

---

# Compte rendu d'activité au sein d'ITREC GESTION

-

## L'engouement des sociétés informatiques pour les technologies Open Source

---



## REMERCIEMENTS

Je tiens à remercier tout particulièrement les personnes suivantes :

**Monsieur *Frédéric Martinez***, ingénieur ENIB et chef de projet chez ITREC GESTION. Désigné comme mon tuteur au sein de l'entreprise depuis le mois d'avril 2004, je tiens à le remercier pour m'avoir encadré durant toute cette période et pour m'avoir formé sur les technologies Open Source JAVA J2EE. Je le remercie également pour tous les conseils qu'il m'a prodigués, pour les corrections qu'il a apportées à mon mémoire, et pour sa grande gentillesse.

**Monsieur *Daniel Ghienne***, directeur technique chez ITREC GESTION. Je le remercie pour m'avoir proposé cette mission au sein de la société, pour ses conseils et sa gentillesse, et pour m'avoir permis de découvrir le monde du service au sein des collectivités locales.

Je tiens enfin à remercier l'ensemble du personnel d'ITREC Gestion pour leur accueil sympathique et chaleureux.

## CONVENTIONS



Ce sigle invite le lecteur à consulter l'annexe correspondante.

\* Ce symbole invite le lecteur à se reporter au lexique pour la définition des mots concernés.

Le texte ainsi encadré reprend les idées fortes des paragraphes précédents.

# SOMMAIRE

<b>INTRODUCTION</b> .....	5
<b>I – CONTEXTE ET OBJECTIFS DU PROJET</b> .....	6
1.1 CARTE D’IDENTITE DU GROUPE ADELIOR .....	6
1.1.1 Entité juridique et organisationnelle .....	6
1.1.2 Un groupe ouvert sur l’Europe .....	8
1.2 CARTE D’IDENTITE DES PRINCIPALES FILIALES D’ADELIOR .....	11
1.2.1 La société ITREC .....	11
1.2.2 La société ITREC GESTION .....	12
1.2.3 La société INFEO .....	23
1.3 LE CONTEXTE DU PROJET .....	24
1.3.1 Déroulement chronologique des missions .....	24
1.3.2 Les sujets traités et les résultats obtenus .....	25
<b>II – L’ENGOUEMENT DES SOCIETES INFORMATIQUES POUR LES TECHNOLOGIES OPEN SOURCE</b> .....	34
2.1 LES TECHNOLOGIES OPEN SOURCE .....	34
2.1.1 Définition de l’Open Source .....	34
2.1.2 Les créateurs du concept Open Source .....	35
2.1.3 Les critères d’une licence Open Source .....	36
2.1.4 L’intérêt des technologies Open Source pour les industriels .....	37
2.2 LES SUCCES BASES SUR L’OPEN SOURCE .....	39
2.2.1 Qui contribue au développement de l’Open Source ? .....	39
2.2.2 De fameux projets basés sur l’Open Source .....	39
2.3 UTILISATION DE L’OPEN SOURCE AU SEIN D’ITREC GESTION .....	40
2.3.1 Pourquoi utiliser l’Open Source dans un projet de développement ? .....	40
2.3.2 Les technologies utilisées dans le développement de GIMA WEB .....	45
2.4 EXPERIENCE RETIREE DE L’UTILISATION DE CES TECHNOLOGIES .....	56
<b>CONCLUSION GENERALE DE L’ETUDE</b> .....	57
<b>ANNEXES</b> .....	59
<i>Les synthèses financières dans GIMA WEB - Code source d’exemple</i> .....	60
<i>Le framework de tests unitaires JUNIT - Code source d’exemple</i> .....	75
<i>General Public License ou GPL</i> .....	76
Introduction .....	76
Licence Publique Générale GNU Version 2, Juin 1991 .....	76
Préambule .....	76
Stipulations et conditions relatives à la copie, la distribution et la modification .....	77
Comment appliquer ces directives à vos nouveaux programmes .....	81
<b>SOURCES / WEBOGRAPHIE</b> .....	83
<b>GLOSSAIRE</b> .....	86

## INTRODUCTION

Ce mémoire de fin d'étude s'intègre dans le cadre de la formation d'ingénieur par apprentissage suivie au centre de formation à la CCI de Rodez. Il présente et fait le bilan de ma dernière période au sein de la filiale ITREC GESTION du groupe ADELIOR.

ADELIOR est constitué de plusieurs filiales présentes en France comme à l'international : ITREC (Paris), ITREC GESTION (Paris/Toulouse), EUVOXA (Nantes), BENELUX (Belgique), ALGORIEL (Paris), OPSI (Paris), SPIDER BUSINESS (Bruxelles/Clermont Ferrand) et INFEO (Paris/Toulouse).

ITREC GESTION est un éditeur de logiciel qui développe depuis plusieurs années GIMA, un logiciel de gestion de patrimoine bâti. En avril 2004, suite à un rapprochement des sociétés, INFEO, la filiale où j'ai effectué la première partie de mon apprentissage, a déménagé et s'est installée dans les mêmes locaux qu'ITREC GESTION à Labège Innopole. Ces deux sociétés sont sous la responsabilité du directeur technique de l'agence de Toulouse, monsieur Daniel GHIENNE, lui-même placé sous l'autorité de la directrice technique du groupe, madame Pascale GIMET et de monsieur Vincent ROUAIX, président directeur général.

Depuis le mois d'avril 2004 je travaille pour ITREC GESTION sous la responsabilité de monsieur Frédéric MARTINEZ, ingénieur ENIB (Ecole Nationale d'Ingénieurs de Brest) en charge du développement du logiciel GIMA WEB (Gestion Intégrée des Magasins et des Ateliers). Cette dernière période passée dans le cadre de mon contrat d'apprentissage m'a permis de travailler avec monsieur MARTINEZ sur le développement de cette application.

Ce dossier reprend les tâches qui m'ont été confiées durant cette période ainsi que le thème de réflexion développé. J'ai ainsi adopté un plan scindé en deux parties distinctes afin de clarifier mon étude : la première partie du mémoire est le compte rendu de la dernière période de mon apprentissage ; l'entreprise est présentée de manière détaillée, ainsi que l'environnement matériel et logiciel ; le déroulement chronologique de la période, les sujets traités et les résultats obtenus y sont également définis.

La seconde partie est un travail de réflexion sur le thème de l'Open Source ; une étude sur l'engouement des sociétés informatiques pour ces technologies sera menée au travers d'exemples et de cas concrets. Les outils Open Source utilisés dans le développement du progiciel GIMA WEB seront également présentés. Enfin je joindrai en annexe les documentations qui m'ont été fournies ainsi qu'un lexique regroupant le vocabulaire technique employé, avec les définitions qui s'y rapportent.

La conclusion de l'étude s'attachera à reprendre les points essentiels de ce mémoire en les synthétisant et en les agrémentant de remarques.

# I – CONTEXTE ET OBJECTIFS DU PROJET

## 1.1 CARTE D'IDENTITE DU GROUPE ADELIOR

### 1.1.1 Entité juridique et organisationnelle

Le groupe ADELIOR regroupe des investisseurs privés et institutionnels autour d'un projet de "Build Up". ADELIOR englobe les sociétés ITREC, ITREC GESTION, EUVOXA, BENELUX, OPSI, ALGORIEL, SPIDER BUSINESS et INFEO. Intégrés dans cette démarche de croissance, ITREC et ITREC GESTION constituent les bases du développement d'ADELIOR autour de leurs domaines d'expertises.

Organisation juridique (Source ADELIOR 2005)

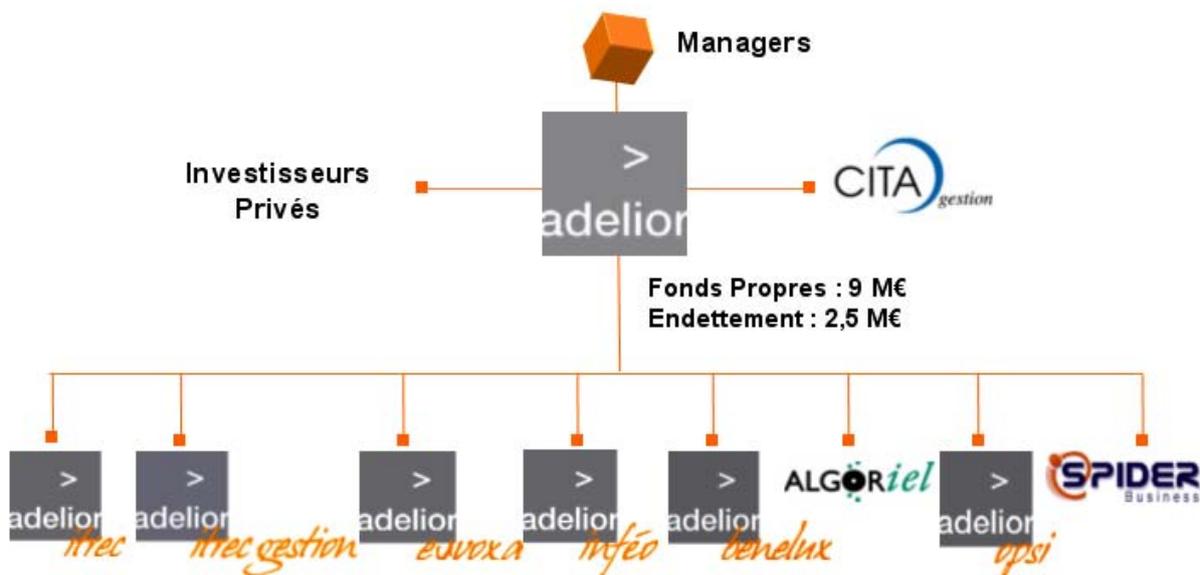


Figure 1 – Organisation juridique du groupe ADELIOR

Le groupe ADELIOR réalise un chiffre d'affaire de 48M€ pour un effectif de 700 employés :

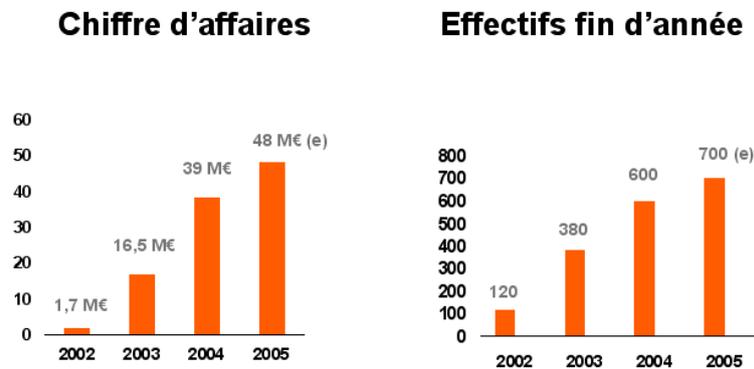


Figure 2 – Chiffre d'affaire et effectif du groupe sur l'année 2005

### Organigramme général du management ADELIOR

Source ADELIOR 2005

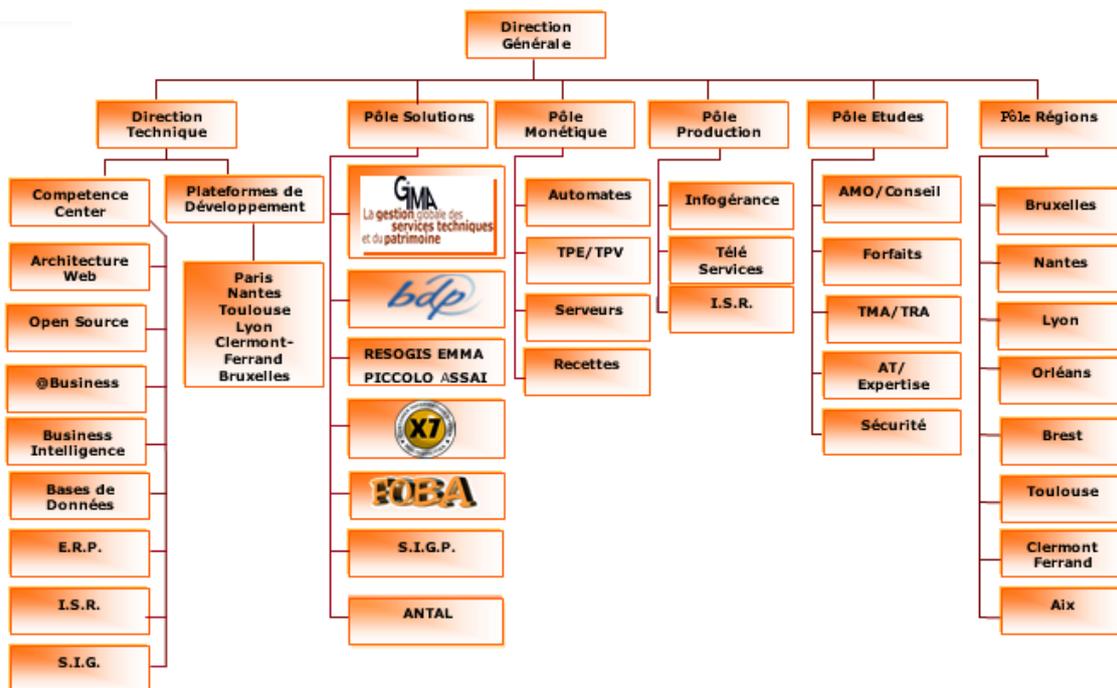


Figure 3 – Organigramme général du management du groupe ADELIOR

## 1.1.2 Un groupe ouvert sur l'Europe

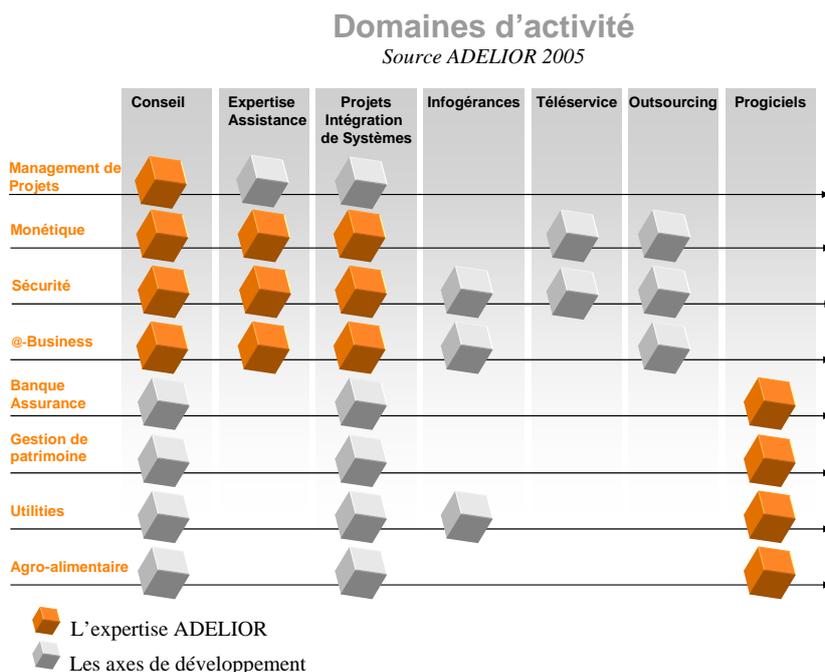
### 1.1.2.1 Implantations

La holding ADELIOR compte sept établissements sur le territoire national et deux implantés en Europe (Belgique et Pays-Bas).



Figure 4 – Carte des implantations du groupe ADELIOR

### 1.1.2.2 Les activités et services



ADELIOR positionne son savoir-faire sur des secteurs d'activité liés aux services et à la Tierce Maintenance. La société tend vers une couverture globale des prestations dédiées au système d'information. L'offre est axée sur les métiers des clients, allant des études à la production en passant par le conseil, l'accompagnement, la formation et les progiciels.

### 1.1.2.3 Les clients

ADELIOR propose ses services auprès des collectivités locales (Mairies, Conseils Régionaux...) mais également auprès des établissements privés et publics :

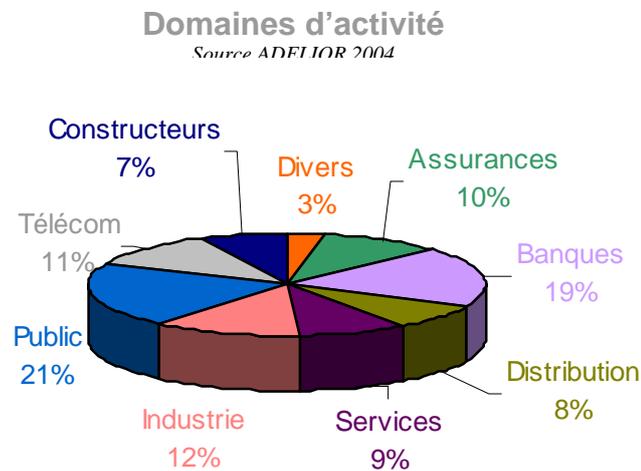


Figure 6 – Domaines d'activité du groupe ADELIOR

### 1.1.2.4 L'offre Produits

L'offre Produits comprend un ensemble de logiciels de gestion à destination des secteurs suivants :

- **Banques/Assurances**

- ↳ SimuEL : simulation d'Épargne Logement,
- ↳ SimuASSUR : simulation d'Assurance,
- ↳ SimuRET : simulation de Retraite,
- ↳ SimuPREV : simulation de Prévoyance.

- **Agro-alimentaire**

- ↳ Logiciel de gestion de production animale pour les coopératives généralistes et les groupements spécialisés.

- **Collectivités locales/Privés**

- ↪ GIMA : solution globale de gestion des magasins, ateliers, interventions et patrimoine,
- ↪ BDP : solution de gestion géographique du patrimoine,
- ↪ RESOCAD : solution de gestion géographique des réseaux (eau, électricité...),
- ↪ URBACAD : solution de gestion géographique appliquée à la maintenance urbaine,
- ↪ VIACAD : solution de gestion géographique appliquée à la voirie,
- ↪ PICOLO : modélisation des réseaux d'eau potable,
- ↪ EMMA : auto surveillance et diagnostic permanent des réseaux d'assainissement,
- ↪ ASSAI : gestion technique et administrative des installations d'assainissement non collectif.

- **Gestion de clientèle**

- ↪ Progiciel X7 : gestion de la relation client et gestion de la facturation et du recouvrement.

## 1.2 CARTE D'IDENTITE DES PRINCIPALES FILIALES D'ADELIOR

### 1.2.1 La société ITREC

Créée en 1979, la société ITREC "*Informatique Temps Réel Études & Conception*" spécialisée dans les domaines des réseaux et télécoms, s'est enrichie en 1985 d'une deuxième société "ITREC GESTION" dédiée à la Tierce Maintenance Applicative\*, aux techniques Client/Serveur et aux Grands Systèmes et Bases de Données.

La filiale ITREC compte aujourd'hui 130 collaborateurs pour un chiffre d'affaire de 11 M€

Depuis la création du groupe ITREC, les dirigeants, ingénieurs et techniciens, participent à une politique d'entreprise basée sur quatre orientations majeures :

- Un savoir-faire pluridisciplinaire permettant de répondre aux besoins d'un schéma directeur informatique,
- La mise en place d'entités de taille humaine facilitant un management responsable et efficace,
- Des investissements continus dans les secteurs de haute technologie assurant la pérennité de l'entreprise,
- Des actions de formation permanente aux techniques informatiques.

Le groupe ITREC développe ses activités dans les domaines suivants :

- Les automates bancaires,
- La monétique,
- Carte à microprocesseurs,
- La sécurité,
- Les méthodes,
- Les systèmes centraux,
- Les architectures distribuées,
- Intranet/Extranet,
- Conseil.

Au mois d'octobre 2002, ITREC et ITREC Gestion ont rejoint le groupe ADELIOR, regroupant des investisseurs privés et institutionnels autour d'un projet de "Build Up".

## 1.2.2 La société ITREC GESTION

C'est au sein de l'équipe ITREC GESTION que s'est déroulée la seconde partie de mon apprentissage. J'ai intégré l'équipe de développement de GIMA / GIMA WEB à mon retour en entreprise au mois d'avril 2004.

### *1.2.2.1 L'équipe*

L'équipe ITREC GESTION se compose de 10 personnes organisées de la manière suivante:

#### GIMA WEB :

- ➔ *Frédéric Martinez* : chef de projet, il conçoit le produit GIMA WEB sur une plateforme J2EE Open Source. C'est aussi un expert NAT SYSTEM.
- ➔ *Ghislain Borie* : chef de projet, il gère le club utilisateur de GIMA, travaille sur le développement de l'*Observatoire du patrimoine\** sur GIMA WEB, et s'occupe des interactions entre GIMA et BDP.

#### GIMA Client/Serveur :

- ➔ *Emmanuel Parant* : chef de projet et responsable de la mairie de Toulouse, il possède d'excellentes connaissances en GMAO (Gestion de Maintenance Assistée par Ordinateur) et contribue au développement de GIMA Client/Serveur. Il gère l'encadrement autour du langage Power Builder et travaille également sur la génération d'SQL en dynamique.
- ➔ *Fabrice Cavagnara* : chef de projet et responsable de la mairie de Lyon, il contribue au développement de GIMA Client/Serveur.
- ➔ *Jean Louis Baule* : il a travaillé sur la Hot Line GIMA et contribue aujourd'hui au développement de nouvelles versions de GIMA Client/Serveur ; il corrige également les bugs présents dans l'application, et forme les clients sur le logiciel.
- ➔ *Odile Higashiyama* : elle travaille à Paris sur le développement de GIMA. C'est l'une des plus anciennes personnes travaillant sur le projet.
- ➔ *Emmanuel Boudet* : expert sur Business Object (outil de reporting), il s'occupe de la gestion des requêtes SQL dans GIMA.
- ➔ *Patrick Orget* : administrateur réseau d'ITREC GESTION, il s'occupe également des installations de GIMA chez le client.
- ➔ *Daniel Ghienne* : directeur technique, il gère l'équipe de développement et l'aspect commercial.
- ➔ *Bernard Esseyric* : s'occupe du suivi des clients vis à vis des bugs et gère la hotline.

### 1.2.2.2 L'environnement logiciel

#### a – Le produit GIMA

##### Présentation

GIMA a été créée au début des années 90 sur un besoin de la mairie de Paris. A ce moment là la demande portait sur la gestion de stock intégré et sur la gestion des interventions. A cette époque on reprochait le coût des cabinets d'architectes; une solution informatisée est alors apparue plus adaptée aux yeux des collectivités locales. LOGISPACE - ancien nom de la société - a alors initié le développement de GIMA, logiciel spécialisé dans la **gestion de patrimoine** client. Traitant de la gestion globale des services techniques et du patrimoine, il se compose de trois modules principaux :

- Le module **demande**, optionnel, traite les demandes de GIMA et de GIMA WEB.
- Un module **opération** qui gère la synthèse des coûts en temps réel. Ce module est généré par des demandes qui ont été validées.
- Enfin, le module **intervention** permet de gérer et de répertorier les lieux, les agents, les coûts, et les bons de commande.

La finalité de GIMA est de proposer un outil de gestion au quotidien permettant de faire du reporting et de localiser facilement l'information.

A titre d'exemple, GIMA a été utilisé par la mairie de Toulouse à la suite de la catastrophe d'AZF pour établir les coûts à réclamer aux assurances.

##### Caractéristiques

Les caractéristiques techniques peuvent être déclinées sous plusieurs aspects :

- C'est un outil de pilotage efficace dans le sens où il permet d'optimiser l'utilisation des budgets et d'anticiper les coûts.
- C'est également un outil de gestion au quotidien : moyens humains, moyens matériels ou encore traitement des demandes (patrimoine, interventions, stock, parcs automobile).
- GIMA est un outil structurant car il est basé sur la standardisation des processus de gestion; il intègre tous les domaines de la gestion tout en proposant une organisation adaptée ainsi qu'une rigueur budgétaire.
- C'est un outil de communication : le module GIMA WEB gère les demandes d'intervention et de fournitures.
- GIMA propose des interfaces comptables : engagement et retour de mandatement.
- Le logiciel propose enfin d'autres types d'interfaces, comme la gestion des fluides, un inventaire comptable, ou encore la gestion du carburant.

##### Le référentiel patrimoine

GIMA s'oriente autour de trois axes :

- Un axe fonctionnel, indispensable au fonctionnement de GIMA : il traite des unités de gestion, des équipements ou encore des zones, ainsi que les descriptifs pour les documents associés, baux, etc.
- Un axe physique, parcellaire et immobilier.
- Un axe technique.

