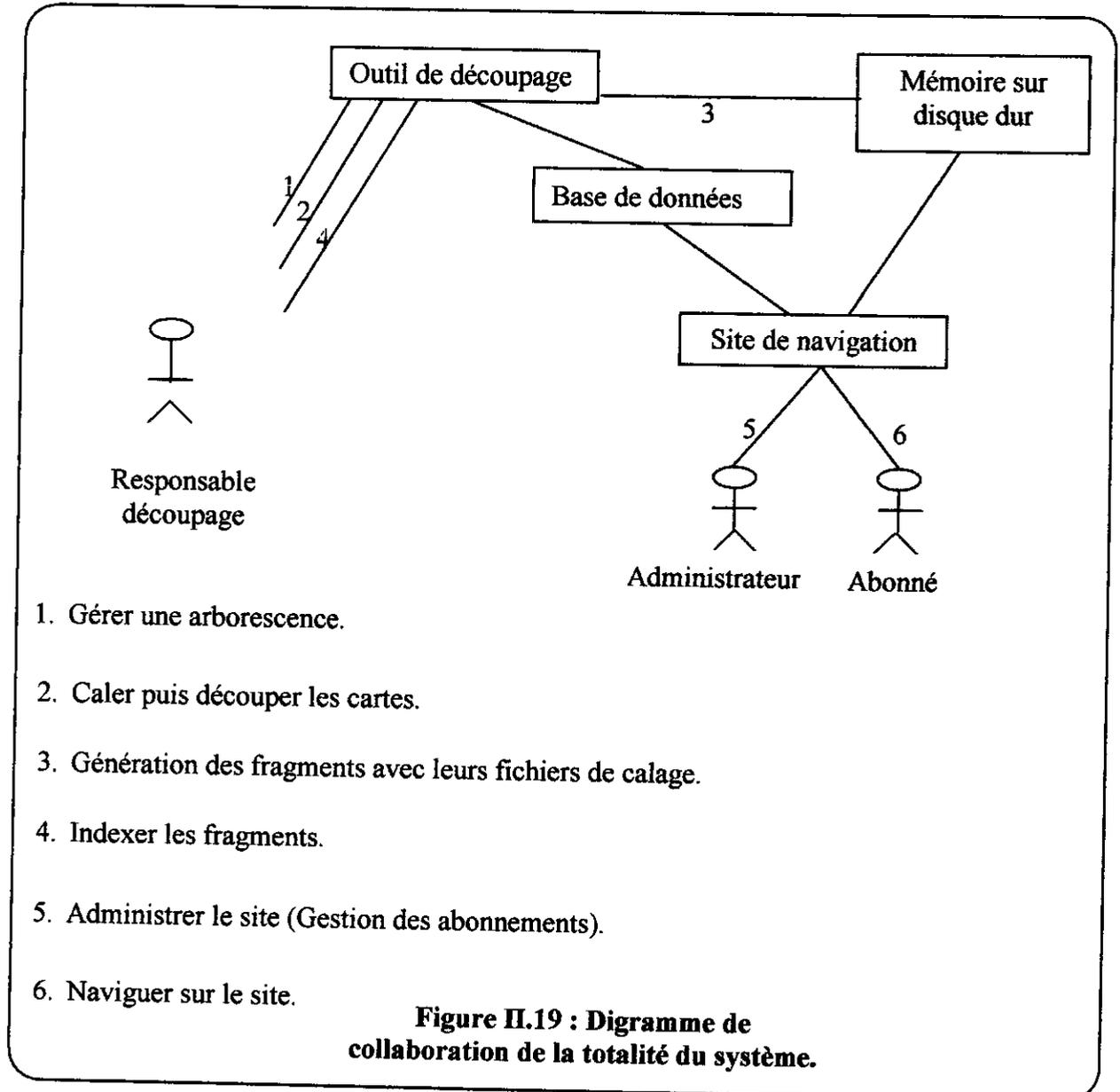
Figure II.18 : Digramme de classe du site de navigation.

V. Conception du Système :

V.1. Digramme de collaboration :

Les diagrammes de collaboration montrent des interactions entre objets (instances de classes et acteurs).

Ils permettent de représenter le contexte d'une interaction, car on peut y préciser les états des objets qui interagissent. [Mull, 1997]



V.2. Diagramme de composants :

Les diagrammes de composants permettent de décrire l'architecture physique et statique d'une application en termes de modules : fichiers sources, bibliothèques, exécutables, etc. Ils montrent la mise en œuvre physique des modèles de la vue logique avec l'environnement de développement.

Les dépendances entre composants permettent notamment d'identifier les contraintes de compilation et de mettre en évidence la réutilisation de composants.

Les composants peuvent être organisés en paquetages, qui définissent des sous-systèmes. Les sous-systèmes organisent la vue des composants (de réalisation) d'un système. Ils permettent de gérer la complexité, par encapsulation des détails d'implémentation. [Mull, 1997]

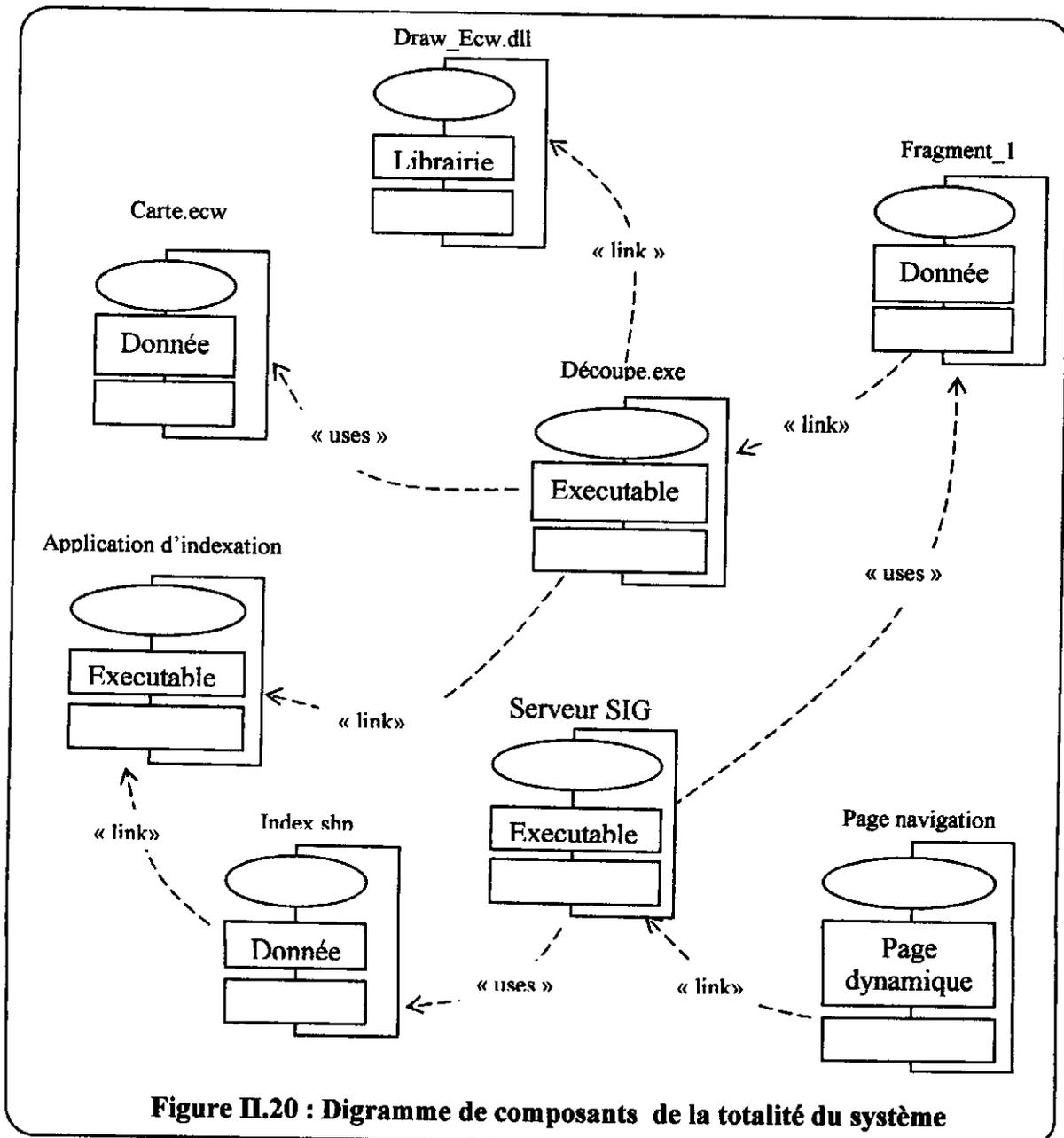


Figure II.20 : Diagramme de composants de la totalité du système

V.3. Digramme de déploiement:

Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.

Les ressources matérielles sont représentées sous forme de nœuds.

Les nœuds sont connectés entre eux, à l'aide d'un support de communication.

La nature des lignes de communication et leurs caractéristiques peuvent être précisées.

Les diagrammes de déploiement peuvent montrer des instances de nœuds (un matériel précis), ou des classes de nœuds.

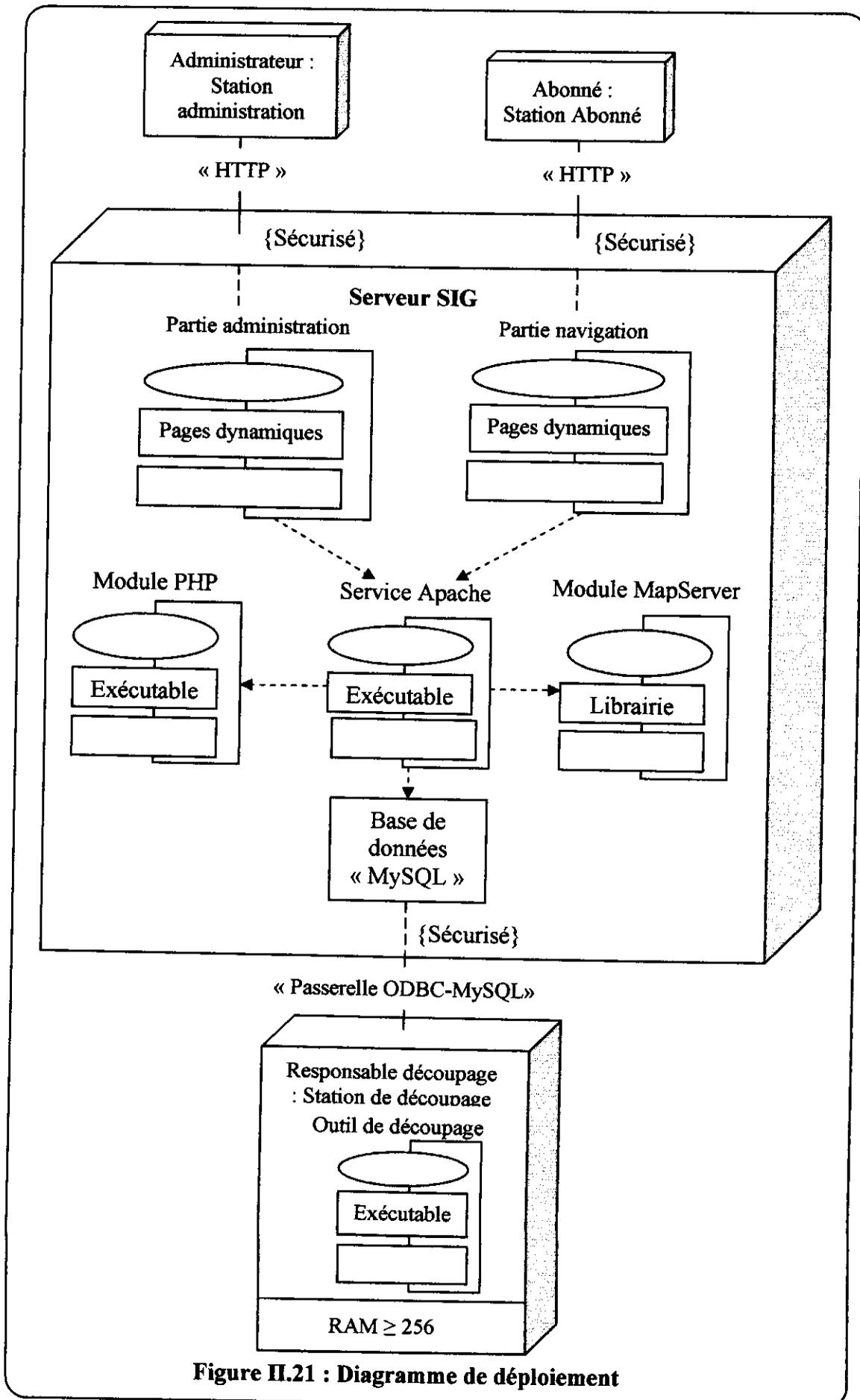


Figure II.21 : Diagramme de déploiement

VI. Implémentation de l'outil :

VI.1. Contexte matériel et logiciel :

Nous supposons que l'entreprise posséderait des P4 équipés si possible d'une carte graphique puissante (ex : Gé force4).

Avant de faire part de notre choix d'outils il serait bon de rappeler la contrainte majeure liée au développement :

- L'outil doit être capable de manipuler des images de grande taille sans mettre en péril le matériel utilisé, tout en étant précis et si possible rapide.

Nous décidons d'utiliser l'environnement « **DELPHI** » pour les raisons suivantes :

- DELPHI offre un environnement de développement d'application visuelle simple, mais efficace.
- DELPHI offre la possibilité de générer des images JPEG.

Ce choix ne satisfait malheureusement pas notre contrainte, d'ailleurs aucun environnement de développement connu (Builder, JAVA, Visual Studio) ne satisfera notre contrainte pour la raison suivante :

Ces environnements de développement ne manipulent pas vraiment les formats compressés, ils les convertissent en format BMP.

Par exemple une image de 17 mégaoctets sous format JPEG vaut environ 160 mégaoctets sous format BMP ce qui engendre une saturation de mémoire, un ralentissement du système voire le blocage total et un réchauffement dangereux de la carte graphique.

Fort heureusement une recherche poussée sur le net nous permet de trouver une bibliothèque qui satisfait notre fameuse contrainte elle s'intitule « **ECWJPEG2000SDK** ».

Cette dernière est une bibliothèque de développement professionnelle réalisée par ER Mapper (Géospatial Imagery Solutions) qui permet de manipuler des images de format ECW et JPEG2000 de taille dépassant 1 Gigaoctet. « www.ermapper.com »

Cette bibliothèque est développée avec « Visual Studio », et ne peut être utilisée ailleurs.

La version utilisée dans notre projet est une version démo gratuite avec laquelle on ne peut ni déboguer ni exporter une image, elle se limite à afficher des images en format ECW ou JPEG2000.

En plus les développeurs de cette bibliothèque nous offrent un convertisseur gratuit du format JPEG vers le format ECW.

La base de données serait stockée par un serveur MySQL.

VI.2. Concept de base de la bibliothèque « ECWJPEG2000SDK » :

L'image est composée d'un ensemble de lignes et de colonnes de données, avec une ou plusieurs couches (selon le nombre de couleurs) pour chaque pixel du tableau de données.

