



**UNIVERSITE D'ANTANANARIVO**  
**ECOLE SUPERIEURE POLYTECHNIQUE**



**DÉPARTEMENT INFORMATION GEOGRAPHIQUE ET FONCIERE**



Mémoire de fin d'études en vue de l'obtention du diplôme d'ingénieur Géomètre-Topographe



**RECHERCHE D'UNE SOLUTION  
APPLICATIVE OPEN SOURCE  
POUR LA REALISATION D'UN  
PROJET WEBMAPPING**

*Présenté le 25 janvier 2011 par :* **ANDRIMPIRENENA Zolalaina Herinarivo Joseph**

*Président du Jury:* ■ **Monsieur RABETSIAHINY**, Maître de conférences Chef du département Information Géographique et Foncière, ESPA

*Examineurs :* ■ **Madame RABEHERIMANANA Lyliane**, Maître de conférences, Enseignant Chercheur  
■ **Madame NARY HERINIRINA IARIVO**, Chef de département cartographie au FTM

*Encadreurs :* ■ **Monsieur RAJAONARISON Jean Désiré**, Ingénieur Civil Géographe, FTM  
■ **Monsieur RABEMALAZAMANANA**, chef de Département photogrammétrie, FTM

**Promotion : 2008**





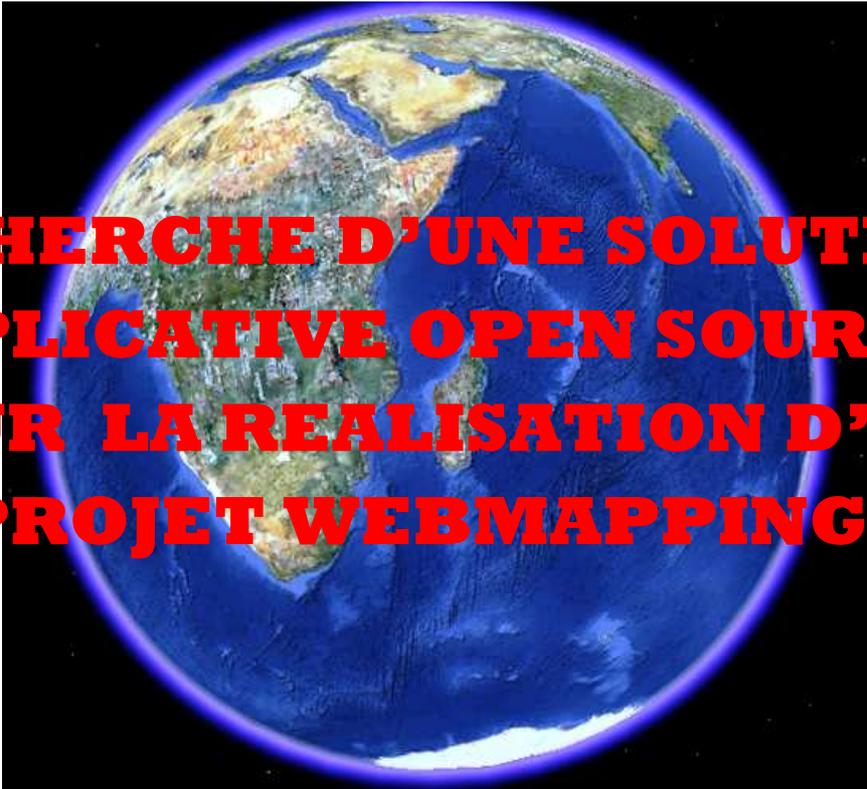
**UNIVERSITE D'ANTANANARIVO**

**ECOLE SUPERIEURE POLYTECHNIQUE**

**DÉPARTEMENT INFORMATION GEOGRAPHIQUE ET FONCIERE**



Mémoire de fin d'études en vue de l'obtention du diplôme d'ingénieur Géomètre-Topographe



**RECHERCHE D'UNE SOLUTION  
APPLICATIVE OPEN SOURCE  
POUR LA REALISATION D'UN  
PROJET WEBMAPPING**

*Présenté le 25 janvier 2011 par :* **ANDRIMPIRENENA Zolalaina Herinarivo Joseph**

*Président du Jury:* ■ **Monsieur RABETSIAHINY**, Maître de conférences Chef du département Information Géographique et Foncière, ESPA

*Examineurs :* ■ **Madame RABEHERIMANANA Lyliane**, Maître de conférences, Enseignant Chercheur  
■ **Madame NARY HERINIRINA IARIVO**, Chef de département cartographie au FTM

*Encadreur :* ■ **Monsieur RAJAONARISON Jean Désiré**, Ingénieur Civil Géographe, FTM  
■ **Monsieur RABEMALAZAMANANA**, chef de Département photogrammétrie, FTM

**Promotion : 2008**

## **REMERCIEMENTS**

Je tiens tout d'abord à remercier Dieu tout puissant parce que ce travail n'aurait pas été possible sans LUI puis les personnes suivantes :

Monsieur ANDRIANARY Philippe Antoine, Directeur de l'École Supérieure Polytechnique d'Antananarivo pour ses efforts afin de nous donner la possibilité d'étudier dans un environnement intellectuel adéquat ;

Monsieur RABETSIAHINY, Chef de Département de la Filière Information Géographiques et foncière, pour son dévouement à la recherche de toute amélioration de notre département, et pour l'honneur qu'il me fait de présider mon jury ;

Monsieur RAJAONARISON Jean Désiré, Ingénieur Civil Géographe au FTM et Enseignant au sein de l'École Supérieure Polytechnique d'Antananarivo pour m'avoir accordé sa confiance, pour sa disponibilité, ses conseils et plus globalement pour son accompagnement durant la réalisation de ce travail de fin d'étude ;

Monsieur RABEMALAZAMANANA, chef de Département photogrammétrie au FTM, pour tout le temps qu'il a consacré à répondre à mes questions, et toute son aide au sujet de la cartographie en ligne. Ses compétences et ses grandes qualités humaines auront fortement contribuées à rendre cette mémoire très enrichissante et très agréable ;

Madame NARY HERINIRINA IARIVO, chef de Département cartographie au FTM qui a bien voulu examiner ce travail ;

Madame RABEHERIMANANA Lyliane, Maître de conférences, Enseignant Chercheur qui a bien voulu aussi siéger parmi les membres de jury ;

L'ensemble des enseignants et des intervenants ayant participé à notre formation au sein de l'ESPA plus particulièrement le département Information Géographique et Foncière ;

Les membres de la promotion 2008, pour les bons moments passés et particulièrement ANDRIAMORAINAMIZAKAZO Adolphe Jean Berthin avec qui j'ai eu des échanges assez réguliers concernant la cartographie en ligne durant la réalisation de ce travail ;

L'ensemble des employés du FTM pour leur gentillesse et leur disponibilité permanente ;

L'ensemble des personnes qui ont à un moment partagé leurs connaissances ou leurs visions de mon travail. Si je ne les mentionne pas tous, je tiens cependant à ce qu'ils soient conscients de l'aide qu'ils m'ont apporté ;

Ma famille plus particulièrement ma mère pour ses encouragements et pour tout le soutien qu'elle m'a apporté durant ces nombreuses années. Merci.

# TABLE DES MATIERES

REMERCIEMENTS.....	i
TABLE DES MATIERES .....	iii
LISTE DES TABLEAUX .....	ix
INTRODUCTION GENERALE.....	x
<b>Chapitre.1. L'INFORMATION GEOGRAPHIQUE ET LE SIG SOUS WEB .....</b>	<b>4</b>
<b>1.1. INFORMATION GEOGRAPHIQUE .....</b>	<b>4</b>
1.1.1. Caractéristiques des données géographiques.....	4
1.1.1.1. La sémantique .....	4
a. Les couches d'informations.....	4
b. L'identifiant.....	5
c. La localisation.....	5
1.1.1.2. Qualité des données dans les Bases de données géographiques.....	5
1.1.1.3. La sémiologie .....	6
1.1.2. Gestion des données .....	6
1.1.3. Les objectifs de gestion de l'information .....	6
1.1.4. Rôle de l'information géographique.....	6
1.1.5. Impact de la technologie sur l'information géographique.....	7
1.1.6. L'information géographique sur Internet .....	7
1.1.7. Besoin en information géographique.....	7
<b>1.2. SYSTEMES D'INFORMATION GEOGRAPHIQUE (SIG) .....</b>	<b>8</b>
1.2.1. Définitions .....	8
1.2.2. Historique des SIG.....	9
1.2.3. Modélisation de l'information géographique .....	9
1.2.4. Représentation et Modèle de données .....	10
1.2.4.1. Modèle Vecteur .....	10
1.2.4.2. Modèle Raster.....	10
1.2.5. SIG et la représentation de la réalité géographique .....	11
1.2.6. Principaux domaines d'utilisation.....	12
1.2.7. Les produits importants .....	12
1.2.8. SIG sous le Web .....	13
1.2.8.1. Internet et SIG .....	14
1.2.8.2. Sites cartographiques statique et dynamique.....	15

1.2.8.3.	Interactivité dans les applications SIG-Web .....	15
1.2.9.	SIG mobiles .....	16
<b>1.3.</b>	<b>LOGICIEL LIBRE ET LES SIG .....</b>	<b>17</b>
1.3.1.	Avantages de modèle de logiciel libre.....	17
1.3.2.	Logiciels libres disponibles pour les SIG .....	20
1.3.3.	Logiciels libres pour les SIG sous Web.....	21
<b>Chapitre.2.</b>	<b>ARCHITECTURE CLIENT-SERVEUR, LES APPLICATIONS DE</b>	
<b>DIFFUSION CARTOGRAPHIQUE ET LE WEBMAPPING.....</b>		<b>22</b>
<b>2.1.</b>	<b>ARCHITECTURE CLIENT-SERVEUR.....</b>	<b>22</b>
2.1.1.	Introduction .....	22
2.1.2.	Le modèle Client serveur .....	23
2.1.3.	Le modèle client-serveur à deux niveaux .....	24
2.1.3.1.	L'approche centrée sur le client .....	24
2.1.3.2.	L'approche centrée sur le serveur .....	26
2.1.4.	L'approche client-serveur à trois niveaux ou architecture 3-tiers .....	27
2.1.5.	Les applications Web.....	28
2.1.5.1.	Les applications Web classique.....	28
2.1.5.2.	Les applications Web de nouvelle génération.....	29
2.1.5.3.	Le Web à trois niveaux.....	29
2.1.6.	Les applications SIG-Web.....	30
2.1.7.	Serveur Web .....	31
2.1.8.	Les applications CGI .....	31
<b>2.2.</b>	<b>LE WEBMAPPING .....</b>	<b>32</b>
2.2.1.	Introduction .....	32
2.2.2.	Principes de cartographie sur Internet .....	33
2.2.3.	Serveur cartographique.....	35
2.2.3.1.	Serveur des cartes.....	35
2.2.3.2.	Serveur de fichiers.....	35
2.2.4.	Principe général de fonctionnement d'un serveur cartographique .....	35
2.2.5.	Fonctionnalités des serveurs cartographiques .....	36
2.2.6.	Les avantages et les inconvénients des serveurs cartographiques .....	37
2.2.6.1.	Avantages .....	37
2.2.6.2.	Inconvénients.....	37

2.2.7.	Les serveurs cartographiques existants.....	38
2.2.7.1.	Les éditeurs classiques .....	38
a.	Avantage .....	38
b.	Inconvénients .....	38
2.2.7.2.	Les solutions Open Source .....	38
a.	Avantages.....	39
b.	Inconvénients .....	39
2.2.8.	Les infrastructures et les logiciels utilisés .....	40
2.2.8.1.	Côté serveur.....	40
2.2.8.2.	Côté client.....	40
2.2.9.	Les outils de développement Web.....	41
2.2.10.	Mise en place d'une application Web.....	41
<b>Chapitre.3.</b>	<b>: DEVELOPPEMENT DU SYSTEME.....</b>	<b>43</b>
<b>3.1.</b>	<b>Introduction .....</b>	<b>43</b>
<b>3.2.</b>	<b>Définitions .....</b>	<b>43</b>
<b>3.3.</b>	<b>Pourquoi modéliser ? .....</b>	<b>43</b>
<b>3.4.</b>	<b>Les éléments d'UML.....</b>	<b>44</b>
<b>3.5.</b>	<b>Les diagrammes d'UML .....</b>	<b>45</b>
<b>3.6.</b>	<b>Les étapes du cycle de vie d'une application.....</b>	<b>45</b>
<b>3.7.</b>	<b>Développement du système.....</b>	<b>47</b>
3.7.1.	L'expression des besoins.....	47
3.7.2.	Spécification des besoins.....	47
3.7.2.1.	Principe de fonctionnement du système .....	47
3.7.2.2.	Fonctionnalité du système .....	48
3.7.3.	L'analyse .....	48
3.7.3.1.	Cas d'utilisation du système.....	48
3.7.3.2.	Le modèle statique.....	49
3.7.3.3.	Le modèle dynamique .....	50
3.7.4.	La conception .....	55
3.7.4.1.	Conception du système.....	55
3.7.4.2.	La décomposition du système en sous systèmes .....	56

<b>Chapitre.4.</b>	<b>: IMPLEMENTATION</b>	<b>60</b>
<b>4.1.</b>	<b>Implémentation : choix des composants logiciels</b>	<b>60</b>
4.1.1.	Les serveurs cartographiques	60
4.1.2.	Les éléments d'une application MapServer	61
4.1.3.	Serveur web	62
4.1.4.	Serveur d'application	63
4.1.5.	Modèle de l'interface	63
4.1.6.	Présentation du CartoWeb	64
4.1.6.1.	Architecture et fonctionnement général de CartoWeb	65
4.1.6.2.	Fonctionnalités de CartoWeb	66
4.1.6.3.	Structure générale de CartoWeb	68
4.1.6.4.	Protocole SOAP	69
4.1.6.5.	Principe de Fonctionnement du CartoWeb	69
4.1.6.6.	Fonctionnement de CartoWeb	69
4.1.7.	SGBDR (Système de Gestion de Base de Données Relationnelles)	70
4.1.7.1.	PostgreSQL	71
4.1.7.2.	PostGIS, principes et utilisation	72
4.1.7.3.	QUANTUM GIS	73
<b>4.2.</b>	<b>Implémentation : Installation et Configuration des composants</b>	<b>73</b>
4.2.1.	Présentation de l'interface et de son développement	74
4.2.2.	Interface de navigation	74
4.2.3.	La liste déroulante d'échelles	76
4.2.4.	Gestion des couches (Onglet Thèmes)	77
4.2.5.	Recherche (Onglet Recherche)	77
4.2.6.	Les outils dessin (Onglet Dessin)	77
4.2.7.	Outils de mesure	78
4.2.8.	Interrogation des objets (Onglet Requête)	78
4.2.9.	Les outils de mise à jour (Onglet Edition)	79
4.2.10.	Impression (Onglet Imprimer)	80
<b>4.3.</b>	<b>Test et validation</b>	<b>80</b>
<b>CONCLUSION ET PERSPECTIVES</b>		<b>82</b>

<b>BIBLIOGRAPHIE.....</b>	<b>84</b>
<b>WEBOGRAPHIE.....</b>	<b>86</b>
<b>ANNEXE.....</b>	<b>89</b>
<b>ANNEXE 1 : Le Fichier « faliarivo.map ».....</b>	<b>90</b>
<b>ANNEXE 2 : La projection pour Madagascar.....</b>	<b>94</b>

## LISTE DES FIGURES

<i>Figure 1 : Les différentes couches d'information géographique .....</i>	<i>5</i>
<i>Figure 2 : Schéma exemplaire d'un SI.....</i>	<i>8</i>
<i>Figure 3: Les deux modèles de représentation de la réalité : vecteur et raster.....</i>	<i>10</i>
<i>Figure 4 : Modèle client- serveur.....</i>	<i>24</i>
<i>Figure 5 : Principe de fonctionnement d'une application côté-client. ....</i>	<i>25</i>
<i>Figure 6 : Principe de fonctionnement d'une application cote-serveur.....</i>	<i>26</i>
<i>Figure 7: Principe de fonctionnement d'une application client-serveur a trois niveaux .....</i>	<i>27</i>
<i>Figure 8: Les composantes d'une application SIG-Web .....</i>	<i>30</i>
<i>Figure 9 : Fonctionnement d'une application CGI .....</i>	<i>31</i>
<i>Figure 10 : La trilogie du Webmapping.....</i>	<i>32</i>
<i>Figure 11 : Principes d'échanges entre un ordinateur client et un serveur .....</i>	<i>34</i>
<i>Figure 12 : Différents diagrammes définis par UML .....</i>	<i>45</i>
<i>Figure 13 : Description globale du système.....</i>	<i>47</i>
<i>Figure 14 : Diagramme des cas d'utilisation.....</i>	<i>49</i>
<i>Figure 15 : Diagramme des classes .....</i>	<i>50</i>
<i>Figure 16 : Scénario d'une simple connexion utilisateur.....</i>	<i>51</i>
<i>Figure 17 : Scénario d'une connexion échouée .....</i>	<i>52</i>
<i>Figure 18 : Diagramme d'états-transitions pour objet « couche ».....</i>	<i>53</i>
<i>Figure 19 : Diagramme d'états-transitions pour objet « utilisateur » .....</i>	<i>53</i>
<i>Figure 20 : Diagramme de collaboration de la fonctionnalité : consultation .....</i>	<i>54</i>
<i>Figure 21 : Diagramme de collaboration de la fonctionnalité : Mise à jour .....</i>	<i>54</i>
<i>Figure 22 : Architecture logiciel du Système .....</i>	<i>55</i>
<i>Figure 23 : Architecture matériel du Système.....</i>	<i>56</i>
<i>Figure 24 : Interface Homme Machine de l'application coté client.....</i>	<i>57</i>
<i>Figure 25 : La place de la conception des objets.....</i>	<i>58</i>
<i>Figure 26 : Le modèle conceptuel de la BD d'authentification.....</i>	<i>58</i>
<i>Figure 27 : Fonctionnement de MapServer .....</i>	<i>62</i>
<i>Figure 28 : Architecture et fonctionnement général de CartoWeb .....</i>	<i>65</i>
<i>Figure 29 : Structure de CartoWeb (source CartoWeb Documentation, Camptocamp) .....</i>	<i>68</i>
<i>Figure 30 : Principe général de fonctionnement de CartoWeb .....</i>	<i>69</i>
<i>Figure 31 : Les principaux composants logiciels du système .....</i>	<i>74</i>
<i>Figure 32 : La vue principale de l'application .....</i>	<i>75</i>

## **LISTE DES TABLEAUX**

<i>Tableau 1 : Tableau comparatif du mode raster et du mode vecteur.....</i>	<i>11</i>
<i>Tableau 2 : Les grands types de sites Web et les niveaux d'interactivité. ....</i>	<i>16</i>
<i>Tableau 3 : Tableau comparatif entre les solutions Open Source et les produits propriétaires. ....</i>	<i>40</i>

## LISTE DES ABREVIATIONS

<b>API</b>	: Application and Programme Interface
<b>ASP</b>	: Active Server Pages
<b>BDG</b>	: Base de Données Géographiques
<b>CGI</b>	: Common Gateway Interface
<b>CPU</b>	: Centre Process Unité
<b>CSS</b>	: Cascading Style Sheets
<b>DHTML</b>	: Dynamic Hypertext Mark-up Language
<b>DRA</b>	: Développement Rapide d'Applications
<b>EPSG</b>	: European Petroleum Survey Group
<b>FTM</b>	: Foiben- Taosaritanin'i Madagasikara
<b>GPL</b>	: Gnu License Public
<b>HTML</b>	: Hypertext Markup Language
<b>HTTP</b>	: Hyper Text Transfer Protocol
<b>IIS</b>	: Internet Information Services
<b>ISAPI</b>	: Internet Server Application Programming Interface
<b>ODBC</b>	: Open Database Connectivity
<b>PDA</b>	: Personal Digital Assistant
<b>PHP</b>	: Hypertext Pre-processor
<b>RPC</b>	: Remote Procedure Call
<b>RTC</b>	: Réseau Téléphonique à Commutation
<b>SGBDR</b>	: Système de Gestion des Bases de Données Relationnelle
<b>SIG</b>	: Système d'information géographique
<b>SOAP</b>	: Simple Object Access Protocol
<b>SQL</b>	: Server Query Language SVG
<b>SVC</b>	: Scalar Vector Graphic
<b>SWF</b>	: Shockwave Flash Format
<b>TCP/IP</b>	: Transmission Control Protocol /Internet Protocol
<b>UML</b>	: Unified Modelling Language
<b>WAN</b>	: Wide Area Network
<b>WFS</b>	: Web Feature Specification
<b>WMS</b>	: Web Map Specification

**WKT** : The Well-Known Text  
**W3C** : World Wide Web Corporation  
**XML** : eXtensible Mark-up Language

## **INTRODUCTION GENERALE**

La montée en puissance des systèmes d'information géographique (SIG), conséquence des progrès informatiques en matière d'acquisition, de stockage, de traitement et de communication des données, les rend indispensables dans notre vie quotidienne surtout avec l'apparition des nouvelles technologies de communication (technologie de client/serveur, technologie de WEB...). En effet, l'aspect communication est décisif en matière de gestion et de prise de décision. Mais il révèle des obstacles d'ordre technique (matériels et logiciels) et humain (compétence et maîtrise).

Le point fort des systèmes d'information géographique est leur capacité de rassembler, dans un seul outil, des données de nature très diverses mais localisées géographiquement. Les SIG ne se limitent pas à rassembler et à communiquer l'information géographique mais permettent aussi d'analyser, manipuler et gérer celle-ci, de simuler divers scénarios d'évolution et de restituer les résultats. La richesse et la disponibilité des outils de communication de l'information géographique corollaire de l'évolution de l'informatique et des réseaux de communication a permis une progression soutenue dans le domaine des SIG. Le progrès qu'a connu ce domaine a engendré une mutation des SIG bureautiques vers les SIG-Web en passant par les SIG sous réseau.

La technologie Internet a considérablement participé à l'échange et à la publication de l'information géographique. Cependant, la normalisation constitue toujours un obstacle et mobilise aujourd'hui des groupes de réflexion à tous les niveaux afin de construire un cadre pour l'interopérabilité et permettre l'échange de ces données.

La publication des données cartographiques sur Internet est devenue un moyen de communication indispensable pour les différents organismes manipulant l'information géographique. Cette technologie basée généralement sur une architecture connue sous le nom « Client/serveur » est largement utilisée pour les applications de diffusion cartographique. Elle est également implémentée avec des outils libres (open source) mis à la disposition du grand public initié par la communauté scientifique internationale. Ces outils libres nécessitent en revanche une maîtrise et beaucoup d'efforts pour pouvoir les utiliser.

Notre objectif majeur à travers ce travail est de proposer une cartographie sur le Web et de disposer d'un outil de consultation et de mise à jour des données spatiales et attributaires à travers un client universel, le plus courant c'est l'Internet Explorer ou Mozilla Firefox. Cette solution sera centrée sur un serveur applicatif appelé serveur spatial, un serveur Web et un serveur de bases de données ; tous issus du monde Open source. L'idée est donc de permettre aux « petits comptes », telles les collectivités locales, d'utiliser ces systèmes sans investissements majeurs.

## **1 Problématique**

La cartographie sur Internet bien qu'elle soit relativement récente a énormément évolué et ce grâce à l'engouement des spécialistes du domaine à vouloir intégrer les technologies issues du monde Internet à leur secteur d'activité. Ainsi, les deux milieux industriels spécialisés et universitaires ont développé les environnements adéquats pour le fonctionnement des solutions de cartographie Web. Notre travail s'inscrit dans l'axe des efforts fournis par la communauté des systèmes ouverts. A cet effet, un certain nombre de problèmes doivent être résolus. Nous citons quelques-uns :

- Quels sont parmi les composants disponibles sur Internet ceux qui sont nécessaires à l'implémentation de la solution arrêtée ?
- Quel format normalisé faut-il adopter ?
- Quels sont les serveurs à déployer ?
- Comment convertir les données de production pour être supportés par les outils Open Source?
- Quelle est l'architecture de la solution finale ? ...

Notre contribution est de répondre à ce genre de questions. Pour cela il a fallu d'abord effectuer une recherche approfondie des outils disponibles sur Internet, de les étudier, de les comparer, d'en choisir les mieux adaptés et enfin concevoir et implémenter notre solution en s'appuyant les composants et outils Open Source soigneusement sélectionnés. Cette implémentation a néanmoins nécessité la réécriture en divers langage de programmation d'un ensemble important de scripts.

## **2 Objectif**

L'objectif est de réaliser une étude des principaux composants Open Source disponibles sur Internet dans le domaine de la cartographie Web, de les analyser et de les intégrer par l'ajout des autres composants dans un processus de rétro ingénierie. Il s'agit de fournir à l'utilisateur final une application opérationnelle, normalisée et sécurisée sans investissements énormes (si le pouvoir public accèderait à fournir les données cartographiques gratuitement).

Cette application est implémentée avec une approche orientée objet en utilisant le langage de programmation PHP pour des raisons de simplicité et de portabilité. C'est également une application client serveur qui fonctionne pour sa partie client avec un simple navigateur Web et qui peut utiliser comme ressource réseau une simple ligne RTC (Réseau Téléphonique à Commutation).

## **3 Méthodologie**

Les SIG constituent un outil important pour les décideurs, cependant cette solution coûte toujours cher et elle reste seulement à la portée des « grands comptes ». Par conséquent, il est intéressant de les mettre à la disposition des « petits comptes » dans une optique de modernisation en mettant à leur disposition un outil simple, de gestion, de planification et d'aide à la prise de décision.

Pour cela, la méthodologie à suivre consiste en :

- Etude et analyse des outils et composants Open Source disponibles.
- Modélisation et architecture de l'application en se basant sur le processus de rétro ingénierie.
- Conception orientée objet (représentation UML, diagramme de classes).
- Implémentation et configuration des différents composants.
- Test et validation de l'application.

## **4 Organisation du mémoire**

Pour bien mener ce travail, il a été structuré en plusieurs chapitres à savoir : Les deux chapitres I et II donnent les concepts fondamentaux de l'information géographique et les systèmes d'information géographique (SIG) en provoquant un nouveau concept en l'occurrence le SIG-WEB (SIG sous Web) qui est le résultat des progrès scientifiques en matière des technologies de l'information et de la communication (Internet, Web, serveur d'application et autres).

Le chapitre III présente un aperçu sur le monde des logiciels libre ou l'Open Source. Cette nouvelle philosophie qui a submergé l'Internet ces derniers temps comme une alternative à l'informatique commerciale fermée considérée comme une boîte noire.

Le chapitre IV concerne l'architecture client-serveur permettant la réalisation des applications de diffusion cartographique.

Le chapitre V décortique la panoplie de logiciels utilisés dans le domaine (SIG-Web), en présentant leurs avantages et leurs inconvénients, notamment les logiciels qui relèvent de l'Open Source. On y trouve un inventaire non exhaustif de ce qu'existe comme solution Web pour la cartographie.

Le chapitre VI est consacré à la conception du système répondant aux besoins exprimés. Ainsi un système (plate forme et interface) a été développé avec des outils et des logiciels libres (Open Source) à savoir : Apache, Php, MapServer, PostgreSQL/PostGIS, CartoWeb...En utilisant le formalisme UML (Unified Modeling Language) pour la phase de conception.

Dans le Chapitre VII, nous détaillons l'implémentation et la mise en œuvre. Ainsi une description de l'ensemble des outils choisis pour implémenter ce système. Ces logiciels sont : Apache (serveur Web), MapServer (serveur spatial), PHP avec le module MapScript (serveur d'application), PostgreSQL (SGBDR) pour gérer les différentes données. La description détaillée de l'interface et des fonctionnalités de ce système est aussi présentée ainsi que le test de fonctionnement.

Nous finirons ce mémoire par une conclusion générale, des perspectives envisagées pour perfectionner ce travail et d'une bibliographie.

# **Chapitre.1. L'INFORMATION GEOGRAPHIQUE ET LE SIG SOUS WEB**

## **1.1. INFORMATION GEOGRAPHIQUE**

L'information géographique est la description d'objets, d'événements, de phénomènes localisés par rapport à la surface de la terre. La connaissance de la position géographique permet d'appliquer un raisonnement spatial.

On rassemble sous la dénomination d'information géographique des données aussi diverses :

- La distribution des ressources naturelles (sols, eaux, végétation, etc.) ;
- Des infrastructures (routes, bâti,...) ;
- Des découpages politiques et administratifs (commune, district, province, etc.) ;
- Des statistiques qui ont une extension spatiale ;
- Description spatiale de la surface de la terre (courbes de niveau, points de coté...).

L'information géographique a connu des évolutions fondamentales, grâce à l'apparition des SIG, la numérisation des données, l'utilisation de l'imagerie à haute résolution et l'exploitation des systèmes de positionnement spatial (GPS, GALILEO,...). Dans sa forme numérique, l'information géographique peut circuler, s'échanger à grande vitesse, sauvegarder et se combiner avec d'autres informations pour en créer une nouvelle. Elle joue un rôle très important dans le développement d'un pays.

### **1.1.1. Caractéristiques des données géographiques**

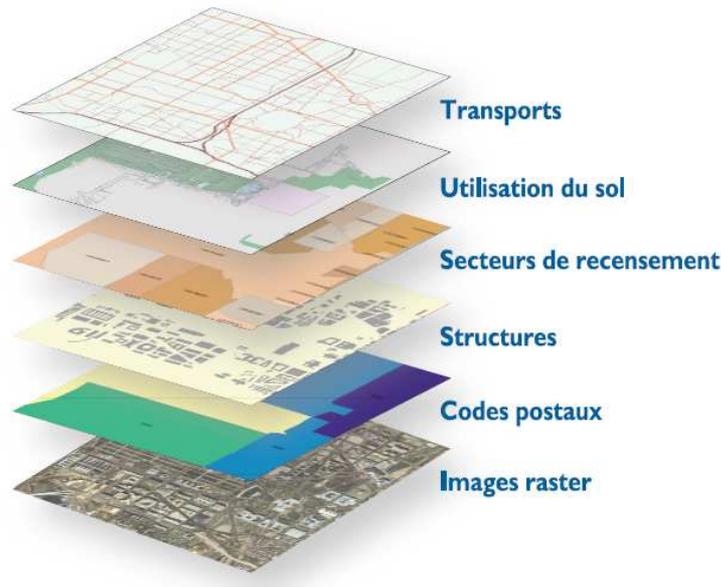
Les données géographiques présentent un certain nombre de caractéristiques qui les rendent particulières lors de leur modélisation et de leur traitement informatique :

#### **1.1.1.1. La sémantique**

Représente la signification des données géographiques qui réfère les phénomènes pouvant être représentés. On examinera seulement quelques aspects, comme :

##### **a. Les couches d'informations**

La diversité des applications géomatiques suggère de structurer les informations géographiques en couches (réseau routier, topographie, hydrographie, orographie ...) (Figure 1).



**Figure 1** : Les différentes couches d'information géographique

## **b. L'identifiant**

Ne réfère qu'un seul objet et il est unique et évite beaucoup de problèmes comme celui de l'homonymie.

## **c. La localisation**

Elle se fait par l'usage des coordonnées (longitude, latitude, x, y...) par rapport à un système de coordonnées bien défini.

### **1.1.1.2. Qualité des données dans les Bases de données géographiques**

Une Base de Données est un ensemble d'informations structuré, géré par un système (noyau) et doté d'un dictionnaire de données.

Les bases de données géographiques (BDG) sont devenues aujourd'hui un outil indispensable pour stocker et gérer l'information géographique qui les englobe et exploitent les données qu'elles contiennent. Ces bases de données sont au cœur d'applications variées tant au niveau de la finalité (système d'aide à la décision, production de données spatiales, publication et diffusion) que des organismes consommateurs des données (collectivités locales, établissements étatiques ou privées, départements ministériels...).

Il existe deux types de qualité :

**Qualité interne** : Ensemble des propriétés et caractéristiques ou service qui lui confère l'aptitude à satisfaire aux spécifications du contenu de ce produit ou du service ;

**Qualité externe** : Adéquation des spécifications aux besoins de l'utilisateur.

### **1.1.1.3. La sémiologie**

L'objectif de la sémiologie (graphique) est de choisir les symboles, les couleurs, le placement des textes, etc., de manière à mettre en évidence les caractéristiques propre de la carte.