## L'approche GIMA

Le logiciel se base sur la standardisation des processus de gestion; l'approche se veut globale et modulaire tout en offrant un positionnement par métier.

### Les concurrents

A l'heure actuelle il n'existe pas de concurrence qui au même titre que GIMA s'occupe de la gestion de grosses collectivités. Deux petites sociétés, *ATAL* et *SBCG*, ont cependant développé un produit semblable.

### L'aspect commercial

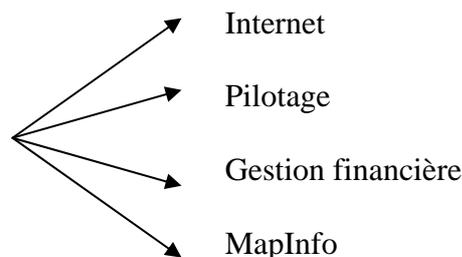
Trois commerciaux gèrent la prospection et la vente du logiciel; à Toulouse, Daniel Ghienne, responsable de l'agence de Toulouse, s'occupe de l'aspect relationnel et commercial avec le client.

### Les clients de GIMA

Les clients de GIMA sont les collectivités locales : mairies, conseils généraux et communautés urbaines. Les villes concernées sont Grenoble (60 utilisateurs), Lyon (300 utilisateurs), Toulouse (200 utilisateurs)... Cependant les modules du logiciel sont vendus séparément. Le coût total pour une mairie dépend du budget. Exemple : 100 000 euros pour la ville de Bordeaux, 500 000 euros pour la ville d'Angers, la moyenne étant à 200 000 euros. Les droits d'accès paramétrés permettent d'enlever un grand nombre de fonctionnalités au logiciel. Il est ainsi possible de descendre à des offres inférieures à 10 000 euros pour une licence GIMA Client/Serveur. GIMA n'a pas été conçu pour viser le secteur privé mais une démarche allant dans ce sens est en cours. De même, les universités peuvent être intéressées par le logiciel car il n'existe pas de produit imposé par le ministère. Les CHU vont également bientôt se constituer comme clients GIMA, d'où la nécessité de développer un nouveau module graphique pour couvrir leurs besoins. De manière plus générale il a été constaté qu'avec le temps et l'expérience, les clients intéressés par GIMA sont arrivés à une maturité telle que lorsque ils font l'acquisition du produit, ils achètent l'intégralité des modules car ils sont amenés à tous les utiliser à plus ou moins long terme.

### L'activité

- Bâtiments
- Parcs automobiles
- Espaces verts
- Fêtes et cérémonies
- Foncier
- Voirie



## Aspect technique de GIMA

Développée sous *Power Builder 9*, l'application repose sur des bases de données *Oracle*, *Sybase SQL Anywhere* (BDD livrée avec *Power Builder*), ou encore *SQL Server 5 et 6*, cela varie selon le client. La *Data Window*, intégrée à *Power Builder*, est un outil qui permet de récupérer les données de la base pour formatage ou traitement. Elle permet également de gérer la connexion aux différentes bases.

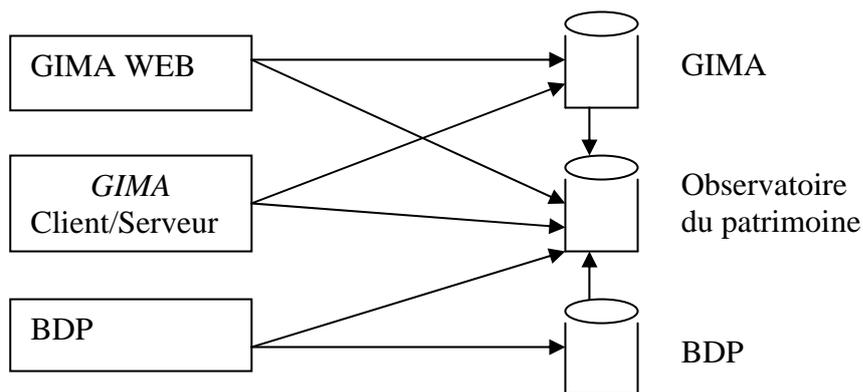


Figure 7 – Interaction entre GIMA WEB et GIMA Client/Serveur

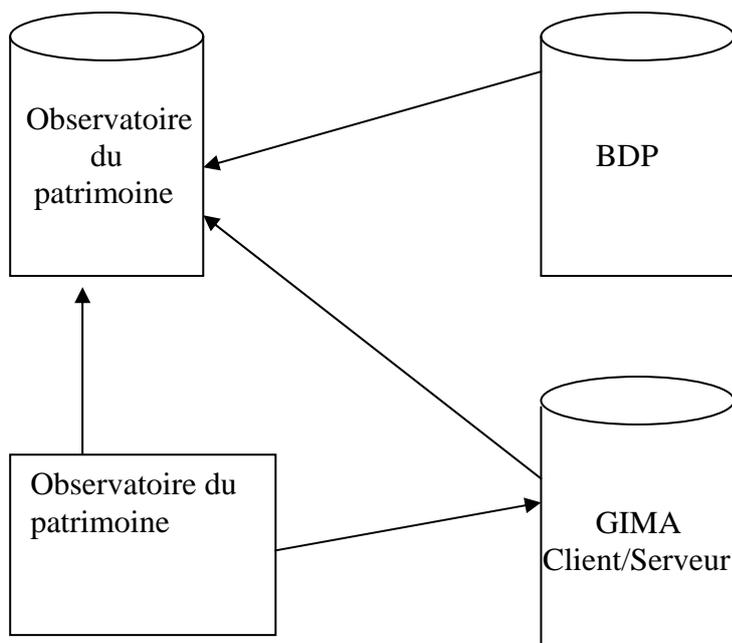


Figure 8 – Interaction entre les différentes bases de données

## Aperçu de l'application



Figure 9 – L'utilisateur doit se loguer pour accéder à l'application

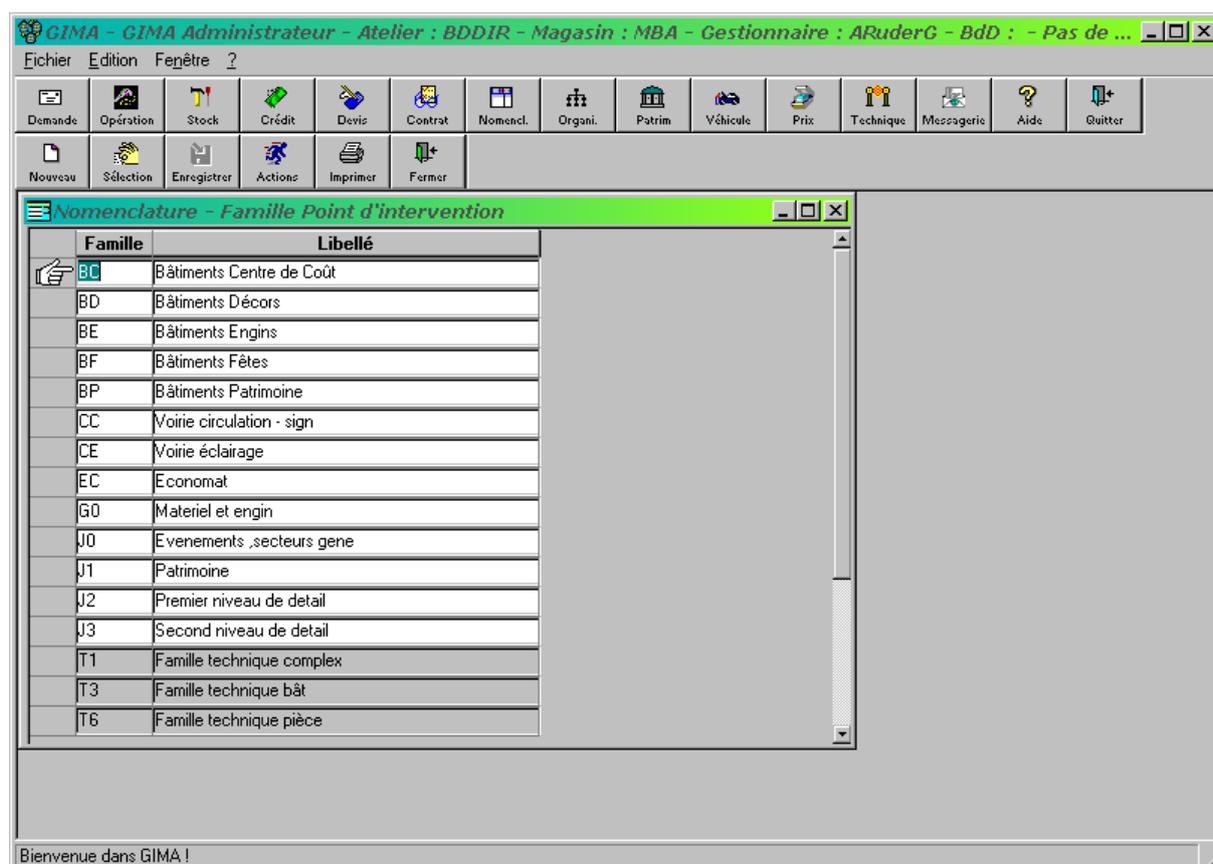


Figure 10 – La section des nomenclatures « Famille Point d'intervention »

DP	DG	Nom	Adresse	C.P.	Ville	Respo
BAT	AA	Angers Agglomération	83 rue du Mail	49100	Angers	HAMON
BAT	AAD	Angers Agglo Développement	8 bd Bessonneau	49100	Angers	BOURDI
BAT	AC	Direction Action Culturelle	bd de la Résistance et de la Déportation	49100	Angers	FLEURY
BAT	ADOC	Archives Documentation	bd de la Résistance et de la Déportation	49100	Angers	BERTOL
BAT	AG	Administration Générale	bd de la Résistance et de la Déportation	49100	Angers	CHUPIN
BAT	APU	Architecture	bd de la Résistance et de la Déportation	49100	Angers	ROUSSE
BAT	BAT	Direction Bâtiments	bd de la Résistance et de la Déportation	49100	Angers	SERVAN
BAT	CAB	Cabinet	bd de la Résistance et de la Déportation	49100	Angers	KOTRAE
BAT	CARA	CARA	26bis avenue Montaigne	49100	Angers	ROTH H
BAT	CCAS	CCAS	bd de la Résistance et de la Déportation	49100	Angers	LAOUR.
BAT	DG	Conseil de Gestion	bd de la Résistance et de la Déportation	49100	Angers	HARCOL
BAT	CMC	Conseil Municipal et Courrier	bd de la Résistance et de la Déportation	49100	Angers	DELEST
BAT	CO	Conseil en Organisation	bd de la Résistance et de la Déportation	49100	Angers	BERNIE
BAT	CRI	Communications et Relations	bd de la Résistance et de la Déportation	49100	Angers	PEREVE
BAT	DGS	Direction Générale des Serv	bd de la Résistance et de la Déportation	49100	Angers	MONIER
BAT	DGSA1	DGSA Développement Soci	bd de la Résistance et de la Déportation	49100	Angers	X
BAT	DGSA2	DGSA Ressources Internes	bd de la Résistance et de la Déportation	49100	Angers	CALVEZ
BAT	DGST	DGST	bd de la Résistance et de la Déportation	49100	Angers	GUEVEL
BAT	ECAC	Achats Economat	bd de la Résistance et de la Déportation	49100	Angers	CHATEL
BAT	EE	Direction Education Enfance	bd de la Résistance et de la Déportation	49100	Angers	KASPRZ

Figure 11 – Il est possible d'effectuer des recherches selon des critères bien spécifiques

GIMA - GIMA Administrateur - Atelier : BDDIR - Magasin : MBA - Gestionnaire : ARuderG - BdD : - Pas de ...

Fichier Edition Rapport Fenêtre ?

Demande Opération Stock Crédit Devis Contrat Nomencl. Organi. Patrim. Véhicule Prix Technique Messagerie Aide Quitter  
 Nouveau Sélection Enregistrer Imp. paramé. Documents Commandes ction gestion irection gosti Technique atrats Assurs trats Msinton Imprimer Fermer

Module organisation - Direction Gestionnaire :

DP :    
 DG :  Nom :   
 Adresse :   
 Code postal :  Ville :   
 Téléphone :  Fax :  Archivé :   
 Responsable :  Code hiérarchie :   
 Observation :

Bienvenue dans GIMA !

Figure 12 – Ici, l'utilisateur veut rajouter une nouvelle entrée dans le module organisation

## *b – Le produit GIMA WEB*

### **Présentation**

GIMA WEB est un produit qui se base sur des technologies Open Source. Frédéric Martinez, chef de projet chez ITREC GESTION, est en charge du développement de cette application depuis septembre 2003. J'ai pour ma part contribué au développement du module *Observatoire du patrimoine* durant ma mission de fin d'études, de février à septembre 2005.

Après 2 années de développement le produit est opérationnel et installé chez plusieurs clients: les mairies des villes d'Angers et de Rennes, et les pompiers de Lyon (SDIS 69). La mairie d'Angers compte 300 utilisateurs GIMA WEB qui se servent du module *Demande* pour gérer leurs fournitures. Le SDIS 69 utilise le progiciel pour le suivi et l'entretien du matériel des pompiers, ainsi que pour la commande de fournitures. Développé en langage JAVA, le produit offre certaines fonctionnalités du logiciel GIMA et les présentent au client de manière plus conviviale. A l'heure actuelle, le produit gère les demandes d'intervention sur site, commandes de fournitures et la maintenance de matériels. En outre, GIMA WEB repose sur la même base de données que GIMA Client/Serveur.

GIMA WEB sera présenté de manière détaillée dans la seconde partie de mon étude, dont le thème est « *L'engouement des sociétés informatiques pour les technologies Open Source* ». Ce dossier présentera également les technologies Open Source utilisées au sein d'ITREC GESTION.

### **Aperçu de l'application**

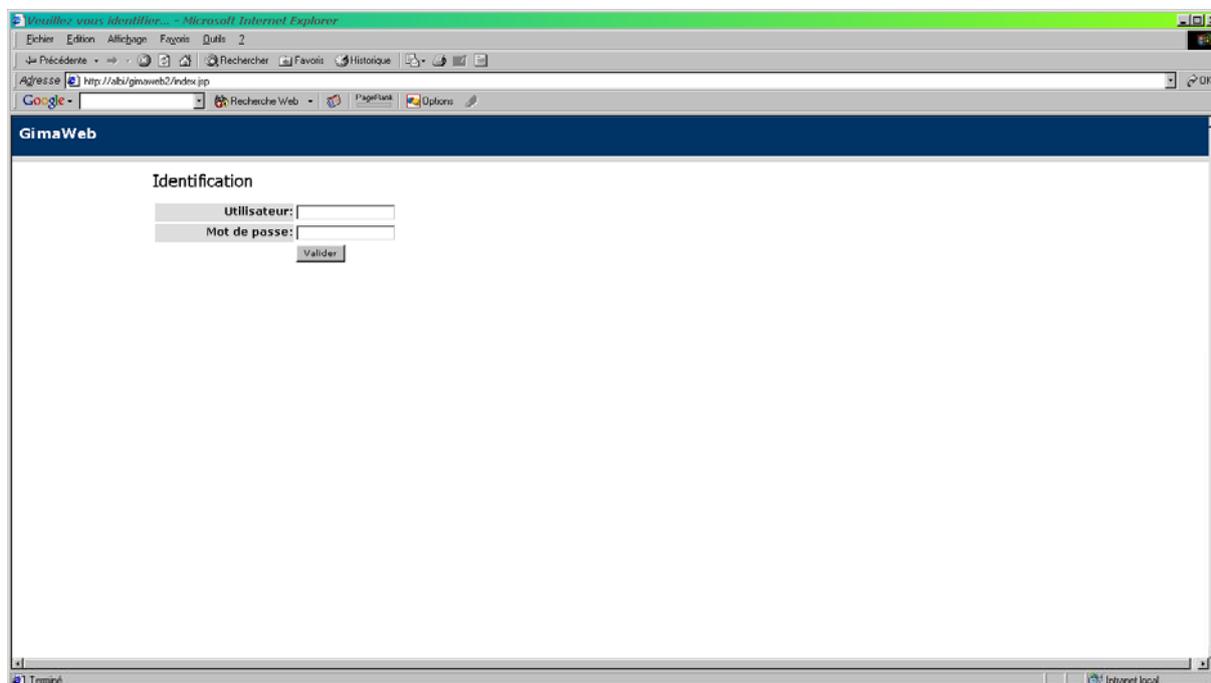


Figure 13 – Le client possède un profil et doit se loguer pour accéder à l'application