Comment réaliser une vue grâce au concept SetFileView :

Ce concept est très puissant, car il offre à l'utilisateur la possibilité de voir une partie d'une image à un niveau de résolution différent du niveau réel de l'image originale.

Exemple :

Supposons que notre application possède une fenêtre de taille 500x300 pixels et nous ouvrirons une image de taille 15000x10000 pixels, puis nous voulons voir une vue d'ensemble de cette dernière.

Lors de l'affichage de l'image nous avons intérêt à ne pas lire l'ensemble des 15000x10000 pixels, car pour une vue d'ensemble de l'image les détails sont négligeables, donc nous donnerons l'illusion que nous chargeons l'image entière alors qu'en fait nous chargeons 500x300 pixels, ce qui permet de réduire considérablement la consommation des ressources matérielles.

Cette dernière deviendra relativement stable quelque soit la taille de la carte, car elle dépendra uniquement de la zone d'affichage.

La figure suivante illustre comment nous devrions spécifier les coordonnées d'une zone de l'image afin de construire la vue souhaitée,

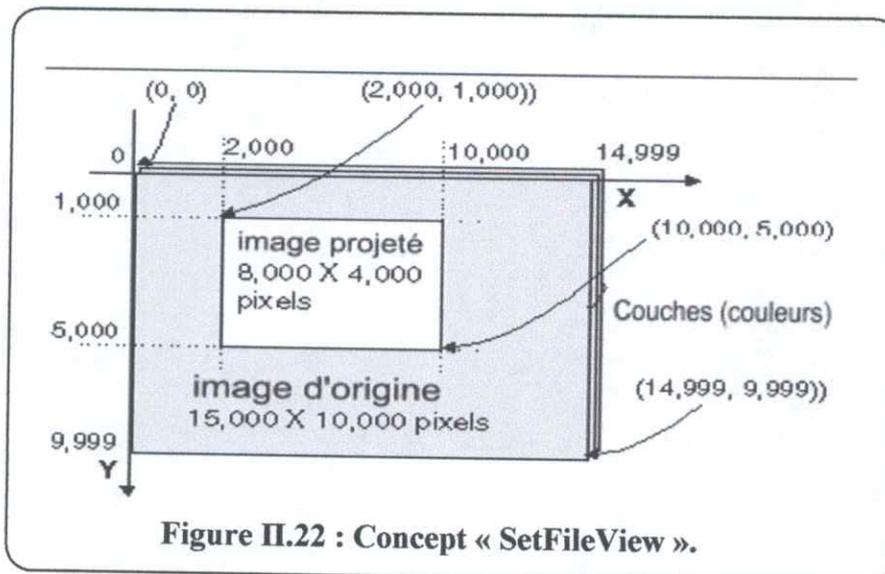
Exemple :

- Pour avoir une vue d'ensemble il suffit de faire :

```
NCScbmSetFileView(pView, nBands, pBandList, 0, 0, 14999, 9999, 500, 300);
```

- Par contre pour projeter une partie de l'image il suffit de faire:

```
NCScbmSetFileView (pView, nBands, pBandList, 2000, 1000, 10000, 5000, 500, 300);
```



pView Pointer vers la vue projetée.

nBands Le Nombre de couleurs à lire de l'image originale.

pBandList Tableau contenant les indices de couleurs,

Exemple :

Si **nBands** égal trois (couleur RGB), alors on déclare **pBandList** comme ceci :

```
INT32 pBandList[3] = { 0, 1, 2 };
```

2000,1000 coordonnées en pixel du coin supérieur gauche de la zone projetée.

10000,5000 coordonnées en pixel du coin inférieur droit de la zone projetée.

500,300 tailles de la zone d'affichage.

VI.3. Traitement de la contrainte principale de l'outil (taille des images):

Nous supposons que l'entreprise aura utilisé le convertisseur pour convertir ces cartes au format ECW.

1. Nous réaliserons dans l'environnement « Visual Studio » une DLL (Bibliothèque dynamique) qui va lire une partie de l'image (Fichier ECW) et l'afficher dans un canevas.

2. Ce dernier serait le canevas d'une image se trouvant dans l'application réalisée en DELPHI, en plus clair :

- L'application réalisée en DELPHI possédera un composant image.
- Chaque composant image possède un pointeur vers son canevas.
- Pour dessiner une partie de la carte nous appelons Notre DLL avec les arguments suivants :
 - Le chemin vers le disque dur de la carte à afficher.
 - La partie à afficher de la carte :
 - Coordonnées du coin supérieur gauche.
 - Coordonnées du coin inférieur droit.
 - Un pointeur vers le canevas de l'image d'affichage.
 - La largeur et la hauteur du canevas (zone d'affichage).

3. L'application réalisée en DELPHI exportera l'image en JPEG.

VI.4. Calage des cartes:

Etant donné que nos cartes sont en UTM ces ajustements se révèlent nécessaires :

- ✓ Rotations.
- ✓ Translations.
- ✓ Changements d'échelle.

En d'autres termes un ajustement polynomial de degré 1 s'impose, et notre système de passage du référentiel image vers le référentiel UTM se définit comme suit :

$$x' = a_0 + a_1x + a_2y$$

$$y' = b_0 + b_1x + b_2y$$

Où :

- x' et y' sont les coordonnées d'un point dans le référentiel du monde réel (en UTM);
- x et y sont les coordonnées d'un point dans le référentiel de l'image (en pixel);
- a et b des coefficients.

Selon la méthode des **moindres carrés**, le nombre minimum d'amers nécessaire à la résolution de ce système est de trois points, donc nous serons confrontés à résoudre les deux systèmes suivants:

$$x' = a_0 + a_1x_1 + a_2y_1$$

$$y' = b_0 + b_1x_1 + b_2y_1$$

$$x' = a_0 + a_1x_2 + a_2y_2$$

ET $y' = b_0 + b_1x_2 + b_2y_2$

$$x' = a_0 + a_1x_3 + a_2y_3$$

$$y' = b_0 + b_1x_3 + b_2y_3$$

Donc

$x' = UA$ et $y' = UB$ tel que :

$$U = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}, \quad A = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \quad \text{et} \quad B = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}$$

Nous serions tenté de faire $A = U^{-1} x'$ et $B = U^{-1} y'$ tel que U^{-1} est la matrice inverse de U, mais cette solution ne s'applique que si le nombre de point est égal à trois point, car le calcul de la matrice inverse de U exige que U soit carrée.

Or nous voulons que le nombre de points de calages (d'amers) soit illimité pour offrir un maximum de précisions à notre calage, alors nous appliquons la solution suivante :

Soit U^t la matrice transverse de U .

$UA = x'$ implique $U^t UA = U^t x'$ donc $A = [U^t U]^{-1} U^t x'$, par similitude $B = [U^t U]^{-1} U^t y'$.

Cette modification est utile car nous savons que le résultat du produit matriciel $U^t U$ est une matrice symétrique et donc bien évidemment carrée, ce qui nous offre la possibilité de calculer la matrice inverse quelque soit le nombre de points de calage.

VL5. Connexion vers la base de données:

L'environnement DELPHI (DELPHI 5 plus exactement) ne prévoit pas de connexion vers une base de données de type MySQL.

Fort heureusement nous avons trouvé sur le net une passerelle ODBC-MySQL grâce à un driver se nomme « `mysql-connector-odbc-3.51.12-win32` ».

Après installation du driver il suffit d'aller dans « Panneau de configuration Windows », puis dans « Outils d'administration », puis « Sources de données (ODBC) », et enfin configurer un alias vers la base de données.

Cette manipulation est évidente pour un informaticien, ceci dit nous aimerions bien automatiser la création de l'alias.

Il faut savoir qu'un alias ODBC n'est qu'un ensemble de valeurs enregistrées dans la base de registres.

Supposons que nous voulons créer un alias se nommant `A_GPS` nous aurons à définir les registres suivants :

Nom du registre	Valeur	Exemple
HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI\A_GPS\Database	Nom de la base de données	GPS
HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI\A_GPS\Driver	path du driver	C:\WINDOWS\system32\myodbc3.dll
HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI\A_GPS\UID	Nom de l'utilisateur	User
HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI\A_GPS\PWD	Mot de passe	Pass
HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI\A_GPS\SERVER	Nom ou Adresse IP du server	LocalHost ou 127.01.01
HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI\ODBC Data Sources	Nom alias	A_GPS

Table II.3 : Registres d'un alias ODBC

VL6. Indexation des fragments générés:

Pour l'implémentation de cette tâche nous utiliserons un outil déjà opérationnel, ce dernier se nomme « gdaltindex », celui-ci fait partie d'une bibliothèque se nommant GDAL qui est livré avec un certain MapServer que nous présenterons plus loin.

Le format d'appel de cet outil est :

```
>> gdaltindex [-tileindex field_name] index_file [gdal_file]*
```

Exemple:

```
>>gdaltindex fichier_index.shp d:/répertoire des images/*.jpeg
```

Cet outil sera appelé directement par l'application réalisée en DELPHI grâce à la fonction **ShellExecAndWait(paramètres)**.

Exemple :

```
ShellExecAndWait ( 'cmd', '/c D:\gdaltindex.exe D:/Index_File/ fichier_index.shp  
D:/ répertoire des images/*.jpg',",0);
```

Nous verrons comment nous utiliserons ces fichiers d'indexations afin d'afficher les vues souhaitées.

VII. Implémentation du site :

VII.1. Contexte matériel et logiciel du site:

Nous supposons que le serveur hébergeant le site soit raisonnablement puissant que les machines ordinaires.

Le *Web Mapping*, ou diffusion de cartes via le réseau Internet, est un domaine en pleine expansion, ceci dit il y'a quelque années le coup du développement d'une solution intégralement indépendante des produits payants reste excessivement élevé.

Aujourd'hui, Le *Web Mapping* connaît une nouvelle source d'enrichissement qui est le développement des solutions *Open Source*, suivant la philosophie GNU qui autorise la copie, la diffusion du logiciel et la modification du code source, ces programmes généralement gratuits et d'utilisation libre émergent à un rythme soutenu.

Au centre de cette sphère en développement permanent né: *MapServer*.
MapServer peut être piloté par plusieurs langages tel que:

- JAVA
- Python.
- PHP.

Nous décidons de développer notre site en PHP, avec MySQL comme serveur de base de données et bien évidemment Apache comme serveur.

VII.2. MapServer(La solution *Open Source*) :

VII.2.1. Présentation de MapServer :

MapServer est un environnement de développement Open Source permettant de construire des applications à référence spatiale, il permet de générer des cartes à partir de données diverses et de fichiers regroupant les paramètres décrivant la façon dont elles doivent être présentées.

MapServer n'est pas un SIG complet, et n'aspire pas à le devenir, le plus souvent, MapServer est utilisé sur un serveur Internet pour générer des images dans des pages web, et ainsi permettre l'affichage mais aussi la modification, d'images cartographiques sur un site Internet. On peut aussi utiliser MapServer sur son ordinateur pour générer des cartes, effectuer des analyses et croisements...

MapServer excelle à afficher les données spatiales sur le web (cartes, images, et données vectorielles), propose un ensemble de fonctions digne des meilleurs outils payants (navigation, requêtes, mais aussi mise à jour à distance), il s'adapte quasiment à tout type d'environnement. Il peut être facilement étendu afin de supporter de nouveaux formats de données, environnements de développement, systèmes d'exploitation ou serveurs *Web*.

MapServer existe sous deux formes concrètes principales :

- Un exécutable, à utiliser à la ligne de commande.
- Programme CGI, et une bibliothèque de fonctions PHP ou Perl : MapScript.

[D, PIN 2006], [L, JEG 2006]

VII.2.2. Origine de MapServer :

MapServer a été développé par l'Université of Minnesota (UMN) dans le cadre du projet ForNet en coopération avec la NASA et le Département des Ressources Naturelles du Minnesota (MNDNR).

A ce jour, le projet MapServer est hébergé par le projet TerraSIP , un projet sponsorisé par la NASA, et réalisé par UMN et un groupement d'acteurs dans le domaine de l'aménagement du territoire.

C'est en fait une bibliothèque de fonctions développée sous licence GPL (General Public Licence) selon les recommandations de l'OGC (Open Geospatial Consortium), il est ainsi régi par des normes publiques qui en assurent sa pérennité.

Le logiciel est maintenu par un nombre grandissant de développeurs (approximativement 20) répartis dans le monde entier.

Il est soutenu par un groupe hétérogène d'organisations qui financent les développements et la maintenance. [[D, PIN 2006], [Mapp, 2007]

VII.3. Présentation du langage PHP:

PHP est un langage de script qui s'inclut dans le langage HTML. La syntaxe du langage PHP provient du C, de Java et du Perl, avec un petit nombre de fonctions inédites par rapport à ces langages. Le but du langage PHP est de permettre aux développeurs de site web d'écrire rapidement des pages web dynamiques. [S, BAK 1999].

VII.4. Organisation concrète d'un site internet :**a. Un site Internet classique (statique) :**

- ◆ L'utilisateur demande à afficher une page dans son navigateur, en donnant son adresse URL.
- ◆ Le navigateur convertit cette demande en une requête HTTP.
- ◆ Cette requête HTTP est convertie par le réseau Internet (DNS et nœuds) en un chemin vers un fichier HTML sur une machine serveur.
- ◆ Le serveur traite cette demande en rassemblant les éléments de la page HTML (fichier html et images, feuilles de styles, scripts, etc., potentiellement utiles).
- ◆ Le serveur envoie au navigateur client les fichiers.
- ◆ Le navigateur traite les fichiers pour afficher la page. [L, JEG 2006]

b. Un site Internet dynamique, avec php et MySQL :

- ◆ L'utilisateur demande à afficher une page dans son navigateur, en donnant son adresse URL.
- ◆ Le navigateur convertit cette demande en une requête HTTP.
- ◆ Cette requête HTTP est convertie par le réseau Internet (DNS et nœuds) en un chemin de fichier PHP sur une machine serveur.
- ◆ Le serveur transfère la requête à PHP.
- ◆ Php traite le fichier demandé, c'est à dire interprète le script en langage php, ce script peut :
 - ◆ écrire du texte, donc générer des balises HTML.
 - ◆ lire et modifier des images.
 - ◆ exécuter des requêtes sur un SGBD et en traiter les données des résultats.
- ◆ Php convertit le résultat de l'exécution du fichier en un fichier HTML, qui est retourné au serveur.
- ◆ Le serveur traite ce fichier en rassemblant les éléments de la page HTML (fichier html et images, feuilles de styles, scripts, etc., potentiellement utiles).
- ◆ Le serveur envoie au navigateur client les fichiers.
- ◆ Le navigateur traite les fichiers pour afficher la page. [L, JEG 2006]

VII.5. Organisation concrète d'un serveur SIG basé sur PHP/MapServer :