### **1.1.2. Gestion des données**

La gestion des informations géographique (numériques) est assurée par le système de gestion de base de données (SGBD). Globalement, les SGBD se consacrent à toutes les tâches de gestion de l'information. Ces systèmes ont donc les fonctionnalités leur permettant de diffuser l'information, de maintenir sa qualité et sa cohérence.

La différence initiale et fondamentale entre un Système de Gestion de Base de Données et un Système de gestion de fichiers est la présence d'un dictionnaire de données (ou catalogue).

### **1.1.3. Les objectifs de gestion de l'information**

#### **❑ Accès à l'information**

Les SGBD constituent une interface entre les données et l'utilisateur ou l'application qui les utilise. Les données sont ainsi accessibles sans se préoccuper de leur localisation physique. Les SGBD ont, de plus, des fonctionnalités d'interrogation qui facilitent la recherche de l'information.

#### **❑ Cohérence**

La diminution de la redondance facilite le maintien d'une base de données cohérente. Elle ne doit pas contenir des informations contradictoires. La gestion des données par un SGBD facilite le contrôle de cette cohérence.

#### **❑ Maintenance**

Les fonctions d'interrogation, la diminution des redondances, ainsi que le maintien de la cohérence facilitent la maintenance de la base de données. Les fonctionnalités d'interrogation permettent de sélectionner les données qui doivent être mises à jour, les contraintes d'intégrités vérifient que les mises à jour réalisées n'introduisent pas d'incohérence et la diminution des redondances aide à la complétude de ces opérations. De plus les SGBD assurent aussi la sécurité des données et les reprises après une panne lors des transactions.

### **1.1.4. Rôle de l'information géographique**

Actuellement, la connaissance du territoire, de son occupation, les paramètres d'environnement liés aux différents phénomènes sont devenus essentiels. L'information géographique est un élément crucial dans l'élaboration des décisions. Cette information est également considérée comme un instrument de gestion de l'espace en utilisant des outils d'assemblage et d'analyse de la connaissance sur le territoire. L'information géographique sert de

support au débat public. Elle permet de visualiser l'impact des choix d'aménagement sur l'espace et sur les hommes qui l'occupent.

### **1.1.5. Impact de la technologie sur l'information géographique**

De nombreux instruments facilitent la collecte de l'information géographique en occurrence : le positionnement spatial (GPS), satellites haute résolution, images aériennes, outils matériels et logiciels (PDA, réseau).

Ces évolutions ont un grand impact sur la production de données notamment la réduction des coûts initiaux de saisie. D'autre part, le développement des nouvelles technologies de communication (Internet, Web...) conduit à des bouleversements plus profonds dans la mesure où les mécanismes d'échange de l'information géographique ont totalement changé.

Le développement de l'information géographique est une des conséquences de ces évolutions notamment le développement des outils matériels et logiciels. La réduction des coûts (logiciels et matériels), l'amélioration de leurs performances et le progrès dans le domaine de communication contribuent à élargir l'utilisation de l'information géographique dans différents domaines.

### **1.1.6. L'information géographique sur Internet**

Le rôle de l'information dans la prise de décision est sans cesse grandissant. L'informatique a permis l'accumulation massive de données et ceci est particulièrement vrai dans le domaine de l'information géographique. Par ailleurs, Internet offre un moyen de mettre ces données directement à disposition (avec une sécurisation d'accès) soit au sein d'une entreprise, soit à l'ensemble des internautes. Généralement, la manipulation de données géographiques implique l'usage et la maîtrise de logiciels SIG. Ces derniers proposent des capacités d'analyse spatiale et de restitution cartographique puissantes, mais au prix élevé. Les évolutions technologiques permettent de proposer à travers un simple navigateur Internet des outils de consultation et de diffusion de données spatiales.

### **1.1.7. Besoin en information géographique**

Le marché de l'information géographique est en croissance continue ces dernières années. La demande a fortement augmenté par les établissements étatiques et les collectivités locales qui représentent un pourcentage élevé de la demande actuelle. Les contraintes du marché sont le prix des données et l'inadaptation des produits offerts par rapport aux besoins. Le développement du marché de l'information géographique a contribué à la création d'entreprises spécialisées dans la production de logiciels et de données. Également, il a poussé la communauté scientifique à développer des outils et logiciels mis gratuitement à la disposition du grand public.

## 1.2. SYSTEMES D'INFORMATION GEOGRAPHIQUE (SIG)

### 1.2.1. Définitions [5] [8] [16]

Actuellement, il existe plusieurs définitions du terme SIG dont on a retenu les suivantes :

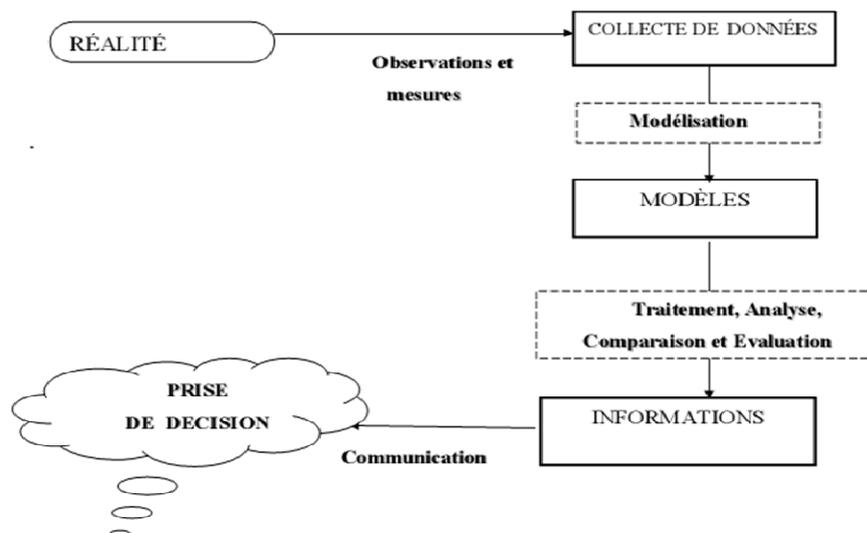
Un SIG est un ensemble organisé de matériels informatiques, de logiciels, de données géographiques et de personnel capable de saisir, stocker, mettre à jour, manipuler, analyser et présenter toutes formes d'informations géographiquement référencées.

C'est un ensemble de processus s'appliquant à des données brutes afin de produire une information pertinente en vue d'une prise de décision. Il utilise indifféremment des données avec ou sans référence spatiale et comprend des modules d'analyse spatiale. Il est conçu afin de permettre la manipulation, l'analyse, la modélisation, la gestion et la représentation de données à référence spatiale en vue de résoudre des problèmes complexes de gestion et de planification.

Les systèmes d'informations géographiques sont considérés comme des outils d'aide à la décision dédiés aux différents opérateurs œuvrant sur le territoire. Ces opérateurs peuvent être des techniciens, gestionnaires, planificateurs ou bien des décideurs. Chacun de ces opérateurs a un niveau d'activité sur les SIG bien spécifique qui peut varier de la simple gestion de données jusqu'à la planification stratégique.

Pour simplifier, un SIG est un ensemble composé de matériel, de logiciel, des utilisateurs, de données et de méthodes. Il regroupe les fonctionnalités d'abstraction, d'acquisition, d'archivage, d'analyse et d'affichage.

Tout système d'information géographique (SIG) réalisé doit assurer, tout d'abord, l'acquisition des données. Ces dernières seront stockées, gérées, analysées, puis traitées et redistribuées aux utilisateurs et autres acteurs (*Figure 2*).



*Figure 2 : Schéma exemplaire d'un SI*

## **1.2.2. Historique des SIG [21]**

Les SIG existaient avant l'apparition de l'informatique. En effet, la combinaison entre l'information spatiale et textuelle n'implique pas forcément l'utilisation de l'outil informatique.

Les premiers SIG sont apparus au début des années 70. Cependant, ce n'est que vers la fin des années 80 qu'une vraie demande de ce genre d'information est réellement apparue, ce qui a donné naissance à un marché de logiciels, de données et de services.

L'évolution et la diffusion des SIG sont fortement liées aux développements de la technologie informatique. On distingue trois périodes principales :

Fin des années 1950 - milieu des années 1970 : début de l'informatique, premières applications de cartographie automatique.

Milieu des années 1970 - début des années 1980 : diffusion des outils de cartographie automatique/SIG dans les organismes d'Etat (armée, cadastre, services topographiques...).

Depuis les années 80 : croissance du marché des logiciels, développements des applications sur PC, utilisation des réseaux informatiques à grande échelle (bases de données distribuées, applications sur Internet, les SIG-Web).

## **1.2.3. Modélisation de l'information géographique**

Une entité spatiale est une entité que l'on veut pouvoir représenter dans l'espace en définissant sa forme (point, ligne ou polygone) et sa localisation. La relation spatiale est un type d'association liant au moins deux types d'entités spatiales et éventuellement des types d'entités non spatiales. Un attribut spatial est une propriété qui varie dans l'espace. (Ex : altitude, profondeur, Z...).

### **☐ Entités spatiales**

On distingue trois types différents d'entités spatiales (SIG 2D) : les points, les lignes et les polygones.

### **☐ Attributs spatiaux et non spatiaux**

Une entité est décrite par des attributs spatiaux (ex : altitude, hauteur...) et non spatiaux (ex : Non, Désignation...).

### **☐ Relations spatiales**

Les relations spatiales sont générées par les SIG. Ces systèmes offrent en particulier des opérateurs spatiaux permettant à l'utilisateur d'effectuer des requêtes qui intègrent les relations spatiales. On distingue deux types d'opérateurs spatiaux : les opérateurs métriques pour effectuer les mesures et les opérateurs topologiques pour traiter les relations entre les entités.

Cinq opérateurs topologiques de base sont définis « disjoint », « touche », « croise », « contient » et « égale ».

### 1.2.4. Représentation et Modèle de données

On distingue généralement deux aspects pour les données des SIG :

- l'aspect sémantique qui donne une description dans différents champs, attributs ou valeur, comme dans un SGBD classique ;
- l'aspect graphique qui est la représentation graphique de l'objet localisé géographiquement.

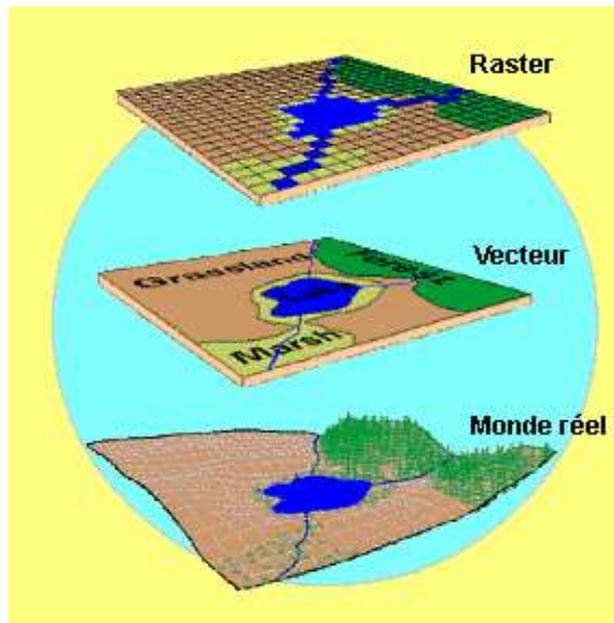
Les systèmes d'information géographique exploitent deux types différents de modèles géographiques :

#### 1.2.4.1. Modèle Vecteur

Utilise des entités géométriques (point, ligne et polygone) pour représenter les entités spatiales. Chaque entité est représentée par un objet localisé dans un système de coordonnées. Cet objet est associé à un enregistrement dans une table d'attributs qui contient les valeurs thématiques décrivant l'entité représentée. Il est particulièrement utilisé pour représenter des données discrètes.

#### 1.2.4.2. Modèle Raster

Divise l'espace avec une grille régulière de cellules (pixels) ordonnées sous forme d'une matrice. Cette dernière produit un espace continu où chaque pixel contient une seule valeur. Donc une couche d'information est représentée par une image (matrice). Il s'adapte parfaitement à la représentation de données continues dans l'espace.



*Figure 3: Les deux modèles de représentation de la réalité : vecteur et raster*

Chacun de ces deux modèles de données dispose de ses avantages et ses inconvénients (*Tableau 1*). Un SIG moderne doit exploiter simultanément ces deux types de représentation.

	<b>AVANTAGES</b>	<b>INCONVÉNIENTS</b>
<b>RASTER</b>	<ul style="list-style-type: none"> <li>-Plus proche à la réalité.</li> <li>-Il est plus facile d'écrire des programmes pour manipuler et traiter les données ;</li> <li>-Modèle assurant une meilleure compatibilité avec les données maillées telles que les images satellitaires numériques ;</li> <li>-Meilleure compatibilité avec certaines sorties de type traceur et imprimante à jet d'encre ou terminaux graphiques.</li> <li>- Adapté à la représentation des phénomènes continus comme par exemple altitude de terrain.</li> </ul>	<ul style="list-style-type: none"> <li>-Requiert beaucoup de mémoire ;</li> <li>-Difficultés à représenter exactement les entités spatiales (lignes topographiques, piste, route, chemins de fer...):</li> <li>-Nécessité de disposer d'un dispositif de conversion vecteur/raster pour intégrer des données en mode vecteur.</li> </ul>
<b>VECTEUR</b>	<ul style="list-style-type: none"> <li>-Beaucoup moins de mémoire requise ;</li> <li>-Possibilité de changer la symbolisation des objets représentés sur la carte ;</li> <li>-Possibilité d'associer à des entités spatiales plusieurs attributs descriptifs ;</li> <li>-Possibilité d'appliquer des requêtes spatiales sur les objets</li> </ul>	<ul style="list-style-type: none"> <li>-Les fonctions spatiales d'analyse sont beaucoup plus complexes ;</li> <li>-Certaines données de variable continues (ex : altitude, images satellitaires) ne peuvent être représentées sans traitement préalable (classification, visualisation 3D...).</li> </ul>

***Tableau 1 : Tableau comparatif du mode raster et du mode vecteur***

Ces objets sont géoréférencés, dans un système de coordonnées. Cela permet la localisation adéquate des objets les uns par rapport aux autres.

Chaque objet (mode vecteur) est lié avec des données attributaires. Ces dernières peuvent être de types différents : numériques, alphanumériques, logiques...Elles sont stockées dans des enregistrements dans une base de données.

### **1.2.5. SIG et la représentation de la réalité géographique**

La grande force des SIG est donc de pouvoir rassembler, dans un outil unique, des données de natures très diverses, mais localisées géographiquement.

Les SIG acceptent les cartes numérisées ou les ressources provenant de la télédétection comme les photographies aériennes et les images satellites.

Les SIG possèdent tous des modalités d'acquisition des données, soit par importation de fichiers, soit par numérisation de documents propres. Ils doivent permettre également l'archivage de données, la garantie de leur sécurité et assurent leur actualisation. Ils doivent offrir également des outils d'analyse spatiale et d'interrogation attributaires des bases de données associées. Ils doivent offrir des outils d'affichage et de visualisation ainsi que la possibilité d'exporter des documents élaborés.

### **1.2.6. Principaux domaines d'utilisation**

Beaucoup d'organismes ou d'entreprises disposent déjà de données informatiques comportant des références géographiques : adresses, localisation, etc. Les outils de gestion de données classiques comme les bases de données et les logiciels de comptabilité gèrent ce type de renseignements passivement et ne permettent d'effectuer que quelques requêtes simples. Les SIG ont de très nombreuses applications pour :

#### **❑ Localisation :**

- Cartographie numérique ;
- Evaluation et localisation des risques naturels et technologiques.

#### **❑ Gestion :**

- Gestion des ressources naturelles ;
- Gestion des réseaux (énergie, réseaux routiers) ;
- Gestion des interventions des services d'urgence ;
- Gestion de l'espace naturel (déforestation, forêt, parcs) ;
- Aménagement de territoire ;
- Plans d'urbanisme ;
- Etudes et projets routiers ;
- Calcul et optimisation des itinéraires ;
- Études d'impacts, avant-projets.

#### **❑ Communication :**

- Cartographie statistique ;
- Cartes thématiques ;
- Illustration de rapports & documents ;
- Partage de l'information.

### **1.2.7. Les produits importants [15]**

Actuellement, le marché des logiciels SIG est dominé par des produits d'origine américaine. En l'occurrence ESRI avec ses produits : ArcInfo, ArcEditor et ArcView...qui occupe la position de leader. Intergraph (GeoMedia) et MapInfo suivent derrière. De grands logiciels progressent en matière d'interfaces, d'affichages en 3D et des outils les accompagnent. Récemment on voit l'apparition de nouvelle génération de produits en l'occurrence, les serveurs cartographiques à distance. Cependant, la question majeure reste celle des droits sur les données, leur propriété, leurs droits d'accès, leur diffusion et leurs coûts.

Il faut dire enfin que l'introduction d'un SIG opérationnel dans une entreprise ou une collectivité doit s'accompagner d'une réflexion sur l'organisation du travail. Les SIG ne fournissent qu'une aide à la décision, le pouvoir final étant bien celui du décideur économique, administratif ou politique.

Une des difficultés de la mise en place d'un SIG réside dans le choix du bon outil notamment lorsqu'on a une grande panoplie de choix. Mais les difficultés de la gestion et de l'utilisation des données sont plus importantes que celles de l'outil SIG proprement dit. En effet l'acquisition et la gestion des données représentent 70 à 80 % du coût global d'un SIG. L'intégration des données existantes, l'enrichissement et la restructuration des données ou leur échange sont des opérations souvent complexes et coûteuses.

### **1.2.8. SIG sous le Web [4] [18]**

Le Web offre à ce jour de nombreuses et diverses techniques permettant de diffuser l'information géographique (carte et données) sur le Web.

Dans sa conception la plus classique, un SIG comporte un ensemble de données structurées au sein d'une base de données et un ensemble de données géographiques organisées en couches.

L'ouverture sur le Web se réalise par le biais d'une application spécifique mettant en format adéquat les données issues du SIG à des fins de consultation, de diffusion ou autres. Il permet non seulement de visualiser les informations géographiques au travers d'une interface Web mais également l'acquisition, la manipulation et la gestion des données géographiques de type point, ligne, surface et de type raster.

L'intégration des modules SIG à travers une interface Web est une innovation actuelle. Elle offre un accès aux fonctionnalités d'un SIG sans la complexité de mise en œuvre des systèmes de serveurs SIG connus. A partir d'un fond cartographique, l'outil permet de saisir via une interface, des données spatiales, de créer et de mettre à jour ces données. L'interface de saisie est conviviale. Elle suit le processus naturel de collecte de l'information:

- Identifiant de l'auteur (lien vers la base de contacts du site) ;
- Choix de la nature de la donnée saisie ;
- Choix entre nouvelle saisie ou mise à jour ;
- Définition du périmètre de saisie à partir d'une carte ou d'une liste de valeur ;
- Affichage de la liste des entités géographiques du périmètre sous forme d'un tableau ;
- Contrôle de la saisie ;
- Validation des données et enregistrement dans la base de données centralisée.

### **1.2.8.1. Internet et SIG**

Internet est un réseau mondial où des ordinateurs reliés parlent la même langue pour communiquer et proposer des services en utilisant le protocole TCP/IP. Le protocole TCP/IP (Transfer Control Protocol/Internet Protocol) est une convention déterminant la façon dont les ordinateurs s'échangent de l'information. Cette convention implique que chaque ordinateur est identifié par une seule adresse (IP) et donc que la donnée peut être envoyée d'une adresse vers une autre (ordinateur) sans ambiguïté. Cette information est divisée en petits paquets et est rassemblée à son arrivée.

Parmi les objets qu'Internet permettra d'échanger, ceux qui relèvent du champ de l'information géographique, c'est à dire qui est référencés par leur position géographique et qui peuvent faire l'objet d'une représentation cartographique.

Outre les progrès en matière de systèmes de positionnement (GPS) et d'observation satellitaire à haute résolution, les solutions de traitement et de communication utilisant les nouvelles technologies de communication modifient considérablement notre vue du domaine.

La publication de données spatiales sur Internet est devenue un moyen de communication indispensable. Il permet d'élargir et généraliser l'accès à l'information géographique. Toutes les données géographiques (Raster ou vecteur) pourront être publiées. En effet, au sein d'un SIG-Web deux modes prédominants de représentation cartographique peuvent cohabiter : le mode Raster et le mode vecteur, selon lesquels la transmission d'une carte peut se faire.

En mode Raster les formats les plus connus sont : GIF, JPEG, PNG, TIF.... ; Pour les formats vectoriels on peut distinguer deux types : le SVG et le SWF dit aussi Flash et ne sont lus par un navigateur Web qu'avec l'adjonction d'un « plug-in » (Plug-in : Élément logiciel que l'on adjoint à une application pour en étendre ses fonctions).

#### **❑ Le format SVG (Scalar Vector Graphic) [26]**

Le format SVG est un format texte, recommandé par W3C (World Wide Web Corporation). Il est apparu en fin 1999 et publié sous forme de recommandation par W3C le 4 septembre 2001 sous sa version 1.0. Il a émergé comme un standard de fait parmi les moyens d'intégrer du graphisme vectoriel sur un site Web. Il est défini comme un langage de type XML décrivant des dessins vectoriels en deux dimensions.

En matière de diffusion de données cartographique le format SVG permet :

- L'affichage dynamique par couches : zonage thématique et intégration des images (images satellitaires, aériennes ...)
- Système de coordonnées relatives pour positionner les objets ;
- Les transformations géographiques élémentaires (translation, changement d'échelle, rotation) grâce à des opérations matricielles applicables à chaque objet ou groupe d'objets ;

- Des fonctionnalités de recherche d'éléments textuels et de localisation relative d'objets au sein d'une carte ;
- L'indexation des données textuelles des cartes, (légendes, étiquettes) par les moteurs de recherche.

Une large interactivité (coté client) à travers la gestion des événements utilisateur classique (onclick, onmousemove ...) :

- Une qualité esthétique de la représentation : Fading, filtre, gestion de transparence accès aux feuilles de style du document ;
- Une grande ouverture : c'est une partie intégrante du standard XML donc normalisé et qui a fait l'objet d'une recommandation W3C depuis le 04/09/2003.

Cependant, à coté de ces avantages, le format comporte également un certain nombre de contraintes :

- Nécessité d'un plug-in qui se charge dans le navigateur lors de la première utilisation ;
- Absence de géoréférencement : les objets sont positionnés en coordonnées relatives ;
- Taille de fichiers dépendant du nombre d'objets à représenter.

#### ❑ Le format Flash SWF (ShockWave Flash Format) :

Le format Flash SWF est un format binaire et propriétaire de la firme Macro-média.

### **1.2.8.2. Sites cartographiques statique et dynamique [18]**

Initialement les sites étaient statiques et proposaient des pages figées à l'avance. Les cartes présentées des images statiques, éventuellement cliquables, insérées dans le code HTML. Puis, sont apparus des sites dynamiques dans lesquelles les pages ont été générées à la volée en réponse à une requête de l'utilisateur. Enfin, on assiste depuis peu à l'émergence de sites interactifs dans lesquels les données géographiques présentées peuvent être interrogées et manipulées (échelle, sélection, mise à jour ...). Cette évolution technologique permet d'aller jusqu'à proposer des vraies fonctionnalités SIG. Les sites cartographiques peuvent se définir comme dynamiques ou statiques et ainsi on peut différencier entre une carte statique et une carte dynamique. En effet, le contenu généré par le serveur peut être fait à partir des données géographiques ou attributaires ou les deux. Une carte peut être statique mais les données attributaires peuvent être extraites d'une base de données à la demande. La carte et les données attributaires sont générées selon la demande de l'utilisateur.

### **1.2.8.3. Interactivité dans les applications SIG-Web**

Un site cartographique est caractérisé par les fonctionnalités qu'il offre à l'utilisateur pour agir sur une carte, par la notion d'interactivité où plusieurs degrés sont différenciés :

Le premier niveau est la visualisation d'une carte avec comme interaction les possibilités qu'offre le HTML. Les fonctions de déplacement sur une carte, pan et zoom, ainsi que les fonctions de

présentation (affichage de couche, changement d'échelle, vue globale) constituent le deuxième niveau.

Enfin, des fonctions se rapprochant de celles d'un SIG de bureau telles que l'acquisition, la manipulation, la gestion et le traitement des données géographiques ou attributaires sont aussi réalisables.

Dans cette démarche on peut distinguer trois types de sites selon les fonctionnalités proposées (Tableau 2) :

Type de site	Niveau d'interactivité	Grande fonctionnalité
Site classique	Pauvre	Publication des cartes produites par un SIG
Site de WebMapping	Moyenne	Des cartes interactives + pages HTML dynamique +des requêtes SQL.
Site Web/SIG interactif	Riche	Fonctionnalité SIG (Mise à jour) Système ouvert, évolutif.

***Tableau 2** : Les grands types de sites Web et les niveaux d'interactivité.*

### **1.2.9. SIG mobiles [13]**

Les applications mobiles des SIG permettent à des utilisateurs en déplacement la collecte et la numérisation des données directement sur le terrain à partir de micro-ordinateurs portables, de tablettes PC ou des PDA (Personal Digital Assistant). Ainsi, les organisations peuvent intégrer à leurs bases de données et leurs applications des informations en temps réel, ce qui permet d'accélérer l'analyse, le traitement, l'affichage et la prise de décision.

## **1.3. LOGICIEL LIBRE ET LES SIG**

Un logiciel libre est un logiciel dont le code source est disponible, il peut ainsi être reproduit, modifié, et redistribué. Généralement issu de milieux universitaires et amélioré par des communautés de développeurs ; Il peut être accompagné de plusieurs exécutables correspondant à différents systèmes d'exploitation (Windows, Unix, Linux..).

La licence du logiciel libre permet à son utilisateur de comprendre et d'étudier le fonctionnement du logiciel, de le paramétrer pour un usage bien particulier et de faire connaître et distribuer à la communauté des utilisateurs les modifications ou améliorations apportées sur ce logiciel.

Les fondements philosophiques du mouvement du logiciel libre sont proches de ceux qui ont servi de base au mouvement scientifique depuis de nombreuses décennies : la mise en commun des idées et du savoir collectif pour permettre la progression de la recherche et l'augmentation de ce savoir.

Le terme "logiciel libre" désigne ce que les anglophones appellent "Free Software" ou "Open-Source Software ". Ces différents termes sont synonymes, et la caractéristique principale de ces logiciels n'est pas d'être gratuit (bien qu'ils le soient généralement) mais bien d'être libre et disponible sous forme de code source. Un logiciel libre n'est pas simplement placé dans le domaine public par son auteur, qui abandonnerait ainsi tous ses droits, mais il est soumis à une licence qui détermine les droits et devoirs de ceux qui l'utilisent. La plus connue de ces licences est la GNU General Public Licence (GNU GPL) qui garantit à l'utilisateur quatre (04) libertés :

- Liberté d'utiliser le programme, quel que soit l'usage ;
- Liberté d'étudier le fonctionnement du programme et de l'adapter à ses propres besoins, ce qui nécessite l'accès au code source ;
- Liberté d'améliorer le programme et de publier ses améliorations ;
- Liberté d'en redistribuer des copies.

Le "logiciel libre" ne doit pas être confondu au "freeware", appellation qui désigne les logiciels en code binaires dont l'usage est gratuit. Ces derniers ne sont donc pas disponibles en code source, et il arrive qu'ils deviennent soudainement payant ou passent en catégorie "shareware", lorsqu'ils rencontrent un certain succès.

### **1.3.1. Avantages de modèle de logiciel libre**

L'énumération ci-dessous est bien loin d'être exhaustive. Parmi les nombreux atouts du logiciel libre, nous retenons ceux qui sont particulièrement importants en milieu académique :

❑ **Coûts et modalités d'utilisation :**

- Les logiciels libres n'occasionnent aucun frais de licence et d'entretien (bien qu'il soit possible, dans certains cas si l'utilisateur le souhaite, de payer pour recevoir une distribution (CDs, documentation) et un certain support.
- On peut déplacer son budget sur les services, l'intégration, la sécurisation, la personnalisation, la formation, l'assistance technique... donc au bénéfice de l'utilisateur.
- Le logiciel libre entraîne par conséquent une diminution de coût total de possession de l'informatique par rapport aux solutions propriétaires.
- Les logiciels libres peuvent ainsi être utilisés sans restriction par les étudiants, enseignants et collaborateurs sur leurs machines professionnelles et privées.

#### ❑ **Qualité, niveau de fonctionnalité et sécurité :**

- Les logiciels libres sont le fruit d'un travail de développement coopératif réalisé par les meilleurs spécialistes de la discipline (souvent issus du milieu académique) et expertisé/validé par des centaines de développeurs.
- Le degré de réactivité de ce modèle de développement est très élevé ; l'implémentation de technologies nouvelles ainsi que la mise à disposition de correctifs sont très rapides (exemples: serveur Web Apache, PHP...).
- Les logiciels libres sont avant tout conçus comme des outils destinés à offrir les services correspondant aux besoins des utilisateurs, et sont développés sous le contrôle d'une communauté ouverte attentive à l'état de l'art en matière de génie logiciel. Par opposition, les logiciels commerciaux sont essentiellement vus par leurs auteurs comme des marchandises, développées dans le secret d'une équipe avec comme souci principal la rentabilité, mais sans garantie de pérennité.
- De l'avis des spécialistes de la sécurité informatique, l'accès au code source d'un logiciel est la seule véritable garantie de sécurité : cela permet de débusquer plus rapidement les erreurs de programmation et empêche l'implémentation de portes d'entrée indiscretes.
- Les mises à jour des logiciels libres peuvent être planifiées sans contrainte, et uniquement lorsque cela est nécessaire (contrairement aux logiciels propriétaires qui nécessitent des mises à jour à répétition et coûteuses).

#### ❑ **Portabilité, souplesse et performances :**

- La disponibilité du code source facilite la portabilité sur différentes architectures.
- Conçus dans un esprit d'ouverture (en particulier multi-plateformes) les logiciels libres sont beaucoup moins intrusifs que les logiciels propriétaires : les imbrications avec le système d'exploitation sont en général beaucoup plus faibles (pas de problèmes de DLLs...), ce qui diminue les risques de conflits entre logiciels installés et facilite les processus d'installation et de mise à jour.
- En général mieux optimisés que les logiciels commerciaux, les logiciels libres sont souvent basés sur l'utilisation d'algorithmes issus des travaux de recherche les plus

avancés et donc performants par nature ; (exemple: Linux tourne bien plus efficacement que Windows sur des machines moins performantes).

#### □ **Ouverture, transparence, flexibilité, échanges, autonomie et liberté de l'utilisateur :**

- Les protocoles et les formats de données des logiciels libres sont publics et bien documentés (standards ouverts), ce qui facilite grandement la compatibilité et l'interopérabilité entre applications de même que les échanges entre utilisateurs.
- L'adaptation des logiciels libres à de nombreuses plates-formes laisse à l'utilisateur toute latitude quant au choix du système d'exploitation (Windows, Unix ou Linux...).
- Du fait du grand nombre de programmeurs participant au développement des logiciels open source, le code est forcément bien documenté ; d'éventuels mauvais choix réalisés par certains développeurs sont rapidement identifiés par d'autres et corrigés ; l'ouverture du code garantit également un respect strict des standards.
- Les contacts entre l'utilisateur final et la communauté développant un logiciel libre (feedback, annonce de bugs, proposition d'améliorations, demande d'aide...) sont très aisés, directs et gratuits.
- Derrière la plupart des logiciels libres se trouve une communauté d'utilisateurs partageant leurs connaissances et développements dans un esprit d'ouverture (forums News dédiés, sites Web, documents FAQ, tutorials...).
- La gestion des logiciels libres est facilitée par l'abondante documentation disponible et par la transparence des procédures d'installation et de mise à jour : toutes les options d'installation et de configuration sont documentées et accessibles (fichiers de configuration lisibles et éditables...), par opposition aux mécanismes opaques du monde Windows (de type "Setup.exe" où tout est caché par l'interface utilisateur graphique) et où l'on reste complètement désemparé lorsque quelque chose ne se déroule pas normalement.
- Le logiciel libre favorise un authentique pluralisme technologique : la diversité des solutions libres disponibles apporte à l'utilisateur un maximum de flexibilité quant au choix de la solution correspondant le mieux à ses besoins ; le cas échéant, l'utilisateur peut même adapter lui-même la solution choisie (modification du code)
- De façon générale le logiciel libre favorise la créativité et les échanges, par opposition aux solutions commerciales propriétaires qui enferment l'utilisateur dans la logique du vendeur.
- Le logiciel libre protège donc ses utilisateurs des risques et désavantages des solutions fermées ou à source unique, leur permettant en particulier d'être indépendants de la politique des éditeurs de logiciels et de ne pas être vulnérable en cas de faillite du fournisseur.