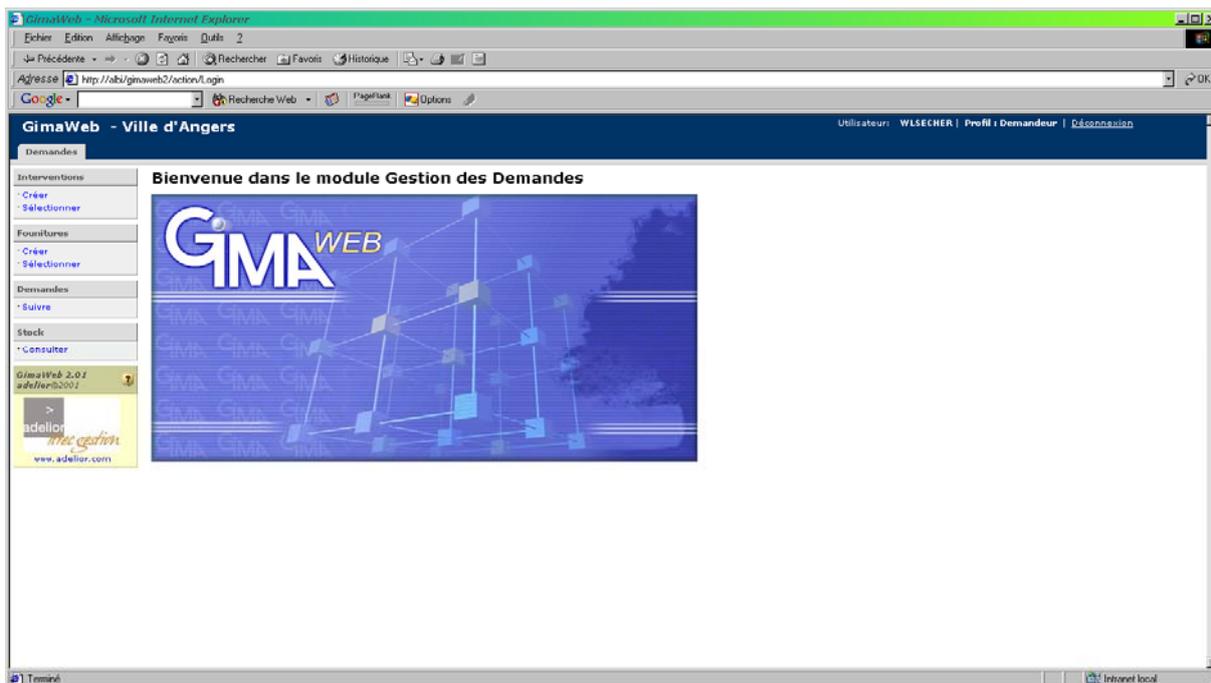


Figure 14 – L'écran d'accueil de l'application GIMA WEB 2.0

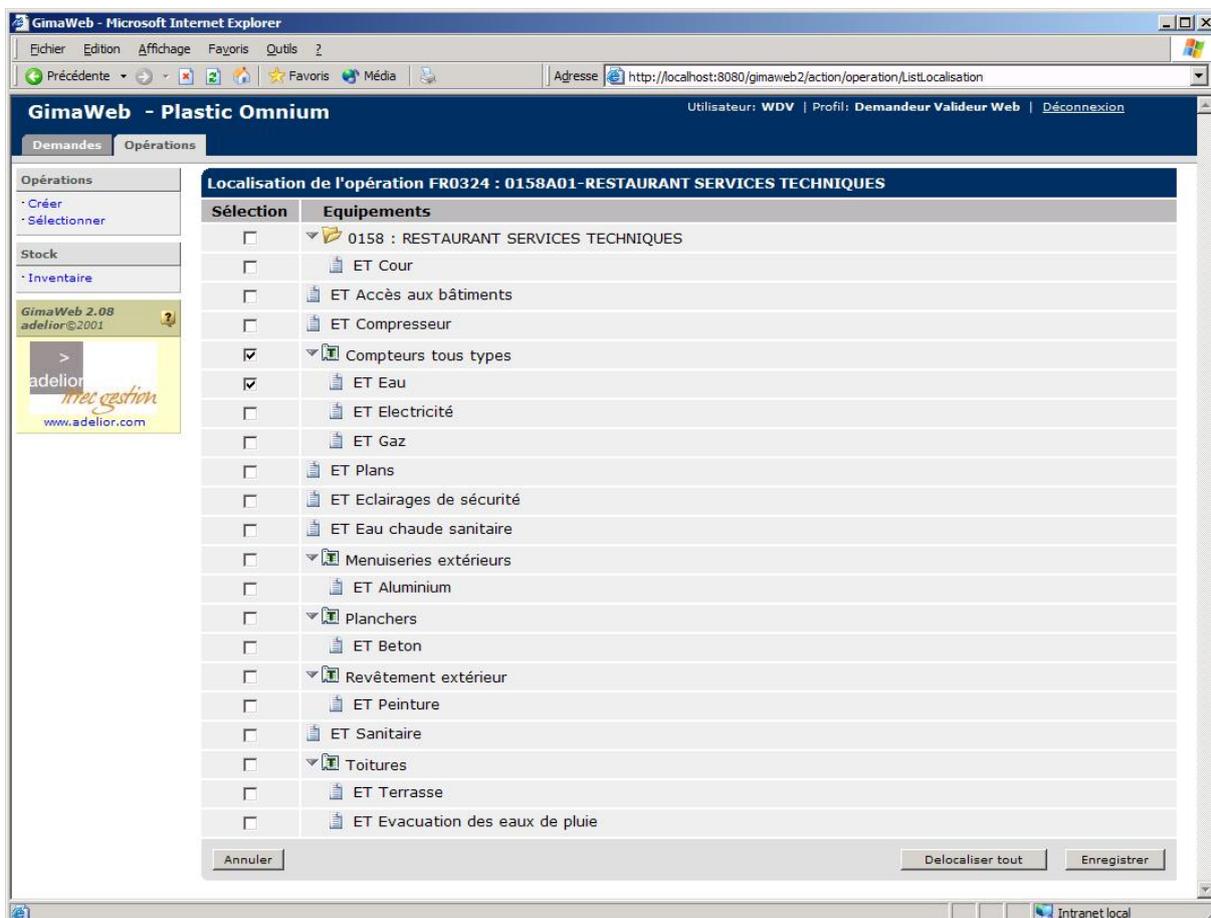


Figure 15 – Localisation sur une opération

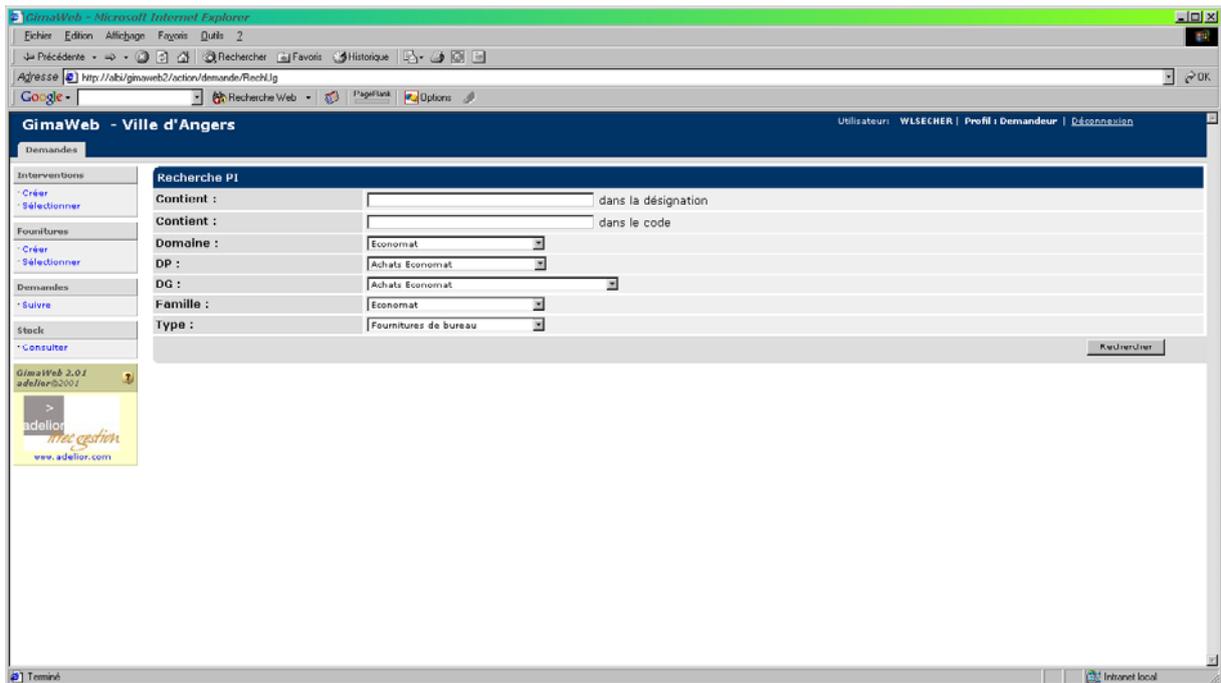


Figure 16 – Critères de recherche pour une demande d'intervention

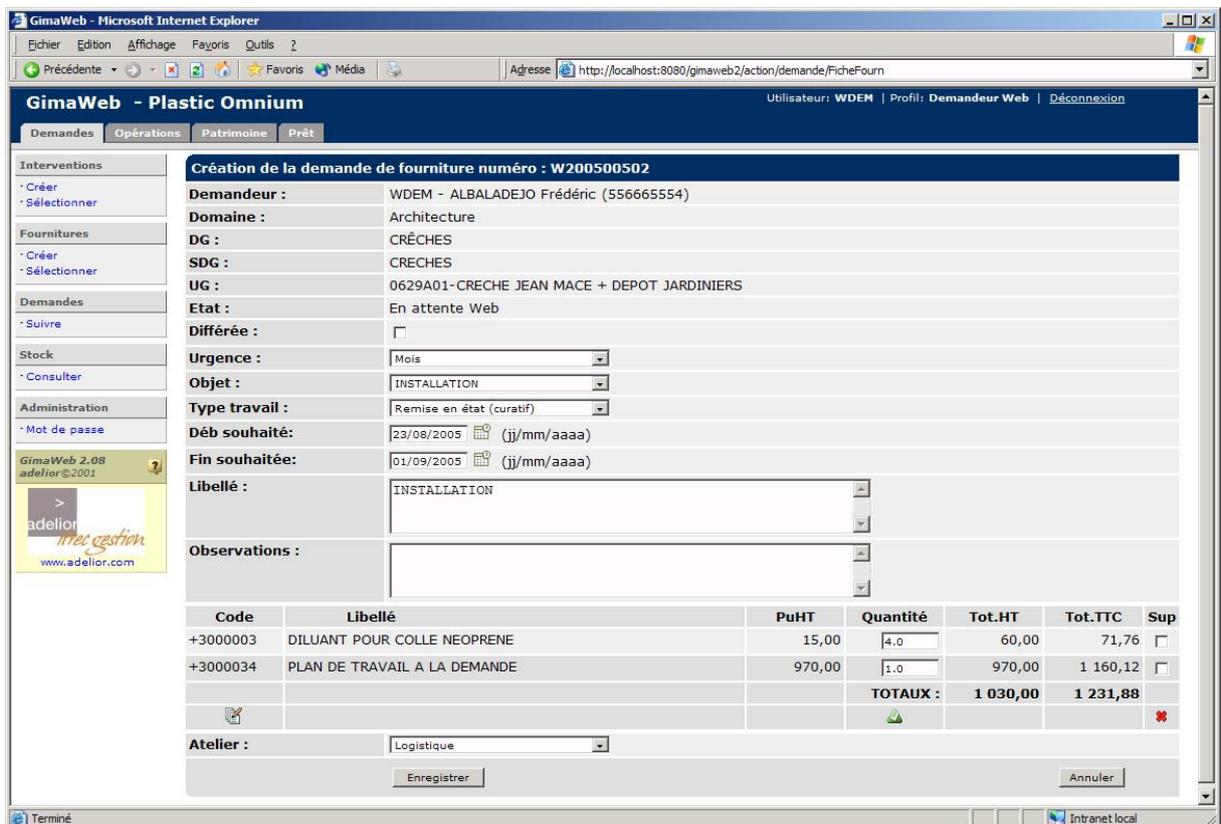


Figure 17 – Création d'une demande de fournitures via GIMA WEB

### *1.2.2.3 L'environnement matériel*

Les postes des développeurs doivent respecter des configurations minimales afin de pouvoir assurer des temps de réponse corrects ainsi qu'un bon confort d'utilisation.

Les postes de développement dédiés à GIMA WEB doivent être équipés de processeurs performants et d'une quantité de mémoire importante ; l'environnement ECLIPSE nécessite d'importantes ressources et chaque modification dans le code implique recompilation et régénération du fichier .WAR qui contient le package des classes du progiciel. Il faut également redéployer le serveur TOMCAT après chaque compilation. La durée de régénération du fichier WAR dure en moyenne 25 secondes et pourrait être beaucoup plus longue si la compilation était lancée sur une machine moins puissante. Il est donc important de travailler sur des machines performantes car cela influe directement sur la productivité des développeurs et donc sur le temps de développement du produit. Dans tous les cas il est recommandé de suivre les préconisations indiquées dans les manuels des différents logiciels installés sur les postes de développement.

A titre d'exemple les stations de développement GIMA WEB sont équipées de processeurs PENTIUM 4 en hyperthreading (simulation d'un bi-processeur), d'au moins 512 Mo de DDRAM, de disques durs SATA et de cartes graphiques avec 64 Mo de mémoire vidéo. Ce type de configuration est nécessaire car la machine virtuelle JAVA et la plateforme de développement ECLIPSE sont gourmandes en ressource.

Le tableau ci-après est un récapitulatif détaillé des stations de développement sur le pôle de Toulouse :

Nom	Utilisateur ou fonction	Carte mère ou modèle	CPU	CPU, MHz	RAM, MB	HDD, MB	OS	Adresse IP	Carte réseau	Ecran
ALBI	Gimaweb prod + sybase	OptiPlex GX100	Celeron	600	127	8001	Windows 2000 Pro	10.32.62.111	3Com EtherLink	non
ARCACHON	Eysseric	OptiPlex GX100	Celeron	600	127	9529	Windows 98	10.32.62.113	3Com EtherLink	Compaq 17p
AYGUESVIVES	Martinez		Intel(R) Pentium(R) 4	2400	512	220705	Windows XP Pro	10.32.62.105	D-Link DFE-530TX	iiyama 17p
BALMA	Portable Borie	Portable Latitude C600	Pentium III	600	256	9076	Windows XP Pro	10.32.62.117	3Com 10/100 Mini	portable
BASTIA	Cavagnara	K8N	AMD Athlon(tm) 64 3000+	2000	1024	78477	Windows XP Pro	10.32.62.146	NVIDIA nForce	yundai 19p
BERGERAC	prochainement Arcachon	VT82C691	Pentium III	600	384	17469	Windows 2000 Pro	10.32.62.128	NETGEAR FA310TX	non
CAMBRAIS	Baule	MS-6337	Pentium III	866	512	38154	Windows XP Pro	10.32.62.122	Realtek RTL8139 Fam	iiyama 17p
CASTANET	Boudet	Asus K8V de Luxe	AMD Athlon(tm) 64 3000+	2000	511	78152	Windows XP Pro	10.32.62.116	Marvell Yukon	Dell 19p
CHASSIRON	DGHIE	Portable Latitude C500	Celeron	900	256	19077	Windows XP Pro	10.32.62.126	Carte Ethernet	Portable
FRONTON	SERVEUR ORACLE 8I	OptiPlex GX110	Pentium III	800	255	19079	Windows 2000 Adv. Server	10.32.62.106	3Com EtherLink	Dell 17p
LAYMONT	Portable Demo	Latitude C810	Intel(R) Pentium(R) III Mobile	1133	512	28616	Windows XP Pro	10.32.62.138	Carte Ethernet	Portable
GOYRANS	serveur Secondaire		Pentium II	450	128	12943	Windows NT Server	10.32.62.47	3Com 3C90x	non
LILLE	Orge	Portable HP OmniBook PC	Pentium III	650	128	5715	Windows 98	10.32.62.140	Xircom CE3 10/100	Portable
LYON	Serveur Oracle EMMA	MS-6309	Celeron	633	512	9620	Windows XP Pro	10.32.62.104	Realtek RTL8139	non
NONTRON	Parant	K8N	AMD Athlon(tm) 64 3000+	2000	1024	58484	Windows XP Pro	10.32.62.144	NVIDIA nForce	yundai 19p
OSSEGOR	serveur Acces	L466cx	Celeron A	466	191	8032	Windows NT Workstation	10.32.62.124	3Com EtherLink	Dell 17p
PAU	Martinez	MS-6309	Celeron	600	128	14605	Windows 98	10.32.62.109	ELNK3 Ethernet	iiyama 17p
RENNES	Portable Demo	Armada V300	Celeron A	450	64	5712	Windows 2000 Pro	10.32.62.107	NETGEAR FA511	Portable
TULLE	serveur de Fichier		Intel(R) Celeron(R)	2800	1024	244568	Windows XP Pro	10.32.62.100	Marvell Yukon	non
VENERQUE	Serveur Principal	HP Vectra	Pentium II	400	256	8062	Windows NT Server	10.32.62.48	3Com EtherLink	non
SRVORCL	Serveur ORACLE GIMA	P4 P800-E DELUXE	Intel(R) Celeron(R)	3000	1024	136000	Mandrake 10.1	10.32.62.49	Realtek 8139	Dell 17p

*Figure 18 – Environnement matériel de la société ITREC GESTION et INFEO Toulouse*

### 1.2.3 La société INFEO

#### *1.2.3.1 Présentation de la société INFEO*

INFEO est une SSII qui s'est spécialisée dans la *gestion de patrimoine* et dans le développement d'*utilities* c'est à dire dans la délégation des services publics (réseaux, eau, téléphonie...).

Jusqu'en décembre 2003, INFEO appartenait au Bureau d'Etudes de SAFEGE et développait des outils informatiques pour améliorer le rendement des utilisateurs SAFEGE, dont le métier est axé sur les réseaux d'eau.

En 2002, INFEO a racheté la société ABSIS : celle-ci, créée en 1994, proposait des solutions autour du logiciel de DAO Drawbase. Implanté dans le sud - ouest, ABSIS comptait parmi ses clients ASTRIUM, SANOFI, TURBOMECA, et FRANCE 3. Ils sont désormais clients du groupe ADELIOR, et ABSIS a été rebaptisée INFEO TOULOUSE.

#### *1.2.3.2 Les produits INFEO*

Les outils développés par INFEO sont les suivants :

- X7 : outil de facturation, gestion des relations - clientèle, gestion de la facturation et du recouvrement, multi-utilities (eau, électricité, gaz...).
- EMMA : qualité de l'assainissement, pluviométrie, auto - surveillance et diagnostic permanent des réseaux d'assainissement.
- PICCOLO : étude de la pollution et du réseau des eaux, modélisation des réseaux d'eau potable. Exemple: étude de la propagation d'une certaine quantité de chlore dans un réservoir d'eau.
- ASSAÏ : logiciel traitant de l'assainissement non collectif. Actuellement 40 licences ont été vendues dans le sud - ouest de la France.
- BDP : logiciel de gestion de patrimoine basé sur la plate-forme AUTODESK (l'objectif étant à terme de faire porter le code par AUTODESK). Intègre des plates-formes graphiques. Le prix du projet BDP avoisine 100 000 euros et le prix d'une licence est de 1 800 euros. Le logiciel BDP se découpe selon 3 axes : un axe *géographique*, un axe *organisation du personnel*, et le *cycle de vie du bâtiment* (conception, architecture d'intérieur, et travaux d'aménagement).
- BDP WEB : INFEO a développé le module BDP WEB qui est un plugin AUTODESK payant (900 euros la licence). Il offre des outils de publication et d'établissement de rapports.

## 1.3 LE CONTEXTE DU PROJET

### 1.3.1 Déroulement chronologique des missions

A mon arrivée chez ITREC GESTION le lundi 31 janvier 2005 j'ai été reçu par Daniel GHIENNE qui m'a présenté mon projet de fin d'études et les différentes missions qui allaient m'être confiées au cours de cette dernière période.

Travaillant aux côtés de mon responsable d'apprentissage en entreprise, monsieur Frédéric MARTINEZ, j'ai dans un premier temps suivi des formations sur les technologies utilisées dans le développement de GIMA WEB. Monsieur Frédéric MARTINEZ m'a ainsi formé sur les frameworks STRUTS et HIBERNATE ainsi que sur les autres technologies utilisées dans le développement du progiciel\* GIMA WEB. Je me suis également formé sur les outils et environnements de développement (serveur TOMCAT, ECLIPSE) afin de mieux comprendre le fonctionnement de l'application. Parallèlement à cela j'ai pu mettre en pratique les connaissances acquises sur ces technologies en travaillant sur un environnement de test.

Cette période m'a permis de me familiariser avec l'environnement de développement et d'apprendre à exploiter les technologies utilisées dans GIMA WEB. J'ai ainsi pu étudier le fonctionnement des frameworks utilisés dans le progiciel puis réaliser des enchaînements de pages par le biais d'appels du poste client au serveur TOMCAT.

Dans un second temps on m'a demandé d'automatiser la conception de pages Web pour la prochaine version du site Internet du groupe ADELIOR et de sa filiale INFEO. Le principe était de faciliter le travail du webmaster chargé de l'intégration de nouvelles informations dans le site. J'ai pour cela eu le choix des technologies à employer. Pour mettre en place ce projet j'ai choisi d'orienter le développement vers les technologies des *Templates*, utilisées sous Dreamweaver. En effet la maison mère dispose déjà de licences Dreamweaver et c'est également ce logiciel qu'utilise le webmaster du site; conserver cet outil ne désoriente donc pas son utilisateur.

J'ai ensuite travaillé au sein de l'équipe production sur le développement du module « Observatoire du patrimoine » de GIMA WEB. Je suis ainsi passé d'un environnement de test à l'environnement réel de développement. J'ai d'abord étudié les spécifications générales et les spécifications fonctionnelles détaillées du produit. Cela m'a permis de mieux comprendre comment j'allais devoir reprendre et poursuivre le développement du module qui avait été initié par Ghislain BORIE. J'ai ensuite modélisé avec Frédéric MARTINEZ les différentes actions que devraient proposer les futures extensions de l'*Observatoire du patrimoine* puis j'ai commencé le développement. Durant cette phase j'ai été encadré par monsieur Daniel GHIENNE sur l'aspect présentation du produit, et par monsieur Frédéric MARTINEZ pour le côté technique (structure des pages et de leurs enchaînements, classes à développer ou à reprendre etc.).

Cette phase de développement a occupé la majeure partie de ma dernière période en entreprise; elle a par ailleurs été entrecoupée de diverses missions qui m'ont été confiées et qui sont indépendantes du principal projet de développement. J'ai ainsi travaillé sur les recettes de GIMA WEB en déroulant de nombreux jeux de test pour vérifier qu'il n'y avait pas de dysfonctionnements sur les versions avant livraison au client. Ce travail permit à monsieur MARTINEZ de corriger plusieurs bugs qui s'étaient générés dans l'application. Cela s'inscrit également dans une démarche qualité car les clients demandent à ce que des tests de débogage soient régulièrement effectués sur les progiciels qu'ils achètent.

## 1.3.2 Les sujets traités et les résultats obtenus

### *1.3.2.1 Formation sur les outils de développement*

Je me suis formé dans un premier temps à l'environnement de développement utilisé pour concevoir le progiciel GIMA WEB. Plusieurs formations sur les outils utilisés m'ont été dispensées par monsieur Frédéric MARTINEZ et j'ai travaillé sur l'écriture de classes de test sous environnement ECLIPSE : cela m'a permis de comprendre le fonctionnement et les différentes interactions entre les fenêtres de l'application ; je me suis ainsi familiarisé avec les frameworks utilisés.

### *1.3.2.2 Aide à la conception du site du groupe ADELIOR*

La seconde mission qui m'a été confiée m'a permis de développer une version HTML du site ADELIOR – INFEO reposant sur l'utilisation de *modèles*, appelés plus communément *Templates* sous Dreamweaver. Le principe est le suivant : le modèle porte l'extension **.dwt** et est localisé dans le dossier *Template* à la racine du site. Il constitue le « socle » et toutes les pages qui vont être créées se réfèrent à ce modèle.

Utiliser un *Template* présente plusieurs avantages :

- Il facilite la conception de pages Web,
- Il définit précisément quelles sont les zones à modifier,
- Il permet la modification simultanée de plusieurs pages.

Il est également possible de créer plusieurs modèles pour adapter un site à une charte graphique particulière. Créer un grand nombre de modèles n'est cependant pas recommandé car l'idée première dans l'utilisation des *Templates* est de faciliter le travail du webmaster.

Par ailleurs, une documentation technique sur l'utilisation des *Templates* sous Dreamweaver est disponible dans l'aide en ligne du logiciel; un utilisateur novice y trouvera ainsi toutes les informations nécessaires pour créer un site reposant sur cette technologie.

La copie d'écran ci après est la page d'accueil du site Web ADELIOR; celle-ci se base sur les *templates*. Le webmaster n'a qu'à modifier les rubriques en rajoutant les informations qu'il souhaite dans la page ; il peut en quelques clics de souris générer de nouveaux modèles de page à intégrer au site. Son travail en est ainsi grandement facilité.

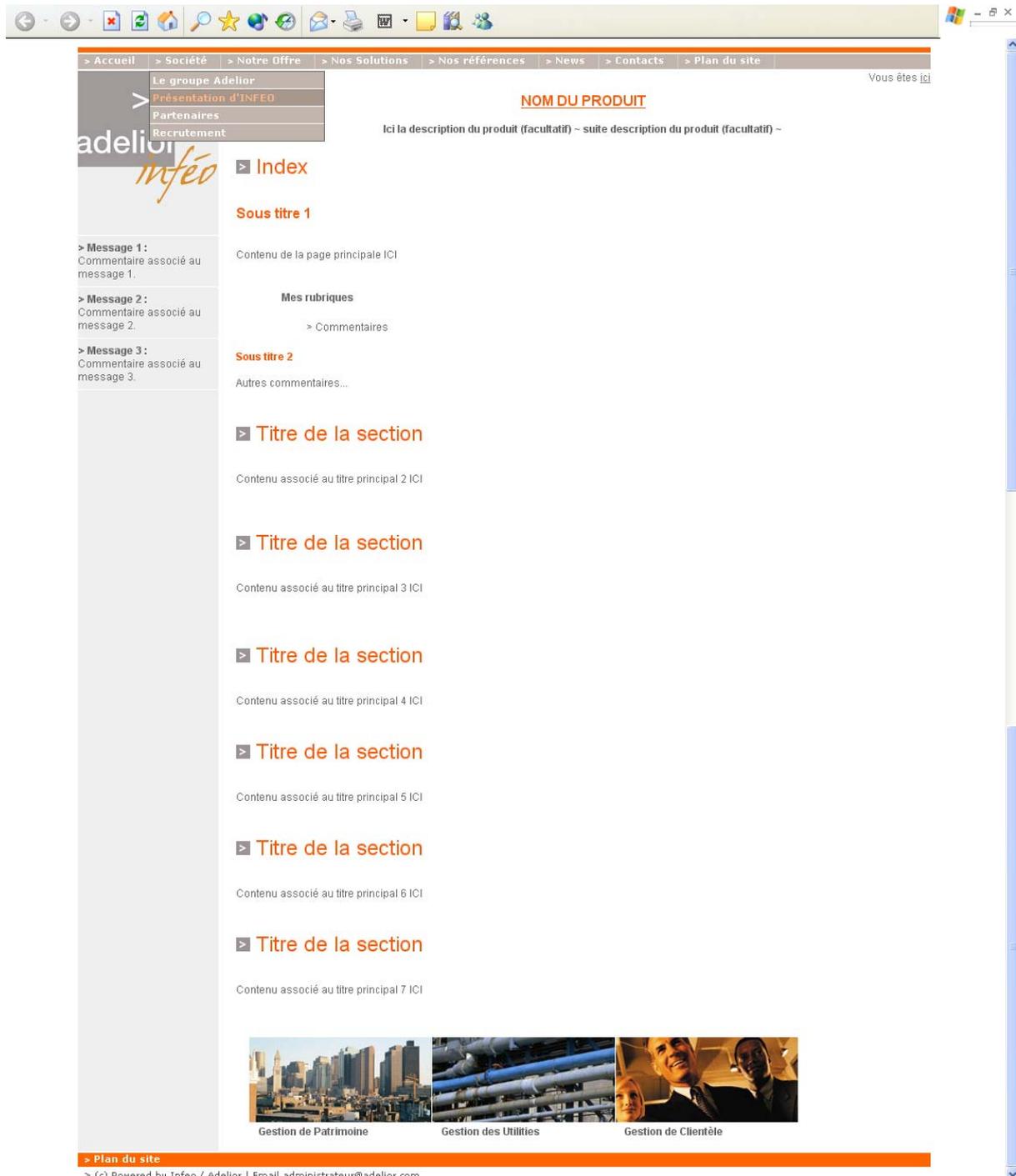


Figure 19 – Page d'accueil du site ADELIIOR, basée sur les templates

Ce site, basé sur l'utilisation des *templates*, a ainsi été utilisé pour concevoir le site Web ADELIOR, comme l'illustre la capture écran ci - dessous (source [www.adelior.com](http://www.adelior.com)) :



Figure 20 – Le site Web du groupe ADELIOR, reprenant les templates

### 1.3.2.3 Développement d'extensions au module « Observatoire du patrimoine »

Pour mener à bien un projet, il est important de définir et respecter des normes de développement. Des méthodes et des règles dites d'*industrialisation* doivent donc être définies.

Travailler sur le module *Observatoire du patrimoine* de GIMA WEB m'a permis d'acquérir des méthodes de conduite de projet. J'ai ainsi pu constater que plusieurs règles devaient être respectées dans la mise en œuvre d'un projet :

- capitaliser le savoir pratique retiré dans les documentations fournies et les différentes ressources mises à disposition,
- avoir un processus de développement bien défini et des procédures de développement,
- être à l'écoute des changements de technologies et adapter le projet.

Ma mission principale lors de cette dernière période en entreprise a été de poursuivre le développement du module *Observatoire du patrimoine* de GIMA WEB initié par monsieur Ghislain BORIE. L'*Observatoire du patrimoine* s'appuie sur le module client/serveur de GIMA Patrimoine et permet de visualiser globalement les dépenses et les recettes générées par une entité physique ou une activité. Le client désireux d'utiliser cette fonctionnalité devra posséder le module client/serveur pour pouvoir utiliser sa version Web.

Après avoir étudié les spécifications générales et techniques du progiciel GIMA WEB j'ai modélisé les différentes actions que devraient proposer les futures extensions de l'*Observatoire du patrimoine*. J'ai ensuite commencé le développement.

Durant cette dernière période j'ai été encadré sur le projet par monsieur Frédéric MARTINEZ qui m'a conseillé sur les aspects techniques, et par monsieur Daniel GHIENNE qui m'a indiqué les préconisations à respecter sur la partie présentation des données.

J'ai plus particulièrement travaillé sur le développement de nouvelles fonctionnalités pour l'*Observatoire du patrimoine* et sur la génération de synthèses financières.

Les synthèses financières permettent de visualiser les dépenses et les recettes gérées dans GIMA Client/Serveur. Le fonctionnement est le suivant : l'utilisateur sélectionne une entité en se positionnant sur un *Complexe Immobilier*, une *Zone*, un *Bâtiment* ou une *Unité de Gestion*. Lorsque l'utilisateur s'est positionné sur un niveau physique et valide son choix, il accède à une liste d'éléments répondant à sa requête. Cette liste présente le code de la cible, le libellé, l'adresse, l'identité, le carnet de santé et la synthèse financière des éléments concernés.

The screenshot shows the search interface for the 'Observatoire du patrimoine' module. It features three main search sections:

- Recherche selon l'axe géographique:** Includes fields for Commune, Quartier, Circonscription, Canton, and Contient (with options to search by parcel number or physical address).
- Recherche selon l'axe fonctionnel:** Includes fields for Domaine, DG, SDG, Secteur, and Ilôt.
- Caractéristiques complémentaires de recherche:** Includes checkboxes for Propriétaire, dropdowns for Propriété, and input fields for Contient (searching by label or code), and a dropdown for Cible (set to 'Parcelle').

A 'Rechercher' button is located at the bottom right of the search area.