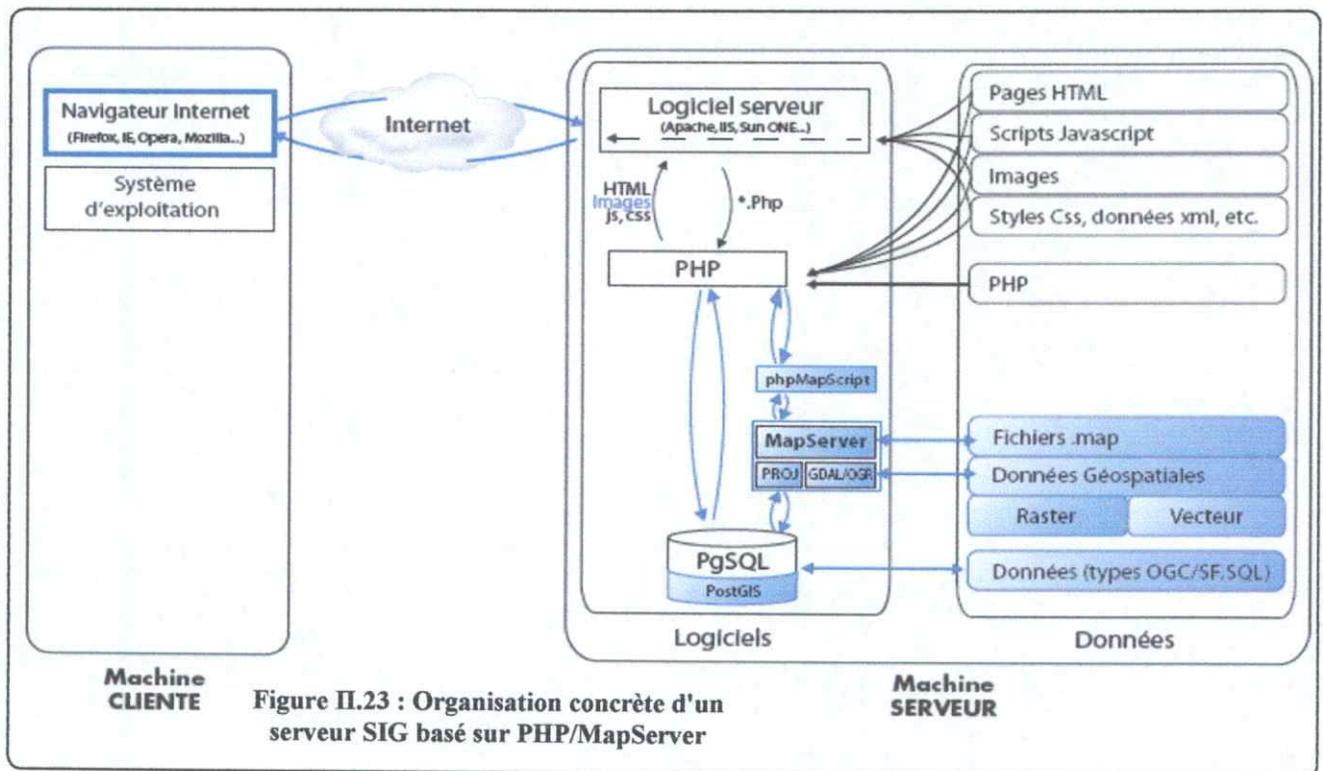


Figure II.23 : Organisation concrète d'un serveur SIG basé sur PHP/MapServer

- ◆ L'Utilisateur demande à afficher une page dans son navigateur, en donnant son adresse URL.
- ◆ Le navigateur convertit cette demande en une requête HTTP.
- ◆ Cette requête HTTP est convertie par le réseau Internet (DNS et nœuds) en un chemin de fichier PHP sur une machine serveur.
- ◆ Le serveur transfère la requête à PHP.
- ◆ Php traite le fichier demandé, c'est à dire interprète le script en langage php, ce script peut :
 - ◆ écrire du texte, donc générer des balises HTML.
 - ◆ lire et modifier des images.
 - ◆ exécuter des requêtes sur un SGBD et en traiter les données des résultats.
 - ◆ Lancer des requêtes SIG a PhpMapScript.
- ◆ PhpMapScript traite les requêtes SIG / MapScript.
- ◆ MapScript traite ces requêtes en préparant un ou plusieurs documents (images), à partir de :
 - ◆ données raster.
 - ◆ données vecteur.

- ◆ données stockées sur serveur SGBD avec extension spatiale.

Ces données sont le plus souvent liées à un **fichier de configuration de type MapFile** d'extension « .map », et peuvent être accessibles en local ou sur d'autres serveurs, distants.

- ◆ Les images générées par MapServer sont ajoutées aux sources de la page rassemblées par PHP.
- ◆ Php convertit le résultat de l'exécution du fichier en un fichier HTML, qui est retourné au serveur.
- ◆ Le serveur traite ce fichier en rassemblant les éléments de la page HTML (fichier html et images, feuilles de styles, scripts, etc., potentiellement utiles).
- ◆ Le serveur envoie au navigateur client les fichiers.
- ◆ Le navigateur traite les fichiers pour afficher la page.

Cette organisation se complique le plus souvent dans les faits par une interface utilisateur, généralement basée sur un formulaire qui va aider à écrire la requête en direction de MapServer, avec notamment la gestion des cliques de la souris sur une image pour demander un zoom, un déplacement, et l'affichage de case à cocher, de listes de champs et de paramètres. [L, JEG 2006]

VII.6. Fichier de configuration mapfile :

Le mapfile détermine les données à utiliser dans une application et à afficher ainsi que les paramètres de requête; il inclut aussi les informations sur la manière de dessiner la carte, la légende, et les cartes qui résultent d'une requête.

Les mapfiles ont normalement une extension **.map** [L, JEG 2006]

VII.7. Mécanisme de couches et de projections de MapServer:

MapServer possède un mécanisme puissant et ingénieux de superposition de couches.

Chaque couche est définie dans un bloque LAYER dans le fichier de configuration mapfile.

Le fichier de configuration mapfile contient un bloc principal, qui peut être considéré comme étant une couche principale.

Les couches sont dessinées dans l'ordre du mapfile, c'est-à-dire que le premier bloc principal du mapfile est dessiné en premier, les suivants viendront en ordre par-dessus sur la carte. L'ordre du mapfile est donc l'ordre inverse de la superposition verticale des sources.

Si nous devons afficher plusieurs couches de projections différentes, la couche principale doit obligatoirement posséder un bloc projection définissant sa projection, suivi de l'**extent**.

L'**extent** décrit les limites de la vue projetée, il se définit en 4 valeurs : MinX, MaxX, MinY, MaxY.

Chaque couche du fichier mapfile doit suivre une projection donnée, ceci n'implique pas forcément que nous devons définir pour chaque couche le bloc projection, car dans le cas où la couche est démunie du bloc projection, MapServer considérera la projection de couche principale comme étant propre à la couche démunie de projection.

MapServer considère l'**extent** du bloc principale comme étant propre chaque couche déclarée à l'intérieur du bloc principale.

Si une couche possède une projection différente de celle du bloc principal, la définition du bloc projection permettra à MapServer de dessiner correctement chaque couche à la position géographique désirée, en appliquant des conversions de coordonnées.

VII.8. Obtention, Installation et test de MapServer :

VII.8.1. Obtention de MapServer :

A l'origine MapServer est un projet OpenSource présenté par l'Université du Minnesota (<http://mapserver.gis.umn.edu/>), comme un code source générant un exécutable à installer sur un serveur Unix ou compatible (linux). Une installation « normale » de MapServer suppose donc les étapes suivantes :

- ✓ téléchargement de la dernière version des fichiers source (en langage C)
- ✓ téléchargement des bibliothèques complémentaires (proj, ogr, freetype, gdal ...)
- ✓ configuration, compilation, installation des bibliothèques complémentaires,
- ✓ configuration des sources de MapServer (./config)
- ✓ compilation par makefile (make)
- ✓ installation (make install)

Ces opérations ne sont pas simples à effectuer sur une machine unix (linux), et peuvent prendre plusieurs jours de travail à un débutant.

Pour simplifier (fortement) cette mise en place, le site MapTools.org propose MS4W MapServer for Windows un dossier compressé permettant d'installer :

- ✓ Apache version 2.2.3
- ✓ PHP version 5.2.0 (MS4W 2.x) or 4.4.4 (MS4W 1.x)
- ✓ MapServer 4.10.0 CGI and MapScript (CSharp, Java, PHP, Python)
- ✓ support GD 2.0.33, FreeType 2.1.10, GDAL/OGR 1.3.2, PROJ, WMS/WFS, Flash, PDF, ECW3.1, PostGIS, GEOS, libcurl 7.15.1
- ✓ mapserver utilities
- ✓ gdal/ogr utilities
- ✓ proj.4 utilities
- ✓ shp2tile utility
- ✓ shapelib utilities
- ✓ shpdiff utility
- ✓ PHP_OGR Extension 1.1.1
- ✓ OWTChart 1.2.0

Cette ressource est disponible à l'adresse : <http://maptools.org/ms4w/>

[L, JEG 2006]

VII.8.2. Installation de MapServer :

Il suffit de décompresser le fichier à la racine d'un disque dur, et de lancer un fichier batch pour installer le serveur apache comme service (c'est à dire application lancée automatiquement par l'OS).

Le démarrage du serveur peut se faire par l'utilitaire « apache Monitor » ou en lançant manuellement le service.

Après l'installation, le pointage d'un navigateur sur l'adresse <http://localhost> aboutit sur la page d'accueil du serveur SIG nouvellement installé.

On dispose ainsi des principaux outils nécessaires au test et à la mise en place d'un SIG en ligne basé sur MapServer. [L, JEG 2006]

VII.8.3. Test de MapServer :

Pour tester si l'installation fonctionne, vous pouvez télécharger le fichier test_intro.zip qui est disponible sur le FTP de ressources pédagogiques du Département de Géo :

ftp://geoprdr:mjjudu@srv-utmftp.univ-tlse2.fr/cours/dess_sigma/sigweb2006/exemples_seances1&2.zip

Ensuite, vous pouvez décompresser ce fichier dans le répertoire racine http du serveur : C:\ms4w\Apache\htdocs

L'accès http à ce répertoire est donc, directement : <http://localhost>. Pour tester si MapServer fonctionne, essayons de faire dessiner une carte simple à MapServer, en utilisant un petit script PHP/MapScript qui lit les paramètres dans simple.map :

<http://localhost/introduction/index.php> [L, JEG 2006]

VIII. Vision détaillée du développement de la totalité du système :**VIII.1. Quelques décisions importantes pour implémenter le site:**

Une question importante n'a toujours pas été abordée, **que va mettre à jour l'outil de découpage concernant la base de données ?**

Pour y répondre nous devons tout d'abord savoir quelles sont les données nécessaires au développement du site.

La base du fonctionnement de notre site est la génération des fichiers de configurations de type « mapfiles », selon les scénarios de navigation deux cas de générations de fichiers mapfiles ont été cités :

1. Le premier cas utilisé pour afficher l'Algérie complète, celui-ci constitue un cas d'exception car nous connaissons à l'avance tous les paramètres du fichier de configuration.
2. Le deuxième est le cas général où nous aurons besoin de définir :
 - Combien de couches raster doit-on générer, ces couches servent à afficher des fragments de cartes pour constituer la vue souhaitée.
 - Pour chaque couche raster nous définirons deux valeurs importantes :
 - La projection de la couche.
 - L'**extent** de la couche qui déterminera les coordonnées de la **zone d'affichage** sur cette couche.
 - Combien de couche vectrice doit-on générer, ces couches servent à afficher les positions des véhicules souhaités.
 - Pour chaque couche vectrice nous définirons aussi deux valeurs importantes :
 - La projection de la couche.
 - Les coordonnées des positions du véhicule.

La définition de l'**extent** pose une question plus importante, **sur quelle projection doit-on déterminer les coordonnées de la zone d'affichage ?**

La réponse à cette question dépendra des deux paramètres suivants :

- Système des coordonnées émises par les kits GPS, ces coordonnées obéissent au système de référence terrestre WGS84, cela dit ils peuvent être convertit en UTM.
- Système des coordonnées des fichiers de calages des fragments, devrions nous les définir en WGS84 ou en UTM ? sachant que l'indexation des fichiers de calages exige que les fichiers indexés doivent partager le même système de coordonnées et la même échelle, or nous savons que l'Algérie est divisée en 4 UTM (UTM29, UTM30, UTM31, UTM32).

Cette analyse nous conduit naturellement vers cette première approche :

❖ **Approche « WGS » :**

- Nous gardons les coordonnées émises par les kits GPS en WGS84.
- Nous regroupons tout les fragments d'une même échelle dans un dossier apparenté.
- Nous générons des fichiers de calage en WGS84.
- Nous générons un fichier d'indexation pour toutes les cartes d'une échelle donnée (ex : index_2000000.shp, index_5000000.shp...).
- Nous définissons la projection du bloc principal du fichier mapfile comme étant WGS84 avec un **extent** défini évidemment sur la même projection WGS84.
- Nous définissons une seule couche raster.
- Nous définissons une couche vectrice pour chaque véhicule, avec des points définis en WGS84 représentant les positions de ce véhicule.

Cette solution est tout à fait réalisable, mais comporte un inconvénient majeur qui est le temps de recherche ou plus exactement le temps d'accès au fichier d'indexation (car les fichiers sont indexés), car MapServeur procédera à la recherche de chaque fragment parmi l'ensemble constituant la carte d'Algérie.

Bien que ces fragments soient indexés, une diminution de temps de recherche améliorerait le temps de réponse de notre site.

Pour cela nous proposons une solution plus complexe :

❖ **Approche « UTM_WGS » :**

- Nous gardons les coordonnées émises par les kits GPS en WGS84.
- Nous regroupons tous les fragments d'une même échèle dans un dossier apparenté.
- Nous regroupons toutes les fragments d'un même UTM dans un sous dossier apparenté.

Exemple :

Dossier racine :

- Echelle200000 :

UTM29

UTM30

UTM31

Alger

Alger_Nord

Alger_Sud

Alger_Est

Alger_West

Blida

UTM32

Batna

- Echelle500000 :

UTM29

UTM30

UTM31

Alger

Alger_Nord

Alger_Sud

Alger_Est

Alger_West

Blida

UTM32

Batna

- Nous générons des fichiers de calage en UTM.
- Nous générons un fichier d'indexation pour l'ensemble des fragments d'un même UTM d'une même échelle (ex : index_25000_UTM31.shp, index_25000_UTM32.shp, index_200000_UTM31.shp...).
- Nous définissons la projection du bloc principal du fichier mapfile comme étant WGS84 avec un **extent** défini évidemment sur la même projection WGS84.
- Nous définissons quatre couches rasters, une couche pour chaque UTM.
- Les couches rasters seront munies de la projection UTM.
- Nous définissons une couche vectrice pour chaque véhicule, avec des points définis en WGS84 représentant les positions de ce véhicule.
- Les couches vectrices seront quand à eux dépourvue de la définition de la projection.

N.B : dans cette approche nous profitons du mécanisme puissant **de couches et de projections** de MapServeur :

- MapServer déduira l'**extent** des couches rasters grâce à la connaissance de :
 - La projection du bloc principal (WGS84).
 - L'**extent** du bloc principal.
 - La projection des couches rasters (UTM).
- MapServer considérera la projection des couches vectrices comme étant celle du bloc principal (WGS84).

Cette approche nous permet d'avoir un meilleur temps de réponse, car le temps d'accès aux fichiers d'indexation sera réduit jusqu'à quatre fois moins en comparaison avec la méthode précédente et ceci grâce à l'optimisation de la taille du fichier d'indexation, qui concernera un UTM parmi les quatre.

Cas échéant : si les fragments recherchés se trouvent à la jonction de deux UTM le temps d'accès est divisé seulement par deux.