#### □ **Pérennité :**

- Le logiciel open source est un gage de pérennité en ce sens que, lorsqu'un logiciel libre est abandonné par ses auteurs, d'autres programmeurs le reprennent en général, lui offrant un

autre destin que celui de finir aux oubliettes ; au pire, la disponibilité du code source permettrait d'engager une personne pour le gérer.

- On peut aussi parler de meilleure pérennité des données en raison du haut degré de standardisation des protocoles et formats utilisés par les logiciels libres.

De nombreuses applications destinées à l'information géographique basées sur des logiciels libres sont apparues. Outre son coût marginal d'acquisition, l'intérêt du logiciel libre dans le domaine géographique réside également dans la référence aux normes internationales (normes ISO, recommandations Open GIS Consortium...), souvent plus utilisées que dans les logiciels du commerce.

Cependant, les logiciels libres réclament un investissement conséquent en temps et en compétences informatiques [CNIG, 2004].

### **1.3.2. Logiciels libres disponibles pour les SIG [9]**

Il n'existe pas de logiciel complet offrant toutes les fonctions utiles à la construction d'un SIG. L'offre gratuite concernant l'information géographique est constituée de très nombreux outils, répondant à des objectifs bien définis, allant des extensions de logiciels commerciaux pour SIG aux applications plus ciblées.

Cependant, on peut trouver une panoplie d'outils spécialisés, efficaces et performants permettant de compléter et d'enrichir les fonctionnalités d'un logiciel pour SIG généraliste.

Ces outils sont souvent disponibles sur le Web sous forme de scripts ou de fragments de codes.

Egalement, de nombreuses bibliothèques, code source et classes plus ou moins spécialisées dans le traitement de l'information géographique sont publiés et accessibles via l'Internet.

Les logiciels libres les plus connus dans le domaine des SIG sont :

- **JUMP** (Unified Mapping Platform): solution SIG bureautique open source, basé sur les standards du Consortium OpenGIS ;
- **SAGA**: "System for an Automated Geo-scientific Analysis" ;
- **LandSurf**: SIG orienté visualisation et analyse de surfaces ;
- **MapScan**: vectorisation de cartes ;
- **Geomatica FreeView** (PCI Geomatics): visualisation de nombreux formats vecteurs et raster, images satellite...
- **Géographie Explorer**: (Blue Marble Geographics) visualisation de formats vecteurs et raster courants ;
- **ArcExplorer** : (ESRI) visualisation et représentation thématique de données ESRI (Shape. SDE) ;
- **MapInfo ProViewer** : visualisation de tables et documents MapInfo (TAB, WOR).

### **1.3.3. Logiciels libres pour les SIG sous Web**

La diversité de ces outils permet de réaliser de véritables applications pour la visualisation des données, la saisie de données et l'interrogation par des requêtes spatiales via une interface Web en naviguant via un explorateur standard qui est pratiquement disponible dans tous les systèmes d'exploitation.

Ainsi, il est possible, avec quelques applications ou bibliothèques libres, d'envisager des applications allant jusqu'à la production et la mise à jour de la donnée et son exploitation via Internet ou via un réseau local.

### **Conclusion partielle**

Aujourd'hui, les solutions open source concernant le domaine des SIG-WEB sont en pleine expansion et développées autour de la philosophie GPL (General Public Licence) qui autorise la copie, la diffusion et la modification du code source mis à la disposition des utilisateurs gratuitement

Dans ce chapitre nous avons présenté les avantages qui nous ont amenés à recourir à l'open source, cependant, l'utilisation des composants permettant de construire des solutions SIG-WEB nécessite d'investir en temps et en compétences pour la maîtrise du domaine.

## **Chapitre.2.ARCHITECTURE CLIENT-SERVEUR, LES APPLICATIONS DE DIFFUSION CARTOGRAPHIQUE ET LE WEBMAPPING**

### **2.1. ARCHITECTURE CLIENT-SERVEUR**

#### **2.1.1. Introduction [6]**

Depuis longtemps déjà, les chefs d'entreprises, les gestionnaires et les décideurs savent qu'il est indispensable pour une entreprise de définir une stratégie informatique pour améliorer sa productivité, pour accéder à de nouvelles sources de revenu et pour satisfaire toujours plus sa clientèle en améliorant les processus de production. Les entreprises utilisent l'informatique à trois niveaux :

##### **❑ Gestion des données :**

Il s'agit là des systèmes informatiques fondamentaux qui contrôlent les processus clés d'une organisation (la comptabilité, le contrôle des stocks, le traitement des commandes, le suivi des tâches, etc.). De nombreuses technologies, allant des systèmes transactionnels à l'architecture client-serveur, ont été appliquées à ces systèmes vitaux pour la pérennité de l'entreprise.

##### **❑ Productivité du personnel :**

L'évolution rapide du PC et des suites bureautiques intégrées a bouleversé irrémédiablement la manière dont chaque employé traite l'information ; elle a également souvent modifié les pratiques et les stratégies des entreprises. Ces outils ont permis d'obtenir des gains de productivité considérables, de rationaliser les activités des entreprises, et mieux rentabiliser les investissements réalisés dans l'informatique.

##### **❑ Groupware :**

L'utilisation de logiciels de communication et de collaboration a permis à la fois aux organisations et aux individus de travailler en partenariat et en équipe. Les systèmes de ce type s'appliquent aujourd'hui à l'ensemble de l'entreprise, avec des effectifs pouvant facilement dépasser plusieurs milliers d'employés, ce qui permet à l'entreprise de définir de nouveaux modes de travail tel le télétravail ou le nomadisme donc plus de souplesse et par conséquent une réduction des coûts.

Toutefois, tous ces avantages ont généralement un coût. Chacun de ces aspects de l'informatique possède ses propres infrastructures, et les entreprises doivent aujourd'hui faire face à une lourde tâche : établir des jonctions entre les différents systèmes et les différentes applications. De ce fait, il a toujours été extrêmement difficile de créer des systèmes qui englobent toutes ces dimensions. De surcroît, pour les entreprises, la multiplicité des infrastructures est synonyme de coûts supplémentaires en matière de logiciels, de matériel, d'assistance technique et de formation.

Au fil des années, un certain nombre d'architectures ont été conçues pour intégrer les différents environnements distribués susceptibles d'être déployées au sein d'une même organisation. Les entreprises tournées vers l'avenir ont ainsi exigé que cette intégration :

- Automatisé les processus de l'entreprise notamment celui lié à la gestion de production afin de réduire les coûts ;
- Offre un accès aisé à toutes les informations de l'organisation (avec toutes les précautions qui s'imposent en matière de sécurité) ;
- Offre de nouvelles opportunités marketing ;
- Supprimé les barrières au sein de l'organisation ;
- Offre une flexibilité qui permet à l'entreprise de réagir rapidement aux changements.

D'où la nécessité à de nouvelles solutions informatiques qui reposent à la fois sur l'architecture client-serveur et sur les technologies Internet :

### **2.1.2. Le modèle Client serveur [1][6]**

Le modèle client-serveur est l'un des modèles les plus présents dans le monde de l'informatique. Il a commencé à se développer quand les informaticiens se sont mis à abandonner les systèmes partagés reposant sur une logique centralisée, au profit de réseaux de station de travail et de serveurs. Le modèle client-serveur suppose de mettre en place une architecture d'application qui permet de subdiviser un processus informatisé en au moins deux tâches moins complexes, avec un mécanisme de communication (protocole) permettant à ces sous-processus de coopérer entre eux. La finalité de cette subdivision est de disposer de couches de fonctionnalités pouvant être écrites par des programmeurs et déployées sur différentes machines d'une manière optimale. On distingue trois types de couches d'application :

**Logique de présentation :** Elle a pour but de définir comment l'utilisateur doit interagir avec l'application ; elle est généralement implémentée à l'aide d'une interface utilisateur graphique simple.

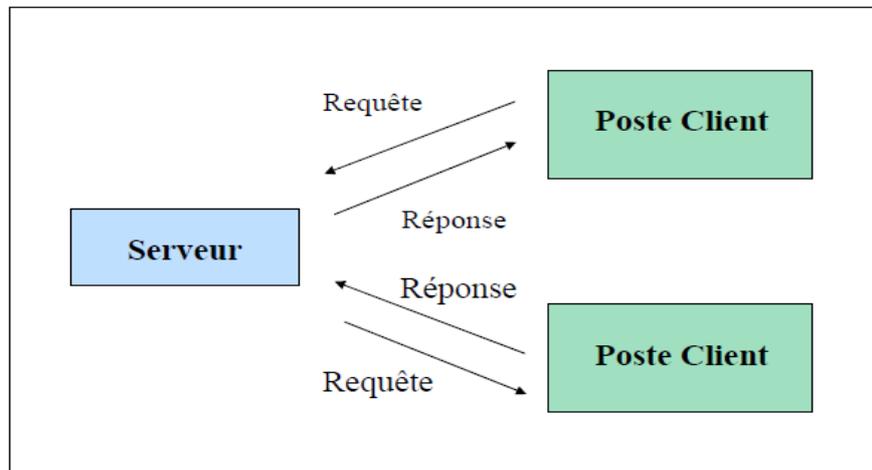
**Logique applicative :** elle vise à définir les mécanismes (ou traitement métier) de l'application.

**Gestion de l'accès aux données :** elle s'intéresse au stockage de données et à leur récupération ; il est primordial que l'intégrité des données soit garantie.

Le développement de couches distinctes impose de prendre très au sérieux la phase de conception et de définir très précisément les limites séparant les différentes couches afin de garantir que la logique sur laquelle chacune repose ne débordera pas sur une autre.

Le modèle client-serveur est considéré comme une technologie très puissante qui permet d'implémenter d'une manière modulaire et flexible des systèmes au sein d'une organisation. Il permet d'abandonner le modèle d'implémentation reposant sur des machines spécifiques installées dans des services isolés, au profit d'une implémentation généralisée à l'ensemble de l'entreprise.

Il existe de nombreuses variantes de l'architecture client-serveur, et la décision la plus importante en matière d'implémentation de systèmes de ce type ; consiste précisément à choisir la variante appropriée.



***Figure 4 : Modèle client- serveur***

### **2.1.3. Le modèle client-serveur à deux niveaux**

La première génération de modèle client-serveur est une version améliorée des applications à partage de fichiers. Avec les nouvelles applications, le serveur de fichiers central est remplacé par un SGBDR (système de gestion de bases de données relationnelles) spécialisé.

Quand le client (une application pour station de travail dotée d'une interface utilisateur graphique) a besoin d'opérer sur des données, il envoie une requête au serveur de données via le réseau. Le SGBD le traite, ensuite, la requête et l'envoie que les données correspondant aux besoins du client.

Comparée aux applications à partage de fichiers (qui envoyaient le fichier complet), cette architecture client-serveur réduit considérablement le trafic sur le réseau. En outre, les SGBD actuels offrent une quantité de fonctions permettant de développer des applications multi-utilisateurs sophistiquées (offrant à plusieurs utilisateurs la possibilité d'accéder simultanément à des données et de les mettre à jour en toute sécurité).

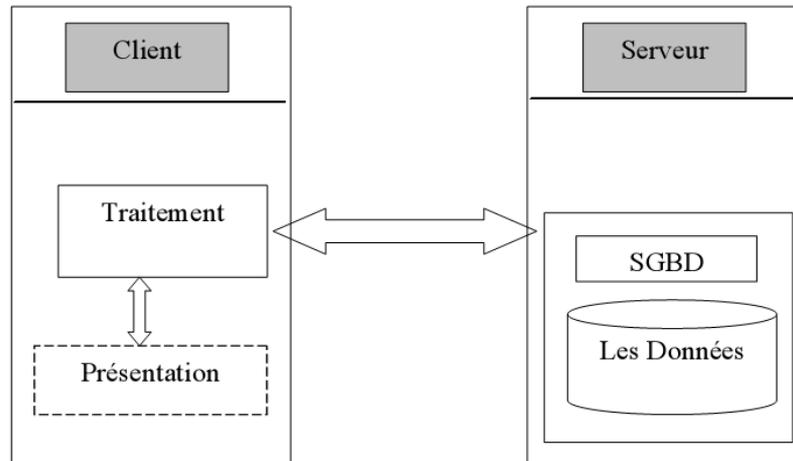
Le traitement des opérations étant divisé en deux couches distinctes, la station de travail et le serveur de données. On parle de modèle client-serveur à deux niveaux (two tier, en anglais) pour désigner ce type d'architecture. Suivant le rôle de chaque niveau on distingue deux approches :

#### **2.1.3.1. L'approche centrée sur le client**

L'implémentation la plus simple et la plus répandue des systèmes à deux niveaux consiste à placer la logique de présentation et les traitements métier sur le client. On parle dans ce cas de l'architecture client lourd /serveur léger. Où le client est associé à une autre application pour

acquérir des fonctionnalités de visualisation de format autre que HTML de type vecteur ou raster, des fonctionnalités SIG (Figure 3).

Les requêtes de bases de données émanant des clients sont généralement implémentées à l'aide du langage de requête et de programmation SQL (Structures Query Language). La requête de base de données est transmise au serveur de données à l'aide d'un protocole de transport pour bases de données distantes.



***Figure 5 : Principe de fonctionnement d'une application côté-client.***

Les systèmes de ce type sont simples à implémenter et par conséquent peu coûteux ; à l'aide de techniques de développement rapide d'applications (DRA). Ils offrent des possibilités très intéressantes pour les environnements de type groupe de travail. Beaucoup d'éditeurs d'outils de programmation proposent aujourd'hui des kits de développement bon marché, avec lesquels l'utilisateur peut créer des applications. C'est le cas par exemple de Microsoft, avec Visual Basic, de Sysbase, avec PowerBuilder et de Borland avec Delphi.

Toutefois, cette intégration étroite entre les traitements métier et le client lourd peut provoquer des problèmes majeurs :

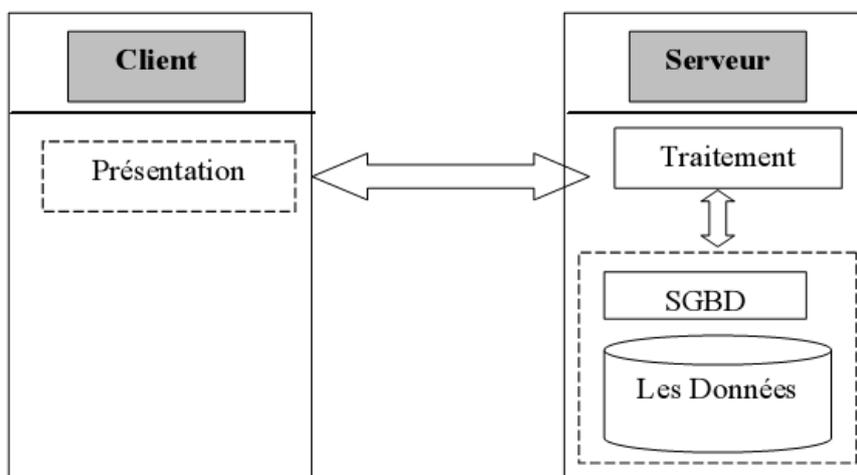
- L'architecture fait peser une charge de traitement considérable sur le client, ce qui signifie qu'il faut parfois des stations de travail avec un CPU puissant et une grande quantité d'espace disque et de mémoire ;
- Les requêtes de base de données peuvent générer des jeux de résultats volumineux ; si les utilisateurs sont nombreux, le trafic sur le réseau peut être sévèrement surchargé ;
- Chaque session ouverte à partir d'une station de travail nécessite une connexion de base de données distincte, ce qui peut absorber complètement les ressources du serveur de données ;
- Le déploiement de traitement métier sur le client peut induire des coûts de déploiement et d'assistance technique très élevés ; si la logique change, les modifications à effectuer pour mettre à jour les logiciels sur un grand nombre de stations de travail peuvent être extrêmement lourdes à gérer.

### 2.1.3.2. L'approche centrée sur le serveur

Il existe une autre implémentation du modèle client-serveur à deux niveaux : l'architecture client léger/ serveur lourd, suivant laquelle les traitements métier sont transférés sur le serveur de données à l'aide de techniques telles que des procédures stockées, des procédures événementielles et des contraintes (*Figure 6*).

Les utilisateurs envoient une demande à un serveur. Ce dernier renvoie la demande à l'application cartographique qui traite la demande et renvoie, par l'intermédiaire du serveur, les résultats inclus dans une page HTML. Toutes les applications ainsi que les données (spatiales et attributaires) restent sur le serveur.

Une procédure stockée est de la logique applicative qui peut être invoquée en envoyant au serveur de données une requête contenant le nom de la procédure stockée et des valeurs de paramètres. Cette requête est transmise à l'aide de protocoles de transports pour bases de données distantes.



**Figure 6 : Principe de fonctionnement d'une application cote-serveur**

Une contrainte est une restriction imposée à une donnée entrée dans une base de données. Une procédure événementielle est une procédure stockée qui est invoquée automatiquement lorsqu'un élément est ajouté, mis à jour ou supprimé dans une table. Ces deux éléments peuvent servir à valider les données et à garantir l'intégrité et la cohérence d'une base de données composée de plusieurs tables liées.

Les procédures stockées permettent de définir et de déployer les procédures de traitement métier dans l'ensemble de l'entreprise. Toutefois, les applications de bases de données à deux niveaux peuvent induire des coûts de développement et de maintenance élevés, dans la mesure où deux kits de développements sont nécessaires l'un pour le client et l'autre pour le langage de programmation des procédures stockées. Les langages de SGBDR étant généralement spécifiques au fabricant du SGBDR. Les concepteurs de bases de données se trouvent enfermés dans leur environnement d'autant que les possibilités en matière de flexibilité et de portabilité sont

extrêmement limitées. Ces langages nécessitent des compétences plus pointues, ce qui a une incidence directe sur les coûts.

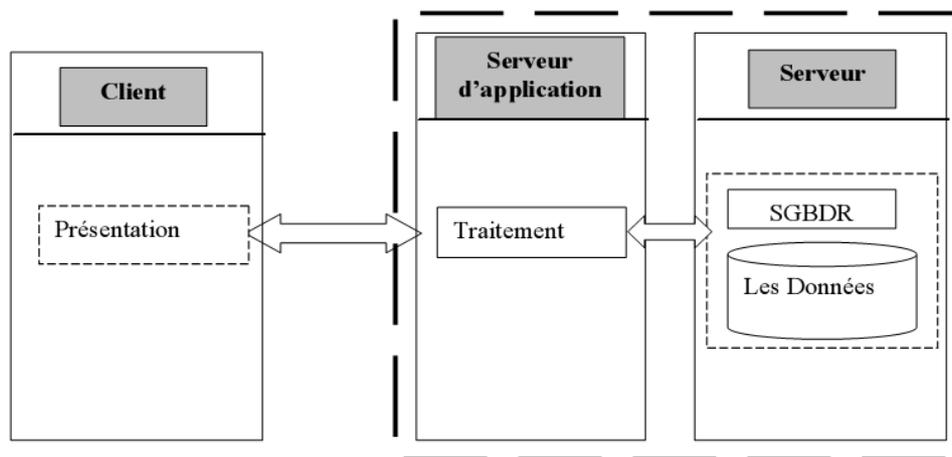
De surcroît, comme dans le cas des systèmes centrés sur le client, le fait que le SGBDR doit établir une connexion de base de données distincte pour chaque station de travail cliente peut avoir des conséquences lourdes pour les ressources du serveur. Aussi les systèmes centrés sur le serveur sont-ils inadaptés à une montée en charge importante, que ce soit au sein d'une entreprise ou sur l'Internet.

En pratique on constate que les performances de ces architectures à deux niveaux diminuent rapidement, dans la mesure où le réseau peut être saturé quand le nombre d'utilisateurs optimal est dépassé. En matière de flexibilité et de montée en charge, ces architectures n'offrent pas les performances requises pour les applications de grande ampleur déployées au sein d'une entreprise ou sur l'Internet.

#### **2.1.4. L'approche client-serveur à trois niveaux ou architecture 3-tiers**

Compte tenu des problèmes évoqués avec l'architecture client-serveur à deux niveaux, cette approche est aujourd'hui considérée comme insuffisante dès lors qu'elle dépasse le cadre d'applications destinées à un groupe d'utilisateurs restreints.

Il a été possible d'obtenir un gain de performances des applications et une réduction sensible du trafic des réseaux en implémentant un nouveau modèle : l'architecture client-serveur à trois niveaux. Cette architecture améliorée a pour particularité d'insérer un niveau supplémentaire entre les deux autres (*Figure 7*).



***Figure 7: Principe de fonctionnement d'une application client-serveur à trois niveaux***

Le client communique avec le niveau du milieu à l'aide de protocoles de communication (comme les protocoles TCP/IP). Le niveau du milieu joue le rôle d'interface, il offre une communication de messages de base, et contient les traitements métier de l'application. Il a pour rôle de :

- Réagir aux requêtes du client, d'appliquer les traitements métier et d'invoquer les requêtes de données ;
- Gérer les réponses de la base de données, d'appliquer des traitements métier supplémentaires, et de générer une réponse pour le client.

De plus, cette approche ne nécessite pas une connexion de base de données distincte pour chaque utilisateur .Elle permet au contraire d'ouvrir un grand nombre de sessions utilisateur avec un petit nombre de connexions de base de données, et préserve les ressources si précieuses du serveur de données.

Puisque l'architecture repose maintenant sur trois niveaux, la logique de présentation, la logique applicative (traitement métier) et la gestion de l'accès aux données sont parfaitement séparées. Il est possible d'améliorer ou de remplacer l'un des niveaux sans toucher les autres. Il est ainsi possible de perfectionner un système en le dotant de nouveaux canaux de transmission. Il est également possible de placer la logique applicative et la base de données sur la même plate-forme et de fait, il est parfaitement possible d'obtenir une solution optimale. Toutefois, pour que le système puisse être dit « à trois niveaux », il faut que des barrières ou des interfaces distinctes séparent les deux couches, sans quoi il n'est pas possible de bénéficier des avantages de ce type d'architecture.

Alors que les systèmes à deux niveaux ont parfaitement leur place dans le monde des applications simples, les systèmes client-serveur à trois niveaux sont aujourd'hui considérés comme la solution idéale pour l'entreprise car :

- Faciles à maintenir ;
- Mieux supportés et adaptés aux besoins évolutifs des entreprises.

### **2.1.5. Les applications Web**

La première utilisation importante du Web dans l'entreprise est apparue avec le développement des réseaux informatiques. Alors que l'Internet est un réseau global et ouvert à tous, l'Intranet est un réseau local fermé au sein de l'entreprise, contrôlant strictement l'accès des utilisateurs. L'intranet utilise au mieux les standards de l'Internet et les navigateurs familiers pour permettre aux utilisateurs d'accéder à l'information.

#### **2.1.5.1. Les applications Web classique**

Étant donné que les applications Intranet fonctionnent en mode client-serveur, l'entreprise n'a pas à déployer des logiciels clients ou à configurer les ordinateurs des utilisateurs (à l'exception des systèmes d'exploitation et des navigateurs). Sur un Intranet les utilisateurs ont le droit de se connecter sur un site Web en interne et d'accéder aux applications et documents sans avoir besoin d'installer ou de configurer quoi que ce soit. Si une application a besoin d'être modifiée, il suffit que le service informatique corrige la faille ou apporte des améliorations sur l'application dans le serveur et toutes les machines clientes bénéficient instantanément de ces modifications.

Cette distribution dynamique des applications présente l'avantage de faire réaliser des économies considérables aux organisations dotées des centaines de machines réparties dans l'entreprise ; et aux entreprises ayant plusieurs annexes sur des sites éloignés.

Les organisations utilisent également l'intranet pour faciliter la collaboration des employés et aider dans la recherche et le traitement des informations. Les navigateurs Web autorisent une approche opérationnelle et une visualisation de l'information cohérente de l'entreprise, quel que soit le format ou le type de données source.

### **2.1.5.2. Les applications Web de nouvelle génération**

On nomme maintenant souvent pages statiques les pages des débuts du Web : ces pages étaient en effet figées en un fichier au format .htm/.html, d'aspect plutôt austère, et dépourvues de la plupart des possibilités d'interaction, auxquelles l'utilisateur est désormais habitué avec les logiciels usuels pour PC.

La nouvelle génération d'applications Web a mis un terme à ces limitations, celles-ci autorisent désormais l'interactivité grâce à l'architecture de type client-serveur distribuée pour le traitement de l'information. Ce sont les navigateurs de la nouvelle génération qui ont permis cette évolution; ils peuvent désormais télécharger des composants logiciels, reconnaître les langages de scripts et gérer l'intégration avec les applications existantes.

Pour parer aux faibles performances liées à l'utilisation des techniques CGI classiques (Common Gateway Interface) (chapitre 4, 1. 7), d'autres technologies facilitant le développement d'extensions serveur ont été créés. Par exemple, plusieurs serveurs supportent l'interface ISAPI (Internet Server Application Programming Interface), laquelle permet aux composants exécutables de résider dans la zone de traitement du serveur Web et de n'être téléchargée qu'une seule fois, lors de la première demande.

Les composants insérés dans les pages Web peuvent être développés dans des langages de haut niveau tel que Java, Visual C++ et Visual Basic. Cela donne aux pages des fonctionnalités sophistiquées que le simple HTML ne permettait pas. De plus, ces composants distribués (également appelés composants répartis) ouvrent la voie vers un véritable environnement de traitement distribué.

### **2.1.5.3. Le Web à trois niveaux**

La forte demande de contenu dynamique a transformé le Web en une variante d'architecture multi-niveau particulièrement souple et indépendante du nombre d'utilisateurs sur le réseau. Il en résulte des applications plus faciles à maintenir et à supporter, tout en minimisant l'impact des modifications qu'il est nécessaire d'apporter à ces applications pour tenir compte de l'évolution des traitements métiers. L'application client est représentée par le navigateur. Elle est responsable de la logique de présentation définie par le document HTML, lequel peut inclure des scripts ou autres composants logiciels.

Le serveur Web correspond au niveau intermédiaire. Il est utilisé pour distribuer les données du client et intégrer les sessions clientes aux applications transactionnelles via CGI/ISAPI.

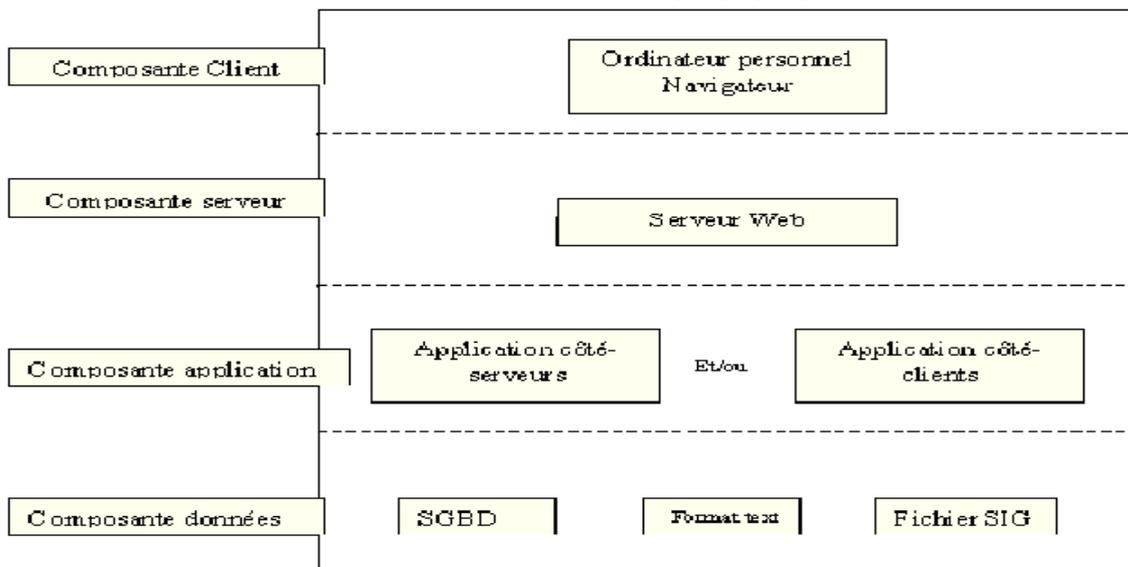
Les applications transactionnelles et la gestion de l'accès aux données doivent être modulées (ou sous forme d'exécutables) pour permettre la distribution des données sur plusieurs machines.

Cette architecture présente le grand avantage de résoudre plusieurs problèmes inséparables aux systèmes client-serveur traditionnels. En réduisant l'application client au HTML et à un langage de script simple, il devient possible de développer une application unique et universelle pouvant être déployée sur différents types de plates-formes (Windows, Mac, UNIX, LINUX et...). Tout le traitement côté client est administré de façon centralisé et déployé dynamiquement, ce qui signifie que toute mise à jour est appliquée automatiquement lorsque l'utilisateur démarre l'application. Cela évite une installation manuelle sur chaque poste client, qui est une opération très coûteuse si le nombre des utilisateurs est important.

### 2.1.6. Les applications SIG-Web

En projetant ces aspects sur les applications SIG sous une architecture Web; on peut identifier sur une application SIG-Web quatre composantes principales (Figure 6) qui sont :

- **Une interface client/navigateur** : se compose d'un ordinateur individuel et d'un navigateur. Cette couche fournit l'interface utilisateur et fonctionne en produisant des demandes au serveur par l'intermédiaire du protocole HTTP.
- **Un serveur Web** : la publication sur un réseau utilisant le protocole TCP/IP, oblige l'intégration d'un serveur Web. Cette couche se compose d'un PC équipé d'un logiciel appelé serveur Web. Cet intermédiaire permet la communication et le transfert de données contenues localement vers une autre machine connectée.
- **Une application cartographique** : deux types d'applications s'imposent ; l'une s'exécute du côté serveur et l'autre du côté client
- **La couche de données** : La donnée peut être gérée et stockée de plusieurs manières (grâce à un SGBD), sous forme de fichier SIG (données géographiques)...



**Figure 8: Les composantes d'une application SIG-Web**

### 2.1.7. Serveur Web

Un service Web est un logiciel, dont le rôle est d'écouter des requêtes d'un type particulier (requête HTTP) provenant de clients que l'on appelle navigateur. (Netscape Navigator, Internet Explorer, Mozilla Firefox ...).

Le terme serveur est généralement employé pour désigner une machine qui propose des services, et donc serveur Web désigne une machine qui fait tourner un logiciel de service Web.

Le protocole utilisé est HTTP (Hyper Text Transfer Protocole) qui permet au navigateur de demander à tout service Web de lui retourner un fichier stocké sur le serveur. La plupart du temps, ces fichiers sont au format HTML.

Le service Web doit être exécuté sur une machine qui possède une identification unique sous la forme d'une adresse IP, afin qu'un navigateur puisse localiser le service Web sans ambiguïté. Ce numéro (ou adresse IP) est codé sur quatre octets, (ex : 127.0.0.1). Bien que seul le numéro IP soit nécessaire, la machine possède généralement un nom enregistré dans un DNS (Domain Name System), qui permet de retrouver l'adresse IP à partir du nom. Pour l'utilisateur humain, il est en effet plus simple de se souvenir d'un nom que d'une adresse IP.