Figure 21 – La page des critères de recherche de l'Observatoire du patrimoine

Cible  
sélectionnée

Accès à la liste des éléments  
correspondant aux critères  
sélectionnés

Accès à la synthèse financière

L'adresse

Code

Le libellé de l'élément

Code	Libellé	Adresse	Sél	Id.	So.	\$.fin.
0017-L-01	ECOLE MATERNELLE BERGSON	19 BLUM (RUE LEON) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0017-B-01	ECOLE MATERNELLE BERGSON	19 BLUM (RUE LEON) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0076-B-01	ECOLE MATERNELLE CHAVASSIEUX	98 CHAVASSIEUX (RUE DE) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0628-B-01	ECOLE MATERNELLE FAURIEL	32 FRANCS MACONS (RUE DES) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0467-B-01	ECOLE MATERNELLE METARE NORD-EST	7 MARCET (RUE BAPTISTE) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0495-B-01	ECOLE MATERNELLE MONTPLAISIR	36 TERRENOIRE (RUE DE) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0514-B-01	EX ECOLE MATERNELLE MICHON	68B ROCHE DU GEAI (RUE DE LA) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0039-B-01	EX MATERNELLE D'ISLY	1 RUEL (RUE ANDRE) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0702-B-01	EX MATERNELLE FENELON	BASLY (RUE) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0066-B-01	LOGEMENT DE FONCTION MATERNELLE	9 JOUVET (RUE LOUIS) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0035-B-02	LOGEMENT DE FONCTION MATERNELLE	5 ALBERT 1ER (BOULEVARD) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0148-B-04	MATERNELLE + EX CENTRE LOISIRS AFL	7 SCHUBERT (RUE FRANZ) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			

Figure 22 – Liste des éléments répondant à la requête de l'utilisateur

Accès au carnet d'identité

Carnet de santé

A partir de cette liste, l'utilisateur peut se rendre sur la *synthèse financière* de l'élément sélectionné mais il peut également y accéder depuis le *carnet d'identité* ou le *carnet de santé*.

Exercice
2006

Figure 23 – Liste des exercices disponibles pour une synthèse sélectionnée

Année d'exercice de la synthèse financière sélectionnée

Accès à la *synthèse financière* depuis le *carnet de santé* :

**Bâtiment N° : 0017-L-01**

**Complexe :** MATERNELLE BERGSON  
**Zone :** Local  
**Libellé :** ECOLE MATERNELLE BERGSON  
**Adresse :** 19 BLUM (RUE LEON) 42000 Saint-Etienne  
**Type :** BATI D'ENSEIGNEMENT

Thématique	Nombre	Coût	Liste Int.
Depuis le 03/01/2002 :	25	1987.39	
Interventions internes :	23	1987.39	🔍
Interventions externes :	2	0	🔍
Interventions en cours :	2	0	🔍
Interventions prévues :	0		
Interventions faites :	23	1987.39	🔍
Consommation des fluides pour 2002 :		1797.78	

**Répartition des interventions**

Interes = 23  
Externes = 2

**Bilan des interventions**

Nombre d'interventions

En cours Prévues Terminées

Retour Synthèse des dépenses et recettes...

Carnet d'entretien

Figure 24 – Représentation graphique des données sur le carnet de santé

Accès à la synthèse financière

Accès à la *synthèse financière* depuis le *carnet d'identité* :

**Bâtiment N° : 0017-L-01**

**Complexe :** MATERNELLE BERGSON  
**Zone :** Local  
**Libellé :** ECOLE MATERNELLE BERGSON  
**Adresse :** 19 BLUM (RUE LEON) 42000 Saint-Etienne  
**Type :** BATI D'ENSEIGNEMENT

Général Fonctionnel Sécurité Accessibilité Métrique

**Construction :** Non renseigné  
**Surface :** 0  
**Niveaux :** 0  
**Hauteur :** Non renseigné  
**Ville propriétaire :** NON  
**Type propriété :** Dernier acte d'acquisition

**Propriétaire :**  
**Valeur :**  
**Surface :**

Code Libellé Adresse

Retour Synthèse des dépenses et recettes...

Figure 25 – Carnet d'identité d'un bâtiment sélectionné

Accès à la synthèse financière

Si aucune *synthèse financière* n'existe pour la cible sélectionnée par l'utilisateur, l'application génère un message d'avertissement :

**GimaWeb - Plastic Omnium** Utilisateur: GIMA | Profil: Administrateur | Déconnexion

Demander Opérations Patrimoine Prêt

**Attention : aucun exercice n'est disponible pour l'élément sélectionné.**

Observatoire  
- Sélectionner

GimaWeb 2.06  
adelior@2001

adelior  
www.adelior.com

**Bâtiment : 54 éléments trouvés(s)...**

Code	Libellé	Adresse	Sél	Id.	Sa.	S.fin.
0017-L-01	ECOLE MATERNELLE BERGSON	19 BLUM (RUE LEON) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0017-B-01	ECOLE MATERNELLE BERGSON	19 BLUM (RUE LEON) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0076-B-01	ECOLE MATERNELLE CHAVASSIEUX	98 CHAVASSIEUX (RUE DE) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0628-B-01	ECOLE MATERNELLE FAURIEL	32 FRANCS MACONS (RUE DES) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0457-B-01	ECOLE MATERNELLE METARE NORD-EST	7 MARCET (RUE BAPTISTE) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0405-B-01	ECOLE MATERNELLE MONTPLAISIR	36 TERRENOIRE (RUE DE) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0514-B-01	EX ECOLE MATERNELLE MICHON	68B ROCHE DU GEAI (RUE DE LA) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0039-B-01	EX MATERNELLE D'ISLY	1 RUEL (RUE ANDRE) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0702-B-01	EX MATERNELLE FENELON	BASLY (RUE) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0066-B-01	LOGEMENT DE FONCTION MATERNELLE	9 JOUVET (RUE LOUIS) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0035-B-02	LOGEMENT DE FONCTION MATERNELLE	5 ALBERT 1ER (BOULEVARD) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			
0148-B-04	MATERNELLE + EX CENTRE LOISIRS AFL	7 SCHUBERT (RUE FRANZ) 42000 Saint-Etienne	<input checked="" type="checkbox"/>			

Retour

Figure 26 – Cas où il n'existe pas d'exercice pour une cible sélectionnée

La fiche de la *synthèse financière*, dont les données sont enregistrées dans la table **gima.ST\_FINANCIERE**, se présente de la manière suivante :

**GimaWeb - Plastic Omnium** Utilisateur: GIMA | Profil: Administrateur | Déconnexion

Demander Opérations Patrimoine Prêt

Observatoire  
- Sélectionner

GimaWeb 2.06  
adelior@2001

adelior  
www.adelior.com

**Synthèse financière de la cible sélectionnée**

Exercice : 2005 Périètre : Batiment '0017-L-01'  
Date de traitement : 12/07/2005 Domaine :

	Poste de coût ou de recette	H.T	TVA	TTC
DEPENSES	Fonctionnement			
	Travaux en régie	10	2.6	12.6
	Total	10	2.6	12.6
DEPENSES	Investissement			
	Travaux entreprise	50	10.3	60.3
	Total	50	10.3	60.3
	TOTAL DEPENSES	60	12.9	72.9
RECETTES	Fonctionnement			
	Loyer	18	3.7	21.7
	Total	18	3.7	21.7
RECETTES	Investissement			
	Cessions	170	35.02	205.02
	Total	170	35.02	205.02
TOTAL RECETTES		188	38.72	226.72

**Dépenses T.T.C.**

Fonct. = 12,6 Invest. = 60,3

**Recettes T.T.C.**

Invest. = 205,02 Fonct. = 21,7

Retour

Figure 27 – La fiche des synthèses financières

Accès à l'impression de la synthèse financière

Dans sa forme imprimable, la *synthèse financière* se présente de la manière suivante :

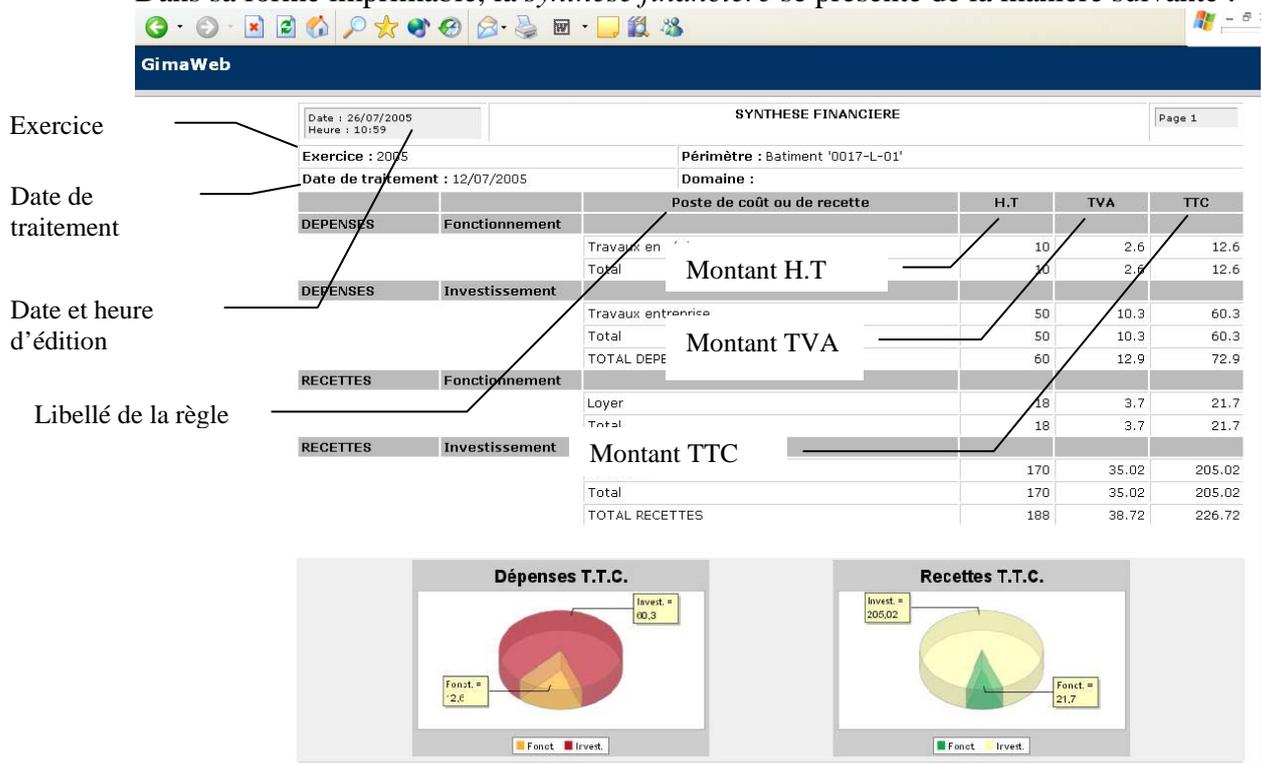


Figure 28 – La page d'édition des synthèses financières

#### 1.3.2.4 Déroulement de jeux de test pour le progiciel GIMA WEB

Au cours de cette période en entreprise j'ai été amené à effectuer du *recettage\** de logiciel, pour l'application GIMA WEB. Il s'agissait plus particulièrement de tester le fonctionnel du *module des demandes* des versions 2.06, 2.07 et 2.08 de l'application avant qu'elles ne soient livrées aux clients. J'ai ainsi été amené à tester le logiciel de manière approfondie afin de relever les différents cas d'erreur ; ceux ci pouvaient être générés par l'insertion de caractères spéciaux dans des clauses de recherche ou en entrant des données absurdes, par exemple des nombres très grands. Dans certains cas les problèmes pouvaient venir de l'enchaînement des pages (basé sur le framework STRUTS) ou de l'absence de tests de contrôle JAVASCRIPT dans les champs de saisie, les listes etc.

Les clients de GIMA WEB (mairies, collectivités territoriales) sont particulièrement exigeant sur le bon fonctionnement de l'application et sur le fait qu'elle soit exempte de bugs : ils transmettent à l'équipe de développement chaque cas d'erreur relevé et font tester le progiciel dans son intégralité avant de le mettre en production au sein de leurs équipes. Mon travail s'est donc inscrit dans une démarche qualité - produit et a permis d'une part de corriger les erreurs présentes dans l'application et d'autre part de fournir au client des justificatifs sur les tests réalisés par l'équipe de développement.

A		B		C		D		E			F			G		
FONCTION				DESCRIPTION DU TEST				RESULTATS								
								Tests 2.06 16-06-05			Tests 2.07 07-07-05			Tests 2.08 17-08-2005		
<b>INTERVENTIONS</b>																
<b>Créer</b>																
Recherche UG				Insertion d'apostrophes et dépassement du nombre de caractères maxi sur chaque champ Test sur le bouton Rechercher et la touche "Entrée"				KO			OK					
Liste UG				Sélection d'une UG directement par l'URL (Code UG)				KO			OK					
Création d'une demande d'intervention				Insertion d'apostrophes et dépassement du nombre de caractères maxi sur libellé Test sur caractères spéciaux dans le libellé				KO			OK					
<b>Sélectionner</b>																
Recherche des demandes d'intervention				Insertion d'apostrophes et dépassement du nombre de caractères maxi sur chaque champ Date de réception antérieure à la date de la demande Test sur le bouton Rechercher et la touche "Entrée"				KO			OK					
Liste des demandes d'intervention				Aucun problème constaté				OK			OK					
Consultation de la demande d'intervention				Accès à la demande directement par l'URL (N° de demande)				KO			OK					
Historique de la demande n.				Accès à l'historique directement par l'URL (N° de demande)				KO			OK					
Modification de la demande d'intervention				Accès à la demande directement par l'URL (N° de demande) Insertion d'apostrophes et dépassement du nombre de caractères maxi sur libellé Test sur caractères spéciaux dans le libellé				KO			OK					
Validation de la demande d'intervention				Accès à la demande directement par l'URL (N° de demande) Insertion d'apostrophes et dépassement du nombre de caractères maxi sur les Commentaires Test sur caractères spéciaux dans les Commentaires				KO			OK					
Suppression de la demande d'intervention				Accès à la demande directement par l'URL (N° de demande)				KO			OK					
<b>FOURNITURES</b>																
<b>Créer</b>																
Recherche UG				Insertion d'apostrophes et dépassement du nombre de caractères maxi sur chaque champ Test sur le bouton Rechercher et la touche "Entrée"				KO			OK					
Liste UG				Sélection d'une UG directement par l'URL (Code UG)				KO			OK					
Création de la demande de fourniture				Insertion d'apostrophes et dépassement du nombre de caractères maxi sur libellé Test sur caractères spéciaux dans le libellé Test sur les quantités des fournitures rattachées				KO			OK					
Recherche des articles en stock				Insertion d'apostrophes et dépassement du nombre de caractères maxi sur chaque champ Test sur le bouton Rechercher et la touche "Entrée"				KO			OK					
Liste des articles trouvés				Aucun problème constaté				OK			OK					
<b>Sélectionner</b>																
Recherche des demandes de fourniture				Insertion d'apostrophes et dépassement du nombre de caractères maxi sur chaque champ Date de réception antérieure à la date de la demande Test sur le bouton Rechercher et la touche "Entrée"				KO			OK					

Figure 29 – Déroulement de jeux de test sur différentes versions de GIMA WEB

Ce travail m'a donc placé du « côté » des utilisateurs de l'application ce qui m'a poussé à tester des aspects du logiciel que le concepteur peut parfois omettre. En effet, un développeur testant sa propre application est souvent accoutumé à tester les cas d'erreurs les plus courants ou ceux qui reviennent fréquemment. Ce travail m'a ainsi permis d'adopter une démarche méthodique pour dérouler des tests de logiciels.

Les différents sujets traités au cours de cette période en entreprise m'ont permis :

- de me familiariser avec les outils développés par le Groupe ADELIOR, plus précisément : GIMA Client/Serveur et GIMA WEB.
- de me former sur les technologies Open Source JAVA/J2EE et sur les frameworks STRUTS et HIBERNATE.
- de mener un véritable projet de développement en équipe et ainsi comprendre les contraintes liées au cycle de vie d'un logiciel : délais, attentes des clients, versioning etc.
- de m'initier à un métier de l'informatique : la recette de logiciels.

## II – L'ENGOUEMENT DES SOCIÉTÉS INFORMATIQUES POUR LES TECHNOLOGIES OPEN SOURCE

Cette seconde partie de mon étude constitue un travail de recherche sur l'utilisation des technologies Open Source, plus particulièrement dans la gestion de projets informatiques.

### 2.1 LES TECHNOLOGIES OPEN SOURCE

#### 2.1.1 Définition de l'Open Source

L'**Open Source** caractérise les logiciels dont le code source est disponible, modifiable et redistribuable sous certaines conditions.

L'utilisation du terme **Open Source** a été suggérée par Christine Peterson du *Foresight Institute* afin de lever l'ambiguïté du mot anglais « free » qui signifie libre au sens de « liberté » mais également « libre accès, gratuité », et signifier ainsi aux utilisateurs qu'un logiciel a un coût. Il s'agissait également d'être plus « Business friendly », le terme *Logiciel libre* ayant une connotation libertaire qui risquait de rebuter certaines entreprises. Par ailleurs, une confusion subsiste parfois entre les termes *logiciel libre* (free software) et **Open Source**. Bien que tous les logiciels libres correspondent aux critères de l'*Open Source Definition*, l'esprit qui anime les développeurs de logiciels libres (« copyleft ») et les règles plus restrictives qui les composent incitent à les différencier.

D'un point de vue économique, la marque **Open Source** contribue à la création d'une nouvelle forme de marché et d'économie, plus sexy que le terme « Logiciel libre » dans le contexte de la nouvelle économie, où l'on parlait de *Business model*, *Business Angel*, *bourse*, etc.

Afin d'éviter que le terme **Open Source** ne soit utilisé à mauvais escient, Eric Steven Raymond a tout d'abord essayé de le déposer. Sa tentative ayant échoué, il créa, avec Bruce Perens, l'*Open Source Initiative* qui délivre désormais le label *OSI Approved* aux licences qui satisfont aux critères définis dans l'*Open Source Definition*. Ce document définit les caractéristiques des logiciels **Open Source**, par exemple le mode de publication des modifications du logiciel, qui sont différentes de celles du « Logiciel libre ».

L'Open Source Initiative a défini les critères nécessaires afin de pouvoir utiliser l'appellation **Open Source** dont voici les éléments essentiels :

- Libre redistribution,
- Code source disponible,
- Travaux dérivés possibles.

Ainsi, le fait de disposer des sources d'un logiciel ne suffit pas à dire qu'il est **Open Source** : dans tous les cas, on doit se référer à la licence d'utilisation du logiciel.

L'histoire et les polémiques soulevées se trouvent dans l'*Open Source Initiative*.

## 2.1.2 Les créateurs du concept Open Source

### Bruce Perens

Rédacteur de la définition, dérivée d'un travail préalablement effectué pour la distribution Debian. Il se présente comme un scientifique spécialisé dans l'informatique.

### Eric S. Raymond

Principal théoricien de l'*Open Source*, convaincu de la nécessité de conduire une campagne d'opinion sur ce thème.

### Larry Augustin

Président co-fondateur de *VA Research VA Linux Systems*.

### Sam Ockman

Président fondateur de Penguin Computing.

### Richard M. Stallman

Fondateur de la Free Software Foundation.

### Linus Torvalds

Créateur du noyau Linux.

### Alan Cox

Responsable du développement du noyau Linux, dont il tient le journal en ligne.

### 2.1.3 Les critères d'une licence Open Source

La licence d'un programme **Open Source** doit correspondre aux critères suivants :

#### ***1. Libre redistribution.***

La licence ne doit pas empêcher de vendre ou de donner le logiciel en tant que composant d'une distribution d'un ensemble contenant des programmes de diverses origines. La licence ne doit pas exiger que cette vente soit soumise à l'acquittement de droits d'auteur ou de royalties.

#### ***2. Code source.***

Le programme doit inclure le code source, et la distribution sous forme de code source comme sous forme compilée doit être autorisée. Quand une forme d'un produit n'est pas distribuée avec le code source correspondant, il doit exister un moyen clairement indiqué de télécharger le code source, depuis *l'Internet*, sans frais supplémentaires. Le code source est la forme la plus adéquate pour qu'un programmeur modifie le programme. Il n'est pas autorisé de proposer un code source rendu difficile à comprendre. Il n'est pas autorisé de proposer des formes intermédiaires, comme ce qu'engendre un pré processeur ou un traducteur automatique.

#### ***3. Travaux dérivés.***

La licence doit autoriser les modifications et les travaux dérivés, et leur distribution sous les mêmes conditions que celles qu'autorise la licence du programme original.

#### ***4. Intégrité du code source de l'auteur.***

La licence ne peut restreindre la redistribution du code source sous forme modifiée que si elle autorise la distribution de fichiers « *patch* » aux côtés du code source dans le but de modifier le programme au moment de la construction. La licence doit explicitement permettre la distribution de logiciel construit à partir du code source modifié. La licence peut exiger que les travaux dérivés portent un nom différent ou un numéro de version distinct de ceux du logiciel original.

#### ***5. Pas de discrimination entre les personnes ou les groupes.***

La licence ne doit opérer aucune discrimination à l'encontre de personnes ou de groupes de personnes.

#### ***6. Pas de discrimination entre les domaines d'application.***

La licence ne doit pas limiter le champ d'application du programme. Par exemple, elle ne doit pas interdire l'utilisation du programme pour faire des affaires ou dans le cadre de la recherche génétique.

#### ***7. Distribution de la licence.***

Les droits attachés au programme doivent s'appliquer à tous ceux à qui le programme est redistribué sans que ces parties ne doivent remplir les conditions d'une licence supplémentaire.

#### ***8. La licence ne doit pas être spécifique à un produit.***

Les droits attachés au programme ne doivent pas dépendre du fait que le programme fait partie d'une distribution logicielle spécifique. Si le programme est extrait de cette distribution et utilisé ou distribué selon les conditions de la licence du programme, toutes les parties

auxquelles le programme est redistribué doivent bénéficier des droits accordés lorsque le programme est au sein de la distribution originale de logiciels.

### **9. La licence ne doit pas contaminer d'autres logiciels.**

La licence ne doit pas apposer de restrictions sur d'autres logiciels distribués avec le programme qu'elle couvre. Par exemple, la licence ne doit pas exiger que tous les programmes distribués grâce au même médium soient des logiciels Open Source.

### **10. Exemples de licences.**

Les licences suivantes sont des exemples de licences considérées conformes à la définition de l'Open Source : GNU GPL, BSD, X Consortium, Artistic et la MPL.



## **2.1.4 L'intérêt des technologies Open Source pour les industriels**

### **2.1.4.1 Pourquoi faut-il choisir l'Open Source ?**

L'**Open Source** est une alternative aux technologies et aux briques logicielles proposées par les éditeurs de technologies. Faire le choix de l'Open Source permet de réduire drastiquement, voire de supprimer les coûts d'acquisition des licences logicielles associées à la mise en œuvre d'une solution e-business. La philosophie de développement des projets Open Source s'appuie sur la dimension communautaire de l'Internet et la constitution de communautés de développeurs qui partagent leurs capacités de travail et d'innovation. Cette philosophie permet à chaque utilisateur d'un projet ou d'une solution de type Open Source d'accéder à son code source afin de l'adapter à des besoins très spécifiques. Ainsi, la totale liberté d'action sur le code source d'une solution ou d'un noyau de solution permet de mettre en œuvre une grande diversité d'enrichissements ou d'innovations.

Ainsi, c'est tout bonnement « en marchant » que le mode de développement Open Source fait la preuve de son efficacité. Le succès - voire l'hégémonie dans certains domaines - des logiciels qui ont été conçus selon ce mode de développement constitue sans doute le meilleur argument auprès des utilisateurs, des programmeurs ou des entreprises.

On a souvent relevé que si les logiciels libres devaient disparaître d'un coup, l'Internet n'y survivrait pas. Malgré les tentatives de propriétérisation des normes de la part de puissants éditeurs (cf. Microsoft, IBM, etc.) de logiciels dits « fermés », l'essentiel des programmes qui constituent le cœur d'Internet répondent aux critères définis par l'*Open Source Definition* et doivent tout à la collaboration de la communauté mondiale des développeurs « libres ».

A titre d'exemple, fin mars 2000, le serveur HTTP *Apache* avait franchi le cap des 60% de parts de marché, laissant loin derrière les applications propriétaires concurrentes. *Bind*, pour sa part, constitue l'implémentation de référence en matière de serveurs de noms (DNS), tandis que le langage *Perl*, particulièrement bien adapté aux technologies du World Wide Web, s'est répandu sur la quasi totalité des plates-formes matérielles et des systèmes d'exploitation.