Ceci dit l'approche nommée « UTM_WGS » ne peut satisfaire l'ensemble des cartes disponibles, en effet l'organisme d'accueil possède des cartes qui ne peuvent être découpé en fragments calés en UTM, pour la simple raison que ces cartes présentent l'Algérie entière et donc les 4 UTM, or nos connaissances concernant l'UTM sont limités, aussi il nous est impossible de savoir séparer les quatre différents parties d'UTM de la totalité de la carte.

En résumer la deuxième approche ne s'applique qu'aux cartes définies dans un UTM unique.

Nous décidons d'adopter les deux approches en même temps en donnant à l'utilisateur de l'outil de découpage la possibilité de choisir la projection d'une échelle donnée.

Cette analyse démontre la nécessité de gérer une arborescence, afin d'organiser la génération des fragments.

Maintenant nous pouvant terminer la conception de l'outil de découpage, et nous devons raffiner la partie navigation du site.

Tout d'abord nous définirons le diagramme de séquence de la gestion d'arborescence puis nous réétudierons la partie découpage d'une carte déjà calée.

Remarque : nous n'avons toujours pas répondu à la question : **que va mettre à jour l'outil de découpage concernant la base de données ?** Car nous y reviendrons plus tard.

VIII.2. Suite de la conception de l'outil de découpage:**VIII.2.1. Gérer une arborescence :****Scénario :**

- L'utilisateur sélectionne dans le menu principal le bouton « Configuration » puis « Gestion de l'arborescence ».
- La fiche « Gestion de l'arborescence » s'affiche.
- L'utilisateur sélectionne le dossier racine où doivent être stockés les dossiers contenant les différents fragments.
- L'utilisateur saisit la valeur d'une échelle exemple '200000'.
- L'utilisateur choisit l'UTM comme projection de l'échelle.
- L'utilisateur clique sur « Ajouter une échelle ».
- Ce qui provoque :
 - o L'ajout de l'échelle dans la liste visuelle 'échelle disponible'.
 - o Une génération automatique du dossier portant le nom de l'échelle exemple '200000'.
 - o Une génération automatique de 4 sous dossiers suivants :
 - UTM29
 - UTM30
 - UTM31
 - UTM32
- L'utilisateur sélectionne avec le bouton droit de la souris un des dossiers provoquant l'apparition d'un menu contextuel contenant :
 - Ajouter un dossier.
 - Supprimer un dossier.
 - Copier un dossier.
 - Coller un dossier.
- L'utilisateur clique sur le sous menu « Ajouter un dossier ».
- Structurer le nom du nouveau dossier pour éviter une redondance avec les dossiers existants exemple : Nouveau Dossier01, Nouveau Dossier02 ...
- Un nouveau dossier est généré sur disque dur avec le nom structuré.
- Un nouveau dossier est ajouté dans l'arborescence en mode édition.
- L'utilisateur édite le nom du dossier.
- Mise à jour du nom du dossier sur disque dur.
- L'utilisateur sélectionne un dossier avec le bouton droit.
- L'utilisateur clique sur le sous menu « Supprimer un dossier ».
- Une boîte de dialogue demandant la confirmation de la suppression s'affiche.
- L'utilisateur confirme en appuyant sur « Entrer ».
- Le dossier est supprimé sur disque dur.
- Le dossier est retiré de l'arborescence.
- L'utilisateur clique sur « Valider ».
- Mise à jour de la base de données.
- La fiche « Gestion d'arborescence » se ferme.

N.B : Si l'utilisateur choisit la projection WGS84 au lieu de choisir l'UTM, ceci produira seulement le résultat suivant:

- o L'ajout de l'échelle dans la liste visuelle 'échelle disponible'.
- o Une génération automatique du dossier portant le nom de l'échelle exemple '200000'.

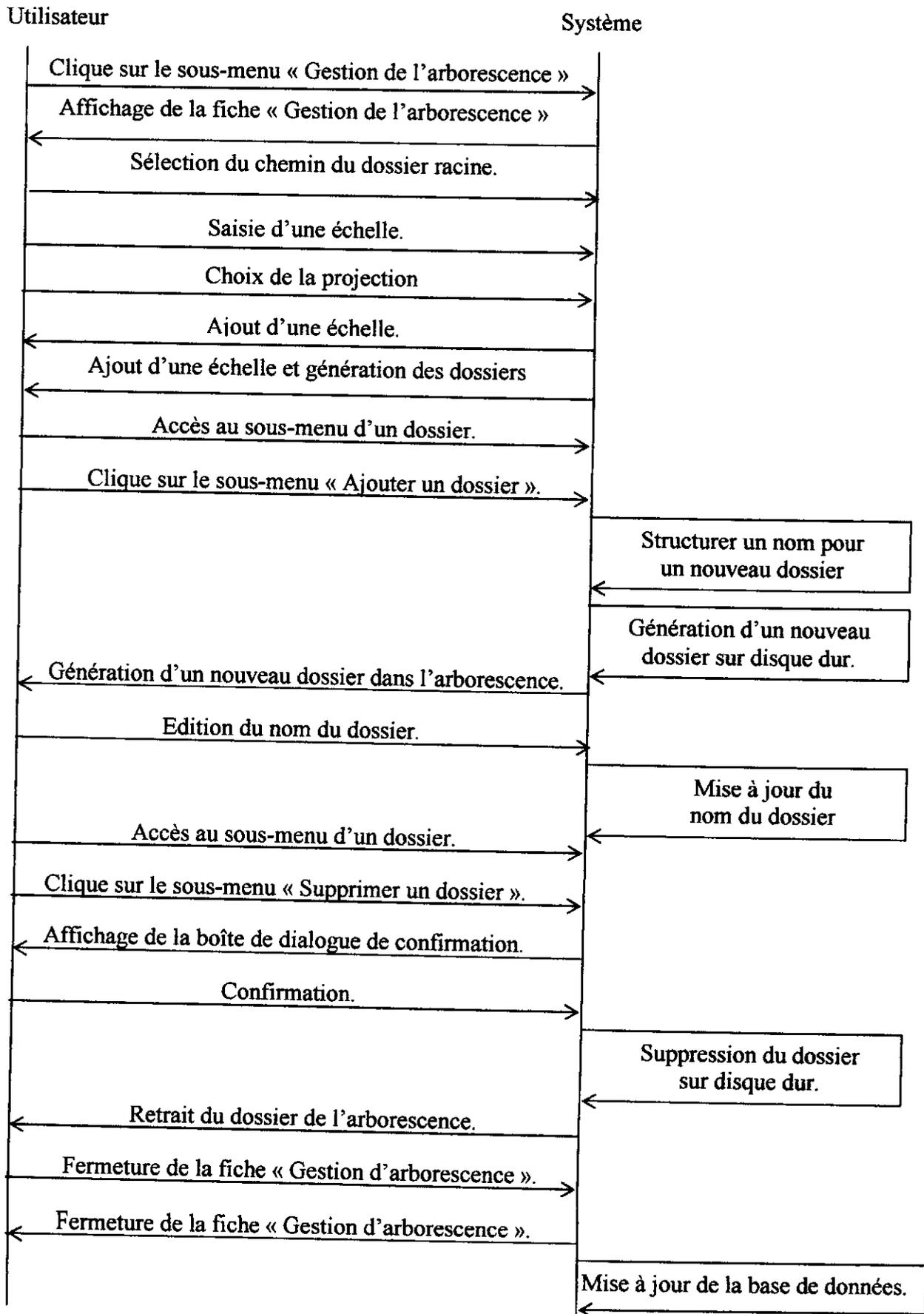


Figure II.24 : Diagramme de séquence Gestion d'une arborescence.

VIII.2.2. Découper une carte déjà calée:**Scénario :**

- L'utilisateur sélectionne l'échelle de la carte.
- Affichage de la projection de l'échelle sélectionnée.
- La projection étant l'UTM l'outil active la partie choix de l'UTM.
- L'utilisateur sélectionne la zone UTM de la carte.
- Affichage d'une partie de l'arborescence, cette partie doit correspondre à l'échelle et la zone UTM sélectionnée.
- L'utilisateur saisit les paramètres de découpage:
 - Taille de fragments (les fragments seront carrés).
 - Taux de compression des fragments (sortie en format JPEG).
 - Nom de la carte.
- L'utilisateur sélectionne un dossier dans l'arborescence où les fragments doivent être enregistrés.
- L'utilisateur clique sur « découper ».
- La progression du découpage est affichée.
- Le programme génère les fragments.
- Le programme génère un fichier de calage d'extension « .WLD » pour chaque fragment.
- Mise à jour de la base de données.
- Un message « fin du découpage » est affiché.

N.B : Si l'utilisateur choisit une échelle dont la projection est le WGS84 alors l'outil désactive le choix de l'UTM et affiche directement la partie de l'arborescence qui correspond seulement à la l'échelle sélectionnée.

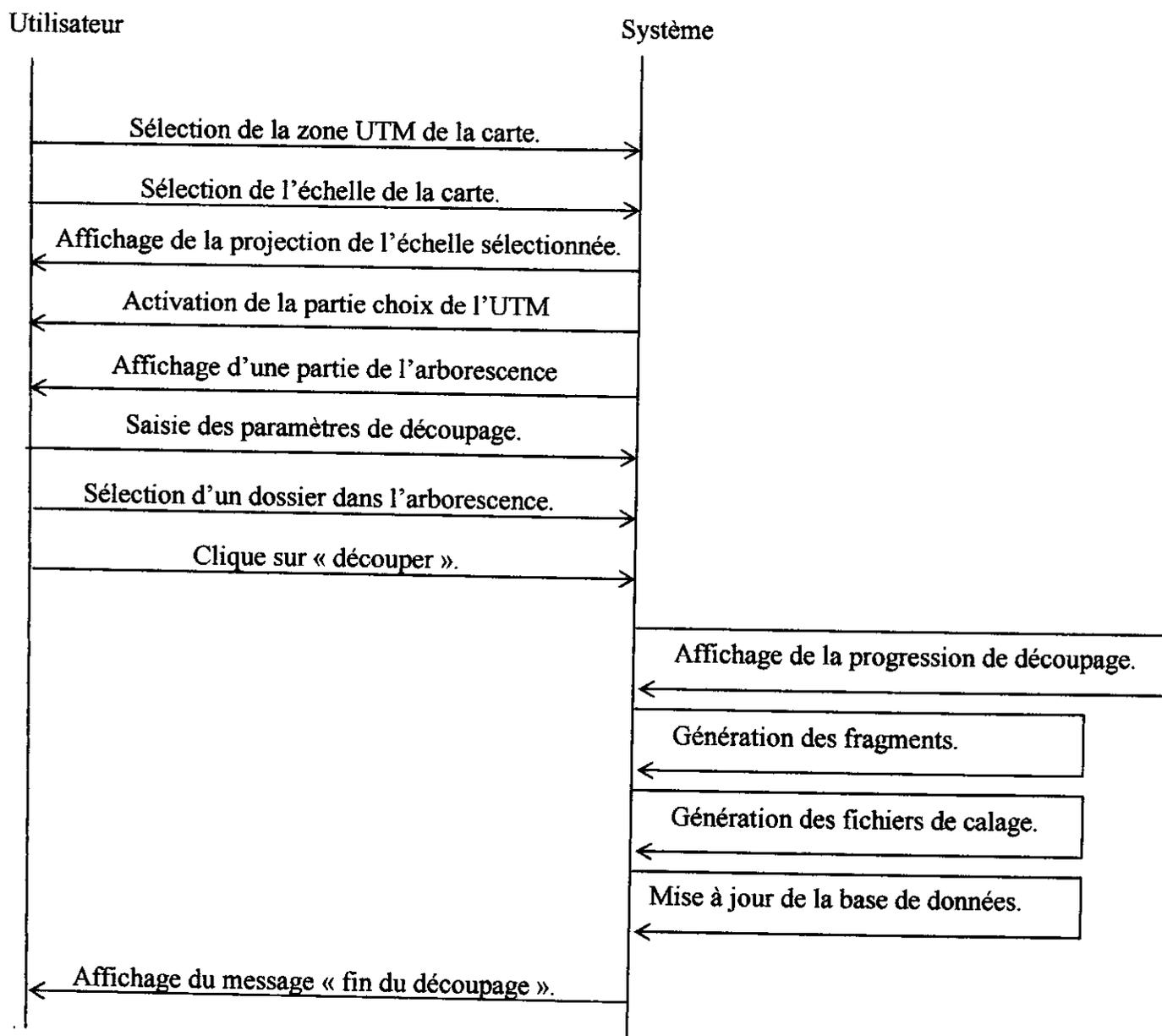


Figure II.25 : Diagramme de séquence « découper une carte déjà calée ».

VIII.3. Conception détaillée de la partie navigation du site :**a. Poursuite en temps réel:****Scénario :**

- L'abonné accède à la page d'accueil du site.
- L'abonné saisit son nom d'utilisateur et mot de passe.
- Vérification du compte réussi.
- L'abonné accède à la page navigation du site.
- Chargement des véhicules de l'abonné.
- Générer un fichier de configuration avec une zone de projection prédéfinie contenant une seule couche de type raster présentant l'Algérie entière.
- Envoi d'une requête SIG liée au fichier de configuration au serveur cartographique.
- Le serveur cartographique exécute la requête et renvoie le résultat.
- Affichage du résultat (vue d'ensemble de la carte d'Algérie).
- L'abonné sélectionne les véhicules à poursuivre.
- L'abonné choisit l'option « Poursuite en temps réel ».
- L'abonné clique sur « Actualiser » (du site non pas du navigateur WEB).
- Chargement de la base de données de la dernière position des véhicules sélectionnés.
- Calcul du centre de gravité des véhicules sélectionnés.
- Calcul de la distance du véhicule le plus éloigné du centre de gravité, parmi les véhicules sélectionnés.
- Déduire les coordonnées de la zone à projeter à partir des deux résultats précédents.
- Déduire la couche d'échelle appropriée.
- Si la projection de la couche est en UTM alors le système génère un fichier de configuration contenant :
 - o Quatre couches de type raster pour les quatre zones d'UTM d'Algérie (UTM29, UTM30, UTM31, UTM32) avec la couche d'échelle appropriée.
 - o Une couche de type vecteur pour chaque véhicule contenant un objet représentant la dernière position du véhicule.
- Envoi d'une requête SIG liée au fichier de configuration au serveur cartographique.
- Le serveur cartographique exécute la requête et renvoie le résultat.
- Affichage du résultat (Vue la plus proche incluant tous les véhicules positionnés).

N.B : Si la projection de la couche est en WGS84 alors le système génère un fichier de configuration contenant :

- o Une couche de type raster en WGS84 pour la couche d'échelle appropriée.
- o Une couche de type vecteur pour chaque véhicule contenant un objet représentant la dernière position du véhicule.