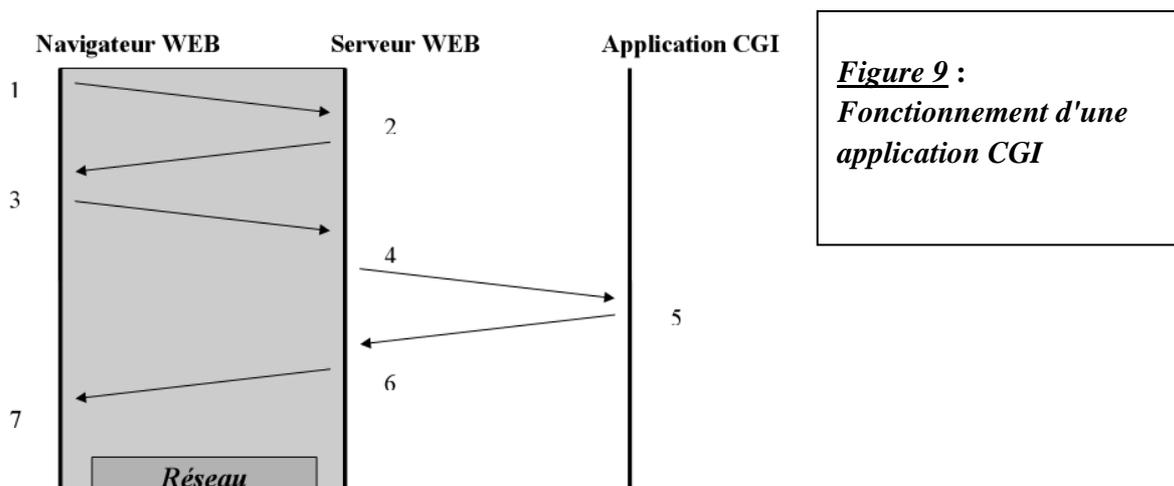
### 2.1.8. Les applications CGI

Les fonctionnalités des serveurs Web de première génération peuvent être étendues grâce à l'interface CGI (Common Gateway Interface) qui permet de faire générer dynamiquement le contenu des pages par des programmes généralement écrits en langage C, PHP ou Perl.

Cette technique permet de personnaliser le contenu des pages présentées à l'utilisateur en les construisant à partir d'informations provenant de bases de données ou d'autre application.

Grâce à CGI la construction des pages dynamiques est devenue possible. La construction des pages HTML est en fait le résultat de l'exécution d'un programme, et non la simple écriture statique de la page par un programmeur.

On peut illustrer le travail d'un programme CGI avec l'utilisation d'une page HTML sous forme d'un formulaire comme suite (Figure 9) :



1. L'utilisateur émet une requête de formulaire.
2. Le serveur établit le formulaire, et l'envoie à travers le réseau au client.
3. L'utilisateur remplit le formulaire, et le retourne au serveur.
4. Le serveur transmet le formulaire à l'application CCI.
5. L'application traite les données, et retourne le résultat au serveur.
6. Le serveur envoie le résultat au client.
7. Le client reçoit le résultat dans son navigateur.

## 2.2. LE WEBMAPPING

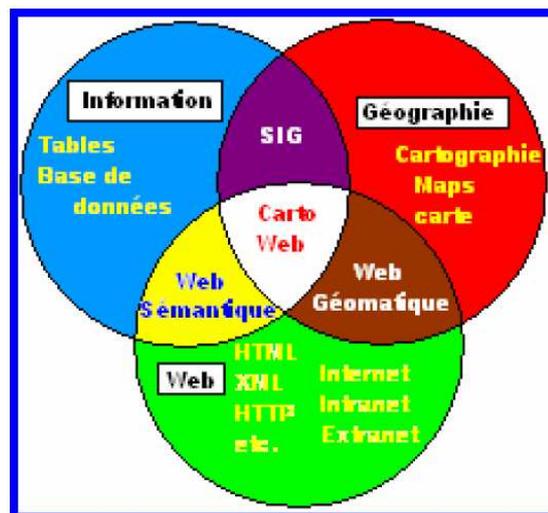
### 2.2.1. Introduction [11] [24]

Qu'on ait besoins de communiquer des informations dans les domaines de gestion des risques, de l'aménagement du territoire, ou bien encore de l'environnement, la publication de données cartographiques sur Internet est un moyen de communication indispensable.

Le Webmapping ou la diffusion des cartes via le réseau Internet, est un domaine en pleine expansion grâce au 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. Le concept de la cartographie Web consiste en trois composantes:

- la géographie,
- l'information,
- le Web (cf. la figure 10).

L'approche du Webmapping est adaptée à un public large non spécialiste : interface simple, orientée « atlas cartographique » et SIG en ligne, consultation par connexion rapide et solution indépendante des navigateurs, sans Plugin.



*Figure 10 : La trilogie du Webmapping*

Elle permet une intégration complète dans un site existant, une simplicité de mise en œuvre et des possibilités d'interactions : la surbrillance des entités, l'information complémentaire, l'affichage des thèmes, etc.

Les fonctionnalités cartographiques sont :

- **Navigation** : à l'intérieur d'un même document.
- **Vue Globale** : vue permettant de localiser la zone de visualisation sur un aperçu d'ensemble du document.
- **Zoom** : l'utilisateur a la possibilité de zoomer la taille de la zone de visualisation.
- **Icônes** : présence de symboles interactifs dans la visualisation.
- **Zones de clic** : zones interactives dans la visualisation déclenchant des événements (pop-up, info bulles, etc.).
- **Gestion de couches** : possibilité de visualiser plusieurs couches (calques).
- **Requêtes utilisateur** : Sémantiques et/ou géométriques, thématiques, prédéfinies ou non.
- **Mise à Jour** : interface de mise à jour des données spatiales et/ou attributaires sans programmation.

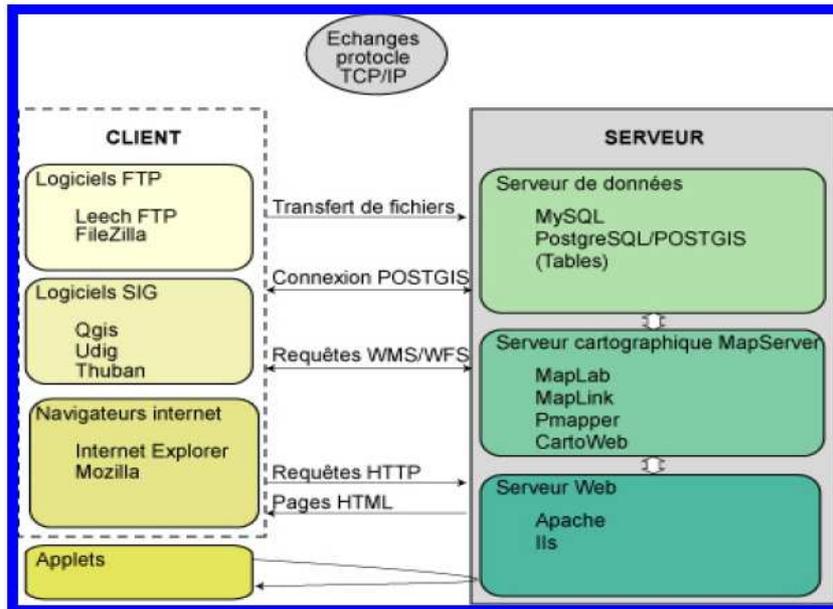
### **2.2.2. Principes de cartographie sur Internet [23]**

La solution la plus répandue actuellement dans le domaine de la mise en ligne de données cartographiques consiste à créer à la volée une image correspondant à la demande de l'utilisateur. Pour cela, il est le plus souvent fait appel à un serveur cartographique.

C'est le protocole de communication TCP/IP\* qui permet à des ordinateurs branchés en réseau d'échanger de l'information via un navigateur web ou de transférer des fichiers via le protocole ftp. L'architecture est de type client/serveur c'est-à-dire qu'il existe une série d'ordinateurs dits clients connectés à un serveur dédié qui lui-même communique vers l'extérieur (www) ou avec des serveurs particuliers par l'intermédiaire de leur matricule : l'IP.

L'utilisateur sur sa machine locale effectue des requêtes pour demander une carte spécifique ; le serveur cartographique interprète cette requête et renvoie la carte sous la forme d'une image matricielle (gif, jpg...) ou vectorielle (svg, flash).

Le serveur cartographique est télécommandé par des langages de script tels que php, javascript, Python ou Perl qui lui permettent de charger dynamiquement une carte en réponse à la requête. L'ordinateur serveur peut chercher cette information soit dans ses propres ressources, soit sur des serveurs de données distants.



**Figure 11 : Principes d'échanges entre un ordinateur client et un serveur**

La consultation de l'information requiert une installation essentiellement côté serveur avec des logiciels tels Apache ou IIS (Internet Information Services) qui tournent en tâche de fond et permettent aux serveurs de cartes d'accéder à l'Intranet et à l'Internet. Il faut aussi rajouter des interpréteurs de scripts comme php\_MapScript et éventuellement un Viewer (visionneuse en français) pour afficher la carte sur le navigateur (navigateur en français) du client.

La visionneuse peut être un applet\* ou un servlet. Dans le cas de l'applet, la visionneuse se télécharge côté client à chaque utilisation, dans le cas du servlet, elle s'exécute directement sur le serveur, la différence se situe au niveau de la rapidité de chargement et de la saturation potentielle des capacités du serveur.

Les requêtes sur un serveur cartographique peuvent être exécutées par le navigateur ou par un programme appelant, nous nous intéresserons dans ce rapport aux requêtes de type WMS/WFS définies par l'OpenGis Consortium (OGC) et utilisées par les serveurs cartographiques OpenSources tels que GeoServer, MapServer et Deegree. Ces moteurs cartographiques sont des programmes dont le rôle est de fabriquer à la volée, selon la demande de l'utilisateur des fichiers images représentant des données géographiques stockées sur le disque dur du serveur ou sur un autre serveur relié.

Au niveau du serveur de données, Les SGBDR tels PostgreSQL ou MySQL, entre autres, peuvent être installés directement sur le serveur contenant le serveur cartographique ou bien à distance. Qu'importe le lieu, l'important est de pouvoir consulter et éditer des données à distance. Nous avons opté pour PostgreSQL et son extension spatiale PostGIS car c'est le système de gestion de bases de données relationnelles le plus abouti dans le domaine du logiciel libre.

PostGIS offre des fonctionnalités approfondies pour traiter les objets géoréférencés et géométriques. En termes de richesse fonctionnelle, la solution PostgreSQL/PostGIS dépasse largement la dernière version de Mysql étendue spatialement, mais elle est plus complexe à mettre en œuvre. Cependant, un tutorial d'installation réalisé par Jean David Techer permet aujourd'hui

de compiler postgresql couplé de PostGIS sans trop d'encombres (<http://www.01map.com/david/doc/postgis090>).

### **2.2.3. Serveur cartographique [16][22]**

Les applications Web de visualisation de données cartographiques relèvent de l'approche client serveur. Cependant selon le rôle assuré par les deux parties (client, serveur) on parlera de :

**Coté serveur :** le client formule seulement la requête et c'est au serveur de la traiter et de préparer les résultats.

**Coté client :** le poste client est doté d'avantage de fonctionnalités locales de traitement. Dans ce contexte, l'analyse des architectures techniques distingue deux grandes familles de solutions : les serveurs de cartes et les teneurs de fichiers.

#### **2.2.3.1. Serveur des cartes**

L'intégralité des traitements est effectuée coté serveur pour envoyer des compositions cartographiques « prêtes à l'emploi » au poste client dont les tâches se limitent à la visualisation ou de navigation simple. Elle nécessite un développement applicatif important sur le serveur.

L'architecture logicielle s'organise autour d'un serveur de données et d'un serveur de cartes et nécessite un dialogue avec les clients au travers d'un serveur Web. Ce dialogue peut parfois nécessiter le chargement d'une application sur le client (Plugging, Applet Java...) qui transmettra les ordres au serveur de cartes. Ces produits sont soit issus des grands éditeurs commerciaux (ESRI avec Arc IMS, CLARITAS avec MapXstream Java...) soit issus du monde du logiciel libre (MapServer...).

#### **2.2.3.2. Serveur de fichiers**

On associe au poste client des fonctionnalités de traitement (plugging, applet Java JavaScript...) pour effectuer le traitement local sans qu'il soit nécessaire pour le serveur de ré-fabriquer une nouvelle carte lors de chaque modification du contexte d'affichage par l'utilisateur. Le serveur prépare des fichiers d'échange de données géographique (en format vecteur ou raster), qui vont être transmis au client en vue de traitements locaux opérés sur tout ou partie des objets transférés. Les traitements sont effectués à partir des instructions encodées dans les pages HTML.

### **2.2.4. Principe général de fonctionnement d'un serveur cartographique**

Le serveur cartographique est le guichet automatique auquel on fait appel pour afficher des cartes sur son poste de travail. Par le protocole de communication Internet, TCP/IP, des ordinateurs branchés en réseau peuvent échanger des données via un navigateur Web ou transférer des fichiers grâce au protocole FTP. L'architecture est de type client/serveur. L'utilisateur, à partir

de son terminal, lance des requêtes pour demander l'affichage d'une carte particulière. Le serveur cartographique l'interprète et renvoie la carte sous la forme d'une image matricielle (png, jpg,...) ou vectorielle (svg, swf,...). 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 de l'utilisateur.

Le serveur cartographique 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 des données 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 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 par un viewer (visionneuse en français), pour afficher la carte (*cf. figure 11*).

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. PostgreSQL et son extension spatiale PostGIS est le système de gestion de bases de données relationnelles le plus abouti dans le domaine du logiciel libre.

PostGIS est désormais inclus dans la distribution officielle de PostgreSQL. Il offre des fonctionnalités approfondies pour stocker et traiter les objets spatiaux géoréférencés. Les différentes briques logicielles ont été testées sous les systèmes d'exploitations Windows XP pour évaluer leur interopérabilité. L'utilisateur disposera des permissions de lecture et d'écriture dans l'arborescence des répertoires de la machine serveur, pour le stockage des solutions côté serveur tandis que les solutions côté client seront placées sur la machine locale dans un environnement Windows XP pro.

### **2.2.5. Fonctionnalités des serveurs cartographiques**

La plupart des serveurs cartographiques envoient une image morte correspondant à la carte demandée ; Cette solution peut être perçue positivement : la donnée ne peut pas être réutilisée, on peut parler d'une forme de sécurité. A l'inverse, elle présente des inconvénients :

- Qualité et temps d'affichage qui peuvent être médiocres.
- Pas de possibilité de mise à jour des données géographiques ou attributaires.

Cependant les serveurs cartographiques permettent au minimum :

- L'affichage d'un jeu de données géoréférencées (avec gestion de différents systèmes de projection) aux formats propriétaires de l'éditeur de logiciel, voir aux formats des concurrents ;
- La sélection des données à afficher ou/et à masquer ;
- La navigation au sein du jeu de données : zoom, déplacement, vue globale ;

- L'interrogation des attributs correspondant à une sélection ;
- La localisation par requête attributaire ;
- La fourniture d'une fiche de renseignement par requête attributaire ;

Sur certains serveurs cartographiques on trouve des fonctionnalités avancées :

- Mise à jour à distance des données (attributaires, spatiales) ;
- Gestion de l'administration des données pour les modifications à distance ;
- Ecriture de remarques textuelles géoréférencées conservées par le serveur et donc partagées ;
- Réalisation des opérations d'analyse spatiale et thématique ;
- Téléchargement des données visualisées ;
- Extraction de données.

## **2.2.6. Les avantages et les inconvénients des serveurs cartographiques**

Quelque soit le type de solution adoptée, tous les moteurs cartographiques présentent les avantages et les inconvénients répertoriés en dessous :

### **2.2.6.1. Avantages**

- Utilisation du jeu de données géoréférencées existant ;
- Facilité de mise à jour du jeu de données ;
- Diffusion dans un réseau (Intranet, Internet) ;
- Utilisation des langages standards en général (Asp, Php, Java, JavaScript, VbScript, C++, Perl...);
- Utilisation de formats d'image standard (tiff, jpeg, png...) ;
- Fonctionnalités proche du SIG bureautique ;
- Sécurité des données en cas de transfert de la carte sous forme d'image.

### **2.2.6.2. Inconvénients**

- Représentation par couche difficile à gérer au niveau visuel ;
- Bande passante importante ;
- Nécessité d'héberger l'application (installer le serveur spatial) ;
- Structure complexe ;
- Développement lourd ;
- Espace disque d'hébergement nécessaire pour les données notamment en format raster ;
- Dans le cas d'affichage en image, on ne peut pas afficher de grandes fenêtres cartographiques (images générées trop lourdes sur le réseau) ;
- Peu de souplesse dans l'enrichissement des fonctionnalités d'interaction entre la carte et l'application cartographique sur le serveur WEB (API proposée trop complexe).

## **2.2.7. Les serveurs cartographiques existants**

### **2.2.7.1. Les éditeurs classiques**

Les grands éditeurs de logiciels SIG ont presque tous lancé un produit serveur cartographique. Ce dernier présente les avantages et les inconvénients suivants :

#### **a. Avantage**

- Lecture native des formats SIG de l'éditeur, voire des concurrents ;
- Proposent des interfaces graphiques pour la création des sites cartographiques (facilité de création apparente) ;
- Formation initiale rapide (possibilité de formation payante chez l'éditeur) ;
- Fonctionnalités très avancées, y compris la mise à jour. Nécessite néanmoins un client spécifique dans certains cas ;
- Proposent des API sous Windows pour le développement.

#### **b. Inconvénients**

- Le prix élevé ;
- Coût des mises à jour ;
- Nombre de clients simultanés limités contractuellement ;
- Parfois des surcoûts pour lire les formats concurrents ;
- Pas toujours compatible avec tous les explorateurs clients (les navigateurs) ;
- Usage de plugins ou de clients spécifiques parfois.

Les produits existants : Ils existent plusieurs produits mais les plus connus sont :

- Mapguide d'Autodesk.
- ApicWeb d'Apic.
- GéoConcept web de Géoconcept.
- Arclms d'ESRI.
- Push'and'See couple à MapXtreme de Claritas.
- WebMap d'Intergraph.
- Toponet de SIRAP.
- STAR NeXt de Star Informati

### **2.2.7.2. Les solutions Open Source**

Issus de milieux universitaires et améliorés par des communautés de développeurs volontaires. Les serveurs cartographiques Open source visent à apporter les mêmes fonctionnalités proposées par les éditeurs classiques, mais avec une philosophie différente.

Les améliorations et les développements spécifiques à chaque système peuvent être réalisés en interne par les administrateurs et techniciens ou en externe par de sociétés de Services en Logiciels Libres, des Sociétés de Services en Ingénierie Informatique ou bien par des sociétés spécialisées. Les applications open source et l'accès à leur code source permettent de créer librement des outils dérivés et des fonctionnalités supplémentaires. De plus, la licence GNU/GPL donne un droit de redistribution à la personne qui a modifié les scripts d'un programme.

Ils présentent les avantages et inconvénients suivants :

#### **a. Avantages**

- Gratuité immédiate ainsi que la mise à jour ;
- Prise en compte de la compatibilité ascendante ;
- Disponibilité de nombreux langages de programmation (Java, Php, Perl,...) ;
- Utilisation de nombreux formats ouverts dont les spécifications ont été diffusées (Shapefile, géotiff, tab Mapinfo...), gestion de nombreuses bases de données différentes (Oracle, Mysql, Postgresql...) ;
- Fonctionnement avec un maximum de navigateurs clients ;
- Grande communauté d'utilisateurs avec partage des travaux de chacun.

#### **b. Inconvénients**

- Nécessite un personnel motivé et auto-formé ;
- Fonctionnalités un peu moins avancées ;
- Documentation technique souvent anglaise ;
- Difficulté d'installation ;
- Gros temps d'investissement humain.

Produits existants : les serveurs cartographiques les plus connus sont :

- MapServer.
- GéoServer.

Plusieurs produits similaires sont disponibles dans le site [www.freegis.org](http://www.freegis.org).

On peut récapituler cette étude comparative entre les solutions Open Source et les solutions propriétaires dans le tableau ci-dessous (*Tableau 3*) :

<b>Critères</b>	<b>Open source</b>	<b>Editeur classique</b>
<b>Prix</b>	Gratuit	Elevé
<b>Format lu</b>	Formats ouverts dont les spécifications ont été diffusées (Shapefile, géotiff, tab Mapinfo...), nombreuses bases de données différentes (Oracle, Mysql, Postgresql...);	Formats SIG de l'éditeur, voir des concurrents
<b>Fonctionnalité</b>	Fonctionnalités un peu moins avancées	Fonctionnalités très avancées, y compris la mise à jour
<b>Développement</b>	Utilise des langages de programmation (Java, Php, Perl,...)	Rapide avec des DLL ou API (composants logiciels prédéfini)
<b>Formation</b>	Nécessite un personnel motivé et auto-formé	Formation initiale rapide

**Tableau 3** : Tableau comparatif entre les solutions Open Source et les produits propriétaires

### **2.2.8. Les infrastructures et les logiciels utilisés**

Toutes les solutions (éditeurs de logiciels SIG ou Open Source) nécessitent au minimum :

#### **2.2.8.1. Côté serveur**

- Serveur Web : IIS ou Apache sous environnement Windows, Unix ou Linux.
- Installation du moteur cartographique (serveur spatial) par programme d'installation ou par compilation.
- Localisation d'un jeu de données sur le serveur Web.
- Hébergement des données.

#### **2.2.8.2. Côté client**

##### **❑ Les solutions des grands éditeurs :**

Ne fonctionne pas avec tous les navigateurs donc nécessitent des clients spécifiques pour les fonctionnalités les plus avancées. Le client spécialisé peut créer un cache des données sur le client (accélère le fonctionnement, utilisation d'un espace disque côté client, confidentialité des données).

##### **❑ Les solutions open source :**

Fonctionnent avec tous les explorateurs Web, pas d'explorateur spécifique pour les fonctionnalités avancées.

L'essentiel du travail se fait côté serveur. Il faut donc une machine puissante et une large bande passante. Si le jeu de données est lourd; il faut aussi prévoir un hébergement de grande capacité. Dans le cas des interfaces spécifiques, le travail s'effectue davantage côté client (gain de temps).

### **2.2.9. Les outils de développement Web**

Les mêmes éditeurs proposent souvent des environnements de développement vous permettant de créer des applications SIG métier, mais aussi des applications Web. Ces outils sont destinés aux développeurs et permettent de répondre à des demandes très spécifiques. Ils présentent cependant les avantages et les inconvénients suivants :

#### **❑ Avantages :**

- Exploitation des bibliothèques de fonctions des logiciels SIG du même éditeur ;
- Possibilité de personnalisation ;
- Utilisation de plusieurs langages différents envisageables (Java, Visual Basic, C++...);
- Multi-usage, pas uniquement Web (Bureautique, Réseau, ...).

#### **❑ Inconvénients :**

- Le prix ;
- Environnements de fonctionnement parfois limité (système d'exploitation, API...);
- Nécessité personnel très qualifié et formé ;
- Manque de flexibilité des bibliothèques de fonctions.

Dans le marché, on peut trouver quelques outils de développement, par exemples :

- MapXtreme de Claritas.
- MapObjets et Internet map server d'ESRI.

### **2.2.10. Mise en place d'une application Web**

La mise en place dépend de plusieurs facteurs à savoir :

- Les objectifs de l'application Web ;
- La conception même de l'application Web ;
- La détermination et la nature des utilisateurs ;
- La préparation des types de données disponibles ;
- La sécurité et les droits d'accès sur ces données.

Suivant ces facteurs suscités, On peut distinguer deux cas :

#### **❑ Cas de site statique simple :**

Quelque soit la solution, la mise en place sera rapide. La différence se situe beaucoup plus au niveau de la ressource humaine. Selon que l'on adopte une solution issue des grands éditeurs ou open source, le profil est très différent :

- Avantage fonctionnalités de l'interface: (grands éditeurs).
- Avantage clients potentiels (navigateurs): (Open source).

#### **❑ Cas de site dynamique complexe :**

L'interface est esthétique et quel que soit la solution, il faut une ressource humaine très qualifiée en développement réseau (langages HTML, PHP, ASP, JAVA...). Le temps de mise en place est

beaucoup plus long car il nécessite un travail en équipe, au minimum un développeur Web et un géomaticien / cartographe.

## **Conclusion**

La cartographie sous Web est une solution réseau déployée sous une architecture de type client-serveur ; et nécessite notamment du côté serveur des composants logiciels de métier permettant la réalisation de fonctionnalités de gestion, traitement et stockage des données à référence spatiale. Pour implémenter une telle solution on fait appel aux différentes solutions logicielles disponibles en occurrence les solutions proposées par les éditeurs SIG classiques ou bien les solutions Open Source. En raison des contraintes du cahier des charges et des avantages intéressants que présentent les solutions open source, notre système sera développé autour des composants du monde open source. La conception et l'implémentation de ce type d'application seront détaillées dans le prochain chapitre.

## **Chapitre.3. : DEVELOPPEMENT DU SYSTEME**

### **3.1. Introduction**

Après avoir présenté l'état de l'art des SIG-Web et particulièrement les architectures sur lesquelles ils sont bâtis, il est temps de se consacrer au développement de l'application projetée. Le développement qui suit un processus (définissant le cycle de vie de l'application) dont les étapes sont définies et présentés ci-après ont été respectées tout le long de la construction de l'application. Pour ce faire, il a été nécessaire de comprendre les besoins de l'utilisateur afin de les transcrire correctement en entités informatisables. Cela a été rendu possible grâce au formalisme UML (Unified Modeling Language) en raison de sa qualité de modélisation à objet et puisqu'il offre un formalisme relativement simple, compréhensible aussi bien par les informaticiens que par les non informaticiens, imposant toute la rigueur nécessaire à la conception du système.

### **3.2. Définitions [14]**

UML est un ensemble de modèles permettant de représenter un système informatique et son utilisation prévue dans l'entreprise. L'usage de ces modèles par des informaticiens vise à améliorer la qualité des applications informatiques qu'ils développent.

UML est un langage standard conçu pour l'écriture de plans d'élaboration de logiciel. Il peut être utilisé pour visualiser, spécifier, construire et documenter les artefacts d'un système à forte composante logicielle.

Un modèle est une simplification de la réalité. Ce peut être un plan détaillé, mais également un plan plus général qui donne une vue d'ensemble du système concerné. Un bon modèle inclut des éléments qui revêtent une grande importance et laissent de côté ceux qui sont inutiles au niveau d'abstraction choisie. Tous les systèmes peuvent être décrits sous différents aspects, au moyen de plusieurs modèles, mais chaque modèle constitue ainsi une abstraction du système, fermée du point de vue sémantique. Ils peuvent être structuraux et souligner l'organisation du système, ou comportementaux et mettre l'accent sur sa dynamique.

### **3.3. Pourquoi modéliser ? [14]**

Les modèles permettent de mieux comprendre le système que l'on développe. La modélisation permet d'atteindre quatre objectifs :

- Les modèles nous aident à visualiser un système tel qu'il est ou tel que nous voudrions qu'il soit ;
- Les modèles permettent de préciser la structure ou le comportement d'un système ;
- Les modèles fournissent un canevas qui guide la construction d'un système ;
- Les modèles permettent de documenter les décisions prises ;

En informatique, la modélisation peut être abordée sous plusieurs angles, les deux principaux sont la perspective algorithmique et la perspective orientée objet.

En règle générale, le développement s'appuie sur une approche algorithmique, dans laquelle les briques principales sont les procédures ou les fonctions. Cette perspective incite les développeurs à se concentrer sur les questions de contrôle et de décomposition des gros algorithmes en procédures plus petites. Ce point de vue n'est pas mauvais si ce n'est qu'il a tendance à engendrer des systèmes fragiles. Au fur et à mesure que les besoins et le système évoluent, ce qui est inévitable, les logiciels centrés sur les algorithmes deviennent très difficiles à mettre à jour.

Aujourd'hui, le développement d'un logiciel s'appuie sur une perspective orientée objet, dans laquelle les principales briques de base sont l'objet ou la classe. Pour simplifier, un objet est un élément généralement tiré du vocabulaire de l'espace problème ou de l'espace solution et une classe est la description d'un ensemble d'objets communs. Chaque objet a sa propre identité, un état (des données associées) et un comportement (il peut subir des transformations et il peut faire subir des transformations à d'autres objets).

En informatique, l'approche orientée objet est largement adopté, tout simplement parce qu'elle a démontré son efficacité lors de la construction de système dans les domaines métier les plus divers et qu'elle englobe toutes les dimensions et tous les degrés de complexité. De plus, la plupart des langages, des systèmes d'exploitation et les outils de développement sont au moins partiellement orientée objet.

L'utilisation de UML permet de :

- Décomposer le processus de développement ;
- Mettre en relation les experts métiers et les analystes ;
- Coordonner les équipes d'analyse et de conception ;
- Séparer l'analyse de la réalisation ;
- Prendre en compte l'évolution de l'analyse et du développement ;
- Migrer facilement vers une architecture objet.

UML n'est pas un langage de programmation visuel, ses modèles peuvent être directement traduits dans différents langage de programmation (Java, C++) voire de l'exprimer à l'aide de tables dans des bases de données relationnelles ou dans la mémoire persistante d'une base de données orientée objet.

### **3.4. Les éléments d'UML**

Brièvement voici les éléments les plus importants dans le formalisme UML :

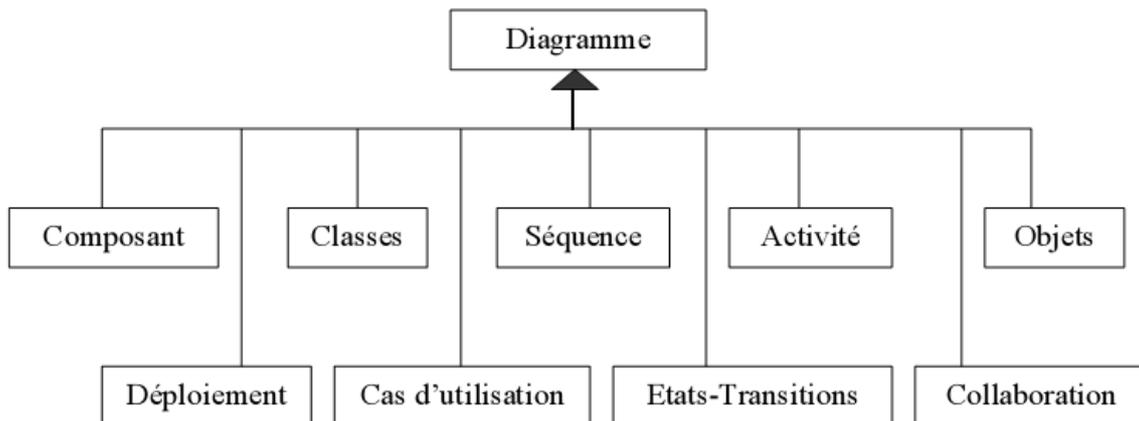
1. Les éléments structurels
  - Une classe
  - Une interface
  - Une collaboration
  - Un cas d'utilisation
2. Les éléments comportementaux
  - Une interaction
  - Un automate à états finis
3. Les éléments de regroupement

- paquetage

#### 4. Les éléments d'annotation

### 3.5. Les diagrammes d'UML [14]

Un diagramme est la représentation graphique d'un ensemble d'éléments qui constituent un système. Il contient les éléments de visualisation qui représentent des éléments de modélisation et leurs caractéristiques.



*Figure 12 : Différents diagrammes définis par UML*

### 3.6. Les étapes du cycle de vie d'une application

Le cycle de vie d'une application passe par les étapes suivantes [MOREL, 1998] :

#### 1) L'expression des besoins

Cette première phase se focalise sur les propriétés externes du logiciel, à savoir :

- Les besoins de l'utilisateur.
- Comportement du système face à l'utilisateur.

#### 2) Les spécifications du système

- Cette phase décrit ce que le système doit être et comment il peut être utilisé.

#### 3) L'analyse

L'objectif est de déterminer les éléments intervenant dans le système à construire, ainsi que leur structure et leurs relations.

Elle doit décrire chaque objet selon trois axes :

- Axe fonctionnel : savoir-faire de l'objet.
- Axe statique : structure de l'objet.
- Axe dynamique : cycle de vie de l'objet au cours de l'application (Etats et messages de l'objet).

Ces descriptions ne tiennent pas compte de contraintes techniques (informatique).