Le simple constat de ce succès constitue la meilleure preuve de la suprématie technique de l'Open Source; toutefois il ne suffit pas de montrer que *ça marche*, encore faut-il assurer que cela peut être *profitable*.

#### *2.1.4.2 Limiter les coups de développement*

Actuellement, de nombreuses entreprises investissent pour empêcher la copie ou faire en sorte qu'elle ne soit pas gratuite, car elles ne conçoivent pas d'autre issue pour rentabiliser leurs coûts de développement. On sait qu'en matière de brevets par exemple, les coûts relatifs à la veille à posteriori représentent l'essentiel des dépenses, bien plus que les droits de dépôt initiaux. En effet, il ne suffit pas de déposer un brevet pour assurer la « protection » de son savoir-faire : il faut mettre en place un réseau de surveillance parfois extrêmement étendu afin de pouvoir, le cas échéant, « piéger » le concurrent indélicat. La plupart du temps cette tâche est confiée à des sociétés spécialisées sauf dans le cas de firmes de dimension internationale. Les fonds consacrés à ces prestations ne constituent pas à proprement parler un investissement productif, bien qu'ils représentent la part principale d'un budget « brevet ». Cette stratégie de la « citadelle assiégée » a pu donner des résultats (permettre d'accumuler des bénéfices assis sur la défense et l'exploitation de technologies propriétaires) mais dans des domaines qui ne progressent pas rapidement. Or dans le secteur informatique cette stratégie a peu de chances de s'avérer payante à moyen et long terme car les évolutions des techniques et des savoir-faire sont au contraire très rapides. Quelques sociétés se sont efforcées de l'adapter en imaginant des protections matérielles contre la copie logicielle. Elles accroissaient ainsi les coûts de développement et de conditionnement de leurs produits sans que leurs clients bénéficient de ces investissements. Enfin elles permettaient à leurs concurrents de développer des solutions équivalentes se concentrant sur l'essentiel à des coûts inférieurs, et surtout moins contraignantes pour les utilisateurs.

#### *2.1.4.3 Réduire le cycle de développement*

Au delà du pur et simple abandon des investissements consacrés à la protection des produits, la méthode de développement Open Source autorise une diminution sensible des coûts par la réduction du cycle de développement.

La démarche Open Source autorise l'exploitation de ressources humaines et logicielles et suscite des économies considérables. Elle peut ainsi rendre accessible le développement de solutions ambitieuses à des sociétés qui n'en n'auraient probablement pas les moyens dans le cadre d'un développement propriétaire fermé. Parallèlement, le constat conserve la même pertinence pour les clients des sociétés de service. Gagnantes sur les délais - donc sur les coûts - ces dernières gagnent aussi en liberté et en indépendance vis-à-vis de leurs fournisseurs. Alors que les solutions propriétaires aspirent la clientèle dans un cycle de mises à jour et de renouvellement (matériel et logiciel) dont il est extrêmement difficile de se dégager, le choix de l'Open Source garantit la possibilité de pouvoir changer à tout moment de prestataire sans pour autant bouleverser le système d'information (adoption de formats de données et d'échange normalisés et pérennes).

Nous savons par ailleurs que 75 % des coûts du cycle de vie d'un logiciel se trouvent dans la maintenance. Dans un contexte propriétaire, cette phase fondamentale peut virer au cauchemar. Alors que les problèmes s'accroissent avec la complexité des solutions mises en oeuvre, il paraît déraisonnable de confier son système d'information à des sociétés qui devront elles-mêmes s'en remettre à des tiers pour maintenir et corriger leur code. Une disposition vraiment fiable pour s'assurer de sa qualité consiste au contraire à en permettre l'examen par des personnes compétentes, ce que l'on a coutume d'appeler dans le monde de l'Open Source la *revue des pairs*.

#### 2.1.4.4 Certains coûts subsistent

La gratuité du droit d'utilisation masque néanmoins certains coûts : une fois un package ou une distribution téléchargée, des compétences pointues - notamment en terme d'intégration - sont le plus souvent nécessaires. Des coûts de formation, de support et de mise à jour sont également à prévoir à moins de posséder ses propres équipes en interne, ce qui constitue une autre forme de dépenses (fixes). D'une manière générale, les coûts sont déportés vers le service.

La méthode de développement *Open Source* autorise une diminution sensible des coûts par la réduction du cycle de développement. Elle autorise ainsi l'exploitation de riches « gisements » de ressources humaines et logicielles et suscite des économies considérables. Le choix de l'*Open Source* garantit la possibilité de pouvoir changer à tout moment de prestataire sans pour autant bouleverser le système d'information. Tous ces avantages ont néanmoins un revers : des compétences pointues sur ces technologies sont nécessaires, notamment en terme d'intégration.

## 2.2 LES SUCCES BASES SUR L'OPEN SOURCE

### 2.2.1 Qui contribue au développement de l'Open Source ?

Plusieurs sociétés contribuent au développement de l'**Open Source**. On compte parmi elles :

- Red Hat Software,
- Madrake Soft,
- Corel, en particulier pour le projet wine,
- Cygnus, qui contribue au développement du fameux compilateur gcc,
- IBM, notamment pour les projets Postfix et Apache.

### 2.2.2 De fameux projets basés sur l'Open Source

De fameux logiciels ont des licences de distribution qui relèvent de l'**Open Source** :

- Sendmail
- Postfix
- Apache
- Bind/named
- le noyau Linux
- Emacs
- XEmacs
- Perl
- Gimp
- Samba
- Eclipse

## 2.3 UTILISATION DE L'OPEN SOURCE AU SEIN D'ITREC GESTION

### 2.3.1 Pourquoi utiliser l'Open Source dans un projet de développement ?

#### 2.3.1.1 Le choix de l'Open Source

Pour concevoir le produit GIMA WEB, il fallait faire un choix entre deux technologies : la technologie MICROSOFT DOT.NET, payante, ou la technologie JAVA. C'est vers la technologie Open Source JAVA J2EE que s'est porté le choix de la société. Mais pourquoi avoir opté pour cette solution pour développer GIMA WEB ?

La première raison est sans doute financière : les logiciels Open Source sont gratuits, ce qui permet de limiter les coûts de développement. Il est ainsi possible de tester une architecture sans achat de licence.

La deuxième est technique : le code source des logiciels Open Source étant disponible, un développeur peut comprendre le fonctionnement des logiciels, corriger des bugs, ou encore procéder à des évolutions. Le fait qu'un logiciel soit Open Source ne signifie pas qu'il soit de mauvaise qualité, au contraire : beaucoup de meneurs des projets Open Source sont des personnes très compétentes qui s'investissent par défi technique et pour la reconnaissance de leurs *pairs*.

Les avantages que l'on retire dans l'utilisation des technologies Open Source sont encore nombreux : les mises à jour sont gratuites et le coût des formations est peu élevé car de nombreuses documentations sont disponibles sur Internet. D'autres arguments plus marketing laissent entendre que les logiciels Open Source sont plus stables et que le respect des standards et la qualité du support sont meilleurs.

Enfin, d'autres critères de sélection ont également pu influencer le choix des technologies utilisées dans le développement de GIMA WEB. Par exemple :

- Les besoins du logiciel (niveau d'API),
- Le niveau de maturité de la solution Open Source choisie,
- Les évolutions possibles,
- Le leadership (et l'organisation de l'équipe de développement).

Il existe par ailleurs des outils permettant d'éviter de re-développer certains aspects d'une application : de nombreux algorithmes JAVA préexistants sont à la disposition du public ainsi que la documentation associée.

Ainsi, c'est l'atelier de développement ECLIPSE, outil offert par IBM à la communauté Open Source, qui a été choisi. IBM voit un grand intérêt à offrir son produit à la communauté Open Source puisque désormais ce sont les utilisateurs qui font vivre ECLIPSE. Cela n'engage donc plus aucun coût pour IBM qui se "contente" de développer WSAD, ensemble de plugins permettant d'enrichir l'atelier de développement ECLIPSE et qui offre également un support aux entreprises (hotline, etc.). Le nombre d'utilisateurs travaillant sur la plate-forme ECLIPSE est donc en accroissement constant et nombre de développeurs ont ainsi participé à l'amélioration du produit. Ce dernier ne cesse d'évoluer et connaît aujourd'hui un grand succès dans la communauté Open Source.

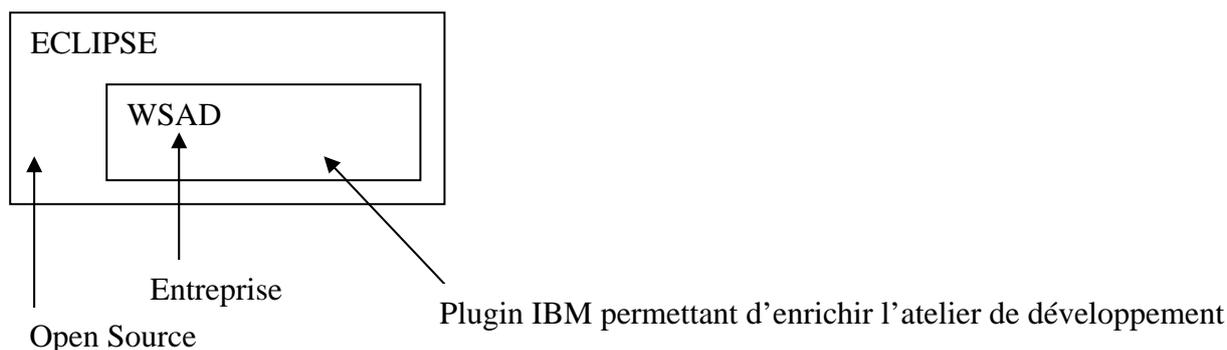


Figure 30 – Positionnement de WSAD par rapport à ECLIPSE

Ces avantages ont cependant un prix car il est parfois difficile pour un débutant d'installer ces logiciels et de les utiliser. C'est le sentiment que l'on peut éprouver lorsque l'on se penche pour la première fois sur l'architecture de GIMA WEB et sur les outils utilisés dans sa conception : le développeur peut se sentir un peu "perdu" devant le nombre de technologies qu'il va devoir apprendre à maîtriser.

Par ailleurs, force est de constater qu'"Open Source" rime souvent avec un support technique délicat à obtenir, et la veille technologique doit être permanente pour suivre les évolutions des produits sur le marché.

### 2.3.1.2 Architecture de GIMA WEB

Au départ l'application GIMA WEB avait été développée à l'aide d'outils NAT WEB (de NAT SYSTEM) et devait pouvoir fonctionner sur différents Systèmes de Gestion de Bases de Données. GIMA WEB a ensuite été porté sur une architecture J2EE (Java 2 Entreprise Edition) et comme nous l'avons vu précédemment, il est indispensable de maîtriser plusieurs technologies pour développer le produit : JAVA (comprenant les frameworks STRUTS et HIBERNATE) côté serveur, et JSP, HTML et JAVASCRIPT côté client. Ces technologies seront présentées de manière détaillée dans la suite de l'étude.

L'architecture JAVA J2EE permet aux entreprises de développer une application de bout en bout. Le framework HIBERNATE est un moteur d'accès aux données génériques. Cela permet ainsi de travailler sur la base de données sans avoir à se soucier de ce qu'il se fait sur les données.

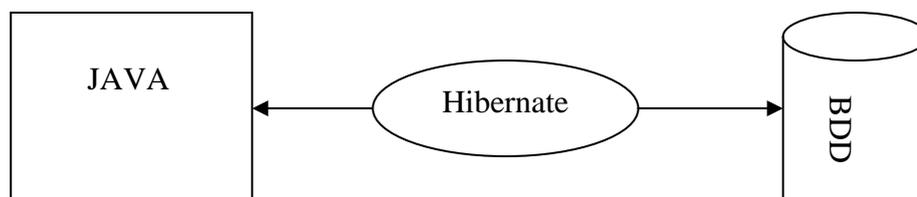


Figure 31 – Positionnement du framework HIBERNATE dans l'application

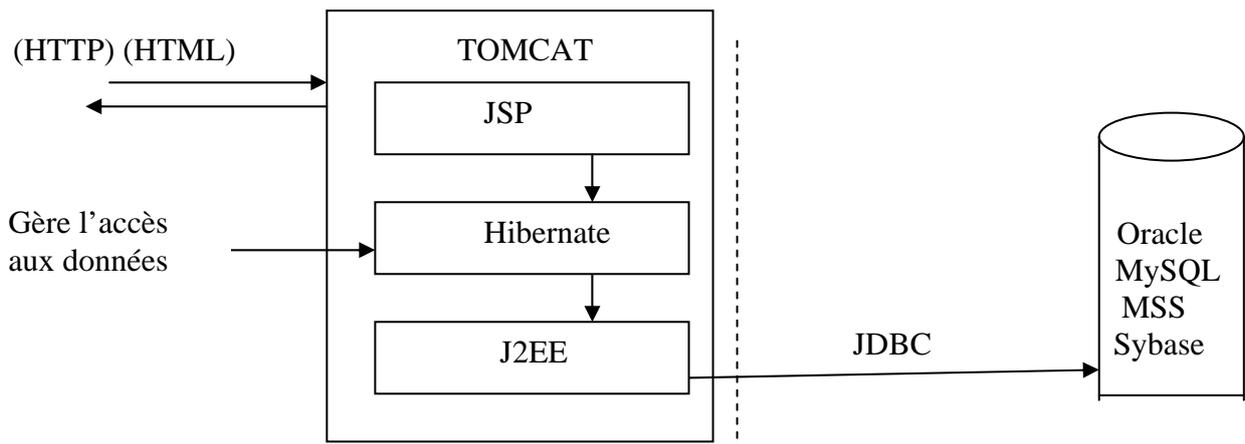


Figure 32 – Représentation schématique du fonctionnement de GIMA WEB

Au cours du développement de GIMA WEB, deux problèmes furent mis en avant :

- La gestion de l'accès aux bases de données,
- La gestion de la persistance des données. Celle - ci est gérée par le serveur d'application qui conserve une image des données chargées en base. Cela permet ainsi d'éviter de générer une requête SQL chaque fois qu'on accède à une information.

Ces problèmes furent résolus par l'utilisation du framework HIBERNATE qui communique avec la base de données : il s'adapte au format spécifique de la base.

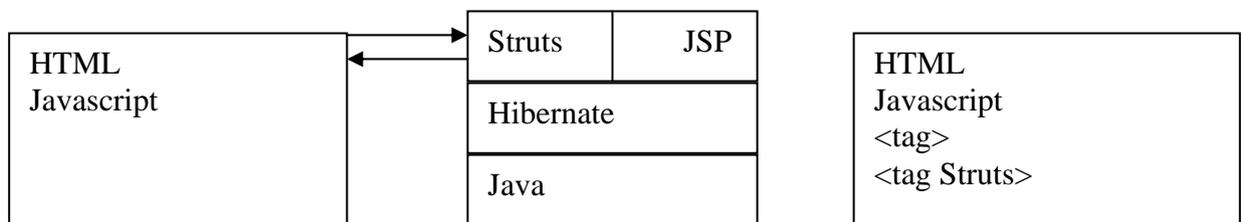


Figure 33 – Représentation schématique du fonctionnement de GIMA WEB

L'architecture de GIMA WEB s'appuie donc sur les frameworks STRUTS et HIBERNATE et sur le langage JAVA. Développer une application avec STRUTS implique l'utilisation de modèles de comportement appelés *templates* : les modèles MVC (modèle – vue – contrôleur) et MVC 2.



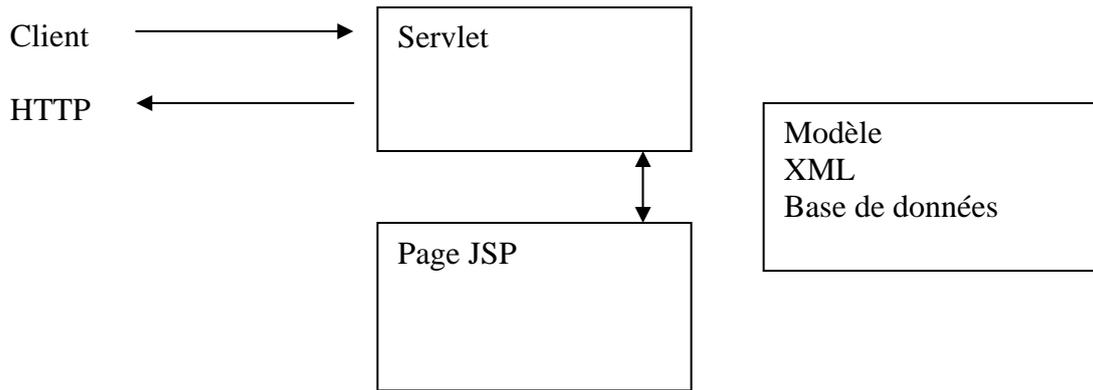


Figure 34 – Le modèle MVC

Dans le modèle MVC, le Servlet prend un message HTTP en entrée et les infos relatives au client. Il effectue les accès à la base et renvoie une réponse HTTP (HTML). Cela reste coûteux en terme de développement car cela nécessite un Servlet par traitement Client/Serveur.

La page JSP, ou Java Server Page, combine dans la même page du code HTML et du JAVA. Le Servlet doit ainsi préparer le traitement pour alimenter la page JSP.

Aujourd'hui nous sommes passé au modèle MVC 2 où l'on n'a qu'un seul Servlet générique, contrairement au modèle MVC où il peut y avoir une multitude de Servlet. Struts est livré avec un Servlet générique conforme au modèle MVC 2.

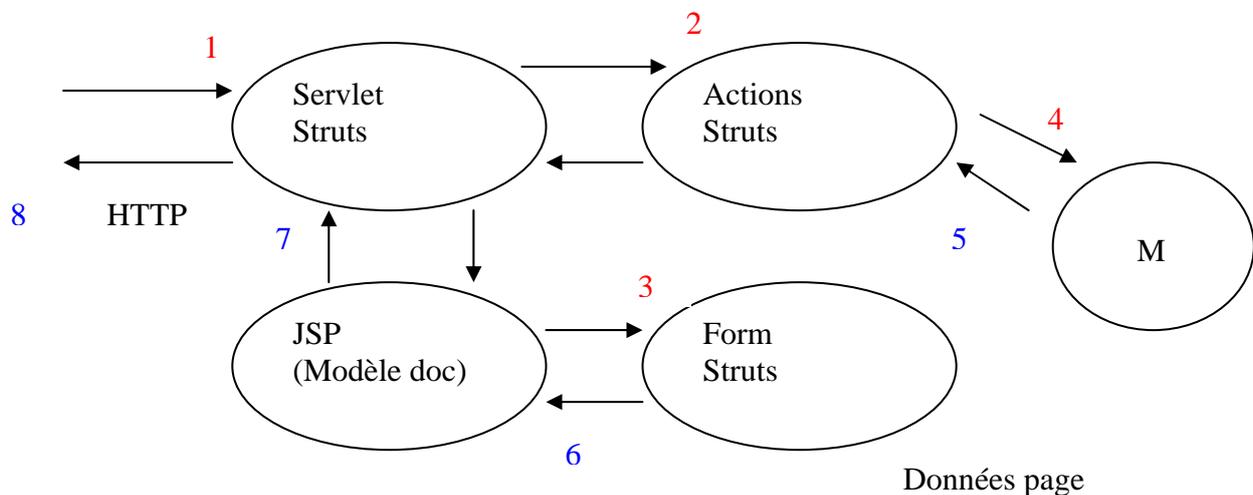
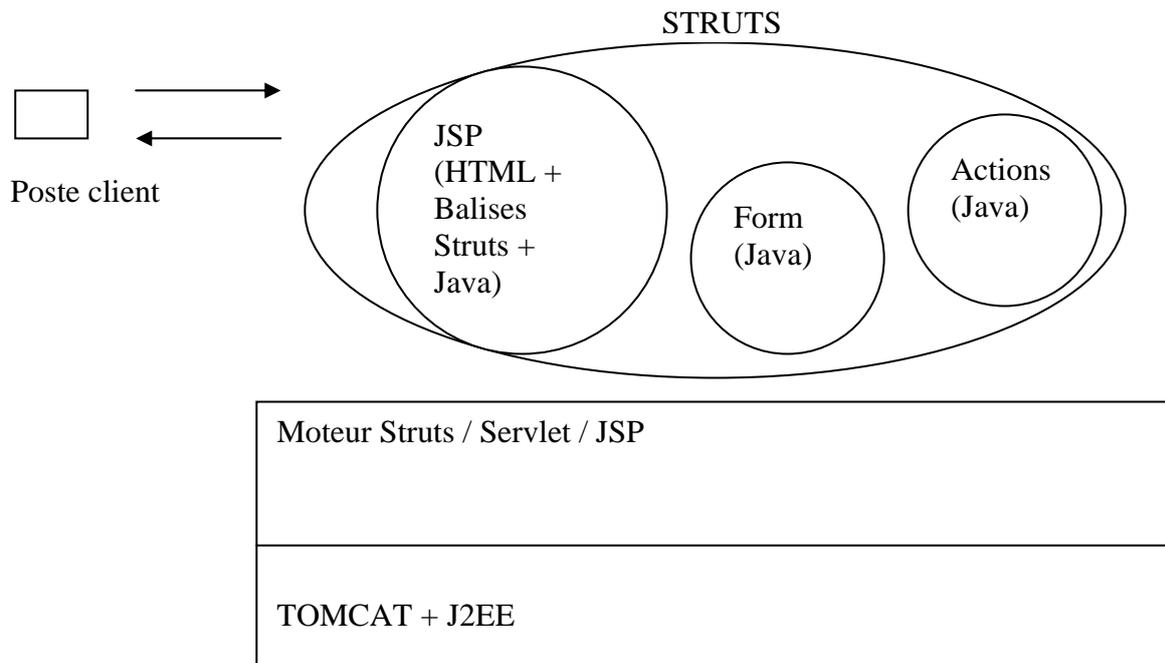


Figure 35 – Le modèle MVC 2

Le Servlet reçoit le flux HTTP, alimente une classe JAVA Formulaire et déclenche une action. L'action effectue un traitement au niveau de la base de données. Le Servlet est ainsi transparent pour l'utilisateur.



*Figure 36 – Architecture globale de GIMA WEB*

## 2.3.2 Les technologies utilisées dans le développement de GIMA WEB

### 2.3.2.1 Le langage HTML

Le HTML (*HyperText Markup Language*) est un système qui formalise l'écriture d'un document avec des balises de formatage indiquant la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents.

Il permet, entre autre, la lecture de documents sur Internet à partir de machines différentes grâce au protocole HTTP, permettant d'accéder via le réseau à des documents repérés par une adresse unique, appelée URL.

En effet le Web est une énorme archive vivante de textes formatés, d'images, sons, vidéos... Ces documents sont organisés grâce à des pages au format HTML et permettant la navigation. Le HTML n'est pas un langage de programmation, c'est un langage contenant des balises permettant de mettre en forme le texte, les images. Une balise est une commande (un nom) encadrée par le caractère inférieur (<) et le caractère supérieur (>) par exemple "<H1>".

Elle peut également avoir des paramètres :

Exemple : "<FONT Size="2" FACE="Arial">"

Les balises HTML peuvent être uniques; la balise <BR> représente par exemple un retour à la ligne. Elles peuvent également fonctionner par paire afin d'agir sur le texte qu'elles encadrent (la balise de fin est alors précédée d'un /) :

Exemple : <marqueur> Votre texte formaté </marqueur>

Ainsi les balises <b> et </b> (équivalant respectivement à <B> et </B>) permettent de mettre en gras le texte qu'elles encadrent :

Exemple : <b> Ce texte est en gras </b>

### 2.3.2.2 Le langage JAVASCRIPT

JAVASCRIPT a été mis au point par *Netscape* en 1995. A l'origine, il se nommait *LiveScript* et était destiné à fournir un langage de script simple au navigateur *Netscape Navigator 2*.

Il a à l'époque longtemps été critiqué pour son manque de sécurité, son développement peu poussé et l'absence de messages d'erreur explicites rendant difficile son utilisation.

Le 4 décembre 1995, suite à une association avec le constructeur *Sun*, *Netscape* rebaptise son langage JAVASCRIPT (un clin d'oeil au langage Java développé par *Sun*). A la même époque, *Microsoft* mit au point le langage Jscript, un langage de script très similaire.