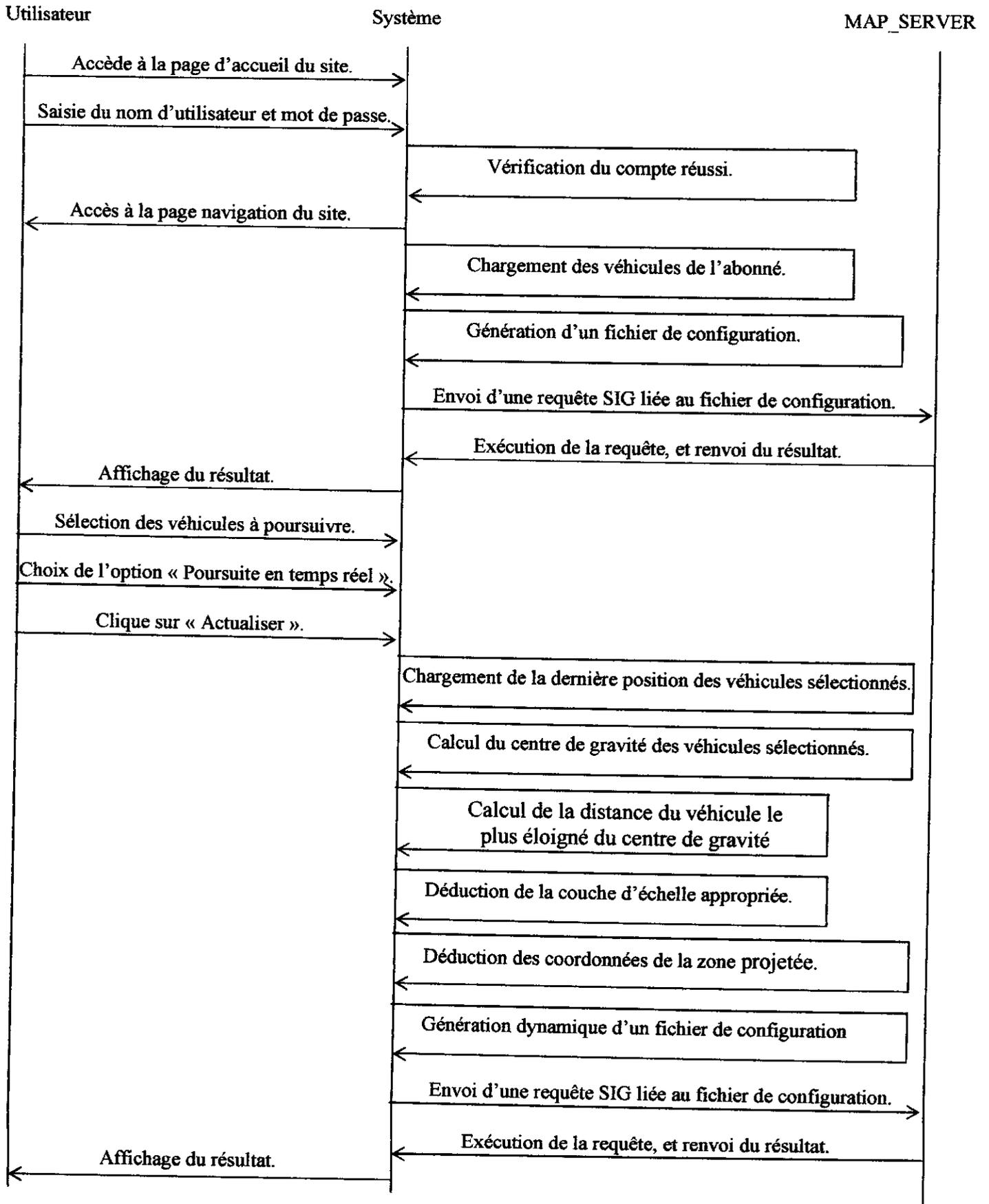


Figure II.26 : Diagramme de séquence de la partie navigation (Poursuite en temps réel).

b. Historique des déplacements:**Scénario :**

- L'abonné accède à la page d'accueil du site.
- L'abonné saisit son nom d'utilisateur et mot de passe.
- Vérification du compte réussi.
- L'abonné accède à la page navigation du site.
- Chargement des véhicules de l'abonné.
- Générer un fichier de configuration avec une zone de projection prédéfinie contenant une seule couche de type raster présentant l'Algérie entière.
- Envoi d'une requête SIG liée au fichier de configuration au serveur cartographique.
- Le serveur cartographique exécute la requête et renvoie le résultat.
- L'abonné sélectionne les véhicules à poursuivre.
- L'abonné choisit l'option « Archives des déplacements ».
- L'abonné choisit la date début et le nombre de jours à consulter à partir de la date début.
- L'abonné clique sur « Actualiser » (du site non pas du navigateur WEB).
- Chargement de la base de données des positions des véhicules sélectionnés prises pendant cette période.
- Pour chaque véhicule on calcule le centre de gravité de ses positions.
- Calcul du centre de gravité global des centres de gravité des véhicules.
- Calcul de la distance de la position du véhicule la plus éloignée du centre de gravité global, parmi toutes les positions des véhicules sélectionnés.
- Déduire les coordonnées de la zone à projeter à partir des deux résultats précédents.
- Déduire la couche d'échelle appropriée.
- Générer un fichier de configuration contenant :
 - o Quatre couches de type raster pour les quatre zones d'UTM d'Algérie (UTM29, UTM30, UTM31, UTM32) avec la couche d'échelle appropriée.
 - o Une couche de type vecteur pour chaque véhicule contenant des objets représentant les positions d'un véhicule.
- Envoi d'une requête SIG liée au fichier de configuration au serveur cartographique.
- Le serveur cartographique exécute la requête et renvoie le résultat.
- Affichage du résultat (Vue la plus proche incluant tous les véhicules positionnés).

N.B : Si la projection de la couche est en WGS84 alors le système génère un fichier de configuration contenant :

- o Une couche de type raster en WGS84 pour la couche d'échelle appropriée.
- o Une couche de type vecteur pour chaque véhicule contenant un objet représentant la dernière position du véhicule.

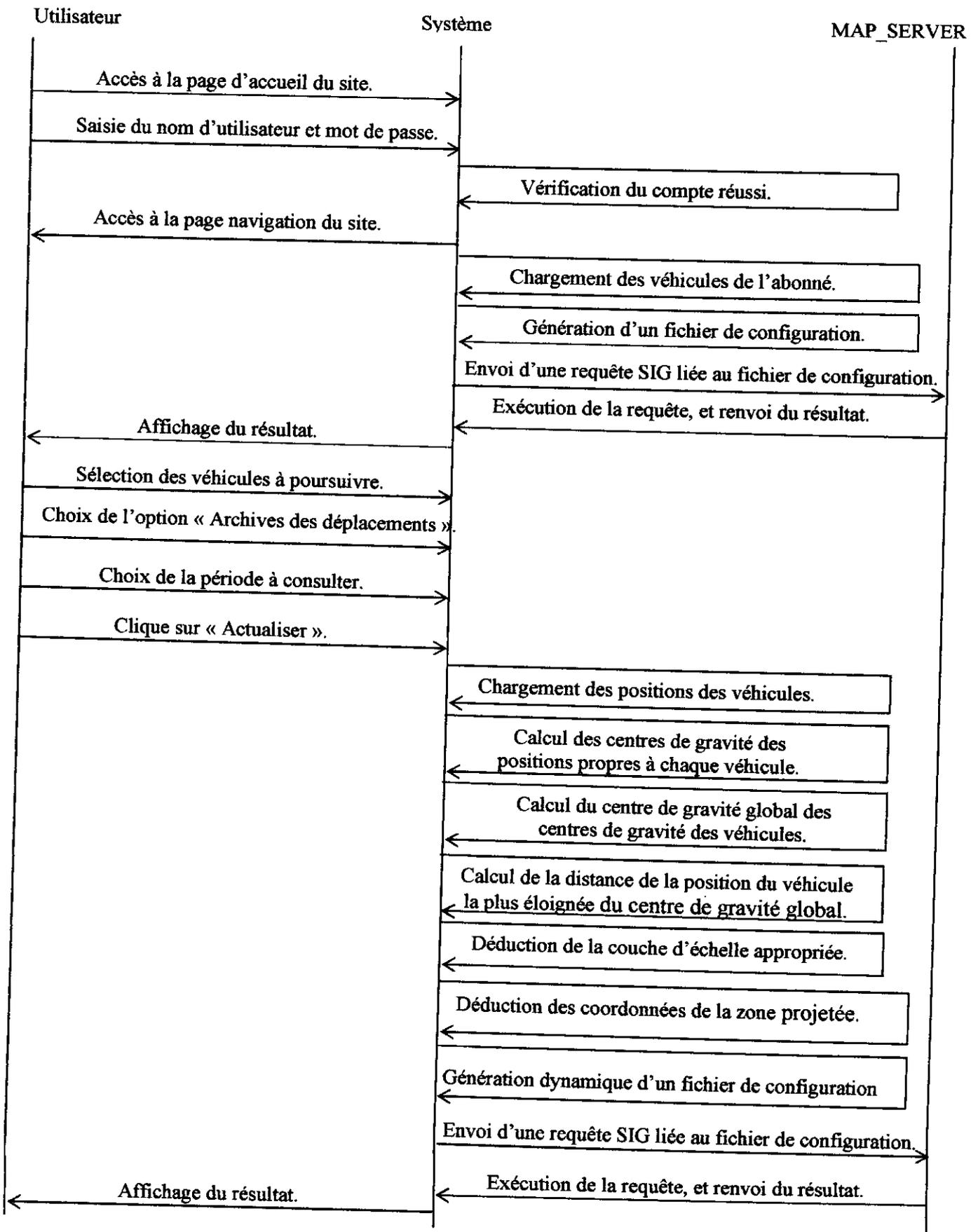


Figure II.27 : Diagramme de séquence de la partie navigation (Historique des déplacements).

VIII.4. Suite des décisions concernant l'implémentation du site:

Nous avons vu que nous devons générer un fichier d'indexation pour chaque fragment d'une même échelle.

Ceci implique que nous devons connaître avant chaque génération d'un « mapfile » quel fichier d'indexation nous voulons désigner afin de répondre aux deux objectifs suivants :

- L'échelle de carte choisie doit être la plus proche de celle de la vue souhaitée, afin de minimiser le nombre de fragments que MapServer va utiliser pour construire la vue souhaitée.
- Eviter que l'échelle de carte choisie soit en dessous de l'échelle de la vue souhaitée, car ceci engendrera un étirement de la carte.

Mais comment connaître l'échelle de la vue souhaitée ? Sachant que l'échelle présente sur les cartes est sous la forme : **1m pour x km**.

Concrètement ya aucun moyen de connaître l'échelle de la vue souhaitée sous la forme **1m pour x km**, par contre nous pouvons calculer l'équivalent d'un pixel de la vue projetée en mètres sur la projection UTM (détaillé en bas).

Ce qui signifie que nous devons connaître pour chaque échelle de carte l'équivalent d'un pixel de celle-ci en mètres sur la projection UTM, afin de comparer avec celui de la vue projetée.

Ceci répond à la fameuse question : **Que va mettre à jour l'outil de découpage concernant la base de données ?**

L'outil de découpage doit mettre à jour la base de données lors:

- De la fermeture de la fiche « Gestion d'arborescence » afin de définir les échelles de cartes existantes.
- Découpage d'une carte afin d'attribuer à l'échelle de cette carte les deux valeurs :
 - Pix_to_wld : équivalent d'un pixel de la carte en mètres sur la projection UTM.

Donc nous devons rajouter la table nommé Echelle à la base de données du site, celle-ci sera définie comme suit :

- Echelle : valeur de l'échelle de type entier.
- Pix_to_wld: équivalent d'un pixel de la carte en mètres sur la projection UTM.
- Projection : indique la projection de l'échelle UTM ou WGS84.

Comment calculer l'équivalent d'un pixel de la vue projetée en mètres sur la projection UTM :

Etant donné que l'extent des couches à projetées sont en WGS84, tandis que l'équivalent d'un pixel de la carte sur la projection UTM est en mètres, ce calcul nécessite la prise de connaissance des informations suivantes:

- Le WGS84 divise la terre en 360 degrés.
- L'unité de l'UTM est en mètres.
- La circonférence de la terre est d'environ 40000km au niveau du méridien.

Ceci implique qu'un degré en WGS84 vaut (40000/360) km au niveau du méridien.

Donc 1deg=111 km, étant donné que l'Algérie est juste au dessus du méridien, nous estimerons que 1deg=100 km=100 000 mètres.

Le rapport largeur (extent)/ largeur (zone d'affichage) définit l'équivalent d'un pixel sur la vue en degré sur le système de référence WGS84.

Donc $\text{pix_to_wld} = (\text{max_extent} / \text{Largeur_zone}) * 100\ 000$ mètre en UTM par pixel sur la vue projetée.

VIII.5. Sécurité du site:

Nous protégerons le dossier administration grâce à une « Authentication-http » de type simple, ceci en réalisant les étapes suivantes :

1. Modifier le fichier de configuration httpd_XXX.conf avec:

```
Alias /gmap/ "/ms4w/apps/gmap/htdocs/"
<Directory "/ms4w/apps/gmap/htdocs/">
AllowOverride AuthConfig
Options Indexes FollowSymLinks Multiviews
</Directory>
```

2. Créer un fichier mot de passe en:

- Ouvrant une fenêtre commande DOS Windows.
- Accède au dossier « BIN » du server Apache.
- Exécuter la commande suivante :

```
.....Apache/bin> htpasswd -c htpasswd.txt nom_utilisateur
```

3. Créer un fichier d'extension « .htaccess » dans le répertoire qu'on souhaite protégé du site, ce fichier contiendra les lignes suivantes :

```
AuthType Basic
AuthUserFile C:/ms4w/Apache/bin/htpasswd.txt
AuthName "Password Required"
require valid-user
```

4. Redémarrer Apache.



Figure II.28 : Authentication-http

Remarque :

Quelques versions de Windows comme « XP » ne supporte pas l'extension « .htaccess », il faut remplacer l'extension « .htaccess » par un autre exemple « .acl », il suffit de modifier le fichier de configuration d'Apache « httpd.conf » en ajoutant la ligne :

```
AccessFileName .htaccess .acl
```

IX. Test et validation :

IX.1. Outil de découpage :

L'interface principale de l'outil de découpage se présentera comme ceci :