#### **4) Conception**

Elle consiste à apporter des solutions techniques aux descriptions définies lors de l'analyse :

- Architecture technique,
- Performances et optimisation,
- Stratégie de programmation.

On y définit les structures et les algorithmes qui seront validés lors des tests.

#### **5) Implémentation**

Habituellement, cette étape est consacrée à la réalisation de la programmation. Dans notre cas, elle est caractérisée essentiellement par le recours aux outils open source. En plus de la réécriture d'un ensemble de scripts, il a été nécessaire d'effectuer un choix judicieux des composants, de les déployer et de les configurer. C'est la raison pour laquelle nous avons délibérément mis cette partie dans un chapitre à part (chapitre VIII).

#### **6) Tests de vérification**

Ils permettent de réaliser des contrôles pour la qualité technique du système. Il s'agit de relever les éventuels défauts de conception et de programmation (revue de code, tests des composants,...). Il faut instaurer ces tests tout au long du cycle de développement et non à la fin pour éviter des reprises conséquentes du travail (logiciels de tests).

#### **7) Validation**

Le développement d'une application doit être lié à un contrat ayant une forme de cahier de charges, où doivent se trouver tous les besoins de l'utilisateur. Ce cahier de charge doit être rédigé avec la collaboration de l'utilisateur et peut être par ailleurs complété par la suite.

- Tout au long de ces étapes, il doit y avoir des validations en collaboration également avec l'utilisateur.
- Une autre validation doit aussi être envisagée lors de l'achèvement du travail de développement, une fois que la qualité technique du système est démontrée. Elle permettra de garantir la logique et la complétude du système.

#### **8) Maintenance et évolution**

Deux sortes de maintenances sont à considérer :

- Une maintenance corrective, qui consiste à traiter les « buggs » ou les failles.
- Une maintenance évolutive, qui permet au système de répondre à de nouveaux besoins ou à des changements technologiques.

#### **9) Retrait**

C'est l'étape ultime du logiciel dans laquelle la décision de retrait de logiciel de l'exploitation est prise.

## 3.7. Développement du système

### 3.7.1. L'expression des besoins

Dans cette phase, nous décrivons globalement notre application et l'apport du futur système; afin de tirer les principales fonctions à développer en partant des besoins globaux des utilisateurs qu'on peut résumer comme suit :

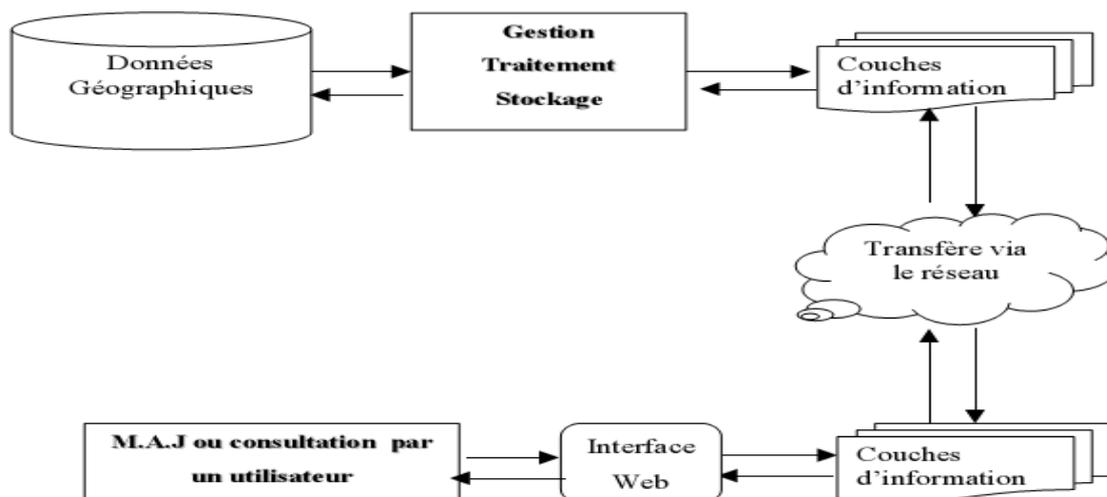
- Disposer d'un outil qui permet la mise à jour à distance des données existantes dans chaque couche thématique de la carte ;
- Assurer la cohérence des données par la centralisation de ces derniers dans un serveur localisé au FTM.
- Se doter d'une application qui servira comme un outil de collecte décentralisé des informations géographiques spécialement pour l'opération de complèment et la collecte de toponymies.
- Faciliter la tâche aux opérateurs habitués avec les logiciels SIG bureautiques en développant des interfaces Web similaires à celles des interfaces bureautiques classiques.
- Offrir un outil bien sécurisé avec des droits d'accès très restreint aux différents utilisateurs.
- Envisager également une utilisation de cette application pour des fins commerciaux (pour les différents produits cartographiques), gestion de personnels ou autres applications réseaux.

La satisfaction de ces besoins peut apporter un gain considérable à l'entreprise par la minimisation des coûts des différentes opérations qui entrent dans le processus de production et par l'amélioration de la gestion des ressources de l'entreprise.

### 3.7.2. Spécification des besoins

#### 3.7.2.1. Principe de fonctionnement du système

Dans cette phase, nous décrivons globalement notre système avec le schéma fonctionnel suivant :



*Figure 13 : Description globale du système*

Ce système est conçu essentiellement pour permettre la mise à jour des données spatiales et attributaires et la consultation à distance. Le transfert de données entre le serveur de données et les postes clients se fait via le réseau internet en utilisant les mécanismes et les protocoles adéquats, tout en respectant les régies de sécurité afin de protéger les données.

Les modifications des données seront immédiates dans la base de données, ces opérations, d'une part ne nécessitent pas un déplacement ce qui apporte un gain considérable à l'utilisateur, et d'autre part ils seront effectués sur des postes clients très légers (on n'a pas besoin d'installer un SIG bureautique pour faire une telle opération).

### **3.7.2.2. Fonctionnalité du système**

Nous distinguons deux fonctions principales à développer :

#### **1) Mise à jour des données spatiales à distance**

La mise à jour est la fonction essentielle du système ; Elle doit prendre en considération :

- La sécurité des données en appliquant un mécanisme d'authentification.
- L'interface homme-machine, qui doit être similaire à une interface de logiciel SIG bureautique.
- Le contrôle de qualité des données, notamment les données attributaires associés aux objets géométriques.

#### **2) Consultation**

Pareille pour la consultation ; cette fonction doit respecter les régies d'authentification ; en plus elle doit offrir tous les outils standards de visualisation (zoom in. zoom out et déplacement). Egalement elle donne aux utilisateurs la possibilité de visualiser les données attributaires et de gérer les différentes couches avec un simple click.

### **3.7.3. L'analyse**

#### **3.7.3.1. Cas d'utilisation du système [21]**

Les cas d'utilisation permettent d'effectuer une délimitation du système et de décrire son comportement. Ils constituent une représentation orientée "fonctionnalités " du système qui sont le résultat de la spécification.

Il existe deux concepts fondamentaux dans la modélisation des uses cases :

- Les acteurs : utilisateurs du système.
- Les uses cases : utilisation du système par les acteurs.

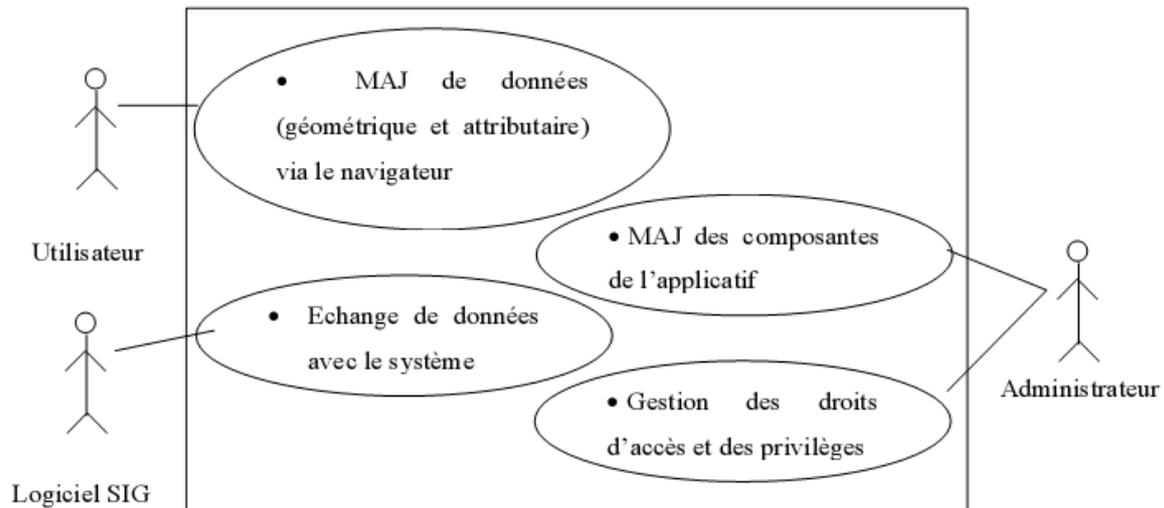
#### **1) Détermination des acteurs :**

Trois acteurs différents interviennent au niveau de notre système :

- a. L'utilisateur final de l'application : c'est la personne qui va utiliser l'application finale via un navigateur WEB standard donc c'est l'acteur primaire.
- b. L'administrateur : administre le système par le contrôle des différents droits d'accès. Ainsi que la mise à jour de l'application au niveau du serveur
- c. Les logiciels SIG bureautiques, fournissent les données géographiques ou autres.

## 2) Détermination des cas d'utilisations (*Figure 14*):

- L'utilisateur fait la mise à jour des données via le réseau (client Web) ; il peut également consulter les cartes on-line.
- L'administrateur est l'acteur manipulant le système d'une manière générale, en définissant les règles de sécurité, les différents utilisateurs, leurs droits d'accès et effectue des configurations concernant les différents serveurs intervenant dans l'application ainsi que la mise à jour des différentes composantes de l'appliquatif
- Le logiciel SIG échange les données avec le système.



**Figure 14 : Diagramme des cas d'utilisation**

Dans cette phase de développement, nous adoptons une démarche basée sur les points suivants :

- Identification des objets et classes ;
- Elaboration du diagramme de classe ;
- Description et représentation des scénarios ;
- Elaboration des diagrammes état-transition ;
- Elaboration des diagrammes de collaboration ;
- Consolidation et vérification des modèles (statique et dynamique).

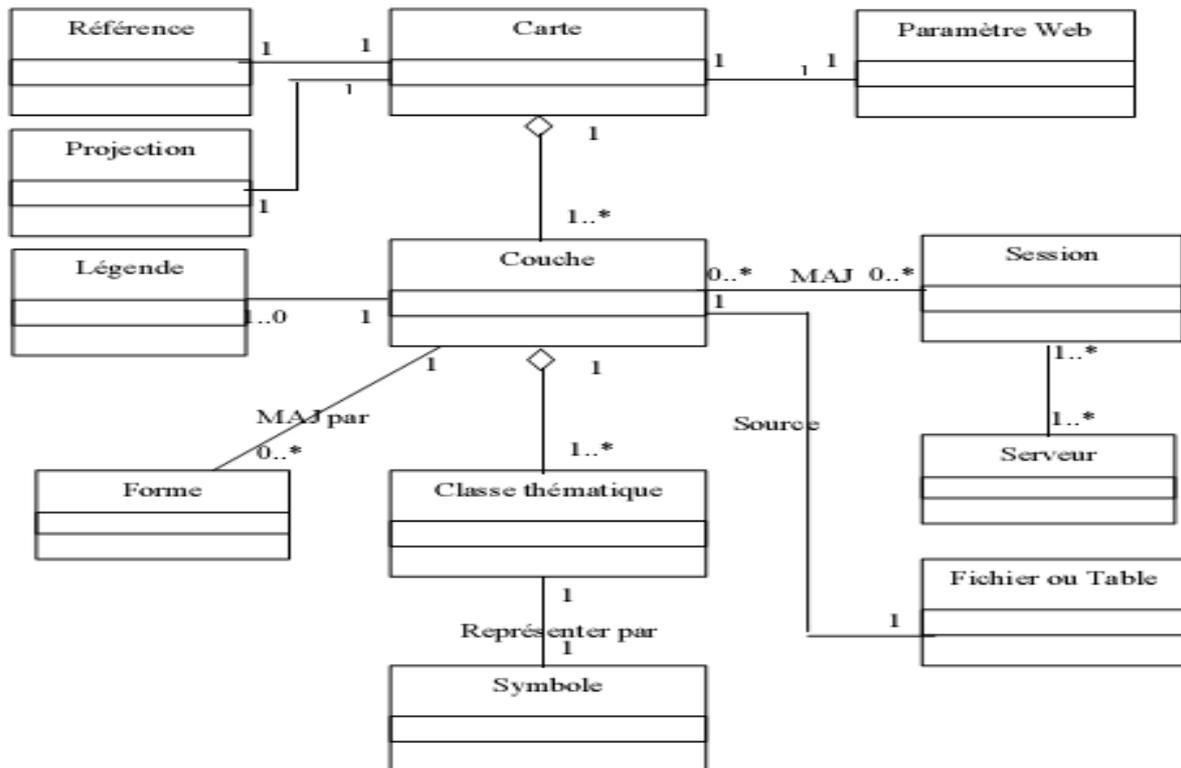
### 3.7.3.2. Le modèle statique

Il est utilisé en phase d'analyse sans rapport avec l'implémentation.

#### 1) Diagramme de classes (*Figure 15*) [14] [25]

Un diagramme de classes est une collection d'éléments de modélisation statiques qui exprime de manière générale la structure statique d'un système, en tenue de classes et de relations entre ces classes. Il montre les aspects statiques du modèle et fait l'abstraction des aspects dynamiques ou temporels.

Les attributs d'une entité n'ont pas de type en rapport avec un langage de programmation. Les cardinalités représentent le nombre d'instances impliquées dans l'association. Le diagramme de classe, ci après représente la structure statistique de notre système.



**Figure 15 : Diagramme des classes**

### 3.7.3.3. Le modèle dynamique

Le modèle dynamique est une vision microscopique du fonctionnement du système. Il sert à mettre en évidence les relations temporelles inter-objets et la représentation sous forme d'un automate du comportement de chaque objet. Il intervient après la définition du modèle statique. Il comprend 03 diagrammes : les scénarios, les états-transitions et le diagramme de collaboration.

#### 1) Les scénarios [14]

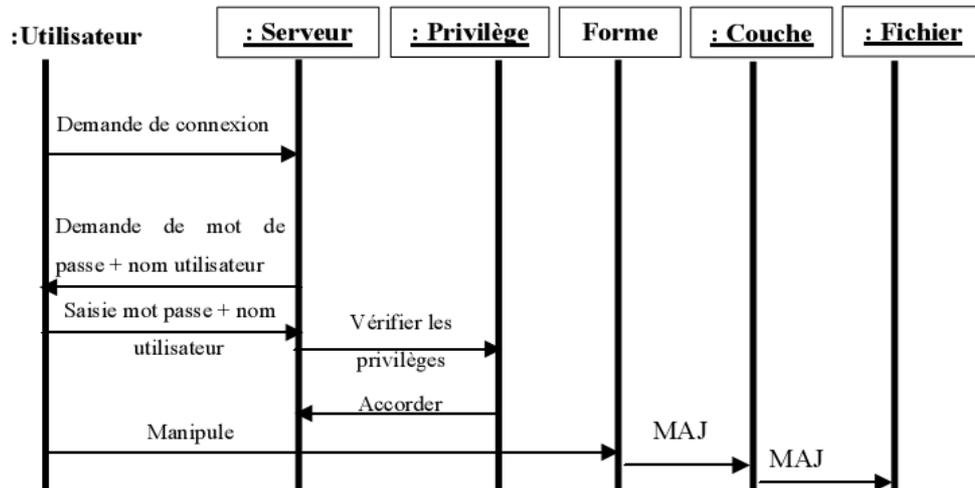
Un scénario est une série d'événements ordonnés dans le temps, simulant une exécution particulière du système. Ce diagramme met en évidence les relations temporelles inter-objet. Un scénario utilise 02 types de concepts :

- des objets faisant partie du système et des objets externes au système ;
- des événements émis et reçus par les objets impliqués dans le scénario.

Il permet d'apporter une meilleure compréhension de l'enchaînement des événements et des messages dans le système et ainsi de comprendre les rôles joués par les objets du système vis-à-vis des autres objets.

Pour identifier les principaux événements impliqués, nous avons choisi d'expérimenter deux scénarios :

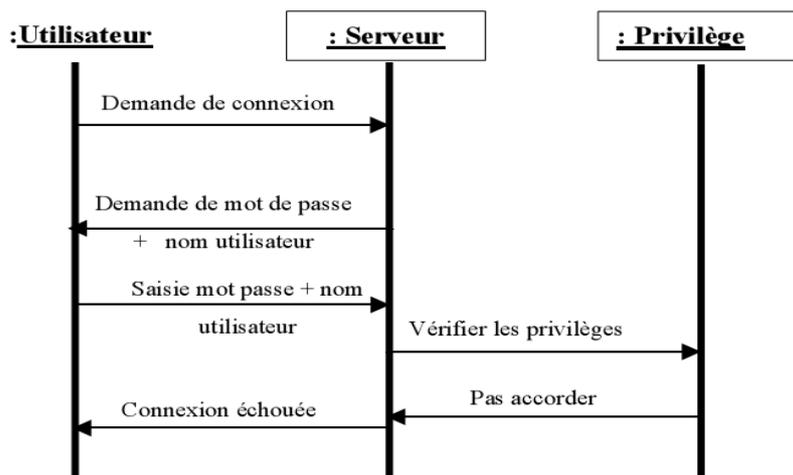
- Connexion réussie + MAJ d'une couche (*Figure 16*) ;
- Connexion échouée (*Figure 17*).



***Figure 16 : Scénario d'une simple connexion utilisateur***

Le scénario «Connexion réussie + MAJ d'une couche » s'exécute suivant la séquence ci-après :

1. L'utilisateur établit une demande de connexion au serveur par une simple connexion http via le réseau ;
2. Le serveur invite l'utilisateur à introduire le nom d'utilisateur et le mot de passe dans un masque de saisie ;
3. L'utilisateur introduit ses informations et les envoie vers le serveur ;
4. Le serveur vérifie ces informations avec la base qui emmagasine les droits et les privilèges d'accès. A l'issue, le système donne l'accord à l'utilisateur ;
5. En manipulant des objets (forme) géométrique, l'utilisateur peut mettre à jour une ou plusieurs couches ;
6. Ces modifications seront enregistrées dans le fichier.



**Figure 17 : Scénario d'une connexion échouée**

Le scénario « connexion échouée » se déroule de la même façon que le scénario « Connexion réussie + MAJ d'une couche » jusqu'à l'étape 4, Après la réception des formations parvenues de l'utilisateur, le serveur procède à une vérification dans sa base ; si la vérification n'est pas réussie (après un nombre bien défini de tentatives) le serveur va interrompre la connexion.

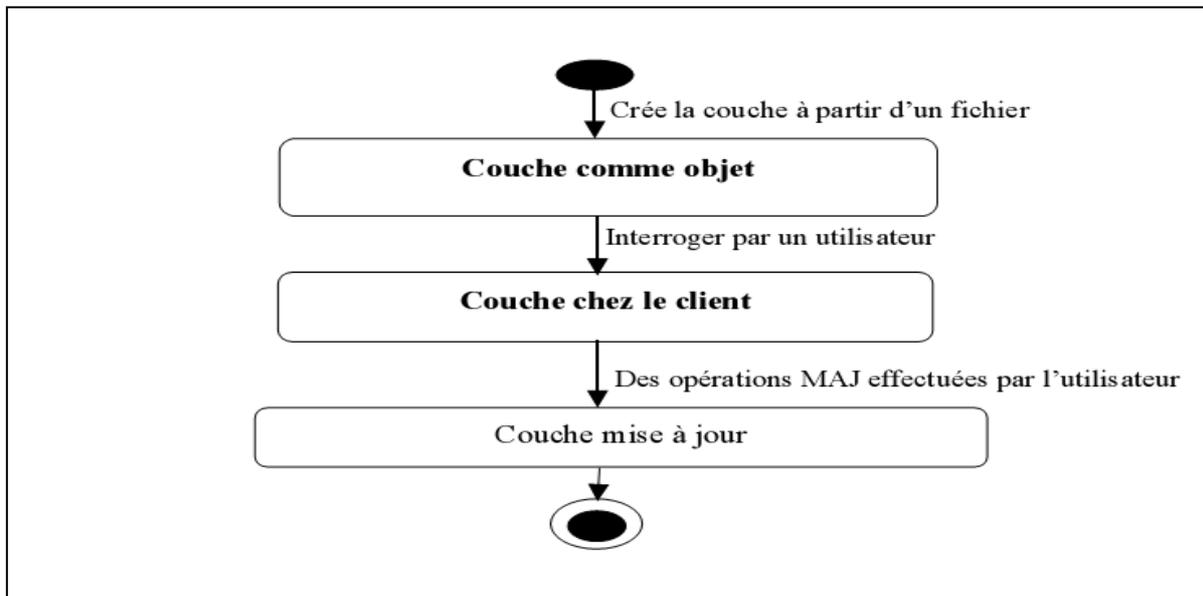
## 2) Diagramme d'états -transitions [12]

C'est un graphe composé de nœuds représentant les états d'un objet reliés par des arcs orientés qui décrivent les transitions. Il découpe un objet en un ensemble d'états. Le passage d'un état à un autre se fait par des événements.

Un diagramme d'états-transitions permet de décrire les changements d'états d'un objet ou d'un composant ayant un comportement remarquable, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.

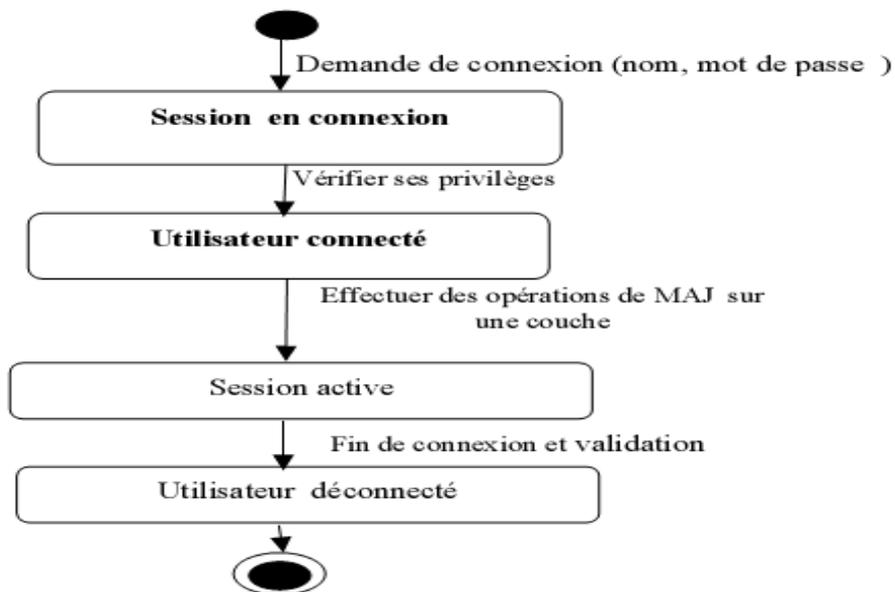
Les diagrammes ci-après (Figure 14 et 15) présentent respectivement les diagrammes états-transition des objets « couche » et « utilisateur »

- Diagramme d'états-transitions d'objet couche indique les différents états de cet objet pendant sa durée de vie.



**Figure 18 :** Diagramme d'états-transitions pour objet « couche »

Diagramme d'état transition d'objet session est le suivant :

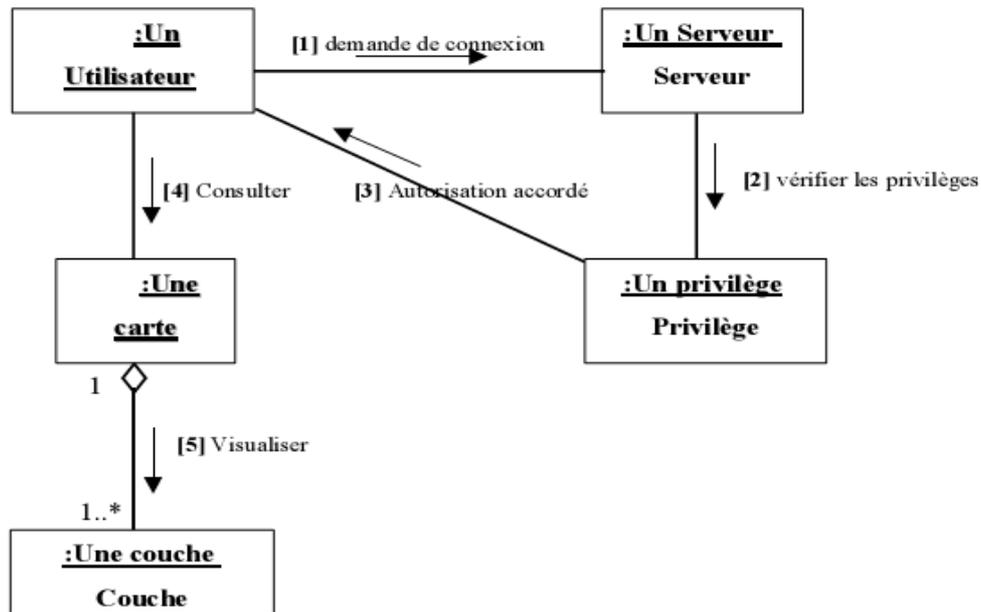


**Figure 19 :** Diagramme d'états-transitions pour objet « utilisateur »

### 3) Diagramme de collaboration [14]

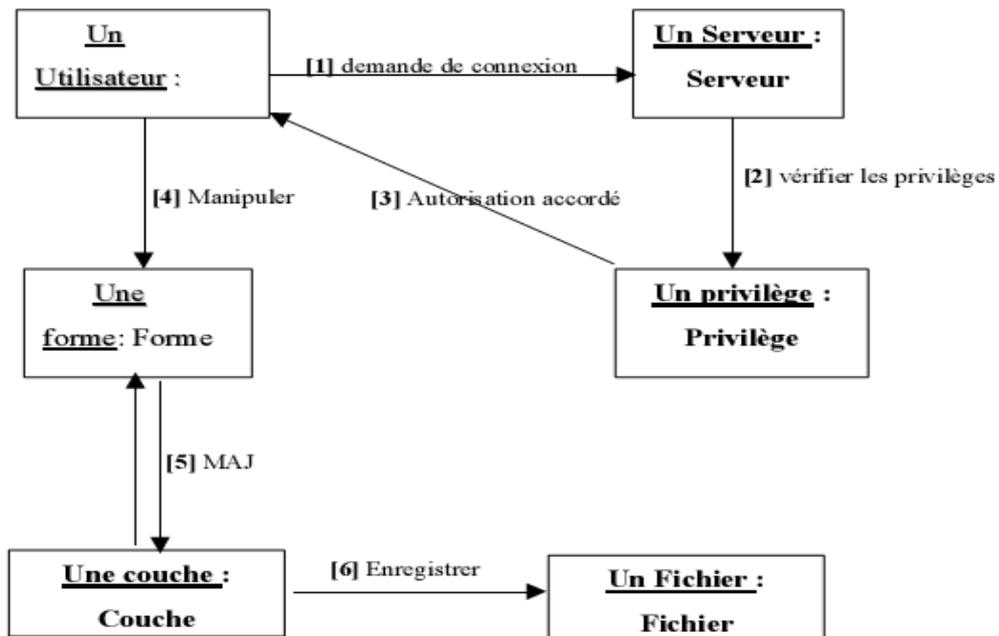
Un diagramme de collaboration entre objets vise à représenter du point de vue statique et dynamique les objets impliqués dans la mise en place d'une fonction de l'application. Nous avons choisi de représenter les deux grandes fonctions de notre système qui sont :

1. la consultation via un browser (Figure 16) ;
2. MAJ des données attributaires et géométrique (Figure 17).



**Figure 20 : Diagramme de collaboration de la fonctionnalité : consultation**

Le contexte de ce diagramme de collaboration fait intervenir l'objet carte qui se compose en plusieurs couches. L'objet carte va être consulté par un utilisateur, ce dernier était l'objet d'une authentification faite par le serveur par un système adéquat.



**Figure 21 : Diagramme de collaboration de la fonctionnalité : Mise à jour**

Le contexte de ce diagramme de collaboration fait intervenir l'objet forme qui sera manipulé par l'objet utilisateur afin de mettre à jour l'objet couche. Ces modifications seront enregistrées dans un fichier correspondant à la couche.

### 3.7.4. La conception

Elle met en place les grands blocs applicatifs. La conception reprend les modèles de l'analyse statique et dynamique et détaille l'organisation des classes dans un langage cible.

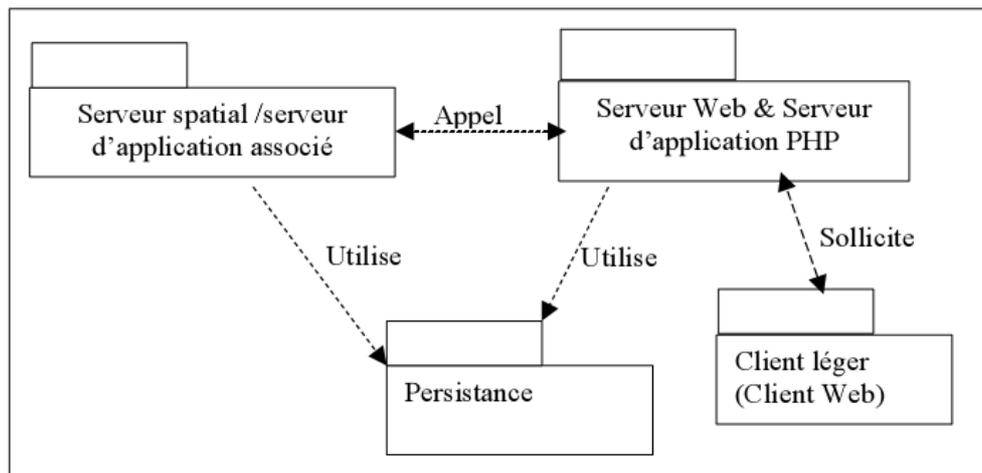
#### 3.7.4.1. Conception du système

Elle vise à définir l'architecture globale du système aux niveaux matériel et logiciel.

##### 1) Architecture logicielle

Les packages décrivent les relations macroscopiques entre les différentes parties du système. Les objets de domaine se regroupent en quatre principaux paquetages (Figure 22):

- Le serveur spatial /serveur d'application spatial (comme MapServer)
- Le serveur d'application PHP ou éventuellement ASP.
- Le client léger (client Web).
- Le module de persistance.



**Figure 22 : Architecture logiciel du Système**

Le module de persistance : c'est un paquetage très important vu que le mode objet ne présente aucune persistance, il peut être une base de donnée relationnelle ou également des fichiers selon un format particulier (\*.shp,\*.tab ...) tout dépend du format d'échange avec le serveur spatial. Ces données peuvent être :

- Des données spatiales ;
- Des données concernant la gestion des utilisateurs, les droits d'accès et l'aspect sécurité : il s'agit d'une base de données ou des fichiers logs.

- Des paramètres nécessaires au fonctionnement et à la configuration du système.