Ainsi, pour éviter des dérives de part et d'autre, un standard a été défini pour normaliser les langages de script, il s'agit de l'ECMA 262, créé par l'organisation du même nom (ECMA, *European Computer Manufactures Association*). Le JAVASCRIPT est un langage de script incorporé dans un document HTML.

Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur Web.

Ainsi le langage JAVASCRIPT est fortement dépendant du navigateur appelant le page Web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage JAVA, avec lequel il a longtemps été confondu.

### 2.3.2.3 Le langage XML

#### 2.3.2.3.1 Présentation

XML (*eXtensible Markup Language*) est un langage à balises étendu ou « langage à balises extensibles ». Contrairement à HTML - qui est à considérer comme un langage défini et figé (avec un nombre de balises limité) - XML peut être considéré comme un métalangage permettant de définir d'autres langages, c'est-à-dire définir de nouvelles balises permettant de décrire la présentation d'un texte.

En réalité les balises XML décrivent le contenu plutôt que la présentation (contrairement à HTML). Ainsi, XML permet de séparer le contenu de la présentation, ce qui permet par exemple d'afficher un même document sur des applications ou des périphériques différents sans pour autant nécessiter de créer autant de versions du document que l'on nécessite de représentations.

XML a été mis au point par le *XML Working Group* sous l'égide du *World Wide Web Consortium* (W3C) dès 1996. Depuis le 10 février 1998, les spécifications XML 1.0 ont été reconnues comme recommandations par le W3C, ce qui en fait un langage reconnu.

#### 2.3.2.3.2 Structure des documents XML

XML fournit un moyen de vérifier la syntaxe d'un document grâce aux DTD (*Document Type Definition*). Il s'agit d'un fichier décrivant la structure des documents y faisant référence grâce à un langage adapté. Ainsi un document XML doit suivre scrupuleusement les conventions de notation XML et peut éventuellement faire référence à une DTD décrivant l'imbrication des éléments possibles. Un document suivant les règles de XML est appelé document bien formé. Un document XML possédant une DTD et étant conforme à celle-ci est appelé document valide.

#### 2.3.2.3.3 Principaux atouts de XML

- La lisibilité : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML,
- Auto descriptif et extensible,
- Une structure arborescente : permettant de modéliser la majorité des problèmes informatiques,
- Universalité et portabilité : les différents jeux de caractères sont pris en compte,
- Déployable : il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme http,
- Intégrabilité : un document XML est utilisable par toute application pourvue d'un parser (c'est-à-dire un logiciel permettant d'analyser un code XML),
- Extensibilité : un document XML doit pouvoir être utilisable dans tous les domaines d'applications.

#### 2.3.2.4 Le langage JAVA

JAVA est un langage objet ressemblant au langage C++. Il a été mis au point en 1991 par la firme *SUN MICROSYSTEMS*. Le but de JAVA à l'époque était de constituer un langage de programmation pouvant être intégré dans les appareils électroménagers, afin de pouvoir les contrôler, les rendre interactifs, et surtout de permettre une communication entre les appareils. Ce programme de développement se situait dans un projet appelé *Green*, visant à créer une télécommande universelle (*Star 7*) comprenant un système d'exploitation capable de gérer l'ensemble des appareils électroménagers de la maison. Etant donné que le langage C++ comportait trop de difficultés, James Gosling, un des acteurs du projet (considéré désormais comme le père de JAVA) décida de créer un langage orienté objet reprenant les caractéristiques principales du C++, en éliminant ses points difficiles, et en le rendant moins encombrant et plus portable (il devait pouvoir être intégré dans n'importe quel appareil). Ainsi, ce langage fut baptisé dans un premier temps 'Oak' (Oak signifiant chêne). Toutefois, puisque ce nom était déjà utilisé, il fut rebaptisé JAVA en l'honneur de la boisson préférée des programmeurs, c'est-à-dire le café, dont une partie de la production provient de l'île Java.

A la même époque, le Web faisait son apparition, or JAVA possédait toutes les caractéristiques faisant de lui un langage approprié pour le Web : le réseau des réseaux rassemblant sur une même structure des machines différentes, il fallait un langage capable de fonctionner sur chacune d'entre-elles : JAVA était conçu pour être portable.

Le Web était limité en bande passante : JAVA était conçu pour être petit. Ainsi, en 1994, l'équipe décida de mettre au point un navigateur (baptisé *HOT JAVA*) intégrant JAVA et capable de faire fonctionner des applets (des petites applications fonctionnant dans un navigateur). C'est ainsi que fin 1995 JAVA eut un terrible essor avec le soutien de *Netscape*, qui ne tarda pas à inclure JAVA dans son navigateur.

JAVA est très proche du langage C++ étant donné qu'il a quasiment la même syntaxe. Toutefois JAVA est plus simple que le langage C++ bien qu'il s'en inspire, car les caractéristiques critiques du langage C++ (celles qui sont à l'origine des principales erreurs) ont été supprimées. Cela comprend :

- Les pointeurs,
- La surcharge d'opérateurs,
- L'héritage multiple,
- La libération de mémoire est transparente pour l'utilisateur (il n'est plus nécessaire de créer de destructeurs),
- Une meilleure gestion des erreurs,
- Les chaînes et les tableaux sont désormais des objets faisant partie intégrante du langage,
- Toutefois JAVA est beaucoup moins rapide que le langage C++, il perd en rapidité ce qu'il gagne en portabilité.

Le fichier source d'un programme écrit en JAVA est un simple fichier texte dont l'extension est par convention *.java*. Ce fichier source doit être un fichier texte non formaté, c'est-à-dire un fichier texte dans sa plus simple expression, sans mise en forme particulière ou caractères spéciaux, c'est-à-dire qu'il contient uniquement les caractères ASCII de base. Lorsque le programme est prêt à être "essayé", il s'agit de le compiler (le traduire en langage machine) à l'aide d'un compilateur. Toutefois, contrairement aux langages compilés traditionnels, pour lesquels le compilateur crée un fichier binaire directement exécutable par un processeur donné (c'est-à-dire un fichier binaire contenant des instructions spécifiques à un processeur), le code source JAVA est compilé en un langage intermédiaire (appelé pseudo code ou *bytecode*) dans un fichier portant le même nom que le fichier source à l'exception de son extension (*.class*).

### 2.3.2.5 Les Java Server Pages

#### 2.3.2.5.1 Présentation

Les JSP (*Java Server Pages*) sont un standard permettant de développer des applications Web interactives, c'est-à-dire dont le contenu est dynamique. C'est-à-dire qu'une page Web JSP (repérable par l'extension .jsp) aura un contenu pouvant être différent selon certains paramètres (des informations stockées dans une base de données, les préférences de l'utilisateur, etc.) tandis qu'une page Web "classique" (dont l'extension est .htm ou .html) affichera continuellement la même information.

Il s'agit en réalité d'un langage de script puissant (un langage interprété) exécuté du côté du serveur (au même titre que les scripts CGI, PHP, ASP, ...) et non du côté client (les scripts écrits en JAVASCRIPT ou les applets JAVA s'exécutent dans le navigateur de la personne connectée à un site).

Les JSP sont intégrables au sein d'une page Web en HTML à l'aide de balises spéciales permettant au serveur Web de savoir que le code compris à l'intérieur de ces balises doit être interprété afin de renvoyer du code HTML au navigateur du client.

Ainsi, les *Java Server Pages* s'inscrivent dans une architecture 3-tiers : ce terme signifie qu'un serveur supportant les *Java Server Pages* peut servir d'intermédiaire (on parle généralement de serveur applicatif) entre le navigateur du client et une base de données (on parle généralement de serveur de données) en permettant un accès transparent à celle-ci.

JSP fournit ainsi les éléments nécessaires à la connexion au Système de Gestion de Bases de Données et à la manipulation des données grâce au langage SQL.

#### 2.3.2.5.2 Le fonctionnement des Java Server Pages

Une page utilisant les *Java Server Pages* est exécutée au moment de la requête par un moteur de JSP, fonctionnant généralement avec un serveur Web ou un serveur applicatif. Le modèle des JSP étant dérivé de celui des servlets (en effet les JSP sont un moyen d'écrire des servlets), celle-ci est donc une classe JAVA dérivant de la classe `HttpServlet`, et utilisant les méthodes `doGet()` et `doPost()` permettant de renvoyer une réponse par le protocole HTTP.

Lorsqu'un utilisateur appelle une page JSP, le serveur Web appelle le moteur de JSP qui crée un code source JAVA à partir du script JSP, compile la classe afin de fournir un fichier compilé `.class`, c'est-à-dire qu'il constitue en fait une servlet à partir du script JSP.

En réalité ce processus est un peu plus perfectionné : le moteur de JSP vérifie si la date du fichier `.jsp` correspond à celle du fichier `.class`. Le moteur de JSP ne transforme et compile la classe que dans le cas où le script JSP a été mis à jour. Ainsi, le fait que la compilation ne se fasse que lors de la mise à jour du script JSP, fait de cette technologie une des plus rapides pour créer des pages dynamiques.

En effet, la plupart des technologies de pages actives (ASP, PHP, etc.) reposent sur un code interprété, ce qui requiert beaucoup de ressources pour fournir la réponse HTTP. Etant donné que les JSP sont compilées (en fait il s'agit d'un *bytecode*) elles sont beaucoup plus rapides à l'exécution.

Actuellement seuls les scripts FastCGI (utilisant du code compilé écrit en langage C) sont plus rapides car ils ne nécessitent pas l'intervention d'une machine virtuelle pour exécuter l'application.

### 2.3.2.5.3 Caractéristiques des Java Server Pages

Les JSP permettent donc d'écrire des servlets en incluant dans des balises spécifiques le code JSP au sein du fichier HTML. De cette façon, elles fournissent une technologie rapide afin de créer des pages dynamiques.

De plus, les JSP étant basées sur JAVA côté serveur, elles possèdent toutes les caractéristiques faisant la force de JAVA :

- Les JSP sont multithreadées,
- Les JSP sont portables,
- Les JSP sont orientées objet,
- Les JSP sont sûres.

### 2.3.2.5.4 Syntaxe des balises JSP

Les pages JSP portent l'extension .jsp. Elles sont compilées sous forme de servlets, et présentent donc des balises spécifiques permettant de manipuler du code JAVA. Ces balises définissent des directives, des déclarations, des scriptlets et des expressions.

#### Exemple :

Commençons par un exemple de directives :

```
<% @ method="doPost" import="java.io.*" %>
<% @ implements="java.io.Serializable" %>
```

Les directives affectent des valeurs à des variables. On peut ainsi implémenter une méthode ou spécifier le type de contenu en sortie (output content), etc. Les différentes variables sont les suivantes : language, method, import, content\_type, implements, et extends.

Les déclarations ont la forme suivante :

```
<SCRIPT runat=server>
public void init (ServletConfig config) throws ServletException
{
// code de la méthode
}
</SCRIPT>
```

Les scriptlets identifient le code JAVA destiné aux servlets, par exemple :

```
<%
String nom = request.getParameter("Nom");
System.out.println(nom);
%>
```

fournit en sortie le contenu du paramètre "Nom".

Enfin les expressions sont une notation raccourcie pour la valeur de retour d'un servlet ainsi,

```
<%= request.getParameter("Nom"); %>
```

 a le même effet que le scriptlet qui précède.

Par ailleurs, les balises spécifiant l'accès à un JavaBean de contenu résultant ont la forme suivante (N.B : l'exemple ne présente pas tous les attributs possibles) :

```
<BEAN name="Nom" varname=monBean introspect="yes" create="no" scope="session">
</BEAN>
```

L'attribut *name* contient la clé permettant de retrouver le JavaBean; l'attribut *varname* contient le nom de la variable locale qui désigne le JavaBean et qui sera utilisée dans les scriptlets; l'attribut *introspect* fixé à "yes" précise que pour toute propriété du JavaBean dont le nom correspond au nom d'un paramètre d'une requête, la méthode de réglage doit être appelée, à chaque requête, pour fixer cette propriété à la valeur du paramètre correspondant; l'attribut *create* fixé à "yes" spécifie que le JavaBean doit être créé s'il n'existe pas (fixé à "no", un message d'erreur est renvoyé) ; enfin, l'attribut *scope* peut être fixé à "request" (dans ce cas, le JavaBean ne sert que le temps d'une requête) ou à "session" (le JavaBean peut alors être utilisé par des requêtes multiples).

### 2.3.2.6 *JUNIT, framework permettant la mise en œuvre de tests unitaires*

Le framework JUNIT est l'œuvre conjointe de Kent Beck (créateur de *XP*) et Erich Gamma (auteur des *Design Patterns*); il est gratuit et téléchargeable à partir du site [junit.org](http://junit.org). Il est écrit pour le langage JAVA, mais des frameworks gratuits inspirés de celui-ci sont également disponibles pour de nombreux autres langages : C++, Python...

JUNIT permet de mettre en œuvre des tests unitaires en langage JAVA. J'ai pu utiliser JUNIT chez ITREC GESTION à la fin de ma seconde année d'études à 3IL; je m'étais alors documenté sur l'utilisation de ce framework. Le principe consiste à écrire une classe de test pour chaque classe que l'on souhaite tester; JUNIT permet ensuite d'exécuter en séquence les tests écrits par le développeur.

Ecrire un jeu de tests unitaires demande un peu de pratique ; l'utilisation d'un framework de tests unitaires est toutefois essentielle à la constitution d'un code robuste; cela permet de placer le code en situation difficile et d'en améliorer la lisibilité. La mise en œuvre de tests unitaires est essentielle lors de la conception d'applications : leur importance dans l'intégration permet ainsi de limiter les effets de bord et la propagation d'erreurs. Par ailleurs les méthodes de test apportent de la valeur documentaire au code.

Avant de livrer une application, il est donc important de prendre le temps de la tester de manière exhaustive; en effet, plus on teste régulièrement des parties d'application, plus le bug est facilement détectable. Il en est de même lorsque des modifications sont apportées à l'application. On peut ainsi passer un temps considérable sur le débogage de quelques lignes de code alors que quelques secondes suffisent à un test pour mettre toute une application à l'épreuve. Enfin, la non régression du code est respectée : on peut modifier un code pour qu'il résiste au pire et l'on sait immédiatement si il y a un bug, où et pourquoi... Cela permet donc au développeur d'avancer plus vite dans son travail.



Le test de logiciel est déterminant pour la qualité du produit.

Réaliser des tests présente les avantages suivants :

- ➔ Analyse de petite granularité des méthodes qui composent l'application,
- ➔ Certitude d'engendrer peu de bugs,
- ➔ Non régression du code (stabilisation du produit),
- ➔ Documente efficacement le code.

Les tests permettent d'éviter :

- ➔ Retard de mise en exploitation,
- ➔ Surcoût de développement lié à la correction des erreurs,
- ➔ Perte de confiance dans l'outil de travail,
- ➔ Rejet de l'utilisateur final.

Le développement de tests unitaires en langage JAVA facilite et optimise le travail du développeur en automatisant les tests unitaires sur les classes : en effet, le temps passé à déboguer une application est souvent long et fastidieux puisque il représente près de 80% du temps passé pour un développeur débutant, et 50% du temps pour un développeur confirmé.

#### 2.3.2.7 JFreeChart, librairie permettant de générer des graphiques en JAVA

Dans de nombreux projets de développement, il est nécessaire de proposer des graphiques statistiques sur l'application et son domaine d'application. Dans la plupart des cas, ces graphiques reposent sur différents modes de représentations typiques : histogrammes, camemberts, courbes de répartitions et graphiques d'évolution en sont quelques exemples.

Ce besoin se retrouve quelque soit le langage et donc, naturellement, en JAVA. La bibliothèque JFreeChart répond à cette problématique en permettant de générer facilement des diagrammes de qualités. JFreeChart est sans doute, dans le monde JAVA, l'outil de génération de graphiques le plus connu et le plus réputé, et ce pour diverses raisons :

- En tant que librairie Open Source, JFreeChart bénéficie d'un avantage important face à ces concurrents : sa gratuité. Par ailleurs, lors du téléchargement, on reçoit les codes sources du produit.
- JFreeChart est un outil facile à utiliser et à intégrer dans une application, qu'elle soit installable, ou utilisée depuis un navigateur web.
- Les résultats fournis sont d'une qualité remarquable (cf. travail réalisé sur les *synthèses financières* dans GIMA WEB).

#### 2.3.2.8 TOMCAT

TOMCAT est un servlet - c'est à dire un programme JAVA tournant sur un serveur - qui gère d'autres servlets et des pages serveur JAVA (JSP ou *Java Server Pages*) permettant de générer des pages Web dynamiques. Il peut être utilisé seul ou couplé avec un serveur (dont la liste est spécifiée dans la documentation). TOMCAT répond aux spécifications de *Sun* concernant les servlets et leurs gestionnaires. Il supporte l'ensemble des classes définies par *Sun* dans le document "*Java API Servlet Specifications*", dont la consultation est fortement recommandée pour un usage avancé de TOMCAT.

TOMCAT est totalement gratuit et généralement utilisé en couplage avec un serveur Apache. Ecrit en JAVA, il nécessite, pour pouvoir fonctionner, la présence d'une machine virtuelle JAVA, et plus précisément du SDK - *Sun Development Kit* - complet. Ceci entraîne le fait que TOMCAT est totalement portable et peut être mis en oeuvre sur des systèmes radicalement différents tels que Linux ou Windows.

Il faut cependant noter que sous Windows, l'utilisation de TOMCAT est moins sécurisée et plus difficile à configurer sur des systèmes Windows 95/98 puisque TOMCAT ne peut y modifier dynamiquement les variables d'environnement.

### 2.3.2.9 JAVA BEANS

La technologie JSP suggère fortement (mais ne requiert pas) l'utilisation des JavaBeans, que l'on peut décrire comme des composants réutilisables, multi plates-formes (c'est le credo du langage JAVA), et résidant sur le serveur d'application. Les servlets peuvent faire appel à des JavaBeans pour traiter les informations qui leur sont transmises par la requête du client. Les JavaBeans, à leur tour, peuvent faire appel à tout autre service dit de "back-end", comme des bases de données ou des *Enterprise JavaBeans* (EJB). Notons que la requête est envoyée à la page JSP la première fois (celle-ci est alors compilée), directement au servlet ensuite (sauf si la page JSP a été modifiée). Notons également que l'on peut envoyer la requête à un servlet faisant appel à un JavaBean, et renvoyer le contenu dynamique sous forme de JavaBean à l'aide d'un autre servlet, celui-ci issu de la compilation d'une page JSP (JSP servlet).

Un des arguments en faveur de l'utilisation des JavaBeans avec des servlets est la séparation du code (proprement dit, c'est-à-dire l'aspect algorithmique) de la présentation. En utilisant ces composants réutilisables, on minimise la quantité de code qui figurera sur la page JSP qui, idéalement, ne contient que les balises pour l'accès au contenu et le rendu (présentation) du contenu.

Un JavaBean se caractérise par des propriétés, des méthodes, et des événements associés. Un événement est la notification au JavaBean que quelque chose est survenu et doit donner lieu à un traitement. Une méthode est une action ou un service réalisé par le JavaBean; les propriétés, enfin, sont définies par une paire de méthodes, une méthode "d'obtention" (getter) et une méthode de "réglage" (setter). Ainsi, dans l'exemple suivant :

```
public Date getTime();  
public void setTime(Date newTime);
```

La propriété Date est définie par le couple de méthode getTime() (méthode d'obtention), et setTime(Date newTime) (méthode de réglage). Ici, la propriété Date est une propriété modifiable par l'utilisateur : la méthode de réglage fait donc elle-même logiquement appel à Date. Les JavaBeans représentent la logique applicative dont les méthodes exposent les services. Un servlet utilise la logique applicative encapsulée dans un ou plusieurs JavaBeans, et le contenu dynamique résultant, stocké lui-même dans un JavaBean, est récupéré par une page JSP pour répondre à la requête initiale. Une balise spécifique permet d'accéder au JavaBean résultant.

### 2.3.2.10 JDBC

La technologie JDBC (*Java DataBase Connectivity*) est un ensemble de classes permettant de développer des applications capables de se connecter à des serveurs de bases de données (SGBD).

Dans un système client/serveur, l'accès aux bases de données avec JDBC peut s'effectuer selon un modèle à deux couches ou bien un modèle à trois couches.

Pour le modèle à deux couches, une application JAVA est intimement liée avec une base de données. A cet effet il faut bien évidemment disposer, pour la base de données concernée, d'un pilote JDBC adéquat. Les instructions SQL sont directement envoyées à la base, cette dernière renvoyant les résultats par un biais tout aussi direct. La base de données peut être exécutée sur la machine locale (celle sur laquelle l'application JAVA fonctionne) ou bien sur tout autre ordinateur du réseau (Intranet ou Internet).

### 2.3.2.11 ORACLE

ORACLE est un SGBD (Système de Gestion de Bases de Données) édité par la société du même nom (*Oracle Corporation*), leader mondial des bases de données.

La société *Oracle Corporation* a été créée en 1977 par Lawrence Ellison, Bob Miner, et Ed Oates. Elle s'appelle alors *Relational Software Incorporated* (RSI) et commercialise un Système de Gestion de Bases de Données Relationnelles (SGBDR ou RDBMS pour *Relational Database Management System*) nommé ORACLE.

En 1979, le premier prototype (RDBMS - RSI1) intégrant la séparation des espaces d'adressage entre les programmes utilisateurs et le noyau ORACLE est commercialisé. Cette version est entièrement développée en langage assembleur. La seconde version (RDBMS - RSI2) est un portage de l'application sur d'autres plates-formes. En 1983, la troisième version apporte des améliorations au niveau des performances et une meilleure prise en charge du SQL. Cette version est entièrement codée en langage C. A la même époque RSI change de raison sociale et devient ORACLE. En 1984, la première version d'ORACLE (ORACLE 4) est commercialisée sur les machines IBM. En 1985, ORACLE 5 permet une utilisation client - serveur grâce au middleware SQL\*Net. En 1986, ORACLE a été porté sur la plate-forme 8086. En 1988, ORACLE 6 est disponible sur un grand nombre de plates-formes et apporte de nombreuses nouvelles fonctionnalités ainsi qu'une amélioration notable des performances. En 1991, ORACLE 6.1 propose une option Parallel Server (dans un premier temps sur la DEC VAX, puis rapidement sur de nombreuses autres plates-formes). En 1992, ORACLE 7 sort sur les plates-formes UNIX (elle ne sortira sur les plates-formes Windows qu'à partir de 1995). Cette version permet une meilleure gestion de la mémoire, du CPU et des entrées-sorties. La base de données est accompagnée d'outils d'administration (SQL\*DBA) permettant une exploitation plus aisée de la base. En 1997, la version ORACLE 7.3 (baptisée ORACLE Universal Server) apparaît, suivie de la version 8 offrant des capacités objet à la base de données. ORACLE est écrit en langage C et est disponible sur de nombreuses plates-formes matérielles (plus d'une centaine) dont :

- AIX (IBM),
- Solaris (Sun),
- HP/UX (Hewlett Packard),
- Windows NT (Microsoft),
- Linux depuis la version 8.0.5 d'ORACLE.

### 2.3.2.12 CVS

#### 2.3.2.12.1 Description générale

CVS (*Concurrent Versions System*) est un outil de suivi de version. Il permet de conserver la trace de l'historique des modifications d'un fichier ou d'un ensemble de fichiers, et de revenir simplement à n'importe quel état antérieur.

Pour garder cet historique, il serait bien sûr possible de conserver chaque version de chaque fichier créés ou manipulés mais cela prendrait un espace disque considérable. L'intérêt que l'on retire dans l'utilisation de CVS est qu'il stocke seulement les différences entre deux versions successives, réalisant ainsi un gain de place appréciable.

Ensuite, CVS enregistre automatiquement la date, l'auteur et un éventuel commentaire explicatif pour chaque modification.

Un autre avantage de CVS est qu'il facilite le travail en groupe. En effet, travailler à plusieurs dans un même répertoire demande une organisation sans faille et une attention de tous les instants afin de ne pas écraser les modifications faites par un autre membre de l'équipe de développement. CVS permet à chacun de travailler dans son propre répertoire, même sur sa propre machine, et prend en charge l'intégration du travail des différents membres du groupe. Notons également que CVS est un logiciel libre, diffusé sous licence GPL. Il fonctionne sur une grande variété de machines et de systèmes d'exploitation : Linux, Solaris, BSD, MacOS, MS Windows, etc.

#### **2.3.2.12.2 Principe de base**

CVS stocke tout l'historique des évolutions d'un fichier sous forme compacte dans un fichier historique. Les fichiers historiques sont conservés dans des *repositories* (dépôt).