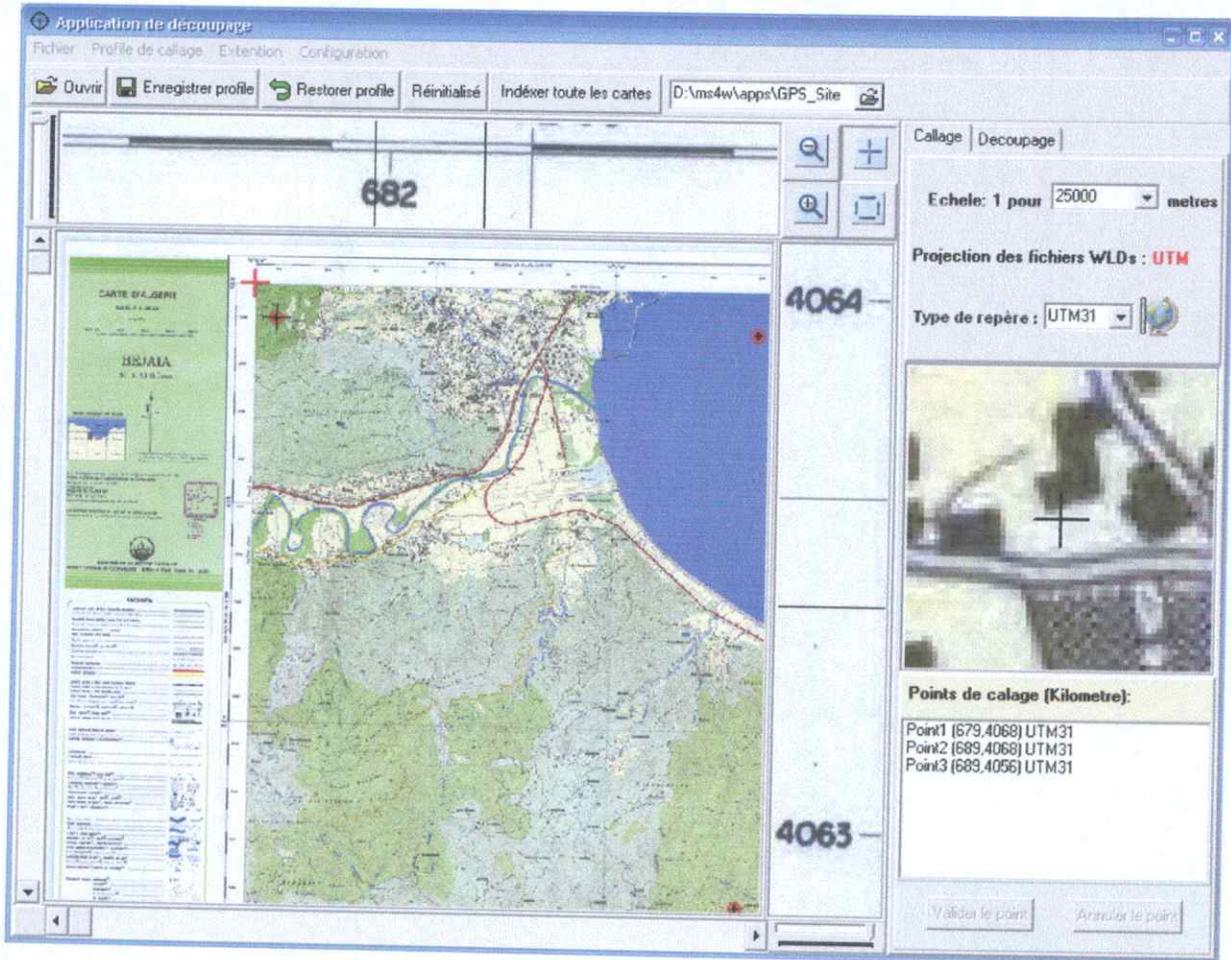


Figure II.29 : Interface principale de l'outil de découpage pendant le calage

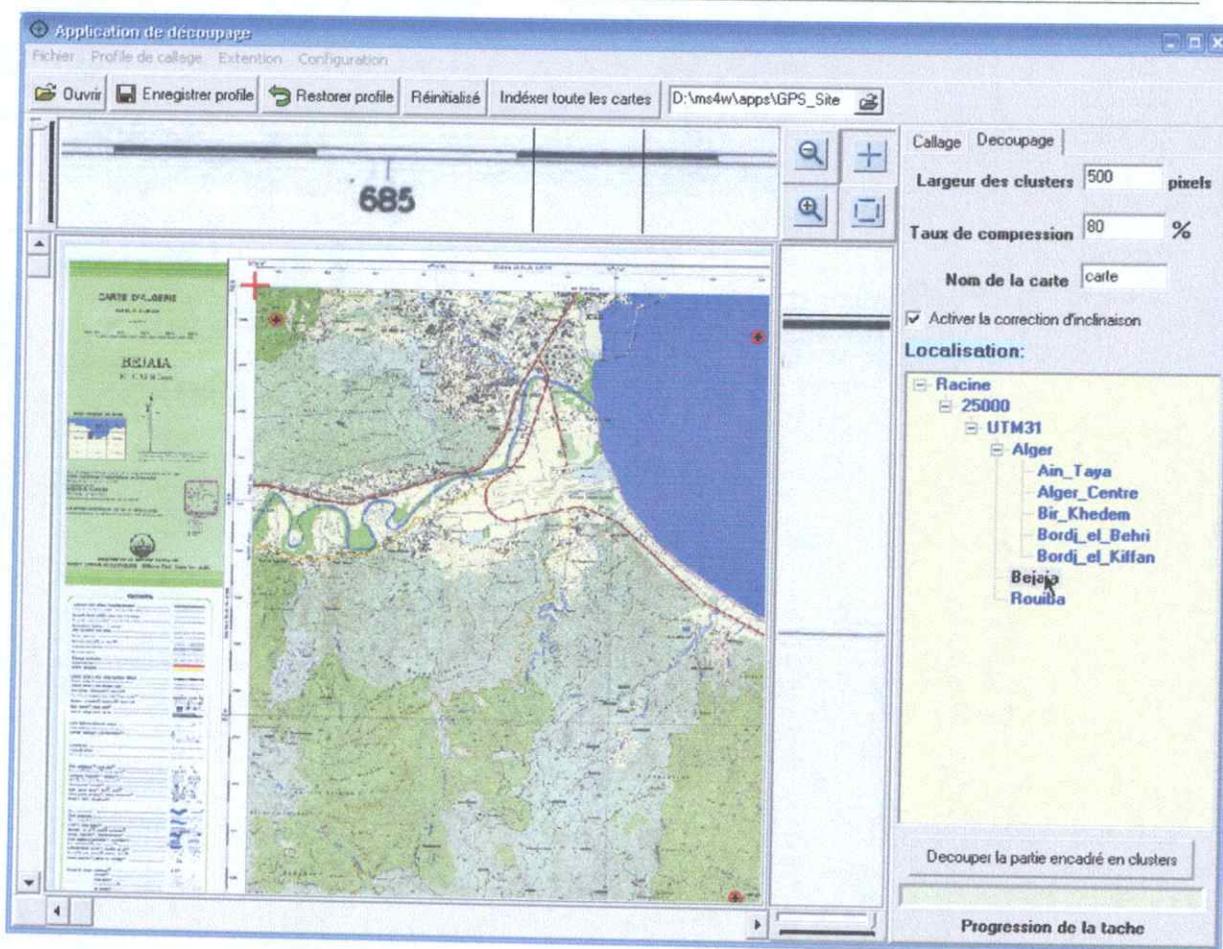


Figure II.30 : Interface principale de l'outil de découpage pendant le découpage

Le résultat final de l'outil est fort satisfaisant car il manipule des images de grandes tailles avec une souplesse impressionnante, néanmoins son exécution nécessite au minimum 256 Mégas de RAM ce qui est tout à fait acceptable.

Les tests de navigation ont démontré que les fragments des différentes cartes stockées collent bien les uns aux autres, même si ses derniers proviennent de différentes cartes et quelque soit l'UTM, ce qui valide le processus de calage mis en place.

Par souci de commodité nous avons rajouté des fonctionnalités qui ne paraissaient pas dans la définition des besoins comme le principe de sauvegarde et restauration d'un profil.

Ce dernier est un fichier qui sera associé à une carte après son calage, afin de la redécouper à plusieurs reprises à différents moments sans pour autant reprendre le calage dès le début.

Les fiches « Gestion d'arborescence » et « Configuration de la connexion vers la base de données » se présenteront comme ceci :

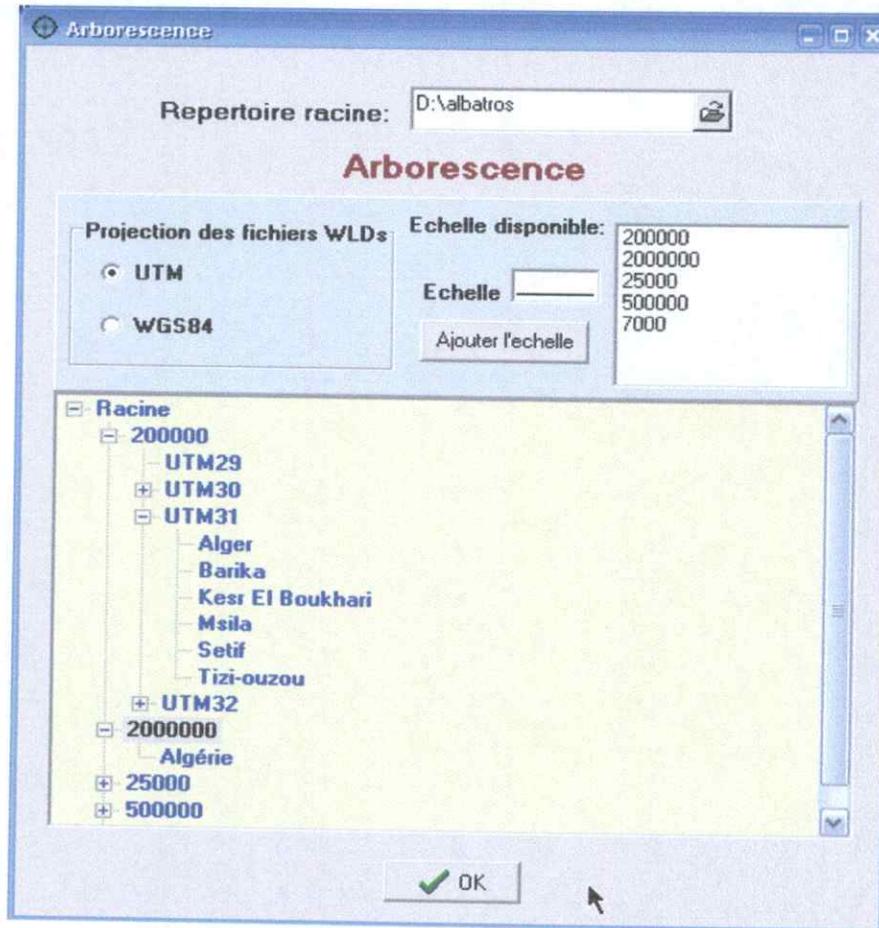


Figure II.31 : Fiche « Gestion d'arborescence »

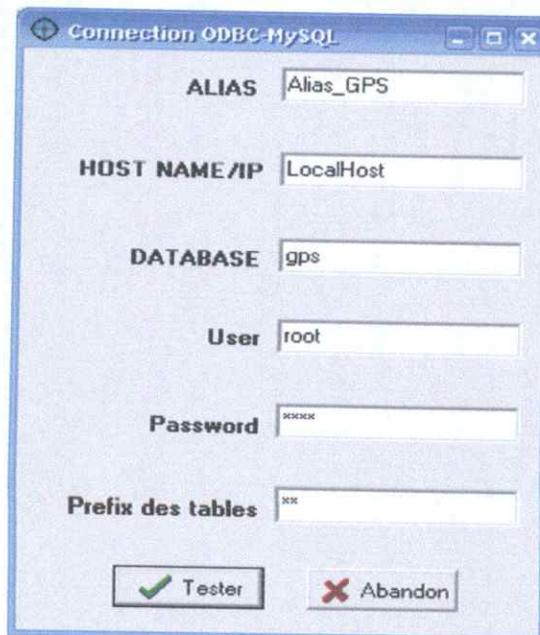


Figure II.32 : Fiche « Configuration de la connexion vers la base de données »

IX.2. Site de Navigation :

IX.2.1. Partie administration :

Nous avons réalisé une administration simple mais fonctionnelle, afin de se focaliser sur la partie navigation du site qui est le cœur de notre thèse.

IX.2.1.1. Gérer les kits GPS:

Numéro du Kit	IMEI
3	kit3
4	kit4
5	kit5
6	kit6

Figure II.33 : Page « Gérer les kits GPS »

IX.2.1.2. Gérer les clients:

Numéro du client	Type de client	Société	Particulier		Numéro téléphone	Compte	
		Intitulé	Nom	Prenom		User	Pass
4	Particulier		Moualek	nassim		na	1
6	Société	SNM				a	a

Figure II.33 : Page « Gérer les clients »

IX.2.1.3. Gérer les abonnements:

Numéro	Matricule	Marque	Nom représentatif	Kit GPS
5	1	Ferari	ferari_Muderna	3
9	2	BMW	Audi	4
10	3	Punto	Punto_Sport	5

Figure II.35 : Page « Gérer les abonnements »

IX.2.2. Partie Navigation :

The screenshot displays a navigation application interface. At the top, it shows 'Algérie' and a map of the region. The interface is divided into several panels:

- Mode de poursuite (Tracking Mode):**
 - Poursuite en temps réel (Real-time tracking)
 - Actualisé tout les (Refresh all) [dropdown]
 - Historique déplacements (History of movements)
 - Jour (Day) [dropdown], Mois (Month) [dropdown], Année (Year) [dropdown]
 - Debut (Start) 01 [dropdown] 01 [dropdown] 2007 [dropdown]
 - Nombre de jours (Number of days) 1 [dropdown]
 - Actualisé (Refresh)
- Vehicule (Vehicle):**

Matricule	Marque
5	Ferari

Vitesse	Altitude	Position prise le
80 KM	100 Metres	01-Jan-2007 à 08:00:36
- Localisation (Location):**
 - Map navigation controls (Zoom in, Zoom out, Home, etc.)
 - Scale: 2^x or 4^x [dropdown]
 - Resolution: 480 x 500 [dropdown]
- Vehicules (Vehicles):**
 - Vehicule_1
 - Vehicule_2
 - Vehicule_3
 - Actualisé (Refresh)

At the bottom of the map, there is a scale bar from 0 to 10 km and a legend for GPRS connection status:

- Connection GPRS presente
- Connection GPRS absente

Figure II.36 : Page navigation

Il suffit que l'utilisateur clique sur une position d'un véhicule pour lui afficher les informations concernant celui-ci au moment de la prise de la position.

Les tests du site aboutissent à:

- Une satisfaction concernant la précision des positions des véhicules, les erreurs sont de quelques pixels donc fort acceptable.
- Un phénomène lié aux zones projetés, en effet quand on projette une vue d'une échelle proche d'une des échelles de cartes disponibles l'affichage est instantané, tandis que plus l'écart des échelles est important plus le temps de réponse s'accroît, car MapServer fera appel à un nombre plus important de fragments afin de construire la vue souhaitée.

Conclusion

La réalisation d'un site web de navigation par satellite s'est révélé être un processus complexe et multidisciplinaire nécessitant des connaissances à la fois :

- En traitement numérique des cartes topographiques.
- En géo référencement (pour aboutir aux équations de calage).
- En conception de site web dynamique orienté SIG (web mapping).

Nous avons démontré au cours de ce travail la nécessité de découper la partie utile à la navigation dans chaque carte scannée en petits fragments afin d'optimiser la construction et le chargement des images générés en temps réel par MapServer en réponse aux demandes de l'interface utilisateur représenté par le site web de navigation. Un fond de carte basé sur ces petits fragments a donc été constitué.

Ce fond de carte est organisé suivant deux dimensions :

- 1- Les échelles : Chaque carte est éditée à une certaine échelle (1/7000, 1/25000, 1/200000...) qui représente le niveau de zoom idéal (ou préférentiel) pour sa lisibilité. En tenant compte de ce facteur, le site web résultant choisi suivant le niveau de zoom demandé par l'utilisateur les fragments de cartes les plus lisibles. Par conséquent, les échelles constituent une première répartition du fond de carte.
- 2- Les systèmes de projections : Chaque carte est aussi éditée suivant un certain système de projection qui est en général l'Universel Transverse Mercator pour l'Algérie, ce qui nous amène à répartir notre fond de cartes aussi suivant les 4 systèmes de projections successifs utm29, utm30, utm31, utm32 qui couvrent ensemble la totalité du territoire algérien. Notre système est ainsi capable de localiser un véhicule n'importe où sur ce territoire.