## 2) Architecture matérielle

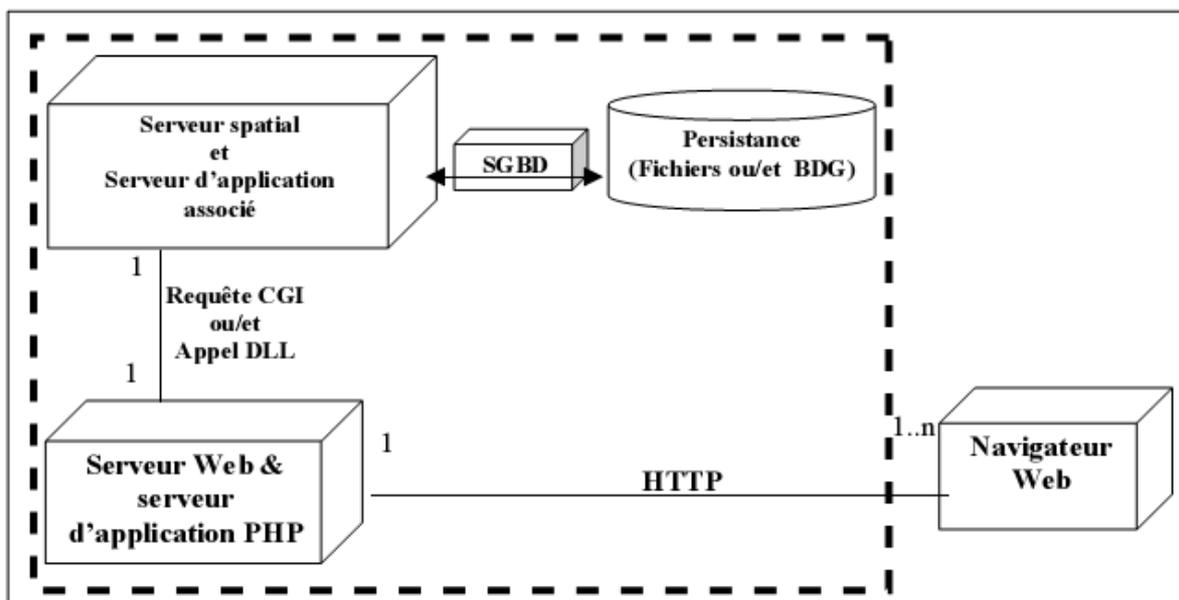
Les différents paquetages seront déployés comme suit :

- Le serveur spatial et le serveur d'application associé sur un poste de travail.
- Le serveur Web sur le même poste que le serveur spatial.
- Les clients légers sur des postes équipés d'un navigateur Web.
- Le module de persistance (les données) ainsi que le SGBD sur le même poste que le serveur spatial ou bien sur un autre poste.

La communication entre les différents composants se fait via le réseau en utilisant les protocoles appropriés.

Pour schématiser ces composants on utilise le diagramme de déploiement qui représente le déploiement des composants sur des dispositifs matériels.

Le déploiement de ces composants sur les dispositifs matériels est illustré sur le diagramme de déploiement suivant (*Figure 23*) :



**Figure 23 : Architecture matériel du Système**

La conception du système est généralement basée sur 03 axes :

### 3.7.4.2. La décomposition du système en sous systèmes

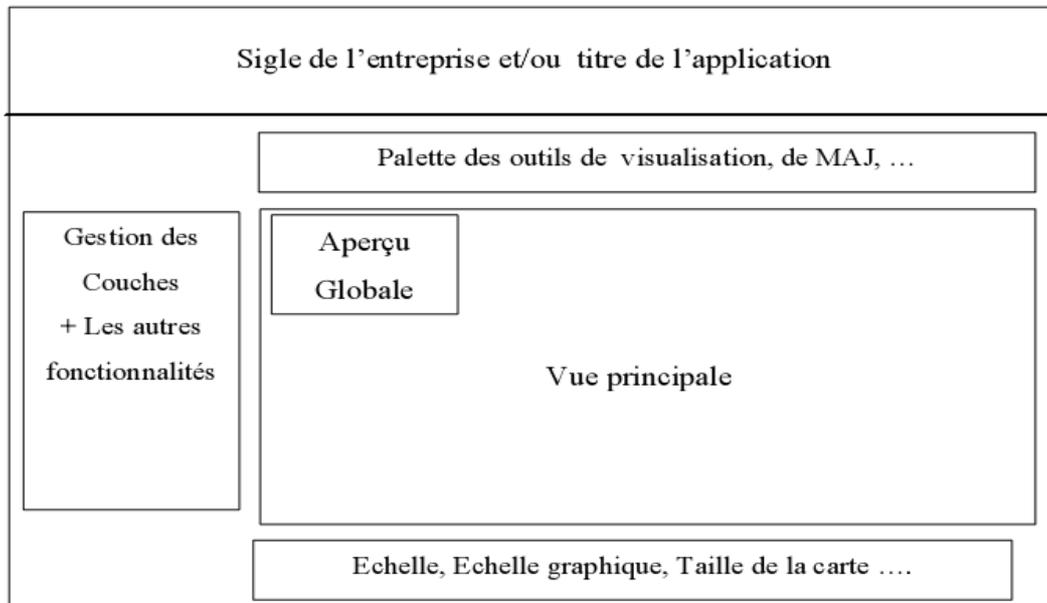
Notre système est décomposé en 03 sous systèmes :

- une couche interface Homme Machine,
- une couche objets métiers.

- une couche infrastructure (base de données attributaire et spatiale...).

#### a. La couche interface Homme Machine (Figure 24)

Elle Comprend les séquences de fenêtres (écrans) qui permettent la communication entre l'utilisateur et le système. On peut schématiser l'interface avec un croquis dégageant les principaux éléments qui constituent l'interface homme-machine.



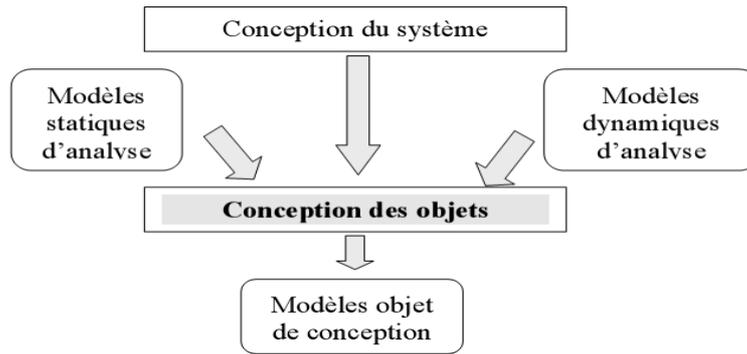
***Figure 24 : Interface Homme Machine de l'application coté client***

#### b. La couche objets métiers

Ce sont les objets appelés par les objets de l'interface, et qui agissent sur la couche infrastructure. Donc ce sont les objets instances des classes décrites dans le diagramme de classes de la phase d'analyse et les objets proposés par le langage de programmation.

#### ☐ La conception objets

La conception objet s'appuie essentiellement sur les diagrammes de la phase d'analyse, et sur la conception du système (**Figure 25**).



**Figure 25 : La place de la conception des objets.**

Dans cette section nous présentons le modèle objet de conception, il s'agit de définir le typage des attributs, c'est-à-dire de déterminer les types primitifs et objet de chaque attribut.

Les principaux modèles de l'analyse dynamique utiles à la conception sont les diagrammes d'états et les diagrammes de collaboration entre objets. En effet, les actions et les activités représentées dans ces modèles sont traduites en méthodes des classes de conception. De plus, les enchaînements des transitions constituent la base de l'élaboration des algorithmes.

Pour les objets : Carte, Couche, Classe thématique, Projection, légende, Référence, Forme et Paramètre Web on a opté pour la réutilisation des objets proposés par PHP\_MapScript (le module Php utilisé pour communiquer avec le serveur spatial MapServer). Dans cette section on présente une brève description des principales classes.

### c. La conception de l'infrastructure (les données)

Une « base de données » n'est rien d'autre qu'une collection structurée de données. On peut créer une telle base à l'aide de fichiers binaires, ou de fichiers textes, cela n'enlève rien à la nature même de la base. Toutefois, cette tenue est employée couramment pour désigner un type bien particulier de base de données: les bases de données dites relationnelles.

Dans notre système on va utiliser une base d'authentification dont le modèle conceptuel décrit dans la figure suivante (**Figure 26**):



**Figure 26 : Le modèle conceptuel de la BD d'authentification**

L'association **A\_privilege** indique qu'un utilisateur peut avoir plusieurs privilèges (cardinalité 0, n). Chaque utilisateur est protégé par un mot de passe qui sera stocké dans la table **Utilisateur** après son cryptage en utilisant les fonctions de cryptage.

#### **d. Gestion des priorités et des données**

En fixant les priorités à plusieurs niveaux :

- La robustesse : il s'agit de mettre en place un système de gestion des erreurs comme par exemple le mécanisme d'exceptions et les messages d'alerte notamment pour contrôler la qualité des données et les canaux d'accès aux données.
- La sécurité : les fonctions d'administration des données de l'application
  - Les données liées aux utilisateurs ;
  - Les mécanismes de protections des données géographique (CGI de MapServer) : toutes les données présentées aux utilisateurs via l'interface Web, sont en format raster quelque soit la source de ces données (vecteur, raster ou BDG). Cette méthode permet la protection des données notamment les données vecteur.
- La performance d'exécution : Afin de diminuer la charge sur le serveur, nous allons utiliser des scripts coté client comme Java Script et DHTML.

## **Chapitre.4. : IMPLEMENTATION**

### **4.1. Implémentation : choix des composants logiciels**

Dans ce qui suit nous allons présenter les différents composants Open Source choisis pour l'implémentation de notre application. Ainsi, des choix ont dû être fait pour le moteur cartographique, le serveur web, le framework et le SGBDR. La prérogative était d'utiliser des outils Open Source, robustes et de suivre les évolutions technologiques.

#### **4.1.1. Les serveurs cartographiques**

L'étude des serveurs cartographiques s'est recentrée autour des deux solutions phares disponibles dans le monde du libre, Geoserver et MapServer.

Il existe d'autres programmes open source avec lesquels il est possible de développer un système de serveur cartographique. Nous avons choisi GeoServer et MapServer car ils sont programmés avec des langages différents. Ces deux langages de programmation se retrouvent dans la plupart des serveurs de SIG sous licence libre. Il nous a alors semblé juste de présenter les avantages et les inconvénients induits par les capacités fonctionnelles de chaque langage utilisé.

D'une part, Geoserver est le serveur cartographique Open Source développé en environnement JAVA, et basé sur la bibliothèque GeoTools. Son avantage réside dans l'intégration de nombreuses spécifications Web Service de l'OGC (Open Geospatial Consortium), tels que WMS (Web Map Specification), WFS (Web Feature Specification), SLD (Style Layer Descriptor)..., mais aussi au niveau de la normalisation notamment pour le support de la norme WFS. Il prend en charge autant les données vectorielles, que les données rasters.

D'autre part, MapServer a été développé au sein de l'Université américaine du Minnesota, ce moteur cartographique permet la publication de données via un serveur Web et propose un ensemble de fonctions important (navigation, requêtes, mais aussi mise à jour à distance). C'est en fait une bibliothèque de fonctions développée sous licence GPL (General Public Licence)..., selon les recommandations de l'OGC. MapServer est ainsi régi par des normes publiques qui en assurent sa pérennité. Actuellement à sa version 5, 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.

Ainsi, le choix du serveur cartographique s'est dirigé vers MapServer pour des raisons de robustesse, d'ancienneté, de développement et de connaissance.

Outre ces arguments sur le choix du serveur cartographique, il semble que les différentes études comparatives effectuées entre ces deux solutions se révèlent proches dans leurs performances. Ainsi, on peut distinguer MapServer comme une boîte à outils permettant le développement d'une application grâce au CGI (Common Gateway Interface) ou php\_MapScript et par sa maturité dans son développement. Alors que Geoserver semble plutôt répondre à la logique d'une solution « prêt à l'emploi » et surtout orienté Web Service.

## 4.1.2. Les éléments d'une application MapServer

MapServer fonctionne généralement comme une application CGI depuis votre serveur HTTP. Sauf si on construit une application avancée avec MapScript (module de PHP contenant des classes et des fonctions pour MapServer) qui accède directement à l'API MapServer.

Les applications CCI MapServer utilisent les ressources suivantes (Figure 23):

1. Un serveur HTTP (Apache ou Internet Information Server) ;
2. Le logiciel MapServer ;
3. Un fichier d'initialisation qui déclenche la première vue d'une application MapServer (facultatif);
4. Un Mapfile qui contrôle ce que MapServer fait avec les données ;
5. Un fichier de mise en forme (template) qui contrôle l'interface utilisateur de l'application MapServer dans une fenêtre de navigateur ;
6. Un ensemble de données (des fichiers SIGSBDG...).

MapServer est installé dans le répertoire CGI-bin du serveur Web, et les fichiers ainsi que les données MapServer sont stockés dans le répertoire des documents du serveur Web. Voici une description des principaux composants d'une simple application MapServer.

### a) Fichier d'initialisation

Ce fichier peut faire partie d'un autre fichier HTML, mais pour plus de simplicité il peut constituer un fichier à part. Ce fichier d'initialisation utilise un formulaire pour envoyer une requête initiale au serveur Web qui retourne un formulaire résultant depuis MapServer. MapServer est dynamique et est lancé et exécuté à chaque fois qu'une requête est reçue, aussi ce fichier d'initialisation est simplement nécessaire pour passer plusieurs paramètres (cachés) à l'application. Le fichier d'initialisation est un fichier HTML, (son extension est .htm ou .html). Une autre méthode consiste à créer un hyperlien vers l'application MapServer. Il devra passer les paramètres de base nécessaires à l'application CCI MapServer.

### b) MapFile

Le MapFile indique les données à utiliser dans une application et à afficher ainsi que les paramètres des requêtes: donc il s'agit du fichier applicatif de configuration. Le Mapfile inclue 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.

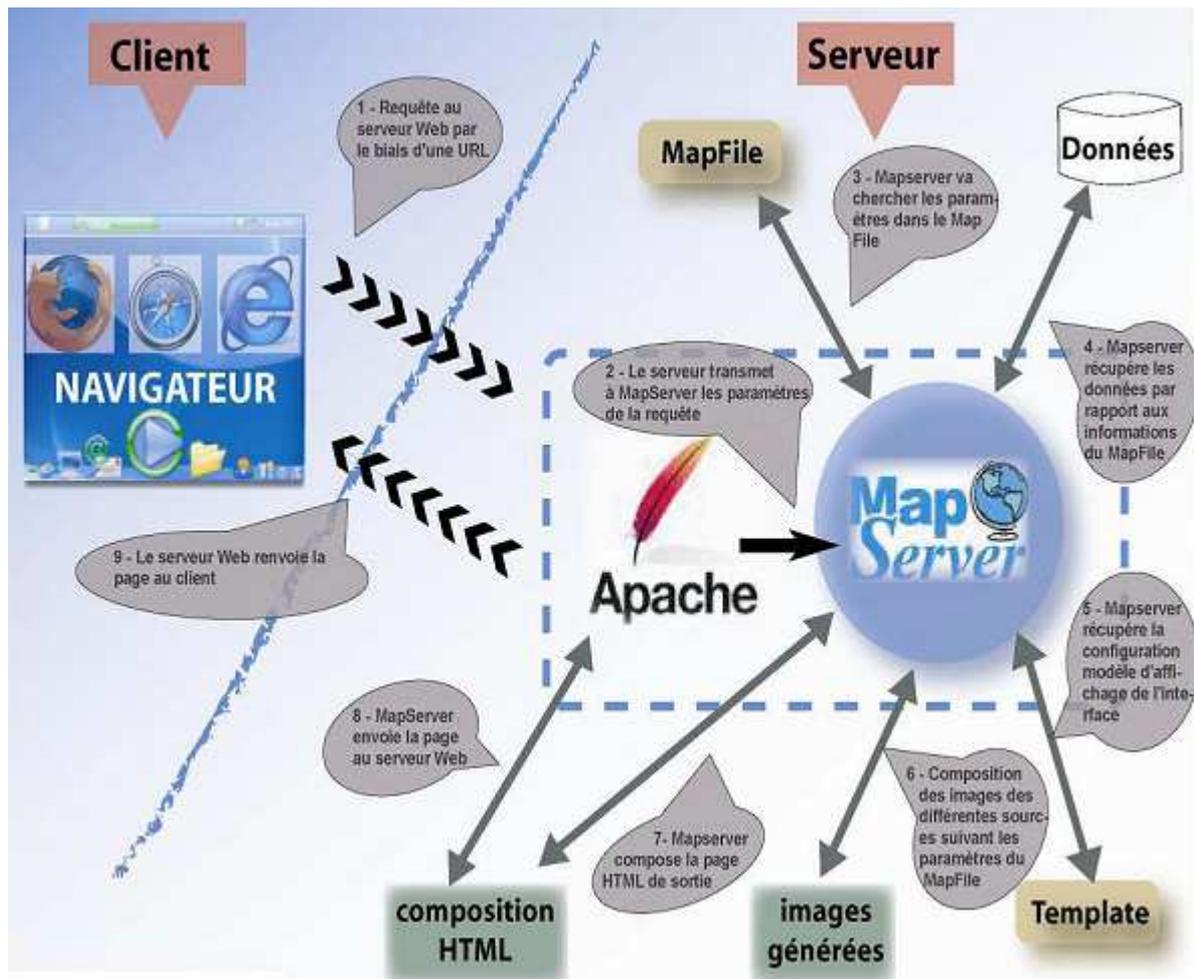
### c) Fichier de mise en forme (Template)

Le fichier de mise en forme contrôle comment les cartes et les légendes issues de MapServer apparaîtront dans une page HTML; il opère comme tout autre fichier HTML sauf que certains champs peuvent être modifiés par le CGI MapServer. Le fichier de mise en forme permet à l'auteur de positionner la carte et la légende sur la page, et détermine de quelle manière l'utilisateur est autorisé à interagir avec une application MapServer (parcourir, interroger, zoomer,...).

#### d) Ensemble de données spatiales

MapServer utilise par défaut le format Shape de ESRI comme format de fichier vecteur. Egalement il peut supporter d'autres formats vecteurs (en utilisant les bibliothèques appropriées). Les données Raster peuvent être traitées par MapServer, selon le mode de compilation.

Les fichiers de données devraient être placés dans un répertoire ou stockés dans une base de données géographique référencée dans le fichier MapFile.



**Figure 27 : Fonctionnement de MapServer**

#### 4.1.3. Serveur web

Pour le serveur Web on a choisi Apache pour les nombreux avantages de ce serveur à savoir :

- Gratuit (fait partie des logiciels libres) ;
- Modulaire ;
- Simple et efficace ;

- Robuste (c'est le plus utilisé dans le monde) ;
- Développement actif.

Les détails de l'installation et la configuration de ce serveur seront présentes annexes.

#### **4.1.4. Serveur d'application**

Pour le serveur d'application la solution potentielle choisie est PHP. PHP (Hypertext Preprocessor ou Personal Home Page) est un langage de scripts généralistes et Open Source, spécialement conçu pour le développement d'applications Web. Il peut être intégré facilement au HTML [1],

Il est à noter la différence avec les autres scripts CGI écrits dans d'autres langages tels que Perl ou C : Au lieu d'écrire un programme avec de nombreuses lignes de commandes afin d'afficher une page HTML, vous écrivez une page HTML avec du code inclus à l'intérieur afin de réaliser une action précise. Le code PHP est inclus entre une balise de début et une balise de fin qui permettent au serveur Web de passer en mode PHP (<?php .. ?>).

Ce qui différencie le PHP des langages de scripts comme le JavaScript est que le code est exécuté sur le serveur. Si vous avez un script similaire sur votre serveur, le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat. Vous pouvez configurer votre serveur Web afin qu'il analyse tous vos fichiers HTML comme des fichiers PHP. Ainsi, il n'y a aucun moyen de différencier les pages qui sont produites dynamiquement des pages statiques.

Le grand avantage de PHP est qu'il est extrêmement simple pour les débutants, mais offre des fonctionnalités avancées pour les experts. Bien que le développement de PHP soit orienté vers la programmation pour les sites Web, vous pouvez en faire bien d'autres usages.

#### **4.1.5. Modèle de l'interface**

Une fois le moteur cartographique choisi, le choix d'une interface a dû être fait. En effet, MapServer n'est pas compilé avec un client web mais il existe un certain nombre d'interfaces Open Source développées à partir de la bibliothèque de fonction php\_Mapscript.

Une recherche nous a permis de dresser un état des lieux sommaire des forces et faiblesses des différentes interfaces. Un premier constat a pu être établi sur le dynamisme impressionnant des développements autour de MapServer, les différents frameworks ont tous suivi les évolutions technologiques et se sont enrichis de fonctionnalités nouvelles en lien avec la demande des utilisateurs. Ce premier constat aidant, on peut cependant remarquer des différences dans la conception et surtout la destination des modèles.

Pour ma part, je ferais ressortir trois modèles du lot, tout d'abord Chameleon développé par DM solutions dont l'architecture, autour des Widgets, permet une grande flexibilité, et facilite l'implémentation, la modification et la création de fonctionnalités. C'est un ensemble de composants qui permet de créer simplement une interface HTML, en réservant des espaces à l'affichage de cartes et de données attributaires et à des outils, comme le zoom, la sélection et de déplacement. Ainsi, Il permet, en quelques clics, de construire une base pour le développement d'une application de visualisation cartographique dynamique (php) sur Internet.

Ensuite, la solution de la société franco-suisse Camp-To-Camp, le CartoWeb. CartoWeb offre des fonctionnalités très avancées et une architecture client/serveur aboutie cependant le seul bémol demeure dans la complexité de son développement, qui me paraît encore lourd dans sa mise en place. C'est la solution qui se rapproche le plus des fonctionnalités d'un SIG classique tout en étant adaptée aux caractéristiques de l'Internet. Il s'intègre facilement dans un environnement Apache, php5, Mapserver 4.5 mais n'est pas compatible php4, ce qui empêche pour l'instant une association avec MapEdit. Il fonctionne grâce aux potentialités de php5 en langage objet par un système de plugin et de core-plugin. CartoWeb est une solution conçue pour le Web, elle permet par son architecture cartoclient/cartoserveur acceptant simultanément plusieurs utilisateurs, une gestion décentralisée de l'information et est compatible avec une connexion à débit limité.

Enfin, le projet OpenLayer, sans doute la solution qui m'a le plus surpris avec son affichage en dalle des images qui en font un modèle très intéressant et prometteur. De plus, son aspect très similaire aux API de Google et Microsoft en font un outil attrayant. Cependant son développement en Javascript me paraît assez lourd par rapport à une solution php\_mapscript.

Ainsi, la solution retenue est le framework développé par la société franco-suisse Camp-To-Camp, CartoWeb3 pour son nombre important d'outils et de fonctions. D'autre part, la communauté et les documentations (en français) autour de CartoWeb sont très importantes, ce qui peut avantager la mise en place d'un projet sous cette plate-forme.

On citera en quelques points des autres avantages du système CartoWeb :

- Mise en valeur des informations spatiales dans un site Web.
- Mise en place facile, peu de programmation.
- Faible coût d'implémentation (pas de licence coûteuse - fédération des coûts d'acquisition des données cartographiques).
- Possibilité de saisir et modifier des informations géographiques via Internet.
- Compatibilité et intégration avec les SIG client serveur du marché.
- Possibilité de louer l'accès aux fonctionnalités (mode ASP).

CartoWeb sera utilisé en particulier pour :

- Diffuser des cartes ou des plans sur le Web avec des informations thématiques spécifiques et toutes les fonctionnalités de navigation (zoom, déplacement...).
- Lier des informations spatiales au contenu informationnel des bases de données.
- Saisir et modifier de l'information géographique en ligne, par l'édition d'objets de type points, lignes ou surfaces.

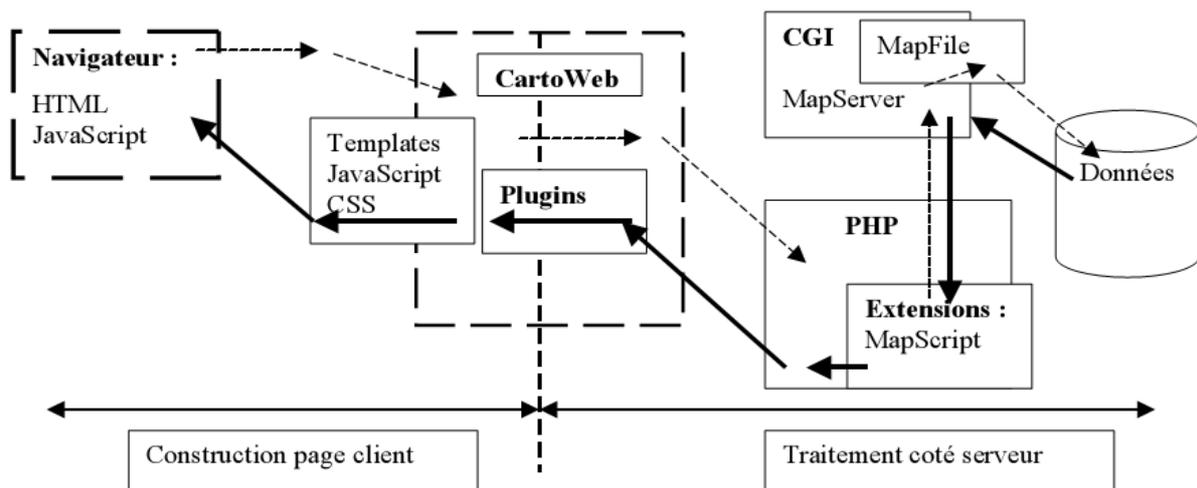
#### **4.1.6. Présentation du CartoWeb**

Cartoweb3 est une solution produite par Camptocamp (<http://www.camptocamp.com>) et disponible depuis le 30 mars 2005 sous licence de logiciel libre. Elle est basée sur le moteur cartographique (serveur Spatial) MapServer et permet d'exploiter la grande majorité des formats de données géographiques (raster et vectoriel) courants.

Elle donne aussi la possibilité de se connecter aux bases de données géographiques PostgreSQL/PostGIS, Arc SDE ou Oracle Spatial. CartoWeb n'est pas réellement un outil de développement, cependant il fournit une structure de base pour le développement d'applications basées sur MapServer et Php/Map Script. Un grand nombre de fonctionnalités et de modules sont prédéveloppés, ainsi que la gestion des flux de données. Ces éléments sont personnalisables et intégrables dans un projet personnel.

#### 4.1.6.1. Architecture et fonctionnement général de CartoWeb

CartoWeb est essentiellement basé sur des scripts php5 qui communiquent avec MapServer via la librairie de fonctions Php/MapScript. Il est le cœur du site (intranet ou internet) de diffusion de données cartographiques, à ce titre, il se place sur le serveur comme se placerait un site Web classique afin que les scripts puissent être exécutés. Pour comprendre comment fonctionne cet ensemble, nous allons suivre étape par étape le cheminement d'un échange d'informations entre le client distant et le serveur, c'est-à-dire depuis une requête émise par le client jusqu'au retour de la réponse (*Figure 28*).



**Figure 28 : Architecture et fonctionnement général de CartoWeb**

Le chemin suivi, depuis la requête jusqu'à l'accès aux données géographiques, représenté par les flèches pointillées.

1. L'envoi de la requête par le navigateur vers le Serveur, en l'occurrence un serveur Apache.
2. Apache analyse la requête et la redirige vers CartoWeb.
3. CartoWeb traite l'information via un grand nombre de scripts php5, ceux-ci sont organisés en plugins. A partir de la requête envoyée par le client, CartoWeb construit sa propre requête destinée à MapServer.
4. Pour que les scripts puissent être exécutés Apache fait appel au module php5.

5. Un certain nombre de fonctions utilisées dans les scripts de CartoWeb sont spécifiquement destinées à la communication avec MapServer. Ces fonctions font partie de la librairie de fonctions MapScript.
6. Les fonctions MapScript sont donc interprétées et traduisent pour MapScan la requête produite par CartoWeb.
7. MapServer réceptionne la requête, l'analyse, et y répond en consultant le Mapfile.
8. Grâce au Mapfile, MapServer accède donc aux données et récupère celles demandées par la requête.

Après cette première étape qui nous a montré comment l'accès aux données était réalisé, voyons comment ces données sont traitées et retournées au client. Ce cheminement est représenté par les flèches en gras sur le schéma. En partant cette fois des données, nous pouvons distinguer les étapes suivantes.

1. Les données sont récupérées et renvoyées à Mapserver. Les modes de communication, entre MapScan et les données, diffèrent suivant la nature des données (formats des couches ou images), ils sont définis dans le Mapfile et Mapserver adoptent en conséquence « la marche à suivre ».
2. Une fois les données récupérées, MapServer les traite pour produire des images et des métadonnées.
3. Les images ainsi produites, ainsi que les informations les accompagnant sont récupérées par CartoWeb via l'utilisation, une nouvelle fois, des fonctions MapScript.
4. CartoWeb traite les informations en fonction de la requête initialement envoyée par le client et transmet l'ensemble à une sous-partie de CartoWeb qui se charge d'organiser les données en vue de l'affichage.
5. Avant le renvoi vers le client, les scripts Php sont interprétés pour produire des pages HTML. Leur mise en page est gérée par un ensemble de fichiers appelés templates accompagnés éventuellement par des fichiers css (styles et mise en forme). Enfin des fonctions javascript peuvent être adjointes à la page pour la rendre plus dynamique.
6. Pour finir l'ensemble est renvoyé par Apache vers le client.

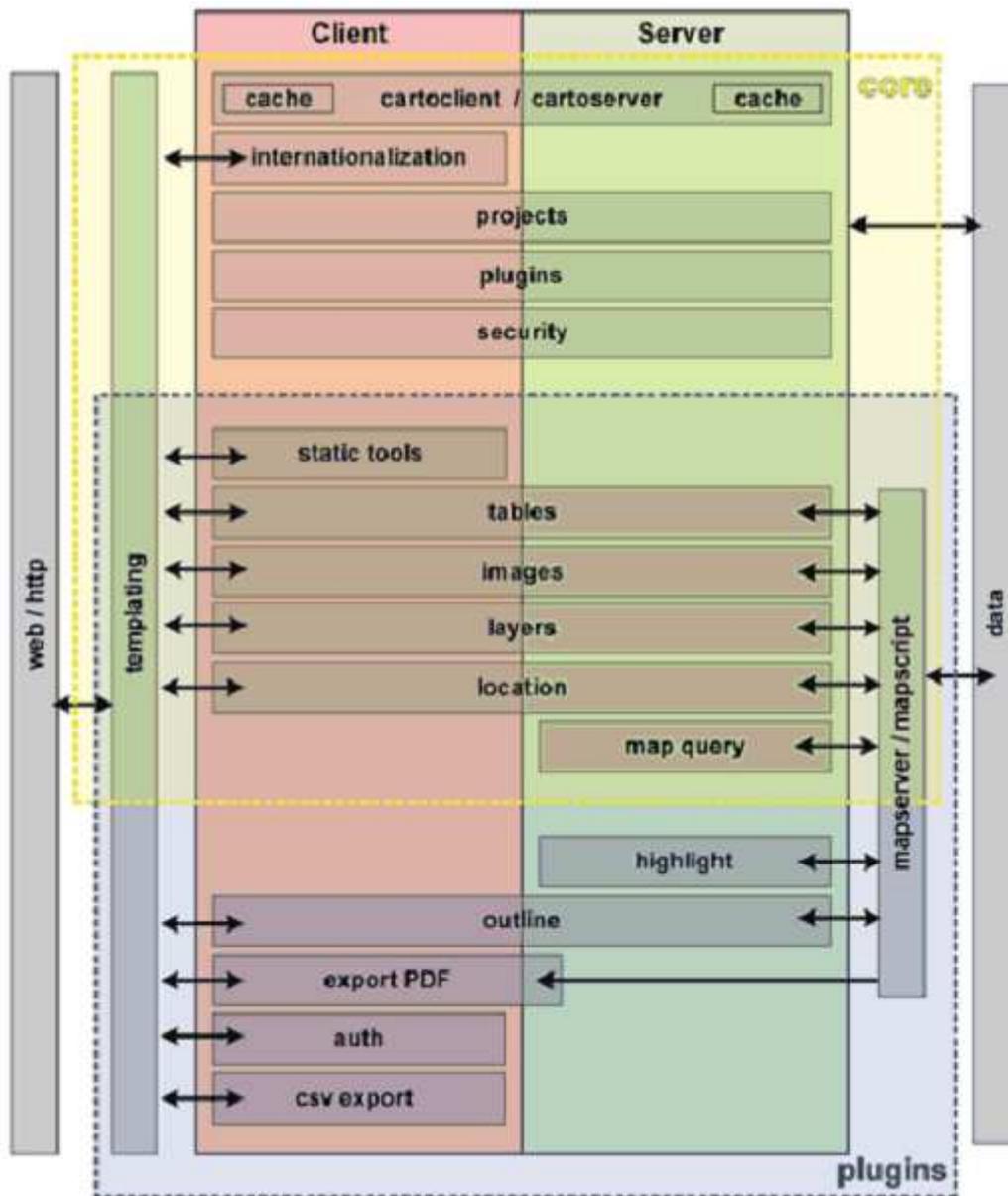
#### **4.1.6.2. Fonctionnalités de CartoWeb**

CartoWeb dans sa forme de base rend possible l'implantation d'un nombre déjà important de fonctionnalités : étant modulable et libre il peut être étendu librement et sans limite, les fonctionnalités potentiellement exploitables sont :

- Accès à de nombreux formats de données raster : TIFF (world file), GeoTIFF, CEOs, DOQ, ECW, ESRI Labelled BIL, Arc/Info Binary Grid, OGDI, Erdas Imagine, GIF, PNG et JPEG.
- Accès à de nombreux formats de données vectorielles : Shapefiles, DGN, SDTS, MIF, TAB et OGDI Vectors.
- Accès à des bases de données spatiales : PostgreSQL/PostGIS, ArcSDE et Oracle Spatial/Locator.
- Compatibilité avec les normes de l'Open GIS consortium.