Chaque fois qu'un utilisateur souhaite travailler sur un fichier ou un ensemble de fichier, il demande à CVS de produire, à partir du contenu du *repository* adéquat, une copie privée des fichiers souhaités dans la version désirée. Cette opération est appelée *checkout* dans la terminologie CVS. C'est sur cette copie personnelle, appelée copie de travail que l'utilisateur effectuer ses modifications, tests, etc.

Une fois satisfait de son travail, l'utilisateur demande à CVS de mettre à jour le *repository* en y incorporant les modifications qu'il a réalisées sur sa copie de travail. Cette opération est appelée *commit* par CVS. L'utilisateur peut ensuite continuer à travailler avec sa copie personnelle et demander à CVS d'incorporer ses modifications au *repository* chaque fois qu'il le souhaite.

Une fois son travail terminé, si un utilisateur ne souhaite pas conserver sa copie de travail, il peut l'effacer et signaler à CVS qu'il ne possède plus de copie personnelle, grâce à un *release*. Il pourra aussi à tout moment recréer une copie de travail à partir du *repository*.

Dans le cas le plus général, plusieurs utilisateurs peuvent posséder une copie de travail des mêmes fichiers, et faire chacun des modifications qui leurs sont propres. Pour le premier qui demandera à CVS d'intégrer ses modifications dans le *repository*, tout se passera comme s'il était seul. Mais lorsque le deuxième utilisateur incorporera ses modifications dans le *repository*, CVS se rendra compte que ces modifications ont été portées sur une version plus ancienne que la version disponible dans le *repository* et combinera ces modifications à celles effectuées par le premier utilisateur. Il en profitera également pour mettre à jour la copie de travail du deuxième utilisateur.

Il est possible cependant que CVS ne sache pas combiner deux ensembles de modifications successives, par exemple si le deuxième utilisateur a modifié une ligne supprimée par le premier. Dans ce cas, il indique au deuxième utilisateur les fichiers en conflit et lui laisse le soin de résoudre manuellement le problème.

Il est à noter que le premier utilisateur n'est pas informé automatiquement des modifications apportées par le deuxième sur les fichiers sur lequel il travaille. S'il ne fait rien de particulier, il n'en aura connaissance que lors de son prochain *commit*. Tout utilisateur possédant une copie de travail peut cependant - et à tout moment - demander à CVS de la mettre à jour par rapport à la dernière version disponible dans le *repository*, autrement dit d'incorporer dans sa copie de travail toutes les modifications apportées par tous les utilisateurs travaillant sur les mêmes fichiers que lui. Cette opération s'appelle un *update*.

### 2.3.2.13 L'IDE ECLIPSE

Enfin je vais présenter l'environnement de développement ECLIPSE, choisi pour développer le progiciel GIMA WEB. Celui-ci à l'avantage de permettre la manipulation de sources JAVA, JSP, HTML, XML, etc., et de fournir une interface conviviale à CVS. Il permet également d'utiliser TOMCAT grâce à un plugin spécifique, et offre de nombreuses autres fonctionnalités.

ECLIPSE est un IDE (*Integrated Development Environment*) ou EDI (*Environnement de Développement Intégré*) développé à l'origine par IBM. Cette plate-forme est maintenant sous licence Open Source et sert comme base aux outils *WebSphere* ou *Together*.

Architecturé autour d'un système à plugins, il est conçu dès le départ pour être extensible. Pour preuve, le plugin JDT, un environnement de développement JAVA très complet (complétion syntaxique, refactoring, changement du code pendant le débogage) sert à développer ECLIPSE lui-même !

Au niveau ergonomique, ECLIPSE n'a rien à envier à ses concurrents. Toutes les fonctionnalités indispensables sont présentes : création de projet, de *Templates*, refactoring, debugage, etc. et remarquablement faciles à prendre en main. Mais la grande force de cet IDE réside dans l'ouverture de son noyau qui permet l'ajout de très nombreux plugins. Il est par exemple possible d'intégrer facilement un serveur d'application pour le debugage, un compilateur de pages JSP, un module de déploiement J2EE, un explorateur de bases de données... Enfin ECLIPSE tire son originalité dans le choix des composants graphiques utilisés lors de son développement. En effet, ECLIPSE ne contient pas d'AWT ni de SWING, mais uniquement des composants SWT/JFace (développés par IBM). Ainsi, l'interface gagne en rapidité et en convivialité.

Il présente toutefois l'inconvénient majeur d'être relativement lent, du fait qu'il est intégralement écrit en JAVA, et nécessite donc une machine assez puissante pour fonctionner correctement.

Ce chapitre a présenté les différentes technologies nécessaires pour mettre en place une application Web de type JAVA/JSP. ORACLE, CVS et ECLIPSE étant bien sûr optionnels mais fortement conseillés car il faut bien souvent stocker des données et dans ce domaine ORACLE reste la référence. Un bon IDE, fiable, convivial et efficace sera un atout supplémentaire surtout si celui-ci est couplé avec un système de gestion de version comme CVS.

## 2.4 EXPERIENCE RETIREE DE L'UTILISATION DE CES TECHNOLOGIES

Travailler sur le projet GIMA WEB m'a permis d'intégrer une équipe de développement sur un projet basé sur les technologies Open Source J2EE. J'ai découvert un nouvel environnement de travail puisque je ne connaissais pas les outils utilisés pour développer GIMA WEB.

J'ai dû fournir un gros effort en amont afin de comprendre quels étaient les mécanismes et les dépendances entre les divers outils utilisés dans le développement du progiciel (ECLIPSE, JAVA, TOMCAT, les frameworks STRUTS et HIBERNATE, CVS, etc.).

Ce qui m'a semblé le plus délicat lors de cette dernière mission en entreprise fut de comprendre le fonctionnement entre ces différentes technologies afin d'être productif et de poursuivre le développement de l'*Observatoire du Patrimoine*.

Il m'a donc fallu fournir un gros effort d'adaptation, mais j'ai également eu le privilège de travailler avec monsieur Frédéric Martinez qui m'a formé sur les produits et a répondu à mes interrogations lorsque je le sollicitais.

Cette mission m'a donc permis de me positionner sur des technologies Open Source qui sont actuellement très utilisées dans le milieu industriel que ce soit par les SSII ou les éditeurs de logiciels. En effet, les postes à pourvoir en qualité d'ingénieur d'étude et développement JAVA J2EE connaissent une forte croissance.

L'expérience que je retire de cette mission est donc très positive : j'ai acquis des compétences sur les technologies Open Source J2EE mais aussi de l'expérience sur le travail en équipe. J'ai enfin eu la chance de pouvoir travailler sur un véritable projet de développement.

## CONCLUSION GENERALE DE L'ETUDE

Cette dernière période en entreprise m'a permis d'acquérir une très bonne expérience professionnelle. J'ai ainsi pu travailler avec l'équipe d'ITREC GESTION, et plus particulièrement avec monsieur Frédéric MARTINEZ sur le développement du progiciel GIMA WEB. Cette période m'a également permis d'aborder de nouvelles missions telles que la recette de logiciels ou le développement en langage JAVA; j'ai ainsi pu élargir mon champ de connaissances et mes compétences techniques en me formant sur de nouvelles technologies très demandées actuellement, et que je n'avais pas abordées : les frameworks STRUTS et HIBERNATE, l'utilisation de la librairie Open Source JFreeChart, ou encore l'utilisation de CVS, un outil de suivi de versions.

J'ai également une meilleure connaissance des priorités et contraintes (économiques) des entreprises, ainsi que des différents acteurs qui travaillent au sein d'une société (qui peut régler tel problème, ou m'aider à avancer dans tel projet ?). Avoir réalisé cette formation d'ingénieur par l'apprentissage a donc facilité mon insertion en milieu professionnel, et j'ai pu remarquer qu'il est très différent du monde étudiant.

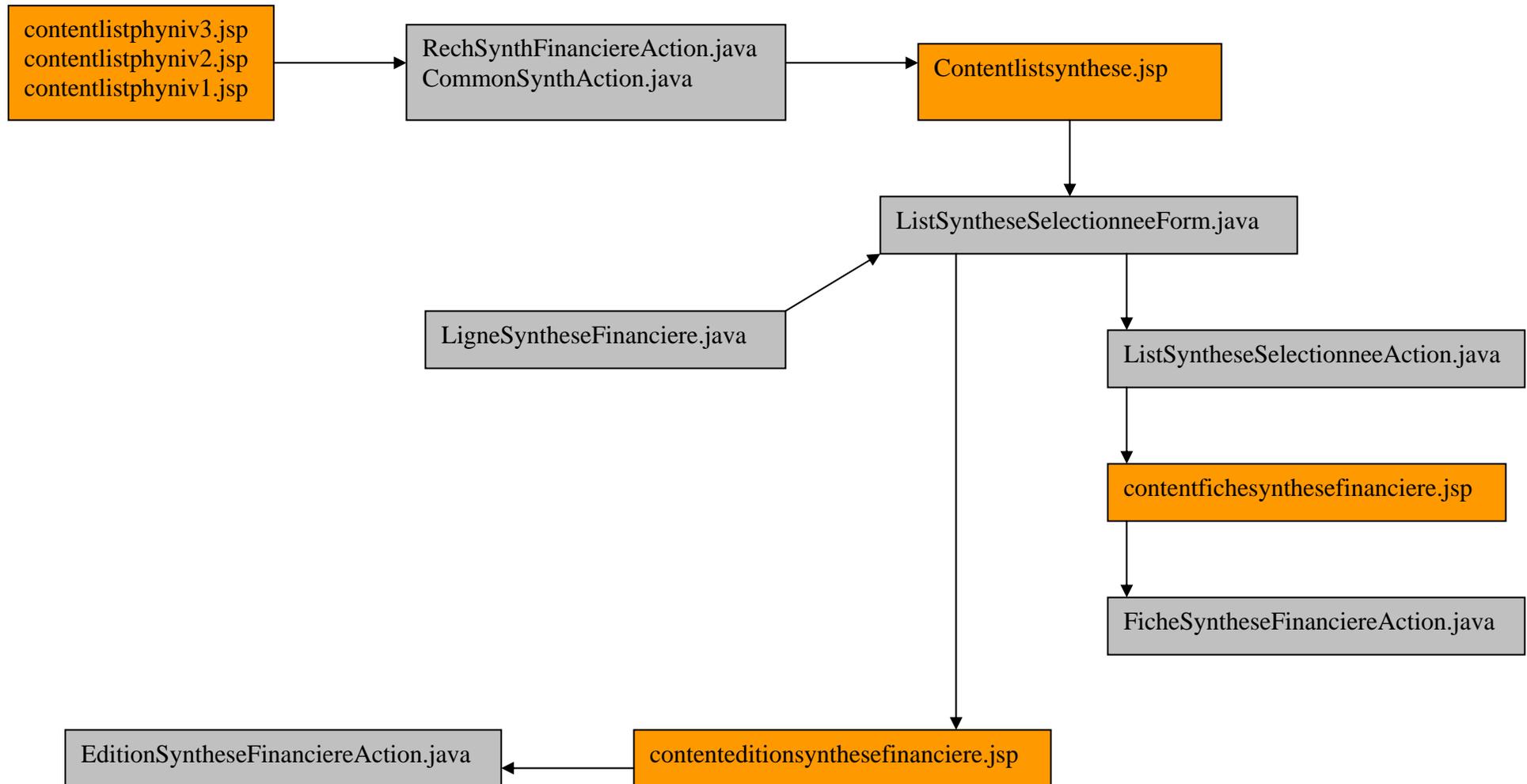
C'est donc un sentiment très positif que j'en retire puisque j'ai véritablement évolué tout au long de ces trois années de formation par l'apprentissage. Je peux le constater par l'évolution de mes compétences techniques mais aussi par les missions qui m'ont été confiées lors de ces différentes périodes en entreprise. Ce sentiment est renforcé par la gentillesse des personnes avec lesquelles j'ai été amené à travailler. Je pense plus particulièrement à mon tuteur en entreprise, monsieur Frédéric Martinez, et à monsieur Daniel Ghienne, directeur technique d'ITREC GESTION.

Enfin, cette dernière période qui clos ma formation d'ingénieur à 3IL m'a permis d'aborder des thèmes (méthodologies de test de logiciels, formation sur de nouvelles technologies WEB) qui constituent un très bon complément aux cours suivis à l'école. D'autre part j'ai eu la chance de travailler en équipe sur un projet de développement en langage JAVA, que nous avons étudié en troisième année.



## ANNEXES

Ci-dessous le schéma d'enchaînement des appels des pages dans le module Observatoire du Patrimoine de GIMA WEB :



## Les synthèses financières dans GIMA WEB - Code source d'exemple

Ci-dessous le code source d'une page que j'ai développée lors de ma dernière période en entreprise ; elle fait référence à la conception de *synthèses financières* pour l'application GIMA WEB.

### **Fichier « ListSyntheseSelectionneeAction.java »**

```
package com.itrec.gimaweb.patrimoine.web.action;

import java.math.BigDecimal;

import javax.servlet.ServletContext;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.itrec.framework.common.util.DateUtil;
import java.util.ArrayList;
import com.itrec.framework.common.util.StringUtil;

import net.sf.hibernate.Session;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.itrec.framework.action.BaseAction;

import com.itrec.gimaweb.main.business.manager.StFinanciereManager;
import com.itrec.gimaweb.main.business.mapping.StFinanciereVO;
```

```

import com.itrec.gimaweb.main.business.common.ListeIHM;
import com.itrec.gimaweb.main.business.mapping.HibernatePlugIn;

import com.itrec.gimaweb.patrimoine.ConstantsPatrimoine;
import com.itrec.gimaweb.patrimoine.web.ihm.LigneSyntheseFinanciere;
import com.itrec.gimaweb.patrimoine.web.form.ListSyntheseSelectionneeForm;

import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PiePlot3D;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.data.general.DefaultPieDataset;
import org.jfree.util.Rotation;
import java.awt.Color;
import java.awt.GradientPaint;
import java.io.File;

/**
 * Gestion de l'action pour l'affichage de la synthèse financière.
 * @author JDAHAN
 * @version 2.06
 * <p>Crée le : 17 mai 2005 par JDAHAN (version 2.06)
 * <p>Modifiée le : 05 juillet 2005 par JDAHAN (version 2.06)
 * <p>Modifiée le : 06 juillet 2005 par JDAHAN (version 2.06)
 * <p>Copyright Adeliior©2001
 * <p>All right reserved
 */

public final class ListSyntheseSelectionneeAction extends BaseAction {
    private BigDecimal totalHt;

```

```

private BigDecimal totalTva;
private BigDecimal totalTtc;

/**
 * Méthode appelée pour generer les blocs de lignes de la synthese financiere
 * @param min le numero de regle minimal
 * @param max le numero de regle maximum
 * @throws Exception
 * <p>Créé le : 8 juil. 2005 par JDAHAN (version 2.05)
 * <p>Modifiée le : xx xxxx xxxx par xxxxx (version 2.xx)
 */
protected void initListe(ArrayList listSyntheseVO, ListeIHM listSyntheseIHM, int min, int max)
throws Exception {
    LigneSyntheseFinanciere ligne = null; // Ligne synthèse financière a afficher
    StFinanciereVO synthese = null;

    int nbr=listSyntheseVO.size(); // nombre d'enregistrements trouves
    for (int cpt=0;cpt<nbr;cpt++) {
        synthese = (StFinanciereVO) listSyntheseVO.get(cpt); // Recuperation de l'objet StFinanciereVO
        if (synthese!=null) {
            // Constitution de la ligne a afficher
            ligne = new LigneSyntheseFinanciere();
            ligne.setNuregle(synthese.getNuregle().toString());
            int NumeroRegle = Integer.parseInt(ligne.getNuregle());

            // cas Depenses - Fonctionnement
            if(NumeroRegle>min && NumeroRegle<max) {
                ligne.setClasse("b");
                ligne.setExercice(synthese.getExercice().toString());
                ligne.setLbnomregle((String)synthese.getLbnomregle());
                ligne.setMtht(synthese.getMtht());
                ligne.setMttva(synthese.getMttva());
            }
        }
    }
}

```

```

        ligne.setMtttc(synthese.getMtttc());
        ligne.setNuregle(synthese.getNuregle().toString());

        totalHt = totalHt.add(synthese.getMtht());
        totalTva = totalTva.add(synthese.getMttva());
        totalTtc = totalTtc.add(synthese.getMtttc());
    }
    listSyntheseIHM.add(ligne);
}
}

```

```

// CALCUL DU SOUS TOTAL DEPENSES - FONCTIONNEMENT
ligne = new LigneSyntheseFinanciere();

```

```

    ligne.setClasse("b");
    ligne.setLibelle1("");
    ligne.setLibelle2("");
    ligne.setNuregle("");
    ligne.setLbnomregle("Total");
    ligne.setMtht(totalHt);
    ligne.setMttva(totalTva);
    ligne.setMtttc(totalTtc);
    listSyntheseIHM.add(ligne);
}

```

```

/**

```

- \* Action principale Struts.
- \* @param mapping ActionMapping
- \* @param form Formulaire de la Recherche UG
- \* @param request HttpServletRequest
- \* @param response HttpServletResponse

```

* @throws Exception Si une erreur survient.
* <p>Crée le : 17 mai 2005 par JDAHAN (version 2.06)
* <p>Modifiée le : 05 juillet 2005 par JDAHAN (version 2.06) -> Insertion des JFreeCharts
* <p>Modifiée le : 06 juillet 2005 par JDAHAN (version 2.06) -> Affichage du perimetre dans la synthese financiere
*/

```

```

public ActionForward execute(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response)
    throws Exception {

    HttpSession session = request.getSession();
    ActionErrors errors = new ActionErrors();
    ListSyntheseSelectionneeForm currentform = new ListSyntheseSelectionneeForm();
    StringUtil sUtil = new StringUtil();
    String forward = "precedent";
    String Id = null;
    ArrayList listSyntheseVO = null;
    Session ses = null;
    ListeIHM listSyntheseIHM = null;
    StFinanciereVO maVO = new StFinanciereVO();
    String label = ""; // # chaine contenant le perimetre a afficher

    // # variables utilisees dans les graphes
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();
    DefaultPieDataset pieDataset = new DefaultPieDataset();
    DefaultPieDataset pieDataset2 = new DefaultPieDataset();

    this.traceUser(session, "ListSyntheseSelectionneeAction : execute(Debut)");

    if (this.userConnected(session)) {

        // recuperation de la VO passee en session dans le CommonSynthAction.java
        maVO = (StFinanciereVO) session.getAttribute(ConstantsPatrimoine.PAOBJLIST_VO);
    }
}

```

```

// recuperation du codeExercice passe en lien dans la JSP associee a cette action
String codeExercice = request.getParameter("codeExercice");

if(codeExercice!=null) {
    maVO.setExercice(Integer.valueOf(codeExercice));
    forward = "success";
}

try {
    ses = HibernatePlugIn.open(request);
    if (ses!=null) {

        // liste contenant mes objets VO
        listSyntheseVO = (new StFinanciereManager()).getAll(ses, maVO);

        if ((listSyntheseVO!=null) && (listSyntheseVO.size(>0)) {
            this.traceUser(session, "ListSyntheseSelectionneeAction : execute()-> Synthese trouvee");
            ServletContext servletContext = getServlet().getServletContext();
            LigneSyntheseFinanciere ligne = null; // Ligne synthèse financière a afficher
            listSyntheseIHM = new ListeIHM(ConstantsPatrimoine.PAOBJLIST_AFFICHSYNTH);
            // Liste des synthèses financières a afficher
            String classe = "a";

            int nbr=listSyntheseVO.size(); // nombre d'enregistrements trouves

            // Montants intermediaires relatifs aux totaux des depenses
            totalHt = new BigDecimal(0);
            totalTva = new BigDecimal(0);
            totalTtc = new BigDecimal(0);

            BigDecimal totalGlobalHt = new BigDecimal(0);

```

```

BigDecimal totalGlobalTva = new BigDecimal(0);
BigDecimal totalGlobalTtc = new BigDecimal(0);

// #####
//
// Cette partie de code gere le calcul des differents totaux
// qui doivent apparaitre dans l'affichage de la synthese financiere
//
// #####

// TITRE PREMIER BLOC
ligne = new LigneSyntheseFinanciere();
ligne.setClasse("a");
ligne.setLibelle1("DEPENSES");
ligne.setLibelle2("Fonctionnement");
ligne.setNuregle("");
ligne.setLbnomregle("");
ligne.setMttht(null);
ligne.setMttva(null);
ligne.setMtttc(null);
listSyntheseIHM.add(ligne);

// PREMIER BLOC : DEPENSES - FONCTIONNEMENT

// Appel de la methode qui factorise l'affichage des lignes
this.initListe(listSyntheseVO, listSyntheseIHM, 0, 10);
totalGlobalHt = totalHt;
totalGlobalTva = totalTva;
totalGlobalTtc = totalTtc;

// Insertion du bloc depenses - fonctionnement JFreeChart
pieDataset.setValue("Fonct.", totalTtc);

```

```
// TITRE SECOND BLOC
totalHt=new BigDecimal(0);
totalTva=new BigDecimal(0);
totalTtc=new BigDecimal(0);
ligne = new LigneSyntheseFinanciere();
ligne.setClasse("a");
ligne.setLibelle1("DEPENSES");
ligne.setLibelle2("Investissement");
ligne.setNuregle("");
ligne.setLbnomregle("");
ligne.setMttht(null);
ligne.setMttva(null);
ligne.setMtttc(null);
listSyntheseIHM.add(ligne);

// SECOND BLOC : DEPENSES - INVESTISSEMENT

// Appel de la methode qui factorise l'affichage des lignes
this.initListe(listSyntheseVO, listSyntheseIHM, 9, 15);

// LIGNE POUR LE CALCUL TOTAL DES DEPENSES
ligne = new LigneSyntheseFinanciere();
ligne.setClasse("b");
ligne.setLibelle1("");
ligne.setLibelle2("");
ligne.setLbnomregle("TOTAL DEPENSES");
ligne.setMttht(totalGlobalHt.add(totalHt));
ligne.setMttva(totalGlobalTva.add(totalTva));
ligne.setMtttc(totalGlobalTtc.add(totalTtc));
listSyntheseIHM.add(ligne);
```

```

// Insertion du bloc depenses - investissement JFreeChart
pieDataset.setValue("Invest.", totalTtc);

// TITRE TROISIEME BLOC
totalHt=new BigDecimal(0);
totalTva=new BigDecimal(0);
totalTtc=new BigDecimal(0);
ligne = new LigneSyntheseFinanciere();
ligne.setClasse("a");
ligne.setLibelle1("RECETTES");
ligne.setLibelle2("Fonctionnement");
ligne.setNuregle("");
ligne.setLbnomregle("");
ligne.setMtht(null);
ligne.setMttva(null);
ligne.setMtttc(null);
listSyntheseIHM.add(ligne);

// TROISIEME BLOC : RECETTES - FONCTIONNEMENT

// Appel de la methode qui factorise l'affichage des lignes
this.initListe(listSyntheseVO, listSyntheseIHM, 14, 22);
totalGlobalHt = totalHt;
totalGlobalTva = totalTva;
totalGlobalTtc = totalTtc;

// Insertion du bloc recettes - fonctionnement JFreeChart
pieDataset2.setValue("Fonct.", totalTtc);

// TITRE QUATRIEME BLOC
totalHt=new BigDecimal(0);
totalTva=new BigDecimal(0);

```

```

totalTtc=new BigDecimal(0);
ligne = new LigneSyntheseFinanciere();
ligne.setClasse("a");
ligne.setLibelle1("RECETTES");
ligne.setLibelle2("Investissement");
ligne.setNuregle("");
ligne.setLbnomregle("");
ligne.setMtht(null);
ligne.setMttva(null);
ligne.setMtttc(null);
listSyntheseIHM.add(ligne);

// QUATRIEME BLOC : RECETTES - INVESTISSEMENT

// Appel de la methode qui factorise l'affichage des lignes
this.initListe(listSyntheseVO, listSyntheseIHM, 21, 23);

// LIGNE POUR LE CALCUL TOTAL DES RECETTES
ligne = new LigneSyntheseFinanciere();
ligne.setClasse("b");
ligne.setLibelle1("");
ligne.setLibelle2("");
ligne.setLbnomregle("TOTAL RECETTES");
ligne.setMtht(totalGlobalHt.add(totalHt));
ligne.setMttva(totalGlobalTva.add(totalTva));
ligne.setMtttc(totalGlobalTtc.add(totalTtc));
listSyntheseIHM.add(ligne);

// Insertion du bloc recettes - investissement JFreeChart
pieDataset2.setValue("Invest.", totalTtc);

// Initialisation des compteurs de la liste IHM