Le site web réalisé au cours de ce travail offre deux fonctionnalités majeures (en plus de quelques fonctions d'administration des abonnements) :

- 1- Le suivi en temps réel : Dans notre système il est possible de suivre un ou plusieurs véhicules en mouvements à la fois. Lorsqu'un seul véhicule est suivi, il est toujours affiché au centre de la page web générée, donnant ainsi l'illusion d'un mouvement sans discontinuités à l'utilisateur au fur et à mesure que les positions se succèdent. Lorsque plus d'un véhicule est suivi à la fois la démarche précédente ne convient pas et ceci d'une part parce que chaque véhicule suit sa propre trajectoire (les trajectoires peuvent même aller dans des sens opposés) et d'autre part parce que chaque véhicule suivi peut être soit à quelques mètres soit à des milliers de kilomètres des autres véhicules. Cela nous a donc amené à imaginer un centre de gravité virtuel qui est calculé à partir des positions des différents véhicules et qui sert de point de repère visuel au centre de la page web générée. La surface géographique retournée dans ce cas à l'utilisateur correspond (en rayon) à la distance entre ce centre de gravité imaginaire et le véhicule qui en est le plus éloigné. Cette seconde démarche garantit en toutes circonstances que la totalité des véhicules suivis sont visibles sur l'écran de l'utilisateur.
- 2- La consultation d'un historique de navigation : L'utilisateur peut à travers l'interface du site web demander l'historique des positions des véhicules depuis une date de début vers une date de fin. Ces positions sont lues à partir de la base de données du site web.

Nous avons constaté l'importance du calage qui nous a permis la mise en conformité géométrique des cartes avec la réalité.

Du point de vue résultats obtenus, les tests de navigation réalisés démontrent que les fragments des différentes cartes stockées collent bien les uns aux autres même quand on passe d'une carte à une autre adjacente ce qui valide le processus de calage mis en place.

On constate que le temps de réponse du site web dépend fortement des échelles de cartes disponibles et de manière générale plus l'échelle (zoom) demandée par l'utilisateur est proche de l'échelle disponible en retour plus le temps de réponse diminue (il peut être quasiment instantané), et inversement, plus l'échelle requise s'éloigne de l'échelle disponible plus le temps de réponse augmente. Ceci s'explique par le fait que MapServer doit lire plus de fragments lorsque l'échelle désirée s'éloigne de l'échelle disponible. Bien évidemment, notre système choisit toujours l'échelle disponible la plus proche de l'échelle désirée.

En conclusion donc, plus il y'aura d'échelles stockées plus la réponse du site sera meilleur mais à condition que la répartition de ces échelles stockées soit uniforme sur une large plage (en pratique on doit avoir des échelles entre 1/5000 et 1/5000000).

Les perspectives que nous envisageons sont les suivantes :

- Introduire une fonctionnalité partielle dans l'outil de découpage afin d'optimiser la reconstitution des fichiers d'indexations.
- Etendre la partie administration du site web en représentant plus de détails (Exemple : Ajout de niveau d'habilité, Ajout de groupe de véhicule...).
- Offrir à l'administrateur la possibilité de paramétrer des zooms prédéfinis avec les meilleurs temps de réponse.

ANNEXE

Fichier de configuration mapfile :

Le mapfile détermine les données à utiliser dans une application et à afficher ainsi que les paramètres de requête; il inclut aussi les informations sur la manière de dessiner la carte, la légende, et les cartes qui résultent d'une requête.

Les mapfiles ont une extension **.map**

Généralités sur les mapfiles :

- ✓ Chaque section commence par un identifiant de type, et se termine par le mot-clé "END", pour former ce que l'on appelle un "bloc".
- ✓ Les mapfiles ne sont pas sensibles à la casse (minuscules / majuscules), sauf pour les noms des champs attributaires, notés entre [crochets].
- ✓ Les chemins de fichiers peuvent être indiqués de manière absolue (à partir de la racine du système), ou de manière relative à l'emplacement du mapfile. Mais attention, les chemins vers les données sont toujours relatifs au paramètre SHAPEPATH.
- ✓ On peut ajouter des commentaires dans un mapfile en commençant la ligne par le caractère #.

Un mapfile possède une structure en sections et une syntaxe spécifique, qui décrit les caractéristiques du document cartographique à générer par **MapServer** :

A. Le bloc PROJECTION :

L'indication d'un géo référencement pour la carte produite par le mapfile nécessite un bloc PROJECTION.

Ce bloc contient les informations de référencement, qui peuvent être de deux formes, une série de paramètres PROJ.4 (Cf. http://proj.maptools.org/gen_parms.html) ou un code EPSG (Cf : <http://www.epsg.org/> ou <http://www.inovagis.org/giserver/epsg.asp>).

Par exemple :

```
PROJECTION
    "proj=utm"
    "ellps=GRS80"
```

```
"zone=15"  
"north"  
"no_defs"  
END Où :  
PROJECTION  
"init=epsg:28992"  
END
```

Notes importantes :

- Si toutes les couches (layers) de votre mapfile sont dans le même système de projection, vous n'avez pas à ajouter de bloc PROJECTION.
- Si vous mentionnez un bloc PROJECTION au niveau racine (donc dans le bloc principal MAP), et seulement un seul bloc PROJECTION dans un bloc LAYER, alors **MapServer** va présumer que toutes les autres couches (layers) sont dans la projection indiquée au niveau du bloc MAP. Ainsi, si vos couches (layers) sont dans de multiples systèmes de projection, il est important de le préciser par des blocs PROJECTION pour chaque couche, chaque bloc LAYER.

B. L'en-tête :

Un Mapfile possède une organisation hiérarchique, basée sur des blocs (ou objets, par référence aux objets de MapScript), à l'intérieur d'un bloc principal, le bloc MAP.

Chaque bloc commence par son nom et se termine par le mot-clé "END".

Au début du mapfile, juste après le mot-clé "MAP" et avant le premier sous-bloc, on trouve un en-tête avec quelques paramètres généraux.

NAME : nom de la carte, préfixe ajouté à tous les fichiers image générés par le mapfile, donc à garder de courte taille.

UNITS : Unité de la carte, utilisée pour les calculs et le dessin de la barre d'échelle.

L'information sur l'unité du fond de carte n'est pas toujours apportée par le bloc « projection », il faut donc essayer de la fournir explicitement à chaque fois.

Valeurs: [feet|inches|kilometers|meters|miles|dd]

EXTENT : Extension de la carte, donc coordonnées des extrémités de la carte dessinée dans l'image générée par le mapfile, exprimées dans l'unité de la projection générale du mapfile.

Valeurs : [xmin] [ymin] [xmax] [ymax]

STATUS : On ou Off, active ou désactive la totalité du mapfile.

FONTSET et **SYMBOLSET** : chemin et nom des fichiers contenant les définitions de polices de caractères (utile sous linux) et de symboles (on peut aussi définir un symbole directement dans le mapfile, mais c'est utile de pouvoir les rassembler dans un seul fichier qui pourra être appelé par différents mapfiles).

SHAPEPATH : Chemin vers le répertoire contenant les fichiers de données (des couches), ou le chemin de base à partir duquel les paramètres DATA iront les chercher.

Ensuite peuvent intervenir les paramètres de génération de l'image :

IMAGETYPE : type de fichier image à générer. La valeur doit correspondre à l'une de celles incluses par défaut par MapServer, ou à une définition explicite par ajout d'un bloc OUTPUTFORMAT. Ce paramètre dépend des options avec lesquelles à été compilé MapServer.

Valeurs : [gif|png|jpeg|wbmp|gtiff|swf|userdefined]

SIZE : Dimensions (largeur et hauteur) de l'image à générer, en pixels, séparées par un espace.

RESOLUTION : Résolution de l'image en dpi, par défaut 72. N'affecte que les calculs d'échelle, cf. le bloc SCALEBAR.

C. Le bloc WEB :

Ce bloc va paramétrer le fonctionnement de l'interface Web autour de la carte générée par MapServer, c'est à dire paramétrer les chemins pour l'image, l'utilisation éventuelle d'un modèle (template) HTML...

IMAGEPATH : Chemin absolu vers le répertoire temporaire où sera stocké le fichier image généré par le mapfile. Doit se terminer par un slash « / », de préférence utiliser des slashes, pas des anti-slashes « \ ».

IMAGEURL : Base de l'url (partie de l'adresse web s'ajoutant au nom du serveur dans l'url) pointant sur le répertoire temporaire contenant les fichiers image générés. Correspond à l'alias défini dans la configuration du serveur Web.

EMPTY : Url vers laquelle l'utilisateur est redirigé lorsqu'une requête MapServer ne retourne pas de résultats.

ERROR : Url vers laquelle l'utilisateur est redirigé lorsque se produit une erreur.

LOG : Chemin et nom du fichier texte dans lequel sera inscrite l'activité de MapServer sur ce mapfile. Contient une trace des opérations effectuées et des éventuelles erreurs survenues. Fonctionne en parallèle avec le paramètre DEBUG et les paramètres LOG des blocs LAYER du mapfile.

MINSCALE : Échelle minimale à laquelle le mapfile sera dessiné. Si une échelle plus petite est demandée, MapServer dessinera la carte à l'échelle précisée par ce paramètre.

MAXSCALE : Idem pour l'échelle maximale.

TEMPLATE : Chemin et nom du fichier modèle HTML éventuel (cf. la question dédiée à ce sujet).

HEADER : Modèle HTML à utiliser avant l'insertion de l'image de la carte.

FOOTER : Idem pour le bas de page.

METADATA : Bloc secondaire utilisé pour stocker des paires noms – valeur. Utilisé par les modèles HTML (pour stocker des variables généralisées) et en mode client et serveur WMS/WFS.

D. Le bloc REFERENCE :

Ce bloc définit les paramètres de la petite carte utilisée comme référence pour la carte principale. C'est une image sur laquelle va être dessiné un rectangle représentant l'extension de la carte principale, ou la localisation des résultats d'une requête, en mode QUERY.

MapServer va faire la relation entre la carte principale et la carte de référence grâce aux paramètres **EXTENT** des deux cartes, et afficher un rectangle (ou un symbole) permettant de localiser et de mesurer l'extension de la carte principale sur la carte de référence. Par défaut le bloc **REFERENCE** a un paramètre **STATUS** à **OFF**, pour activer cette carte il faut donc penser à avoir une ligne « **STATUS ON** ».

Paradoxalement, **MapServer** a besoin d'une image fixe représentant la petite carte de référence, il ne va pas la générer. Cela permet par contre l'utilisation d'une image externe. Il faut donc alors, pour produire cette image, utiliser un mapfile simplifié (uniquement les contours de la couche principale par exemple), réglé pour générer une image de petites dimensions. On peut naturellement créer un mapfile spécifique pour générer cette image, puis l'utiliser comme référence dans le mapfile principal.

COLOR [r] [g] [b] Couleur de fond du rectangle de référence tracé sur la carte. Une valeur à -1 indique que le rectangle doit être évidé. La couleur par défaut est le rouge.

EXTENT [minx] [miny] [maxx] [maxy] Etendue spatiale de la carte de référence.

IMAGE [fichier] Nom de fichier complet de l'image de la carte de référence. Doit être une image de type GIF.

MARKER [integer|string] Numéro ou nom du SYMBOL (dans le fichier SYMBOLSET du mapfile) utilisé lorsque le rectangle de référence devient trop petit (voir les valeurs MINBOXSIZE et MAXBOXSIZE ci-dessous). Un croisillon par défaut.

MARKERSIZE [integer] Taille du SYMBOL utilisé en remplacement du rectangle.

MINBOXSIZE [integer] Si le rectangle de référence est plus petit (largeur ou hauteur en pixels) que cette valeur, alors c'est le SYMBOL défini dans MARKER et MARKERSIZE qui est utilisé (ou le croisillon par défaut).

MAXBOXSIZE [integer] Si le rectangle de référence est plus grand (largeur ou hauteur en pixels) que cette valeur, alors il n'est pas dessiné.

OUTLINECOLOR [r] [g] [b] Couleur du contour du rectangle de référence. Une valeur à -1 indique qu'aucun contour n'est dessiné.

SIZE [x][y] Taille en pixels de la carte (image) de référence.

STATUS [on|off] Indique si la carte de référence doit être utilisée. *Attention*, ce paramètre est OFF par défaut.

E. Le bloc LEGEND :

MapServer peut générer trois types de légendes :

- légendes simples sous forme d'images ;
- légendes basées sur un modèle (template) de légende HTML ;
- légendes en HTML pur.

Les légendes simples sont des images, incluses ou pas dans l'image de la carte principale, reprenant chaque classe *nommée* des layers du mapfile et son figuré. Il faut donc penser à nommer toutes les classes que l'on veut voir apparaître en légende (paramètre NAME).

Lorsque la légende simple est incluse dans l'image de la carte (paramètre : STATUS EMBED), on peut préciser l'endroit où la légende sera dessinée avec le paramètre POSITION.

Ce paramètre prend une valeur correspondant à un code composé de deux lettres, la première pour le haut / bas, u pour « upper », l pour « lower », la seconde pour gauche / droite, l pour « left », r pour « right », et c pour « center » :

ul	uc	ur
.	.	.
ll	lc	lr

Les caissons (rectangles colorés) de légende pour les couches de polygones sont réglables en taille avec le paramètre KEYSIZE (valeurs : largeur hauteur) et leur espacement avec le paramètre KEYSPPACING (valeurs : écart horizontal espace écart vertical). Il n'est pas possible de titrer une légende avec le seul mapfile (la générer sous un titre texte dans la page html qui contient la carte ou utiliser un modèle html).

IMAGECOLOR [r] [g] [b] Couleur de fond de la légende.

LABEL Il est possible d'ajouter un bloc LABEL dans le bloc LEGEND pour en paramétrer les libellés.

OUTLINECOLOR [r] [g] [b] Couleur de contour des caissons.

POSITION [ul|uc|ur|ll|lc|lr] Position de la légende dans l'image de la carte, cf. ci-dessus.

KEYSIZE [x][y] Taille en pixels des caissons de légende, 20x10 pixels par défaut.

KEYSPACING [x][y] Ecart entre les caissons (Y) et les libellés (X). 5 pixels partout par défaut.

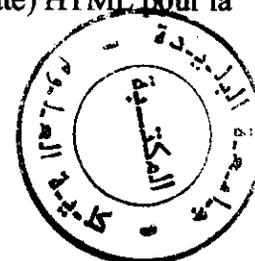
POSTLABELCACHE [true|false] Demande à MapServer de dessiner la légende après que tous les libellés en cache soient dessinés. Utile pour

ajouter des filets et autres éléments de décoration. A
False par défaut.

STATUS [on|off|embed] Permet d'activer ou non la légende. "embed" permet d'intégrer l'image de la légende dans l'image de la carte principale, plutôt que de générer un autre fichier image.

TEMPLATE [filename] Permet de préciser un fichier modèle (template) HTML pour la légende.