- Sorties de données/images aux formats : PNG, GIF JPEG, PDF, WBMP, GeoTIFF et SWF.
- Gestion automatique des systèmes de projection.
- Fonctionnalités de visualisation avancées en mode DHTML.
- Compatibilité avec les anciens navigateurs en mode HTML simple.
- Navigation : zoom, recentrage, déplacement et panning.
- Mesure de distance et de surface.
- Gestion dynamique de la classification des couches, de l'iconographie et des labels.
- Vignette de positionnement dynamique et barre d'échelle.
- Choix dynamique de l'échelle et de la taille de la carte.
- Création, modification et suppression d'objets de type point, ligne ou polygone.
- Requête spatiale par point ou polygone avec retour des informations (attributs ou bases de données externes).
- Localisation d'objets (centrage sur objet) par recherche dans les attributs ou base de données externes.
- Système de cache dynamique permettant de stocker les états de visualisation fréquents et de diminuer la charge de MapServer.
- Impression au format PDF.
- Accès direct à des objets, des vues et des états.
- Gestion dynamique et hiérarchique des légendes, avec accès aux définitions de métadonnées externes.
- Interface multilingue.
- Regroupement hiérarchique des couches en thèmes.

### 4.1.6.3. Structure générale de CartoWeb



**Figure 29 :** Structure de CartoWeb (source CartoWeb Documentation, Camptocamp)

Dans un premier temps, CartoWeb se compose de deux modules principaux : Cartoclient et Cartoserver (**Figure 29**). Cartoserver traite les données et communique avec Mapserver, Cartoclient gère, quant à lui, le lien avec le client, réception des requêtes et mise en forme des réponses. Ces deux modules permettent un mode de fonctionnement de type client-serveur, où Cartoserver peut faire le lien avec Mapserver et fournit les résultats pour le compte de plusieurs Cartoclient ; ceci via le protocole SOAP (Simple Object Access Protocol).

#### 4.1.6.4. Protocole SOAP

SOAP est un protocole de transmission de messages. Il définit un ensemble de règles pour structurer des messages qui peuvent être utilisés dans de simples transmissions unidirectionnelles, mais il est particulièrement utile pour exécuter des dialogues requête- réponse RPC (Remote Procédure Call). Il n'est pas lié à un protocole particulier. Il n'est pas non plus lié à un système d'exploitation ni à un langage de programmation, donc, théoriquement, les clients et serveurs de ces dialogues peuvent tourner sur n'importe quelle plate-forme et être écrits dans n'importe quel langage du moment qu'ils puissent formuler et comprendre des messages SOAP. En tant que tel, il s'agit d'un important composant de base pour développer des applications distribuées qui exploitent des fonctionnalités publiées comme services dans l'intranet ou l'internet.

Toutefois CartoWeb peut fonctionner comme un module unique, où la communication entre Cartoserver et Cartoclient se fait directement (sans utiliser le protocole SOAP).

#### 4.1.6.5. Principe de Fonctionnement du CartoWeb

Pour proposer ses diverses fonctionnalités, CartoWeb applique le principe de modularité, chaque module appelé plugin et est chargé de réaliser des tâches bien précises. Deux critères permettent de décrire ces plugins :

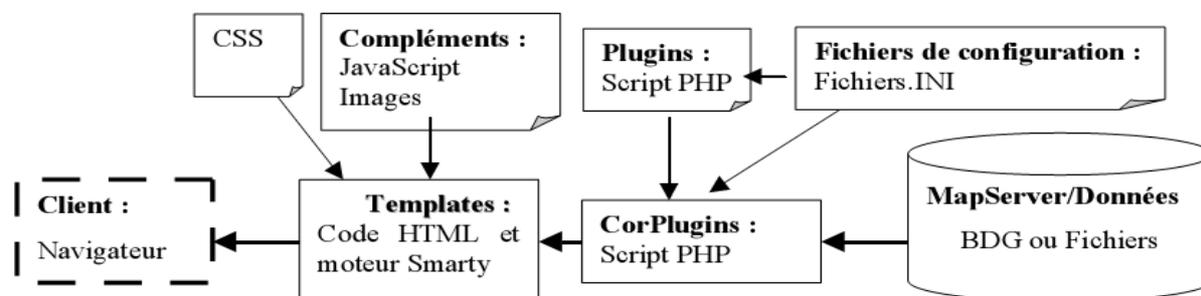
- leur appartenance à Cartoclient ou à Cartoserver;
- leur appartenance au cœur du système (Coreplugins) ou leur aspect annexe.

Les Coreplugins sont indissociables et indispensables au fonctionnement de CartoWeb. Les autres plugins sont totalement indépendants et peuvent être implémentés ou non, suivant les besoins. Leur appartenance à Cartoserver ou à Cartoclient définit simplement leur niveau d'action.

Ce double découpage est présent à tous les plans, les scripts php, les fichiers de configurations et les templates sont structurés en fonction de ce découpage. Ceci permet de retrouver facilement les divers éléments et d'entrevoir leurs fonctions, mais en contrepartie fragmente beaucoup l'application et il est nécessaire de manipuler un nombre important de fichiers.

En respectant cette structure il est tout à fait possible de développer ses propres plugins, répondant à des besoins spécifiques.

#### 4.1.6.6. Fonctionnement de CartoWeb



*Figure 30 : Principe général de fonctionnement de CartoWeb*

Le schéma ci-dessus (*Figure 30*) présente le principe général de fonctionnement de CartoWeb, voyons le rôle des éléments représentés.

- **Mapserver/Données** : Il s'agit de la source des données, que Cartoweb aura en charge de traiter.
- **Fichiers de configuration** : Les fichiers de configuration définissent, quels sont les plugins à activer et paramètrent leur fonctionnement.
- **Coreplugins** : Les coreplugins constituent les éléments indispensables au fonctionnement de Cartoweb. Accompagnés des plugins activés, ils traitent et transforment les données pour permettre leur affichage. Plugins : Les plugins complètent l'action des coreplugins par l'apport de fonctionnalités supplémentaires.
- **Templates** : Les templates définissent la mise en page grâce au HTML et à des commandes supplémentaires interprétées (des templates dynamiques en utilisant Smarty).
- **Css** : Les Css (Cascading style sheets), définissent les styles (texte, blocs, etc...) à appliquer lors de la mise en page.
- **Compléments** : Les compléments sont des éléments qui viennent agrémenter la page produite et y implémente des fonctionnalités supplémentaires.

Le principe est donc simple. Les données transmises à CartoWeb sont traitées en fonction des plugins activés et de leur configuration. Les données ainsi transformées sont mises en page et transmises au navigateur du client.

#### **4.1.7. SGBDR (Système de Gestion de Base de Données Relationnelles)**

Le système de gestion de bases de données relationnelles est un aspect important du projet. De ce fait, le choix de ce dernier doit être réfléchi et soumis à un test de compatibilité avec MapServer d'une part, et de puissance notamment au niveau de sa cartouche spatiale, d'autre part. Le comparatif s'est limité à MySQL et PostgreSQL qui sont sans conteste les deux produits les plus répandus et robustes du marché. De plus, ils ont l'avantage d'offrir un module spatial avec MyGIS et PostGIS. Les conclusions des tests réalisés sur l'insertion des données, les requêtes, la rapidité d'exécution... permettent d'établir une hiérarchie autour de ces produits.

- MyGIS de MySQL est encore un projet jeune, en cours de développement et n'implémente pas de façon complète les normes. Il permet néanmoins le stockage et l'indexation d'objets spatiaux et offre de très bonnes performances pour les requêtes mettant en œuvre des objets spatiaux. A terme, si MyGIS implémente la norme OGC, il représentera un concurrent sérieux à PostGIS. Son avantage est cependant une communauté de développeurs et de supports très active, une très bonne intégration dans les projets Web avec un système de base de données (MySQL) le plus utilisé dans le monde.
- PostGIS, la cartouche spatiale de PostgreSQL implémente la norme OGC et offre un grand nombre de fonctionnalités spatiales. Il n'offre, cependant pas autant de fonction

qu'Oracle. La communauté Open Source de ce projet est très active, tant pour le développement que pour l'animation des forums. Le temps de résolution d'éventuels bugs est très court et peu de problèmes restent sans réponse. Des développements sont en cours concernant le support de la topologie dans PostGIS et des réflexions sont menées pour l'intégration et le support des rasters. De plus, des modules externes à PostGIS permettent de gérer le calcul d'itinéraire (PgRouting) et le géocodage, cela est une source d'évolution possible de la solution vers d'autres problématiques.

Aujourd'hui PostgreSQL qui est son héritier OpenSource se vante d'être le SGBDR le plus avancé au monde, car il possède des fonctionnalités très étendues et le moyen de les étendre facilement par la programmation (langages de macro puissants et accessibles). PostGIS est pour une bonne partie codé dans un de ces langages de macro, plPgSQL.

#### **4.1.7.1. PostgreSQL**

PostgreSQL est un système de gestion de bases de données relationnelles objet (ORDBMS) fondé sur POSTGRES (version 4.2). Ce dernier a été développé à l'université de Californie au département des sciences informatiques de Berkeley.

POSTGRES est à l'origine de nombreux concepts qui ne seront rendus disponibles au sein de systèmes de gestion de bases de données commerciales que bien plus tard.

PostgreSQL est un produit Open Source original de Berkeley. Il supporte une grande partie du standard SQL tout en offrant de nombreuses fonctionnalités modernes, permettant :

- Effectuer des requêtes complexes ;
- Lier les tables en utilisant des clés étrangères ;
- Lancer des déclencheurs (triggers) ;
- Offrir des vues ;
- Assurer l'intégrité des transactions ;
- Contrôler les accès concurrents (MVCC ou multiversion concurrency control).

De plus, PostgreSQL est extensible par l'ajout, par exemple de :

- Nouveaux types de données ;
- Nouvelles fonctions ;
- Nouveaux opérateurs ;
- Nouvelles fonctions d'agrégat ;
- Nouvelles méthodes d'indexage.

Et grâce à sa licence libre, PostgreSQL peut être utilisé, modifié et distribué librement, quelque soit le but visé, qu'il soit privé, commercial ou académique.

##### **a) Concepts de base de PostgreSQL :**

PostgreSQL utilise un modèle client/serveur. Une session PostgreSQL est le résultat de la coopération des processus (programmes) suivants :

- ❑ Un processus serveur, qui gère les fichiers de la base de données, accepte les connexions à la base de la part des applications clientes et effectue sur la base les actions des clients. Le programme serveur est appelé postmaster.

□ L'application cliente des utilisateurs qui veulent effectuer des opérations sur la base de données. Les applications clientes peuvent être de natures très différentes : un client peut être un outil texte, une application graphique, un serveur Web qui accède à la base de données pour afficher des pages web ou un outil spécialisé dans la maintenance ou l'administration de bases de données.

Comme souvent avec les applications client/serveur, le client et le serveur peuvent être sur des hôtes différents. Dans ce cas, ils communiquent à travers une connexion réseau TCP/IP.

Le serveur PostgreSQL peut traiter de multiples connexions simultanées depuis les clients. Dans ce but, il démarre (fork) un nouveau processus pour chaque connexion. A ce moment, le client et le nouveau processus serveur communiquent sans intervention de la part du processus postmaster original. Ainsi, le postmaster s'exécute toujours, attendant de nouvelles connexions clientes.

Dans PostgreSQL, en manipulant des commandes SQL (langage standard pour manipuler les bases de données) on peut :

- Créer une base de données et assurer l'accès à cette base,
- Créer des tables et les mettre à jour (Modification, insertion et suppression),
- Faire des jointures entre les tables et appliquer des fonctions de calcul ou d'agrégat sur les données de ces tables.

#### **b) Extensions de PostgreSQL**

Dès l'origine, PostgreSQL a été conçu pour être extensible. Grâce à cela, les extensions ont un comportement identique à celui des fonctionnalités natives du serveur. Parmi ces extensions, on trouve des outils de conversion, une indexation de texte complète, des outils XML ainsi que des types de données et des méthodes d'indexation additionnelles. D'autres extensions sont développées indépendamment comme PostGIS.

#### **4.1.7.2. PostGIS, principes et utilisation**

PostGIS est développé par Réfractons Research inc., comme projet de recherche dans la technologie des bases de données spatiales. Réfractons Research est une compagnie de consultation dans les domaines de SIG et de base de données dans Victoria, Colombie britannique (Canada).

PostGIS est une extension de PostgreSQL qui permet à des objets spatiaux d'être stockés et manipulés dans une base de données relationnelle.

Il a été développé pour servir de module de fonctionnalité spatial à PostgreSQL, à la manière SDE d'ESRI ou les extensions spatiales pour Oracle. Son développement a commencé en Mai 2001. Le projet est maintenu sous licence GPL (Gnu licence Public). A partir de la version de PostGIS (0.8.0) les normes édictées par l'OpenGIS Consortium sont totalement respectées.

Le principe est donc de pouvoir :

- stocker des informations géographiques,
- réaliser des opérations de type SIG comme des mesures de distance, d'intersection,

- créer de nouvelles géométries en les décrivant dans un format reconnu, ou comme résultat d'opérations des géométries existantes,
- retourner des informations géographiques (géométries et/ou attributaires) selon un format précis, sur le réseau.

De nombreux outils d'administration et de gestion de PostgreSQL sont disponibles. Le plus populaire est **PgAdmin**, **QUANTUM GIS** et d'autres qui sont commerciaux.

#### **4.1.7.3. QUANTUM GIS**

**QUANTUM GIS** «QGIS » est un logiciel Open Source doté d'une interface conviviale, QGIS permettant de se connecter facilement à des sources PostgreSQL/PostGIS afin de récupérer la géométrie et les valeurs attributaires des couches vectorielles. Des outils d'édition permettent ensuite de modifier les attributs, de rajouter des formes et ces modifications s'inscrivent directement dans les tables (<http://qgis.sourceforge.net>). Par ailleurs, le projet en cours peut être sauvegardé sous forme de **mapfile** dans l'optique de mettre à jour le serveur spatial (MapServer). Cependant, celui-ci n'est pas géré correctement et nécessite une intervention manuelle notamment pour spécifier le type de connexion.

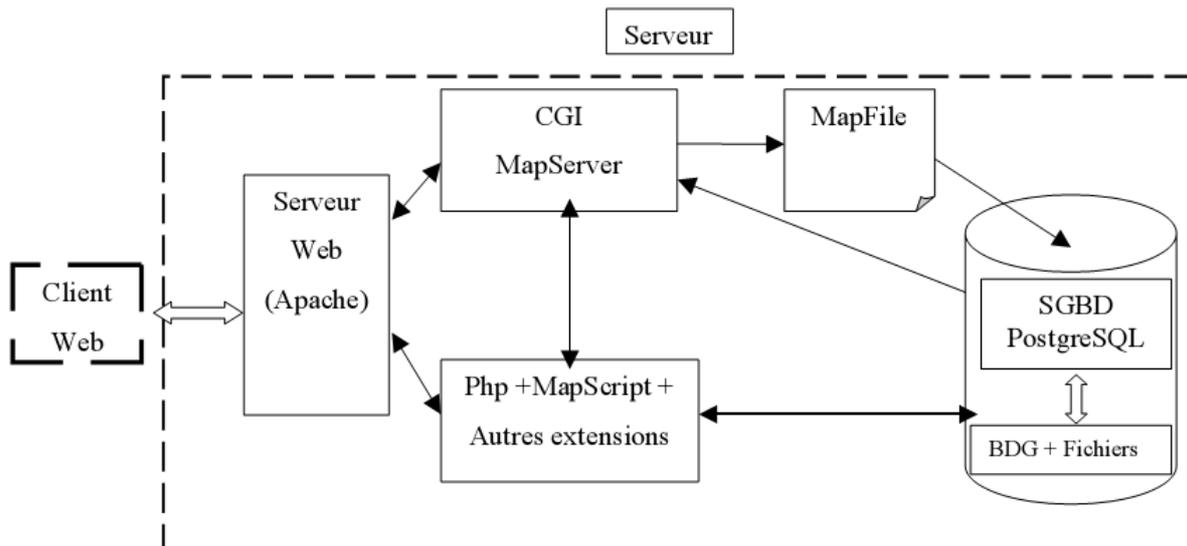
QGIS possède des fonctions simples de discrétisation, propose des outils basiques de création d'analyses thématiques. Les différentes cartes sont imprimables. Pour le moment, on ne peut pas effectuer des requêtes attributaires ou spatiales, même en langage SQL. Ainsi, en local, il est possible de récupérer des données distantes stockées dans une base de données, les représenter thématiquement, rajouter des couches diverses, les modifier et les exporter vers un PostGIS distant.

L'objectif principal derrière l'utilisation de QGIS est de mettre à jour le serveur spatial par la création et la modification de fichier MapFile qu'est le noyau de MapServer. Egalement il nous permet d'éditer les tables (géométrie et attributs) dans la base de données.

## **4.2. Implémentation : Installation et Configuration des composants**

L'implémentation est la phase au cours de laquelle les structures de données et les algorithmes et les composants définis pendant la phase de conception sont transcrit dans un langage de programmation et implémentés dans la plate forme. Elle comporte également l'installation de la plate forme (logiciel et matériels) et son paramétrage.

Globalement les composants de la plate forme logiciels sont schématisés comme suite (*Figure 31*) :



**Figure 31** : Les principaux composants logiciels du système

#### 4.2.1. Présentation de l'interface et de son développement

Le portail cartographique répond à un objectif commercial et l'attrait de l'outil passe bien entendu par des spécificités techniques mais aussi par une ergonomie et un graphisme agréables.

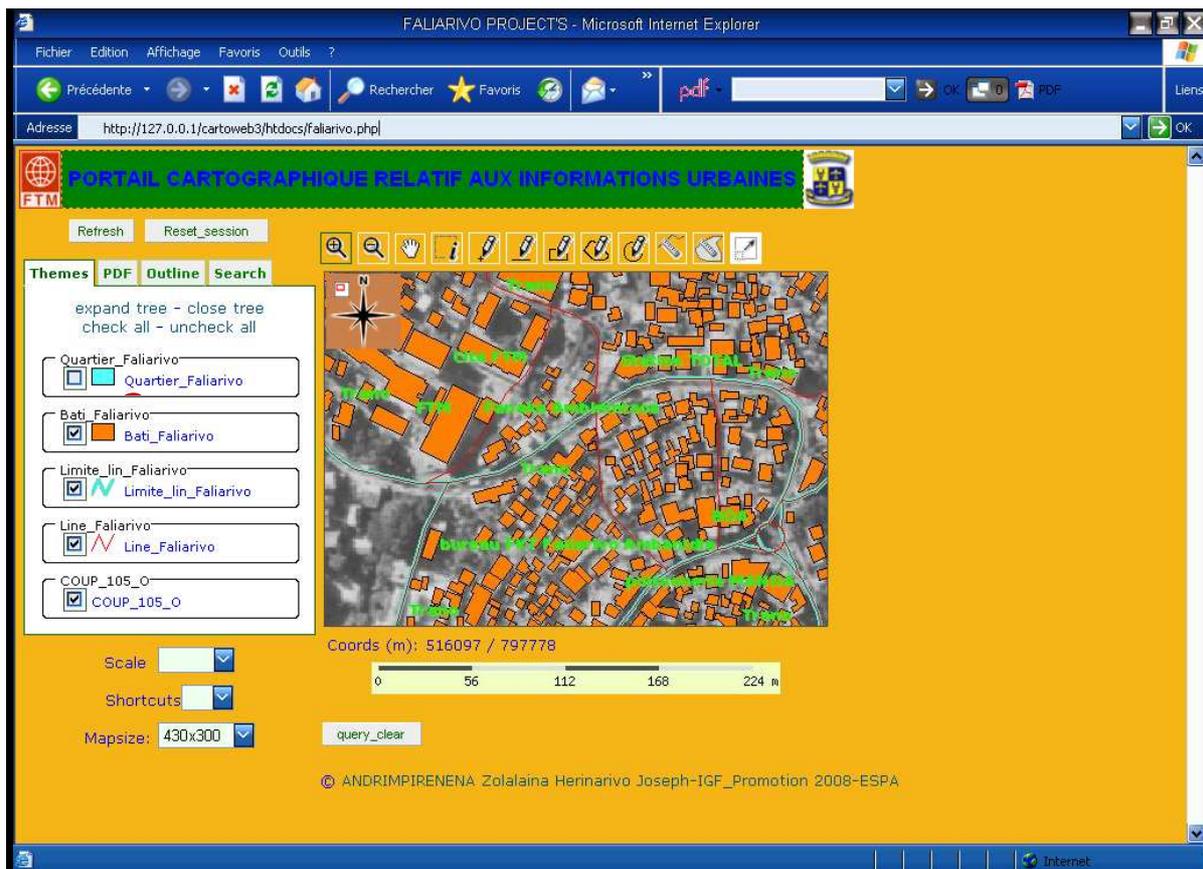
L'interface comprend les séquences d'écrans permettant la communication entre le système et l'utilisateur. Elle permet à l'utilisateur d'exprimer ces besoins à l'application, de configurer le système convenablement et de mettre à jour la couche infrastructure du système (les données). D'autre part elle permet au système d'exposer clairement les résultats pour qu'ils soient compréhensibles par l'utilisateur.

#### 4.2.2. Interface de navigation

##### □ La vue principale (Figure 36)

La vue principale est l'endroit de la fenêtre qui est le plus important puisque c'est elle qui va monopoliser l'attention de l'utilisateur. Elle doit donc être simple d'utilisation et fournir un temps de réponse rapide.

C'est pourquoi on a privilégié un "affichage en temps réel". L'utilisateur positionne la souris à n'importe quel endroit de la carte, clique et se déplace, pour amener l'ensemble du document dans la direction et à la distance choisie. Le document n'attend pas que le navigateur ait relâché la souris pour s'afficher dans sa nouvelle position, mais se déplace bien en temps réel en même temps que la souris. Cette méthode du "glisser-déposer" offre un confort maximum à l'utilisateur qui peut visualiser la carte en un temps très réduit. De plus on privilégie une habitude d'utilisation courante que l'on retrouve très souvent dans les logiciels SIG bureautiques.



***Figure 32 : La vue principale de l'application***

Un portail cartographique, au même titre qu'une carte, suit une logique graphique. Le sens de lecture va de gauche à droite, c'est dans cette optique que la lecture de l'interface a été pensée.

Tout d'abord, on regarde l'information, la carte (1). Une fois l'information visualisée et imprimée dans la mémoire, on va se référer à la légende (2) (à gauche de la carte) pour décrypter l'information lue. Le sens est ainsi respecté et l'utilisateur récupère l'information de manière naturelle. Ensuite intervient l'organisation des outils, avec en haut à gauche la navigation et le dessin (3), que l'on va visualiser en premier et qui constituent les éléments de base de l'application. Et on peut utiliser les outils en option (2) pour approfondir l'interrogation cartographique.

#### **□ La vue globale dynamique :**

La vue globale présente l'intégralité du document affiché. La carte est adaptée à l'espace réservé à cet effet. L'utilisateur dispose d'un curseur lui permettant de se déplacer d'un point à l'autre sur l'ensemble du document. Comme pour la vue principale, le déplacement se fait également en temps réel, à la différence que c'est le curseur qui se déplace par rapport à la vue. Le curseur représente exactement la visualisation de la vue principale et sa taille s'adapte donc au zoom en cours. Egalement, le déplacement sur la vue globale entraîne automatiquement le déplacement de la vue principale et inversement.

## ☐ Les outils de navigation (zoom et déplacement)



De nombreuses applications utilisent la fonction loupe qui permet d'agrandir ou de diminuer la fenêtre de visualisation par paliers successifs. On peut aussi envisager un "zoom à la demande" grâce à une boîte de sélection.



Cliquez sur cet outil et cliquez sur la carte, sur le centre de la région qui vous intéresse ou dessinez un "rectangle de zoom" autour de cette zone. L'outil de zoom peut être utilisé plusieurs fois pour zoomer sur des zones de plus en plus petites.



L'outil de zoom arrière vous permet de vous positionner sur une surface de la carte plus étendue que l'état actuel. Cliquez sur cet outil et cliquez sur la carte au centre de la région qui vous intéresse. Le zoom arrière peut être utilisé plusieurs fois, jusqu'à arriver à la vue générale.

Pour se déplacer dans la carte on peut envisager deux modes :

- 1) Déplacement par des flèches
- 2) Déplacement en temps réel. La solution 2

La solution 2 présente évidemment plus d'intérêt mais nécessite une programmation plus approfondie.



L'outil de déplacement vous permet de vous déplacer sur la carte sans changer l'échelle de la carte. Cliquez sur l'outil de déplacement et déplacez le centre de visualisation en faisant un "Cliquez\_déplacer".

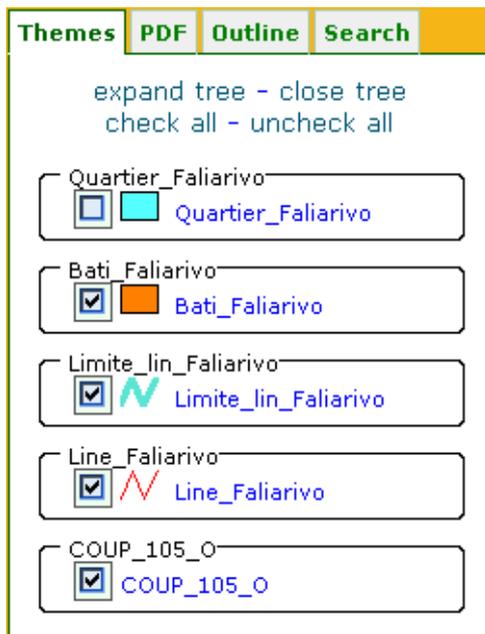


Le zoom vers la carte générale vous permet de quitter le zoom actuel et de revenir plein zoom. Il ne nécessite pas de clic sur la carte pour activer le déplacement. Il est activé dès que vous cliquez dessus.

### **4.2.3. La liste déroulante d'échelles**

Le choix de l'échelle est lié aux documents que l'on souhaite afficher et les types de personnes qui seront amenés à les visualiser. Pour un plan de ville par exemple, il peut être opportun de faire varier l'échelle (1/100, 1/200, 1/500, 1/1000, 1/2000, 1/4000, 1/5000). On préfère indiquer les zooms en pourcentages, plus parlant pour la majorité du public.

#### 4.2.4. Gestion des couches (Onglet Thèmes)



L'interface permet de gérer l'affichage d'une carte sous forme de couches superposées. Cette fonctionnalité permet à l'utilisateur de construire sa carte comme il le souhaite pour faire ressortir certaines particularités ou mieux visualiser le document.

Suivant la nature des documents que l'on souhaite présenter, nous pouvons également permettre aux utilisateurs de visualiser les informations sous forme de couches. Cette fonctionnalité peut servir à présenter sur un fond de carte commun des informations de nature différentes comme des données de population ou de répartition d'activité par de nature différentes comme des données de population ou de répartition d'activité par exemple. Cet outil est fort pratique pour l'analyse des données et offre une visualisation qui confère plus de souplesse dans l'utilisation de l'interface. On peut également envisager la gestion de couches avec des icônes.

L'application supporte une hiérarchie complexe de couches, avec un nombre de niveaux infini. Pour afficher une couche, il suffit de la sélectionner.

 Une fois votre couche choisie, cliquez sur le bouton "OK" pour actualiser l'affichage.

#### 4.2.5. Recherche (Onglet Recherche)

Cet onglet vous permet de déplacer la zone affichée vers celle que vous voulez. Vous pouvez faire cela de différentes manières :

- En sélectionnant un raccourci vers un lieu géographique ;
- En entrant les coordonnées X et Y d'un point (pour lancer la recherche, tapez sur la touche .).
- En faisant une recherche sur un nom. Pour cela, sélectionnez la couche qui vous intéresse, tapez le mot ou le début du mot qui selon vous correspond à un objet de la couche, et suivez les instructions.

#### 4.2.6. Les outils dessin (Onglet Dessin)

Dans l'application, il est possible de dessiner des points, des lignes, des rectangles ou des polygones sur la carte, et d'y attacher des étiquettes. Ces objets sont persistants : au prochain zoom ou déplacement, ils seront toujours affichés. Cet onglet vous permet d'obtenir des

informations sur les objets dessinés et de basculer vers le mode masque, dans lequel tout est masqué, sauf à l'intérieur de l'objet dessiné.

Pour dessiner les objets voulus, utilisez les outils de dessin

#### 4.2.7. Outils de mesure



On peut également offrir à l'utilisateur le calcul d'un trajet entre plusieurs points de la carte. On trouve très souvent cette fonctionnalité dans des applications de cartes routières ou de recherche d'itinéraires. Les distances et les surfaces peuvent être mesurées sur la carte avec ces outils.

L'accès aux différents éléments de l'application (impression, mise à jour ...) peut être accordé ou refusé selon les droits accordés à l'utilisateur. Les fonctionnalités et les données peuvent être concernées par ces restrictions.

#### 4.2.8. Interrogation des objets (Onglet Requête)

Nous pouvons également permettre aux utilisateurs de visualiser les informations attributaires associés aux objets spatiaux en sélectionnant l'objet en question avec le curseur de souris, en utilisant l'outil Requête , vous pouvez faire une recherche géographique sur des objets. Les objets trouvés sont mis en évidence et leurs attributs sont affichés.

Les requêtes sont persistantes (vous pouvez ajouter de nouveaux objets à ceux déjà sélectionnés).

Note : seulement les couches spécifiées comme interrogeables par l'administrateur du site Web sont concernées par la requête.

Cet onglet vous permet de paramétrer votre requête. Pour chaque couche interrogeable, les options suivantes peuvent être mises :

- **Mise en évidence** : permet d'activer/désactiver la mise en évidence des objets ;
- **Attributs** : si elle est cochée, les attributs des couches peuvent être interrogés ; Sinon, seulement les identifiants des objets seront retournés ;
- **Table** : permet d'afficher le résultat d'une requête ;
- **Union sélection** : quand vous sélectionnez un groupe d'objets, ceux qui étaient déjà sélectionnés le restent et ce qui ne l'était pas encore sont sélectionnés ;
- **Xor** : quand vous sélectionnez un groupe d'objets, ceux qui ont été sélectionnés sont désélectionnés et ceux qui ne l'étaient pas sont sélectionnés (par défaut) ;
- **Intersection** : quand vous sélectionnez un groupe d'objets, seulement ceux qui étaient déjà sélectionnés seront conservés dans la sélection ;
- **Dans la requête** : si cette option est cochée, vous forcez à inclure cette couche dans la requête ;
- **Masque** : si elle est cochée, vous appliquerez un masque au lieu de faire une simple Si l'option "Interroger toutes les couches sélectionnées" est cochée, la requête portera sur toutes les couches interrogeables.

Si l'option "Interroger toutes les couches sélectionnées" est cochée, la requête portera sur toutes les couches interrogeables.