F. Le bloc SCALEBAR :



MapServer gère les échelles selon une technique assez particulière. En effet, il part du principe que la carte sera au final une image possédant des dimensions en pixels, et qui sera visualisée au moyen d'un écran qui possède une certaine résolution (72 ou 90 dpi, selon le système d'exploitation). Par ailleurs, la carte doit être dessinée dans un rectangle d'extension maximale donné par le paramètre EXTENT. L'échelle finale de l'image doit donc être définie selon ces paramètres. Le paramètre EXTENT prime sur le paramètre SCALE, car c'est lui qui définit plus précisément ce que doit contenir la carte à dessiner.

Pour dessiner une échelle indiquant une certaine longueur terrain, il faut donc déterminer combien de pixels cette longueur va représenter, et l'indiquer dans le paramètre SIZE du bloc.

Le bloc SCALEBAR permet de dessiner des barres d'échelle, dans l'image de la carte ou comme une image distincte. Il possède les paramètres suivants :

POSITION : Code à deux lettres définissant l'endroit où sera dessinée l'échelle, ce code est le même que celui utilisé par le paramètre POSITION du bloc LEGEND.

SIZE : Dimensions en pixels (largeur espace hauteur) du rectangle contenant la barre d'échelle. Important car détermine la longueur totale de la barre, et donc ses indications chiffrées.

INTERVALS : Nombre de subdivisions à afficher.

STATUS : Inclusion (EMBED), image à part (ON) ou désactivation (OFF).

STYLE : Apparence de la barre, 0 donnant une barre de rectangles pleins, 1 une barre fine munie de repères (barbules).

UNITS : Unités pour le calcul de la longueur des intervalles de la barre d'échelle.

Toutes unités possibles : [feet|inches|kilometers|meters|miles] sauf degrés décimaux.

IMAGECOLOR : Couleur RGB du rectangle qui contient l'échelle.

BACKGROUNDCOLOR : Couleur RGB de la barre d'échelle et de ses libellés.

COLOR : Couleur alternative à BACKGROUNDCOLOR si barre de type 0 et que plusieurs intervalles sont spécifiés.

OUTLINECOLOR : Couleur RGB de la réserve autour de la barre d'échelle (mais pas autour des libellés).

TRANSPARENT : Valeur booléenne (ON / OFF) qui précise si le rectangle contenant l'échelle est transparent.

G. Le bloc LAYER :

Ce bloc va définir les propriétés de création et d'affichage d'une couche de données SIG par MapServer. Les blocs LAYER sont dessinés dans l'ordre du mapfile, c'est à dire que le premier bloc du mapfile est dessiné en premier, les suivants viendront par-dessus sur la carte. L'ordre du mapfile est donc l'ordre inverse de la superposition verticale des sources.

- **Paramètres généraux :**

NAME : Nom de la couche, utilisé comme identifiant par l'interface web. Doit être unique dans le mapfile et d'une longueur maximale de 20 caractères.

- GROUP** : Groupe auquel le LAYER appartient. Utilisé dans les modèles HTML pour activer/désactiver les couches par groupes.
- METADATA** : Bloc secondaire utilisé pour stocker des paires nom – valeur. Utilisé par les modèles HTML et en mode serveur WMS.
- STATUS** : Statut (visibilité) du layer. Valeurs : default, on, off. Doit prendre la valeur « default » pour que le layer soit visible lorsque l'on utilise MapServer en mode Map (le STATUS ON ne suffit pas à le rendre visible, il faut que le layer soit expressément requis).
- TYPE** : Type d'objet géométrique ou modalité selon laquelle la couche doit être dessinée. Valeurs : point|line|polygon|circle|annotation|raster|query. Ce paramètre peut prendre une valeur différente du type géométrique des objets contenus dans la couche d'origine, par exemple une couche de polygones peut être représentée comme un LAYER de type POINT, ce qui affichera les centroïdes des polygones (mais pas l'inverse, naturellement).
- MINSCALE** : Échelle minimale à laquelle la couche sera dessinée. Si une échelle plus petite est demandée, MapServer dessinera la couche à l'échelle précisée par ce paramètre.
- MAXSCALE** : Idem pour l'échelle maximale.
- SYMBOLSCALE** : Échelle à laquelle les symboles et/ou les textes apparaissent à leur taille normale. Ce paramètre permet un dimensionnement dynamique de ce type d'objets selon l'échelle de la carte, dans les limites des deux paramètres précédents. Obligatoire pour l'utilisation du paramètre SIZEITEM dans un bloc CLASS.
- TRANSPARENCY** : Degré de transparence de la couche, exprimé en pourcentage (sans le signe %), de 100 – opaque à 0 – totalement transparent.
- OFFSITE** : Le numéro d'index de la couleur d'une couche raster à traiter comme transparent. Cela permet de ne garder que la région utile d'une couche raster.
- CLASSITEM** : Nom de la colonne attributaire qui est utilisée dans les expressions de sélection des blocs CLASS.
- LABELITEM** : Nom de la colonne attributaire qui fournira le texte des étiquettes.
- TEMPLATE** : Nom du fichier modèle HTML qui prend en compte cette couche. Obligatoire pour rendre cette couche interrogeable par requête, même si on n'utilise pas de modèle HTML.

- **Paramètres de données**

Voir les pages dédiées à l'utilisation des données vecteur, raster et distantes, cf. la question sur les types de données utilisables.

Le bloc LAYER peut contenir des blocs de hiérarchie inférieure (cf. les questions dédiées) :

- bloc CLASS
- bloc LABEL

H. Le bloc CLASS :

Ce bloc, placé dans un bloc LAYER, permet de définir des classes thématiques dans la couche, classes qui vont pouvoir être affichées différemment sur la carte globale. Les blocs CLASS sont traités dans l'ordre du fichier map, donc les premières classes sont les premières dessinées.

Ce bloc peut contenir les paramètres suivants :

NAME : Nom de la classe, à préciser si l'on veut trouver cette classe dans la légende.

EXPRESSION : Critère de sélection des objets de la couche qui vont être inclus dans la classe en cours. Ces sélections peuvent utiliser quatre méthodes : comparaison de chaînes de caractères (en utilisant CLASSITEM), comparaisons logiques simples (avec opérateurs), expressions régulières, fonction de chaîne length().

COLOR : Couleur de fond des objets possédant une surface, exprimée en RGB, trois valeurs entières séparées par des espaces.

OUTLINECOLOR : Couleur de contour des objets, en RGB aussi.

SYMBOL Numéro ou nom du symbole, qui doit être défini dans le mapfile par un bloc SYMBOL ou dans un fichier SYMBOLSET lié au mapfile.

SIZE : Taille du symbole ou de la trame, uniquement utilisable avec les symboles redimensionnables.

MINSIZE et **MAXSIZE** : tailles mini et maxi (en pixels) de dessin des symboles. En dehors de cette fourchette les symboles sont dessinés à la valeur la plus proche.

SYMBOLESCALE : Échelle à laquelle les symboles et/ou les textes apparaissent à leur taille normale. Obligatoire dans le cas de symboles proportionnels avec le paramètre **SIZEITEM**.

TEXT : Nom de la colonne attributaire contenant le texte à utiliser pour l'étiquetage des objets de la classe (On peut aussi créer des expressions combinant plusieurs attributs entre crochets).

TEMPLATE : Chemin et nom du fichier modèle HTML éventuellement utilisé.

DEBUG : Valeur On ou Off. Si le paramètre **LOGFILE** est défini, les messages de débogage détaillés seront ajoutés au fichier de log, en plus des messages d'erreur

Chaque classe définie par ce bloc peut prendre un aspect différent à l'aide de sous-blocs **STYLE**.

I. Le bloc STYLE :

Le bloc **STYLE** permet de préciser les paramètres d'aspect de dessin d'une classe, on le retrouve donc à l'intérieur des blocs **CLASS**. Il peut contenir les paramètres suivants :

ANGLE : L'angle, en degrés, de dessin des lignes des objets **SYMBOL**, à zéro par défaut.

ANGLEITEM : Champ attributaire contenant les angles à utiliser.

ANTIALIAS : Permet de dessiner les textes (polices TrueType uniquement) et les symboles avec un anticrénelage. Attention, lorsque trop de couches sont dessinées avec ce paramètre, la génération de la carte peut échouer.

Valeurs : [true|false]

BACKGROUNDCOLOR : Couleur d'arrière-plan pour les symboles non transparents.

COLOR : Couleur de la classe ou du symbole, en RGB.

MINSIZE et **MAXSIZE** : Tailles mini et maxi de dessin des symboles, en pixels (par défaut 0 et 50, respectivement).

MINWIDTH : Largeur minimale de dessin des lignes, en pixels.

OFFSET [x] [y] : Valeurs de décalage pour le dessin des ombres, des symboles creux.

OUTLINECOLOR : Couleur de contour des polygones et de certains symboles, en RGB.

SIZE : Hauteur en pixels des symboles, ou écart entre les éléments d'une hachure.

SIZEITEM : Champ attributaire contenant les tailles à utiliser.

SYMBOL : Le nom ou l'index symbolset de l'objet symbol à utiliser. Cet objet doit être défini par un bloc SYMBOL dans le MapFile, ou dans un fichier SYMBOLSET lié, indiqué dans l'en-tête du bloc principal MAP.

WIDTH : Largeur de dessin des lignes (et des lignes composant les hachures), en pixels.

J. Le bloc STYLE :

Ce bloc permet de configurer l'étiquetage des éléments de la classe. Il possède lui aussi toute une série de paramètres.

1. Paramètres de base :

TYPE : Type de police de caractères à utiliser :

- Avec la valeur « bitmap », MapServer utilise ses propres polices internes, mais elles ne peuvent subir de rotation ni être dimensionnées précisément (cf. ci-après pour la taille de police).
- Avec la valeur « truetype », MapServer utilise une police vectorielle, au format truetype. Le paramètre FONT indique l'alias utilisé pour indiquer le fichier ttf dans le fichier FONTSET. Par exemple, pour utiliser la police truetype Arial, il faut un fichier déclaré par le paramètre FONTSET (du bloc MAP), et ce fichier (texte) comprendra une ligne « arial chemin/arial.ttf » indiquant l'alias (nom utilisé dans le bloc LABEL) de la police et l'endroit où trouver le fichier correspondant sur la machine.

COLOR : Couleur du texte de l'étiquette, en RGB.

SIZE : Taille du texte, valeur entière pour la taille en pixels des polices TrueType, ou valeur texte pour les polices bitmap de MapServer, parmi [tiny|small|medium|large|giant].

MINSIZE et **MAXSIZE** : Tailles mini est maxi de dessin des étiquettes, en pixels.

MINFEATURESIZE : Taille minimale (valeur entière en pixels) d'un objet de la couche pour qu'il soit étiqueté. Correspond à la longueur pour les objets ligne, à la surface du rectangle d'encombrement pour les objets

polygone. La valeur « auto » indique à MapServer de n'étiqueter que les objets au moins aussi gros que leur étiquette.

2. Paramètres d'effets d'affichage du texte (la présence du paramètre active la fonction) :

ANTIALIAS : Anticrénelage, réduction de « l'effet d'escalier ». Alourdit l'image produite car utilise des dégradés de couleur. Peut bloquer la génération de la carte dans le cas d'une complexité trop grande des éléments à dessiner.

OUTLINECOLOR : Couleur de la réserve autour du texte, en RGB.

SHADOWCOLOR et **SHADOWSIZE** : Couleur (RGB) et décalage (en pixels) pour le dessin des ombres sous les étiquettes.

BACKGROUNDCOLOR : Couleur du rectangle qui va contenir l'étiquette.

3. Paramètres de positionnement :

POSITION : Valeur texte correspondant à la position de l'étiquette par rapport au centre de l'objet qu'elle renseigne, selon le schéma suivant :

ul	uc	ur
cl	cc	cr
ll	lc	lr

Pour la valeur « auto » MapServer va tester les 8 positions externes pour choisir celle qui interfère le moins avec les autres étiquettes de la classe.

ANGLE : Angle en degrés par rapport à la verticale, ou « auto » pour les objets ligne, dans ce cas MapServer alignera l'étiquette sur l'objet.

OFFSET : Valeur X Y de décalage entre le coin bas-droite de l'étiquette et le centre de l'objet renseigné.

MINDISTANCE : Distance minimale (valeur entière en pixels) entre deux étiquettes du même objet.

BUFFER : Zone tampon (en pixels) autour des étiquettes, pour éviter qu'elles se touchent.

FORCE : Force le dessin des étiquettes de la classe, sans tenir compte des contraintes de proximité. Valeur : true/false (à false par défaut).

PARTIALS : Valeur true/false qui détermine si MapServer peut dessiner des étiquettes incomplètes (coupées par les bords de l'image). Vaut false par défaut.

WRAP : Précise le caractère (entre guillemets) utilisé dans le texte des étiquettes pour indiquer un passage à la ligne. **[D, PIN 2006]**

Références

- [P.BOR, 2002] : Patricia Bordin; SIG concept, outils et données, Paris, Hermès science publication, 2002.
- [A.CAZ, 1994] : Anny CAZENAVE et Kurt FEIGL, " Formes et mouvements de la Terre, satellites et géodésie" 1994.
- [CSCI, 1990] : Comité scientifique du colloque intégration de la photogrammétrie et de la télédétection dans les SIG SFPT, Strasbourg 1990.
- [CFCI, 1988] : Comité Fédéral de Coordination Inter-agences pour la Cartographie Numérique, 1988, USA.
- [T, CON 2006] : Thomas Condom (Institut EGID – Bordeaux III).
- [CIGS, 2001] : Coordination de l'information géographique et des systèmes d'information géographique, Suisse 2001.
- [P, SIL 2000] : Patrick Sillard, 2000, Les Projections et référentiel cartographiques, école des sciences cartographiques France.
- [SEIG 2007] : www.seig.ensg.ign.fr (Serveur éducatif de l'IGN et de l'Education sur l'information géographique France).
- [J.GIL 2000] : J-M Gilliot, Introduction aux systèmes d'informations géographiques, Institut Agronomique Paris-Grignon France 2000.
- [IRD] : Institut de Recherche pour le développement, L'IRD est un établissement public à caractère scientifique et technologique, placé sous la tutelle des ministres chargés de la Recherche et de la Coopération.

- [V, GOD 2006] : www.ipt.univ-paris8.fr, Vincent GODARD, 2006, Département de Géographie, Université de Paris 8.
- [L, JEG 2006] : Laurent Jégou, Atelier Cartographie novembre 2006 Département de Géographie / UTM 2006.
- [D, PIN 2006] : Didrik Pinte, <http://mapserver.gis.umn.edu/> serveur de documentation officiel de MapServer, Université du Minnesota 2006.
- [S, BAK 1999] : Stig Sæther Bakken Copyright © 1997, 1998, 1999 au groupe de documentation PHP.
- [MULL, 1997] : Muller ; Modélisation objet avec UML, Eyrolles 1997.
- [P, ROQ 2003] : Pascal Roques ; Modéliser un site e-commerce objet avec UML ; Les cahiers du programmeur 2003.
- [Mapp, 2007] : www.mappemonde.mgm.fr site de revues SIG ; 2007.
- [R, SAN 1996] : R. Santerre et J.-M. Bilodeau, Manuel de foresterie (1996).