#### 4.2.9. Les outils de mise à jour (Onglet Edition) :

La mise à jour des objets (attributaires ou géométriques) est réservée aux utilisateurs ayant ce droit (cette fonctionnalité est protégée par un mot de passe). Pour cela, cliquez sur le bouton Connexion en haut à droite ou bien dans la zone d'action du module si un message vous demande de vous connecter. La page suivante s'affiche :



Entrez votre identifiant et votre mot de passe puis cliquez sur Soumettre.

Les outils de mise à jour offerts par l'application sont conditionnés par le type de couche à mettre à jour (point, ligne, polygone).

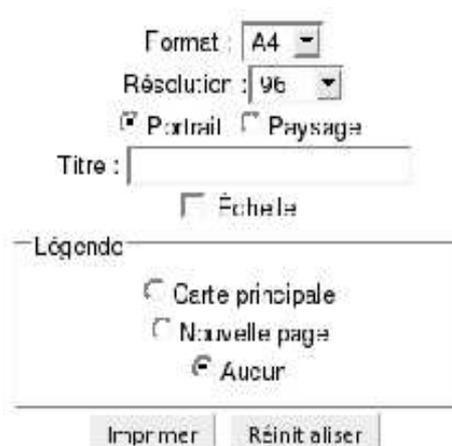
Vous avez des icônes qui sont apparues après avoir choisi la couche à éditer. Ces icônes sont les suivantes :



Dans l'ordre :

- Création d'un nouvel objet (un point dans notre cas, mais il est possible d'avoir une icône ligne et polygone en fonction de la géométrie de la couche  ),
- Edition d'un objet en manipulant ses sommet (vertex) par :
  - Déplacement d'un sommet.
  - Ajout d'un sommet.
  - Suppression d'un sommet.
- Sélection d'un objet.
- Suppression d'un objet.
- L'icône pour afficher l'étendu maximale du projet.
- Activation de l'aimantation.

## 4.2.10. Impression (Onglet Imprimer)



L'application permet des exports au format PDF. Vous pouvez choisir dans cet onglet, les éléments de la carte que vous voulez voir s'afficher (barre d'échelle, vue générale, résultats d'une requête, légende) et des options spécifiques à l'export, il existe différentes options :

- le format : A4, A3, ...
- la résolution : 96, 150, ...
- la disposition : portrait ou paysage ;
- un champ pour entrer le titre de la carte ;
- une option pour afficher l'échelle graphique ;
- une option pour afficher la légende dans une nouvelle page ou dans la même page.

La configuration du système permet de définir plusieurs formats d'export, avec plusieurs configurations possibles. Voici une capture de l'interface du module :

Quand vous avez choisi vos options, vous pouvez cliquer sur le bouton Imprimer. Il y a trois moyens pour obtenir votre fichier en fonction de la configuration du système :

- le fichier PDF s'ouvre dans votre navigateur ;
- une fenêtre s'ouvre pour vous demander si vous voulez ouvrir ou sauver le fichier ;
- enfin une page s'ouvre avec un lien sur lequel cliquer pour télécharger ou ouvrir le fichier.

## 4.3. Test et validation

Pour tester cette application, on a besoin des données que l'on peut grouper en deux types : les données de références et les données thématiques. L'application a été donc testée avec le BD5 FTM du quartier Faliarivo Ambanidia 2<sup>e</sup> Arrondissement Antananarivo Renivohitra.

### □ Le BD5

La BD5 est une BD relationnelle géoréférencé à une échelle de saisie et de précision de 1/5000. Elle est un ensemble structuré d'informations existantes d'une région donnée à grande échelle. C'est-à-dire une entité dans laquelle il est possible d'utiliser directement par les différents utilisateurs. La BD5 est issue de la restitution photogrammétrique de photo aérienne et la numérisation sur des orthophotos avec une précision de 1m environ.

Cette base de données a pour objectif de donner les informations des éléments existants dans un terrain à grande échelle pour mettre à la disposition des entités communales de données structurées.

Les données sont structurées en plusieurs couches d'informations :

- la planimétrie : voie de communication, équipement urbain, limites diverses ;
- l'hydrographie ;
- l'altimétrie ;
- l'occupation du sol.

Il contient aussi plusieurs informations particulières telles que les poteaux électriques, les bornes fontaines, les petites ruelles, les axes ainsi que les bords routes ; les caractéristiques des bâtiments tels que les épiceries, les gargotes, les magasins, les salons de coiffure, les écoles, les églises.

#### ❑ **Situation géographique du quartier Faliarivo**

Le quartier Faliarivo est situé :

- au Nord : le quartier d'Antanimora ;
- au Sud-ouest : le quartier d'Ankazotokana ;
- au Sud-est : le quartier de Miadanarivo ;
- à l'Est : le quartier de Tsiadana ;
- à l'Ouest : le quartier d'Antanimora.

Ce quartier constitue quatre secteurs bien répartis tels que VB, VC, VD, VE et s'étend sur une superficie de 22ha.

L'ensemble de ces tests nous a permis de valider l'ensemble des fonctionnalités de l'application, notamment les fonctionnalités de visualisation. L'accès à la base PostGIS (les utilisateurs, mots de passe) a été également testé.

Outre les tests suscités, d'autres tests ont été effectués sur la plate forme logicielle et matérielle. Ces tests simulent les conditions réelles de fonctionnement de l'application :

Les tests sur la plate forme Logiciel de base :

- Serveur Apache ;
- Serveur d'application PHP ;
- Serveur spatial : CGI, Module PhpMapScript ;
- SGBD PostgreSQL /PostGis.

Les autres tests sont effectués sur l'interface client sur les différentes marques de PC et station disponibles à l'institut, avec les navigateurs Web les plus utilisés (Internet Explorer de Microsoft, Mozilla Firefox).

## CONCLUSION ET PERSPECTIVES

Ce travail rentre dans le cadre de la réalisation au profit du FTM d'une solution logicielle afin de mettre en place un cadre de travail coopératif entre les personnels de ses différentes annexes. L'idée est d'exploiter les potentialités des nouvelles technologies de l'information et de communication via le réseau internet de leur organisme pour moderniser à moindre coût le processus de production cartographique, d'où le recours à l'offre de l'Open Source.

L'information géographique a ses propres spécificités, où les applicatifs issus d'Internet manipulant ce genre d'information doivent prendre en charge la dimension spatiale que représente ce genre d'information. Il s'agit des SIG-Web qui regroupent ce genre d'applicatif. Notre contribution a porté sur la proposition de solutions pratiques pour la mise en place d'un prototype SIG-Web en présentant une étude théorique sur les concepts liés au SIG WEB à savoir : client/serveur, serveur d'application, application CGL serveur spatial ou cartographique. Pour cela un inventaire non exhaustif de l'existant en matière de solutions Web cartographie a été établi en mettant en exergue les avantages et les inconvénients de chaque solution, pour montrer aux géomaticiens les outils potentiels disponibles pour réaliser des systèmes de cartographies sous Web ou SIG-Web.

Le développement de la solution finale, pour des raisons évidentes de clarté et de méthodologie, a nécessité le suivi des étapes classiques du cycle de vie d'une application. En effet, sur le plan:

- ❖ Conceptuel, les différentes architectures des applications sous réseau et sous le Web ont été abordées. La conception du SIG-Web envisagée en architecture trois tiers a été proposée.
- ❖ Pratique, un prototype d'un SIG-Web a été réalisé. Ce prototype prend en charge les tâches classiques des SIG conventionnels, avec possibilité de la mise à jour à distance des données en toute sécurité.
- ❖ Expérimental, une panoplie de logiciels Open Source téléchargés à partir d'Internet a été implémentée. personnalisés et testé et ont donné de bons résultats.

Pour cela, il a fallu identifier puis étudier et enfin architecturer les composants nécessaires au développement de notre solution. Cette dernière a été développée entièrement avec des outils Open Source (PHP, Apache, MapServer, PostgreSQL..), ce qui représente un gain considérable à l'entreprise en matière, d'une part de coût (logiciels gratuits) et d'autre part de valorisation des compétences locale de l'entreprise. Cette solution sera également un support fiable et sécurisé permettant la consultation et la mise à jour des données produites par l'organisme (FTM).

Par ce travail, l'importance des outils libres a été mise en évidence. Il insiste par ailleurs sur la nécessité de disposer de compétences avérées en informatique pour pouvoir bénéficier des outils Open Source et par conséquent assurer la maintenance des solutions ainsi développées.

En guise de perspective, nous préconisons :

- ❖ l'utilisation de notre application via Internet au profit d'un nombre important d'utilisateurs notamment les collectivités locales et les services compétents de l'état, et ce dans le but de leur offrir un outil d'aide à la décision et leur permettre la collecte de données fiables et cohérentes.

- ❖ La sécurisation de cette application et de l'accompagner d'une infrastructure d'authentification solide.

## BIBLIOGRAPHIE

- [1] **A FEDEROV**, “ASP 2.0 ”, Editeur: Eyrolles, France, 2000.
- [2] **André E, Chauvin C.**, 2004. Information géographique et gestion forestière : état des lieux et perspectives. CEMAGREF-Grenoble. France.
- [3] **Baquiast J. P.**, 1999. Utiliser les SIG. Le dossier des SIG, Admiroute. France.
- [4] **BERNIER E, BEDARD Y et LAMBER M**, cartographie sur demande sur le Web et bases de données multidimensionnelles, Revue internationale de géomatiques, 13, n°3, 2003.
- [5] **Blomac F.**, 1994. ArcInfo: concepts et applications en géomatique, Hermès.
- [6] **BOULAHIA Morad**, «Réalisation des terminaux de paiement électronique dans les sites Web dynamiques » MASTER en réseau .INSA Lyon .2000.
- [7] **BOULAHIA Morad**, «Réalisation des terminaux de paiement électronique dans les sites Web dynamiques » MASTER en réseau, INSA, Lyon ,2000.
- [8] **Christophe Tufféry**, « Les SIG dans les entreprises », Hermès, Paris, 1997.
- [9] **CNIG, 2004**. Fiche technique du n° 77 : le logiciel libre et les SIG. Paris, France.
- [10] **CNIG, 2001**. Actualisation fiche 9 – thème de sensibilisation. Paris, France.
- [11] **De Blomac F**, L'open source, véritable alternative pour les SIG ? Document PDF, 4p, 2001.
- [12] **Etienne MOREL**, “ Modélisation Objet avec UML ”, Editeur: Eyrolles, France, 1998.
- [13] **ESRI, 2004**. Applications SIG nomades : ArcPad et périphériques.
- [14] **Grady Booch, James Rumbau et Ivar Jacobson**, « le guide de l'utilisation UML », Eyrolles, 2003.
- [15] **Guiet Y.**, 2004. SIG et cartes : enjeux et initiatives. Qu'est ce qu'un SIG ? Académie de Limoges.
- [16] **Goodchild M. F., Kemp K. K., Thériault M., Roche Y.**, 1996. Systèmes d'information géographique, Notes de cours Volume1, Notions de base. LATIG, Département de géographie, Université Laval, Québec.
- [17] **Jonglez D, Lestrat L**, la cartographie sur Internet : introduction à un état de la technique, <http://www.sig.cwriter.org>
- [18] **Kraak M et Brown A, EDS**, Web cartography, Londres, Taylor and Francis, 2001.
- [19] **Kraak M**, setting and needs for Web cartography, Web cartography, Londres, Taylor and Francis, 2001.
- [20] **Laurent Jégou**, Cartographie et SIG interactifs en ligne, Séance 3 : Approfondissement de l'utilisation de MapServer, décembre 2009.
- [21] **Maguire D., Goodchild M. F. and Rhind D.W.Eds.**, 1991. Geographical Information Systems : Principles and Applications, Longman, London.
- [22] **Marshall J**, developing internet-based GIS application, <http://www01.gisafe.com>
- [23] **Marie COUTARD, Jean-Pascal KLIPFEL, Samuel BLANC**, « Recherche de solutions applicatives open source autour de mapserver », Projet DESS SIGMA du 24/01/05 au 25/02/05.
- [24] **Mohamed Sami Ghanem et Dr. Mohamed-Khireddine KHOLLADI**, Solution CartoWeb pour le Web Mapping, [www.asd-kholladi.org](http://www.asd-kholladi.org)
- [25] **Nathalie LOPEZ, Jorge MIGUEIS et Emmanuel PICHON**, “ Intégrer UML dans vos projets ”, Editions Eyrolles, 1997.

[26] **Neumann A et Winter A.M**, la cartographie en mode vectoriel sur le Web: les possibilités de SVG, <http://carto.net/papers/svg/>

[27] **Robert Laurini, Françoise Milleret-Raffort**, « Les bases de données en géomatique »; Hermes, 1993.

## WEBOGRAPHIE

### ❑ **OGC (Open Geospatial Consortium) et OpenSources**

Site officiel de l'OGC : <http://www.opengeospatial.org/>

Site officiel DM Solutions Group : <http://www.dmsolutions.ca/>

Guide pour distribuer des données via le protocole WMS :

<http://oceanesip.jpl.nasa.gov/esipde/guide.html>

Schémas de l'OGC : <http://schemas.opengis.net>

### ❑ **Open Source**

Site officiel : <http://www.opensource.org/>

Ressource pour les développeurs de l'Open Source : <http://sourceforge.net/>

Projet de promotion de l'Open Source pour l'information géographique : <http://www.freegis.org>

Site SIGLE (Systèmes d'Information Géographique LibRE), espace de diffusion, de, transfert et d'échange dans les domaines des SIG libres : <http://www.projet-sigle.org/index.php3>

Banque de données géospatiales canadiennes en accès libre : <http://geogratis.cgdi.gc.ca/>

### ❑ **Clients cartographiques légers :**

OpenLayers – Site Officiel : <http://openlayers.org/>

MapFish – Site Officiel : <http://trac.mapfish.org/trac/mapfish/wiki>

Cartoweb – Site Officiel : <http://www.cartoweb.org/>

Cartoweb Communauté Francophone : <http://cartoweb-community.globe.org/doku.php?id=fr:start>

### ❑ **Sites Internet et forums sur les technologies Open Source :**

#### **Généralistes :**

Forum sur les SIG: <http://forumsig.org>

Site de développement de logiciels Open Source: <http://sourceforge.net>

Site sur les SIG libres : <http://sig-libre.org>

#### **Mapserver**

Site officiel de MapServer : <http://mapserver.gis.umn.edu/>

Répertoire de téléchargement des versions de Mapserver et programmes associés

<http://dl.maptools.org/dl/>

Nouvelles documentation de Mapserver : <http://users.frii.com/sgillies/projects/mapscript/>

Documentation de MapServer : <http://terrasip.gis.umn.edu/projects/tutorial>

Description du mapfile (Neapoljs) : <http://umn.mapserver.ch>

MapServer et la symbologie : <http://mapserver.gis.umn.edu/doc44/cartographic-symbols.html>

#### **PostgreSQL/PostGIS**

Site officiel de PostgreSQL : <http://www.postgresql.org/>

Quelques outils pour postgres et PostGIS, regrouper et mis en ligne par Jean David Techer :

<http://www.01map.com/download/>

PostgreSQL, site de la communauté francophone autour du SGBDR PostgreSQL, *français* :

<http://www.postgresqlfr.org/>

PostGIS, site de la communauté francophone autour du cartouche spatial de PostgreSQL,

PostGIS, *français* : <http://www.postgis.fr/>

Articles et tutoriaux sur PostGIS (site de Jean-David TECHER) : <http://techer.pascal.free.fr/>

### **Logiciels SIG :**

Site référence de Quantum GIS: <http://qgis.org>

Projet-SIGEE (Systèmes d'information Géographique Libre) sur les logiciels SIG libres, et en particulier sur OpenJUMP et GvSIG : <http://www.projet-sigle.org>

Site de Thuban : <http://thuban.intevation.org>

Site du logiciel Udig: <http://udig.refractions.net/confluence/display/UDIG/Home>

### **❑ Applications autour de Mapserver**

Pour télécharger PHP, Apache : <http://www.phpindex.com/telechargement/index.php3>

Amein : <http://www.terrestris.de/hp/en/downloads.php>

CartoWeb : <http://www.cartoweb.org>

Maptools : <http://www.maptools.org>

MapStorer : <http://www.mapstorer.org/eng/main.html>

P.mapper : <http://pMapper.sourceforge.net/>

Wortomap : <http://www.tydac.ch/download/wor2map.zip>

Chameleon, <http://chameleon.maptools.org/>

### **❑ Aide pour le développement :**

**Aide MapFile**, Tutorial pour la rédaction d'un MapFile, un bon complément du site officiel de MapServer, *français* : [http://fa.vdb.free.fr/MapServer/doc/mapfile-reference\\_fr.html](http://fa.vdb.free.fr/MapServer/doc/mapfile-reference_fr.html)

**Gdal**, Lien vers les forums et les téléchargements de la bibliothèque de fonctions, *anglais*.

<http://www.gdal.org/>

**Projection**, Code EPSG de toutes les projections, *anglais* :

<http://spatialreference.org/ref/epsg/?page=14>

**OSGEO**, Projet de création et de maintenance de format de données libres, *anglais* :

<http://asgeo.netlantis.org/>

**Refraction**, Code et aide pour PostGIS et MapServer, *anglais* : <http://www.refractions.net/>

**BostonGIS**, Excellent site sur PostGIS et sa liaison avec MapServer, *anglais* :

<http://www.bostongis.com/>

**PHP France**, développement PHP, *français* : <http://www.phpfrance.com/>

**Le php facile**, développement PHP et apprentissage, *français* : <http://www.lephpfacile.com/>

**Php astux**, développement PHP et apprentissage, *français* : <http://www.php-astux.info/>

Documentation pour CartoWeb, Yves Jacolin, *français* : <http://www.alolise.org/wiki/index.php>

Découverte de mapserver : [www.sig-gps.net](http://www.sig-gps.net)

Description illustrée de la structure des MapFiles : <http://www.mapserver.org/mapfile/>

Documentation PHP : <http://www.nexen.org>

Dictionnaire de la cartographie : <http://www.microplan.fr/Dictionnaire.htm>

Forum de la cartographie sur Internet : <http://forumsig.symen.ch/>

Forum PHP, Javascript : <http://www.mangue.org>

Liste d'utilisateurs de MapServer : <http://lists.maptools.org/mailman/listinfo/maplab-users>

❑ **Exemples d'applications cartographiques sur le Web :**

Galerie de sites : <http://www.directionsmag.com/webmapgallery/?industry=0&page=1>

Eoportals : <http://maps.eoportal.org/>

Centre Suisse de la Cartographie de la Faune : <http://lepus.unine.ch/carto/>

CitéVision de Renne : <http://www.citevisions.rennes.fr/>

Exemple d'utilisation de CartoWeb : <http://www.cartoweb.lom.camcom.it/>

# **ANNEXES**

## ANNEXE 1 : Le Fichier « faliarivo.map »

```
MAP
NAME "Faliarivo"
EXTENT 515701.950000 797311.513896 516330.443654 797978.820000
IMAGETYPE PNG
IMAGECOLOR 255 255 255
STATUS ON
UNITS METERS
FONTSET "fonts.txt"
SYMBOLSET "symbols.txt"
SHAPEPATH"C:/ms4w/apps/cartoweb3/projects/faliarivo/server_conf/faliarivo/data"
OUTPUTFORMAT
  NAME png
  DRIVER "GD/PNG"
  MIMETYPE "image/png"
  IMAGEMODE PC256
  EXTENSION ".png"
  FORMATOPTION "INTERLACE=OFF"
  TRANSPARENT OFF
END
OUTPUTFORMAT
  NAME jpeg
  DRIVER "GD/JPEG"
  MIMETYPE "image/jpeg"
  IMAGEMODE RGB
  EXTENSION ".jpg"
  FORMATOPTION "INTERLACE=OFF"
  FORMATOPTION "QUALITY=80"
END
OUTPUTFORMAT
  NAME GTiff
  DRIVER "GDAL/GTiff"
  MIMETYPE "image/tiff"
  IMAGEMODE RGB
  FORMATOPTION "TFW=YES"
  FORMATOPTION "COMPRESS=PACKBITS"
END
REFERENCE
  IMAGE images/reference.png
  EXTENT 515701.950000 797311.513896 516330.443654 797978.820000
  STATUS ON
  COLOR -1 -1 -1
  OUTLINECOLOR 200 0 0
  SIZE 290 220 #220 150
END
SCALEBAR
  POSTLABELCACHE TRUE
  STYLE 0
  UNITS meters
  SIZE 350 3
  TRANSPARENT TRUE
  COLOR 77 77 88
  IMAGECOLOR 242 255 195
  BACKGROUNDCOLOR 222 222 222
  LABEL
    TYPE BITMAP
    SIZE SMALL
    COLOR 0 0 0
    POSITION UR
    BUFFER 10
  END
END
PROJECTION
  "init=epsg:29700"
  "init=epsg:29702"
END
LEGEND
  KEYSIZE 20 15
END
```

```

#====START OF LAYER SECTION =====#
LAYER
  NAME 'COUP_105_O'
  TYPE RASTER
  EXTENT 515701.950000 797311.513896 516330.443654 797978.820000
  DATA'C:/ms4w/apps/cartoweb3/projects/faliarivo/server_conf/faliarivo/data/COUP_105_O.tif'
  METADATA
    "force_imagetype" "jpeg"
  END
  STATUS ON
END # LAYER
#
# LAYER quartier_faliarivo
#
LAYER
  NAME 'quartier_faliarivo'
  TYPE POLYGON
  TEMPLATE "tnt"
  EXTENT 515701.950000 797311.513896 516330.443654 797978.820000
  #DATA 'quartier_faliarivo'
  CONNECTIONTYPE postgis
  CONNECTION "dbname='TEST_GIS' host=127.0.0.1 port=5432 user='postgres' password='1234567'"
  DATA 'the_geom FROM "quartier_faliarivo" USING UNIQUE gid USING srid=29702'
  STATUS OFF
  TRANSPARENCY 100
  OPACITY 0
  PROJECTION
    "init=epsg:29700"
  END
  CLASS
    NAME 'quartier_faliarivo'
    STYLE
      WIDTH 4
      OUTLINECOLOR 0 0 0
      COLOR 85 255 255
    END
    TEMPLATE void
  END
END # LAYER
#
# LAYER bati_faliarivo
#
LAYER
  NAME 'bati_faliarivo'
  TYPE POLYGON
  TEMPLATE "tnt"
  EXTENT 515726.595644 797332.700000 516326.200000 797966.459720
  #DATA 'bati_faliarivo'
  CONNECTIONTYPE postgis
  CONNECTION "dbname='TEST_GIS' host=127.0.0.1 port=5432 user='postgres' password='1234567'"
  DATA 'the_geom FROM "bati_faliarivo" USING UNIQUE gid USING srid=29702'
  STATUS OFF
  TRANSPARENCY 100
  PROJECTION
    "init=epsg:29700"
  END
  LABELITEM "nom"
  CLASS
    NAME 'bati_faliarivo'
    STYLE
      WIDTH 2
      OUTLINECOLOR 0 0 0
      COLOR 255 128 0
    END
  END

```

```

        LABEL
        COLOR 10 10 255
        SHADOWCOLOR 175 175 100
        #SHADOWSIZE 0 0
        TYPE TRUETYPE
        FONT VeraBd
        SIZE 10
        ANTIALIAS TRUE
        POSITION CC
        PARTIALS FALSE
        MINDISTANCE 150
        END
    END
END # LAYER
#
# LAYER limite_lin_faliarivo
#
LAYER
    NAME 'limite_lin_faliarivo'
    TYPE LINE
    TEMPLATE "ttt"
    EXTENT 515701.950000 797311.513896 516330.443654 797978.820000
    #DATA 'limite_lin_faliarivo'
    CONNECTIONTYPE postgis
    CONNECTION "dbname='TEST_GIS' host=127.0.0.1 port=5432 user='postgres'
password='1234567'"
    DATA 'the_geom FROM "limite_lin_faliarivo" USING UNIQUE gid USING srid=29702'
    STATUS OFF
    TRANSPARENCY 100
    PROJECTION
    "init=epsg:29702"
    END
    CLASS
    NAME 'limite_lin_faliarivo'
    STYLE
    WIDTH 4
    COLOR 90 229 210
    END
    TEMPLATE void
    END
END # LAYER
#
# LAYER line_faliarivo
#
LAYER
    NAME 'line_faliarivo'
    TYPE LINE
    TEMPLATE "ttt"
    EXTENT 515701.950000 797305.290000 516330.443654 797978.820000
    #DATA 'line_faliarivo'
    CONNECTIONTYPE postgis
    CONNECTION "dbname='TEST_GIS' host=127.0.0.1 port=5432 user='postgres'
password='1234567'"
    DATA 'the_geom FROM "line_faliarivo" USING UNIQUE gid USING srid=29702'
    STATUS OFF
    TRANSPARENCY 100
    PROJECTION
    "init=epsg:29700"
    END
    CLASS
    NAME 'line_faliarivo'
    STYLE
    WIDTH 1
    COLOR 255 0 0
    END
    END
END # LAYER

```

```

LAYER
NAME "cartoweb_point_outline"
TYPE POINT
CLASS
STYLE
SYMBOL "circle"
COLOR 0 0 205
SIZE 10
END
LABEL
TYPE TRUETYPE
FONT "Vera"
SIZE 7
COLOR 0 0 0
OUTLINECOLOR 255 255 255
POSITION lc
END
END
END

LAYER
NAME "cartoweb_line_outline"
TYPE LINE
TRANSPARENCY 100
CLASS
STYLE
OUTLINECOLOR 255 0 0
SYMBOL "line-dashed"
SIZE 3
END
LABEL
TYPE TRUETYPE
FONT "Vera"
SIZE 7
COLOR 0 0 0
OUTLINECOLOR 255 255 255
ANGLE auto
POSITION uc
END
END
END

LAYER
NAME "cartoweb_polygon_outline"
TYPE POLYGON
TRANSPARENCY 60
CLASS
STYLE
COLOR 255 153 0
OUTLINECOLOR 0 0 0
END
LABEL
TYPE TRUETYPE
FONT "Vera"
SIZE 7
OUTLINECOLOR 255 255 255
COLOR 0 0 0
POSITION cc
END
END
END # MAP

```

## ANNEXE 2 : La projection pour Madagascar

Dans chaque pays, il existe une projection officielle fixée par l'Etat et qui devient dès lors la projection de référence. La cartographie de Madagascar utilise une projection local la projection Laborde. Cette projection appartient à la catégorie des projections conformes (conserve les angles) et se nomme Laborde Madagascar. Très proche de la projection de type Mercator Oblique, elle est bâtie à partir de l'Ellipsoïde International 1924. D'après le Foiben-Taosarintanin' i Madagasikara (FTM, Institut Géographique et Hydrographique National), la différence entre la projection Laborde et la projection Mercator Oblique n'excède pas 11 m dans le pire des cas à Madagascar.

Il s'agit de sujets réellement intéressant, je vous recommande très activement la lecture du mémoire de fin d'études de monsieur Misan'ny farany Nirina Andrianarison, intitulé « *Tout sur la projection Laborde et l'utilisation du GPS à Madagascar* », très complet à ce sujet.

### Centre de projection :

◆ **Coordonnées géographiques :**

$$\varphi_0 = -21 \text{ grades}, 18^{\circ}56' \text{Sud}$$

$$\lambda_0 = 49 \text{ gr Est de paris}$$

◆ **Coordonnées en représentation :**

$$X_{V0} = 400\,000 \text{ m}$$

$$Y_{V0} = 8000\,000 \text{ m}$$

Par rapport à Greenwich, il faut ajouter la longitude de Paris par rapport à Greenwich  $2^{\circ}20'14''.025$  ou  $2.5969213$  gr. Donc,  $\lambda_0 = 46^{\circ}26'14.025''$  Est de Greenwich ou  $51,5969213$  gr Est de Greenwich.

### Point fondamental (Observatoire Ambohidepona) :

Le point fondamental du système de projection Laborde Madagascar est matérialisé par le pilier de l'astrolabe auprès de l'observation d'Antananarivo. Ce point fondamental a comme coordonnées :

◆ **Coordonnées géographiques :**

$$M_0 = -21^{\circ}01'8228828 \text{ gon Sud}$$

$$L_0 = 50,238295075 \text{ gon Est}$$

◆ **Coordonnées cartésiennes :**

$$X_V = 517\,353.56 \text{ m}$$

$$Y_V = 797\,717.34 \text{ m}$$

$X_V$  vers l'Est et  $Y_V$  vers le Nord.

### Coefficient de réduction d'échelle : $k_0 = 0,9995$

## **RESUME**

Le travail réalisé est exposé de manière synthétique, dans le but d'aborder les différents concepts et méthodes appliqués pour la mise en place d'un portail cartographique.

L'objectif de ce travail a donc été de développer un outil permettant de faciliter l'accès aux données, par la dématérialisation et le partage de l'information. De plus, le projet rentre dans la volonté du FTM de diversifier ses activités en lien avec une demande croissante de la part des collectivités territoriales.

Comme le marché ne propose pas de solution entièrement satisfaisante, il s'avère nécessaire d'en développer une. Après une étude de l'existant, il apparaît que les logiciels libres offrent les meilleures possibilités, surtout d'un point de vue du développement.

Le portail cartographique se veut simple d'accès et d'utilisation, et est totalement dégagé des contraintes de propriété. En cela, l'aspect Open Source du projet est l'une des prérogatives essentielles au choix des outils utilisés.

## **SUMMARY**

The job accomplished during this training period is displayed in a synthetic way, with the intention of approaching different concepts and methods applied to the development of the cartographic portal.

The goal of this job was to develop a tool permitting an easier access to data, by dematerializing and sharing of information. This project answers the wish of the FTM that is to diversify his activities and new request from local administration services.

Market doesn't offer fully satisfactory solution, so it appears necessary to develop such a system. A study of existing solutions showed that free software proposes the best possibilities from the development point of view.

The cartographic portal should have a simple access and use, and be absolutely devoid of license. In it, the Open Source aspect of the project is one of the essential prerogatives for tools chosen.



**Nom** : ANDRIMPIRENENA  
**Prénoms** : Zolalaina Herinarivo Joseph  
**Titre** : Recherche d'une solution applicative open source pour la réalisation d'un projet Webmapping  
**Nombre de pages** : 94  
**Nombre de figures** : 36  
**Nombre de tableaux** : 5

### RESUME

Le travail réalisé est exposé de manière synthétique, dans le but d'aborder les différents concepts et méthodes appliqués pour la mise en place d'un portail cartographique.

L'objectif de ce travail a donc été de développer un outil permettant de faciliter l'accès aux données, par la dématérialisation et le partage de l'information. De plus, le projet rentre dans la volonté du FTM de diversifier ses activités en lien avec une demande croissante de la part des collectivités territoriales.

Comme le marché ne propose pas de solution entièrement satisfaisante, il s'avère nécessaire d'en développer une. Après une étude de l'existant, il apparaît que les logiciels libres offrent les meilleures possibilités, surtout d'un point de vue du développement.

Le portail cartographique se veut simple d'accès et d'utilisation, et est totalement dégagé des contraintes de propriété. En cela, l'aspect Open Source du projet est l'une des prérogatives essentielles au choix des outils utilisés.

**MOTS CLEFS:** SIG, WEBMAPPING, SGBD, LOGICIEL LIBRE, OPEN SOURCE

### SUMMARY

The job accomplished during this training period is displayed in a synthetic way, with the intention of approaching different concepts and methods applied to the development of the cartographic portal.

The goal of this job was to develop a tool permitting an easier access to data, by dematerializing and sharing of information. This project answers the wish of the FTM that is to diversify his activities and new request from local administration services.

Market doesn't offer fully satisfactory solution, so it appears necessary to develop such a system. A study of existing solutions showed that free software proposes the best possibilities from the development point of view.

The cartographic portal should have a simple access and use, and be absolutely devoid of license. In it, the Open Source aspect of the project is one of the essential prerogatives for tools chosen.

**KEY WORDS:** GIS, WEBMAPPING, DBMS, OPEN SOURCE, FREE SOFTWARE

**Rapporteurs** : Monsieur RAJAONARISON Jean Désiré, Ingénieur Civil Géographe, FTM  
Monsieur RABEMALAZAMANANA, chef de Département photogrammétrie, FTM

**Adresse de l'auteur** : Lot VS 10 BC Bis Ambolonkandrina ANTANANARIVO 101

**Contacts** : 0328795238/0331974531/jo\_lalaina@yahoo.fr