```

```

listSyntheseIHM.initBloc();

// # 06-07-05 : Gestion de l'affichage du perimetre dans la synthese financiere :
// prise en compte des regles de gestion
// # Algorithme pour afficher le type de perimetre pour l'UG
StFinanciereVO synthese = (StFinanciereVO) listSyntheseVO.get(0);
if (synthese!=null) {
    if(!(synthese.getNuequi().equals("0000000000")||synthese.getTyequi().equals("0")))
        label = "Unité de Gestion " + synthese.getNuequi() + "-" + synthese.getTyequi() + "";

    // # Fin algorithme pour afficher le type de perimetre pour l'UG
    else {
        // # Algorithme pour afficher le type de perimetre approprié
        // selon le niveau (0, 1, 2, 3) sur lequel on se trouve
        if(!synthese.getCdcoim().equals("000000")) {

            if(!synthese.getCdzone().equals("00")) {

                if(!synthese.getCdbati().equals("00"))
                    label = "Batiment " + synthese.getCdcoim() + "-" +
                        synthese.getCdzone() + "-" + synthese.getCdbati() + "";
                else
                    label = "Zone " + synthese.getCdcoim() + "-" +
                        synthese.getCdzone() + "";
            }
            else {
                label = "Complexe " + synthese.getCdcoim() + "";
            }
        }
        else {
            label = "Erreur sur la cible !!!";
        }
    }
}

```

```

    }
    // # Fin algorithme pour afficher le type de perimetre approprié
    // selon le niveau (0, 1, 2, 3) sur lequel on se trouve -->
}
}

currentform.setLabel(label);
currentform.setExercice(synthese.getExercice().toString());
currentform.setDttraitement((new DateUtil()).toString(synthese.getDttraitement()));

// Stockage de la ListeIHM dans le formulaire ListSyntheseSelectionneeForm
currentform.setListeSynthese(listSyntheseIHM);

session.setAttribute("palistsyntheseselectionneeform", currentform);

session.setAttribute(ConstantsPatrimoine.PAOBJLIST_VO, maVO);

//
#####
#
// # PARTIE DE L'APPLICATION SOUS JFREECHART
// # DEPENSES
*****
JFreeChart chart = ChartFactory.createPieChart3D("Dépenses T.T.C.", // Titre
                                                pieDataset, // Dataset
                                                true,
                                                true,
                                                false // Afficher la legende
                                                );

File file = new File(servletContext.getRealPath("/images")+"/png/PieChart.jpg");

```

```

PiePlot3D pieplot3d = (PiePlot3D)chart.getPlot();
pieplot3d.setDepthFactor(0.2);
pieplot3d.setStartAngle(290D);
pieplot3d.setDirection(Rotation.CLOCKWISE);
pieplot3d.setForegroundAlpha(0.4F);

GradientPaint gp0 = new GradientPaint(0.0f,0.0f,new Color(0xFFAE21),0.0f,0.0f,new
Color(0xFFAE21));
pieplot3d.setSectionPaint(0,gp0);

GradientPaint gp1 = new GradientPaint(0.0f,0.0f,new Color(0xBB1120),0.0f,0.0f,new
Color(0x4FDA06));
pieplot3d.setSectionPaint(1,gp1);

file.createNewFile();

ChartUtilities.saveChartAsJPEG(file, chart, 320, 200);
// # FIN DEPENSES
*****

// # RECETTES
*****

JFreeChart chart2 = ChartFactory.createPieChart3D("Recettes T.T.C.", // Titre
pieDataset2, // Dataset
true,
true,
false // Afficher la legende
);

File file2 = new File(servletContext.getRealPath("/images")+"/png/PieChart2.jpg");

PiePlot3D pieplot3d2 = (PiePlot3D)chart2.getPlot();

```

```

pieplot3d2.setDepthFactor(0.2);
pieplot3d2.setStartAngle(290D);
pieplot3d2.setDirection(Rotation.CLOCKWISE);
pieplot3d2.setForegroundAlpha(0.4F);

// # couleur de la premiere portion du PIECHART3D
GradientPaint gp2 = new GradientPaint(0.0f,0.0f,new Color(0xBAB44),0.0f,0.0f,new
Color(0xCDE21));
pieplot3d2.setSectionPaint(0,gp2);

// # couleur de la seconde portion du PIECHART3D
GradientPaint gp3 = new GradientPaint(0.0f,0.0f,new Color(0xFFFFAA),0.0f,0.0f,new
Color(0xFFCC88));
pieplot3d2.setSectionPaint(1,gp3);

file2.createNewFile();

ChartUtilities.saveChartAsJPEG(file2, chart2, 320, 200);
// # FIN RECETTES
*****

// # FIN PARTIE DE L'APPLICATION SOUS JFREECHART
//
#####
#
}

ses.disconnect();
ses=null;
}
} catch (Exception e) {
if (ses!=null) ses.disconnect();ses=null;

```

```
        e.printStackTrace();
    } finally {
    }

    if (!errors.isEmpty()) {
        saveErrors(request, errors);
        this.traceUser(session, "ListSyntheseSelectionneeAction : execute(Fin)->Erreur!");

        return mapping.getInputForward();
    }

    this.traceUser(session, "ListSyntheseSelectionneeAction : execute(Fin)->OK");
}

return (mapping.findForward(forward));
}
}
```

## Le framework de tests unitaires JUNIT - Code source d'exemple

La méthode à tester `getCalendar()` appartient à la classe `DateUtil.java` :

```
/**
 * Convertie une chaine date en GregorianCalendar.
 * @param sdate Chaine date
 * @param format Chaine de formatage
 * @return GregorianCalendar.
 * @throws ParseException Si une erreur survient.
 * <p>Crée le : 28 janv. 2004 par FMART (version 2.00)
 * <p>Modifiée le :    xx xxxx xxxx par xxxxx (version 2.xx)
 */
public GregorianCalendar getCalendar(String sdate, String format)
throws ParseException {
    if((sdate!=null)&&(sdate.length())>0)&&(format!=null)&&(format.l
    ength())>0) {
        Date date = this.toDate(sdate, format);
        GregorianCalendar calendar = new GregorianCalendar();
        calendar.setTime(date);
        return calendar;
    } else return null;
}
}
```

Test unitaire écrit pour cette méthode :

```
/*
 * Test pour GregorianCalendar getCalendar(String, String)
 * Convertie une chaine date en GregorianCalendar.
 * @param sdate Chaine date
 * @param format Chaine de formatage
 * <p>Crée le : 08 juill. 2004 par JDAHAN (version 2.00)
 * <p>Modifiée le :    xx xxxx xxxx par xxxxx (version 2.xx)
 */
public void testGetCalendarStringString() throws ParseException {
    Date dateComplete = new Date(System.currentTimeMillis());
    /* retourne la date au format JJ/MM/AAAA sans le temps */
    String maDateJour = dateFormat.format(dateComplete);
    /* retourne la date complète avec le temps initialisé à zero */
    Date dateTronquee = dl.toDate(maDateJour);
    long millisec = dateTronquee.getTime();
    /* je teste la fonction getCalendar() */
    GregorianCalendar monCalendar = dl.getCalendar(dl.toString(new
    Date()), "dd/MM/yyyy");
    /* je vérifie que les valeurs des variables sont égales */
    assertEquals(millisec, monCalendar.getTime().getTime());
}
}
```

## General Public License ou GPL

### **Introduction**

Voici (<http://www.linux-france.org/article/these/gpl.html>) une adaptation non officielle de la Licence Publique Générale du projet GNU. Elle n'a pas été publiée par la Free Software Foundation et son contenu n'a aucune portée légale car seule la version anglaise de ce document détaille le mode de distribution des logiciels sous GNU GPL. Nous espérons cependant qu'elle permettra aux francophones de mieux comprendre la GPL.

### **Licence Publique Générale GNU Version 2, Juin 1991**

Copyright © Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307

États-Unis, 1989, 1991.

La copie et la distribution de copies exactes de ce document sont autorisées, mais aucune modification n'est permise.

### **Préambule**

Les licences d'utilisation de la plupart des programmes sont définies pour limiter ou supprimer toute liberté à l'utilisateur. À l'inverse, la Licence Publique Générale (General Public License) est destinée à vous garantir la liberté de partager et de modifier les logiciels libres, et de s'assurer que ces logiciels sont effectivement accessibles à tout utilisateur.

Cette Licence Publique Générale s'applique à la plupart des programmes de la Free Software Foundation, comme à tout autre programme dont l'auteur l'aura décidé (d'autres logiciels de la FSF sont couverts pour leur part par la Licence Publique Générale pour Bibliothèques GNU (LGPL)). Vous pouvez aussi appliquer les termes de cette Licence à vos propres programmes, si vous le désirez.

*Liberté* des logiciels ne signifie pas nécessairement *gratuité*. Notre Licence est conçue pour vous assurer la liberté de distribuer des copies des programmes, gratuitement ou non, de recevoir le code source ou de pouvoir l'obtenir, de modifier les programmes ou d'en utiliser des éléments dans de nouveaux programmes libres, en sachant que vous y êtes autorisé.

Afin de garantir ces droits, nous avons dû introduire des restrictions interdisant à quiconque de vous les refuser ou de vous demander d'y renoncer. Ces restrictions vous imposent en retour certaines obligations si vous distribuez ou modifiez des copies de programmes protégés par la Licence. En d'autres termes, il vous incombera en ce cas de :

- transmettre aux destinataires tous les droits que vous possédez,
- expédier aux destinataires le code source ou bien tenir celui-ci à leur disposition,
- leur remettre cette Licence afin qu'ils prennent connaissance de leurs droits.

Nous protégeons vos droits de deux façons : d'abord par le copyright du logiciel, ensuite par la remise de cette Licence qui vous autorise légalement à copier, distribuer et/ou modifier le logiciel.

En outre, pour protéger chaque auteur ainsi que la FSF, nous affirmons solennellement que le programme concerné ne fait l'objet d'aucune garantie. Si un tiers le modifie puis le redistribue, tous ceux qui en recevront une copie doivent savoir qu'il ne s'agit pas de l'original afin qu'une copie défectueuse n'entache pas la réputation de l'auteur du logiciel.

Enfin, tout programme libre est sans cesse menacé par des dépôts de brevets. Nous souhaitons à tout prix éviter que des distributeurs puissent déposer des brevets sur les Logiciels Libres pour leur propre compte. Pour éviter cela, nous stipulons bien que tout dépôt éventuel de brevet doit accorder expressément à tout un chacun le libre usage du produit.

Les dispositions précises et les conditions de copie, de distribution et de modification de nos logiciels sont les suivantes :

### **Stipulations et conditions relatives à la copie, la distribution et la modification**

---

- Article 0

La présente Licence s'applique à tout Programme (ou autre travail) où figure une note, placée par le détenteur des droits, stipulant que ledit Programme ou travail peut être distribué selon les termes de la présente Licence. Le terme *Programme* désigne aussi bien le Programme lui-même que tout travail qui en est dérivé selon la loi, c'est-à-dire tout ouvrage reproduisant le Programme ou une partie de celui-ci, à l'identique ou bien modifié, et/ou traduit dans une autre langue (la traduction est considérée comme une modification). Chaque personne concernée par la Licence Publique Générale sera désignée par le terme *Vous*.

Les activités autres que copie, distribution et modification ne sont pas couvertes par la présente Licence et sortent de son cadre. Rien ne restreint l'utilisation du Programme et les données issues de celui-ci ne sont couvertes que si leur contenu constitue un travail basé sur le logiciel (indépendamment du fait d'avoir été réalisé en lançant le Programme). Tout dépend de ce que le Programme est censé produire.

---

- Article 1.

Vous pouvez copier et distribuer des copies conformes du code source du Programme, tel que Vous l'avez reçu, sur n'importe quel support, à condition de placer sur chaque copie un copyright approprié et une restriction de garantie, de ne pas modifier ou omettre toutes les stipulations se référant à la présente Licence et à la limitation de garantie, et de fournir avec toute copie du Programme un exemplaire de la Licence.

Vous pouvez demander une rétribution financière pour la réalisation de la copie et demeurez libre de proposer une garantie assurée par vos soins, moyennant finances.

---

- Article 2.  
Vous pouvez modifier votre copie ou vos copies du Programme ou partie de celui-ci, ou d'un travail basé sur ce Programme, et copier et distribuer ces modifications selon les termes de l'article 1, à condition de Vous conformer également aux conditions suivantes :
  - a) Ajouter aux fichiers modifiés l'indication très claire des modifications effectuées, ainsi que la date de chaque changement.
  - b) Distribuer sous les termes de la Licence Publique Générale l'ensemble de toute réalisation contenant tout ou partie du Programme, avec ou sans modifications.
  - c) Si le Programme modifié lit des commandes de manière interactive lors de son exécution, faire en sorte qu'il affiche, lors d'une invocation ordinaire, le copyright approprié en indiquant clairement la limitation de garantie (ou la garantie que Vous Vous engagez à fournir Vous-même), qu'il stipule que tout utilisateur peut librement redistribuer le Programme selon les conditions de la Licence Publique Générale GNU, et qu'il montre à tout utilisateur comment lire une copie de celle-ci (exception : si le Programme original est interactif mais n'affiche pas un tel message en temps normal, tout travail dérivé de ce Programme ne sera pas non plus contraint de l'afficher).

Toutes ces conditions s'appliquent à l'ensemble des modifications. Si des éléments identifiables de ce travail ne sont pas dérivés du Programme et peuvent être raisonnablement considérés comme indépendants, la présente Licence ne s'applique pas à ces éléments lorsque Vous les distribuez seuls. Mais, si Vous distribuez ces mêmes éléments comme partie d'un ensemble cohérent dont le reste est basé sur un Programme soumis à la Licence, ils lui sont également soumis, et la Licence s'étend ainsi à l'ensemble du produit, quel qu'en soit l'auteur.

Cet article n'a pas pour but de s'approprier ou de contester vos droits sur un travail entièrement réalisé par Vous, mais plutôt d'ouvrir droit à un contrôle de la libre distribution de tout travail dérivé ou collectif basé sur le Programme.

En outre, toute fusion d'un autre travail, non basé sur le Programme, avec le Programme (ou avec un travail dérivé de ce dernier), effectuée sur un support de stockage ou de distribution, ne fait pas tomber cet autre travail sous le contrôle de la Licence.

- 
- Article 3.  
Vous pouvez copier et distribuer le Programme (ou tout travail dérivé selon les conditions énoncées dans l'article 1) sous forme de code objet ou exécutable, selon les termes des articles 0 et 1, à condition de respecter l'une des clauses suivantes :
    - a) Fournir le code source complet du Programme, sous une forme lisible par un ordinateur et selon les termes des articles 0 et 1, sur un support habituellement utilisé pour l'échange de données ; ou,
    - b) Faire une offre écrite, valable pendant au moins trois ans, prévoyant de donner à tout tiers qui en fera la demande une copie, sous forme lisible par un ordinateur, du code source correspondant, pour un tarif n'excédant pas le coût

de la copie, selon les termes des articles 0 et 1, sur un support couramment utilisé pour l'échange de données informatiques ; ou,

- c) Informer le destinataire de l'endroit où le code source peut être obtenu (cette solution n'est recevable que dans le cas d'une distribution non commerciale, et uniquement si Vous avez reçu le Programme sous forme de code objet ou exécutable avec l'offre prévue à l'alinéa *b* ci-dessus).

Le code source d'un travail désigne la forme de cet ouvrage sous laquelle les modifications sont les plus aisées. Sont ainsi désignés la totalité du code source de tous les modules composant un Programme exécutable, de même que tout fichier de définition associé, ainsi que les scripts utilisés pour effectuer la compilation et l'installation du Programme exécutable. Toutefois, l'environnement standard de développement du système d'exploitation mis en oeuvre (source ou binaire) -- compilateurs, bibliothèques, noyau, etc. -- constitue une exception, sauf si ces éléments sont diffusés en même temps que le Programme exécutable.

Si la distribution de l'exécutable ou du code objet consiste à offrir un accès permettant de copier le Programme depuis un endroit particulier, l'offre d'un accès équivalent pour se procurer le code source au même endroit est considéré comme une distribution de ce code source, même si l'utilisateur choisit de ne pas profiter de cette offre.

---

- Article 4.

Vous ne pouvez pas copier, modifier, céder, déposer ou distribuer le Programme d'une autre manière que l'autorise la Licence Publique Générale. Toute tentative de ce type annule immédiatement vos droits d'utilisation du Programme sous cette Licence. Toutefois, les tiers ayant reçu de Vous des copies du Programme ou le droit d'utiliser ces copies continueront à bénéficier de leur droit d'utilisation tant qu'ils respecteront pleinement les conditions de la Licence.

---

- Article 5.

Ne l'ayant pas signée, Vous n'êtes pas obligé d'accepter cette Licence. Cependant, rien d'autre ne Vous autorise à modifier ou distribuer le Programme ou quelque travaux dérivés : la loi l'interdit tant que Vous n'acceptez pas les termes de cette Licence. En conséquence, en modifiant ou en distribuant le Programme (ou tout travail basé sur lui), Vous acceptez implicitement tous les termes et conditions de cette Licence.

---

- Article 6.

La diffusion d'un Programme (ou de tout travail dérivé) suppose l'envoi simultané d'une licence autorisant la copie, la distribution ou la modification du Programme, aux termes et conditions de la Licence. Vous n'avez pas le droit d'imposer de restrictions supplémentaires aux droits transmis au destinataire. Vous n'êtes pas responsable du respect de la Licence par un tiers.

- Article 7.  
Si, à la suite d'une décision de Justice, d'une plainte en contrefaçon ou pour toute autre raison (liée ou non à la contrefaçon), des conditions Vous sont imposées (que ce soit par ordonnance, accord amiable ou autre) qui se révèlent incompatibles avec les termes de la présente Licence, Vous n'êtes pas pour autant dégagé des obligations liées à celle-ci : si Vous ne pouvez concilier vos obligations légales ou autres avec les conditions de cette Licence, Vous ne devez pas distribuer le Programme.

Si une partie quelconque de cet article est invalidée ou inapplicable pour quelque raison que ce soit, le reste de l'article continue de s'appliquer et l'intégralité de l'article s'appliquera en toute autre circonstance.

Le présent article n'a pas pour but de Vous pousser à enfreindre des droits ou des dispositions légales ni en contester la validité ; son seul objectif est de protéger l'intégrité du système de distribution du Logiciel Libre. De nombreuses personnes ont généreusement contribué à la large gamme de Programmes distribuée de cette façon en toute confiance ; il appartient à chaque auteur/donateur de décider de diffuser ses Programmes selon les critères de son choix.

- 
- Article 8.  
Si la distribution et/ou l'utilisation du Programme est limitée dans certains pays par des brevets ou des droits sur des interfaces, le détenteur original des droits qui place le Programme sous la Licence Publique Générale peut ajouter explicitement une clause de limitation géographique excluant ces pays. Dans ce cas, cette clause devient une partie intégrante de la Licence.

- 
- Article 9.  
La Free Software Foundation se réserve le droit de publier périodiquement des mises à jour ou de nouvelles versions de la Licence. Rédigées dans le même esprit que la présente version, elles seront cependant susceptibles d'en modifier certains détails à mesure que de nouveaux problèmes se font jour.

Chaque version possède un numéro distinct. Si le Programme précise un numéro de version de cette Licence et « toute version ultérieure », Vous avez le choix de suivre les termes et conditions de cette version ou de toute autre version plus récente publiée par la Free Software Foundation. Si le Programme ne spécifie aucun numéro de version, Vous pouvez alors choisir l'une quelconque des versions publiées par la Free Software Foundation.

- Article 10.  
Si Vous désirez incorporer des éléments du Programme dans d'autres Programmes libres dont les conditions de distribution diffèrent, Vous devez écrire à l'auteur pour lui en demander la permission. Pour ce qui est des Programmes directement déposés par la Free Software Foundation, écrivez-nous : une exception est toujours envisageable. Notre décision sera basée sur notre volonté de préserver la liberté de notre Programme ou de ses dérivés et celle de promouvoir le partage et la réutilisation du logiciel en général.

---

## LIMITATION DE GARANTIE

---

- Article 11.  
Parce que l'utilisation de ce Programme est libre et gratuite, aucune garantie n'est fournie, comme le permet la loi. Sauf mention écrite, les détenteurs du copyright et/ou les tiers fournissent le Programme en l'état, sans aucune sorte de garantie explicite ou implicite, y compris les garanties de commercialisation ou d'adaptation dans un but particulier. Vous assumez tous les risques quant à la qualité et aux effets du Programme. Si le Programme est défectueux, Vous assumez le coût de tous les services, corrections ou réparations nécessaires.

- Article 12.  
Sauf lorsqu'explicitement prévu par la Loi ou accepté par écrit, ni le détenteur des droits, ni quiconque autorisé à modifier et/ou redistribuer le Programme comme il est permis ci-dessus ne pourra être tenu pour responsable de tout dommage direct, indirect, secondaire ou accessoire (pertes financières dues au manque à gagner, à l'interruption d'activités ou à la perte de données, etc., découlant de l'utilisation du Programme ou de l'impossibilité d'utiliser celui-ci).

---

## FIN DES TERMES ET CONDITIONS

### Comment appliquer ces directives à vos nouveaux programmes

Si vous développez un nouveau programme et désirez en faire bénéficier tout un chacun, la meilleure méthode est d'en faire un Logiciel Libre que tout le monde pourra redistribuer et modifier selon les termes de la Licence Publique Générale.

Pour cela, insérez les indications suivantes dans votre programme (il est préférable et plus sûr de les faire figurer au début de chaque fichier source ; dans tous les cas, chaque module source devra comporter au minimum la ligne de « copyright » et indiquer où résident toutes les autres indications) :

---

((une ligne pour donner le nom du programme et donner une idée de sa finalité))  
Copyright (C) 19xx ((nom de l'auteur))

Ce programme est libre, vous pouvez le redistribuer et/ou le modifier selon les termes de la Licence Publique Générale GNU publiée par la Free Software Foundation (version 2 ou bien toute autre version ultérieure choisie par vous).

Ce programme est distribué car potentiellement utile, mais SANS AUCUNE GARANTIE, ni explicite ni implicite, y compris les garanties de commercialisation ou d'adaptation dans un but spécifique. Reportez-vous à la Licence Publique Générale GNU pour plus de détails.

Vous devez avoir reçu une copie de la Licence Publique Générale GNU en même temps que ce programme ; si ce n'est pas le cas, écrivez à la Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, États-Unis.

---

Ajoutez également votre adresse électronique, le cas échéant, ainsi que votre adresse postale.

Si le programme est interactif, faites-lui afficher un court avertissement du type de celui-ci à chaque invocation :

---

...(nom du programme) version 69, Copyright (C) 19aa nom de l'auteur

...(nom du programme) est fourni sans AUCUNE GARANTIE.  
Pour plus de détails, tapez `g'.

Ce programme est libre et vous êtes encouragé à le redistribuer sous certaines conditions ; tapez `c' pour plus de détails.

---

Les commandes hypothétiques `g' et `c' doivent afficher les sections appropriées de la Licence Publique Générale GNU. Bien entendu, vous pouvez implanter ces commandes comme bon vous semble : options dans un menu, ou bien accessibles d'un clic de souris, etc., tout dépend de votre programme.

Si vous officiez en tant que programmeur, n'omettez pas de demander à votre employeur, votre établissement scolaire ou autres de signer une décharge stipulant leur renoncement aux droits qu'ils pourraient avoir sur le programme :

---

...(employeur, école...) déclare par la présente ne pas revendiquer de droits sur le programme « (nom du programme) » réalisé par ...((nom de l'auteur)).  
((signature du responsable)), ...((date)), ...((nom et qualité du responsable)).

La Licence Publique Générale ne permet pas d'inclure votre programme dans des logiciels sous licence commerciale spécifique. Si votre programme est une fonction de bibliothèque, vous jugerez probablement plus judicieux de le faire relever de la Licence Générale de Bibliothèque GNU (LGPL) plutôt que de la présente.

## SOURCES / WEBOGRAPHIE

**Lien vers le site Web du groupe ADELIOR :**

<http://www.adelior.com>

**Liens Web ayant pour thème l'Open Source :**

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

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

<http://www.linux-france.org/>

<http://solutions.journaldunet.com/dossiers/libre/sommaire.shtml>

**Sites Web des technologies présentées dans mon étude :**

	Documentation sur HTML Site de référence : <a href="http://www.w3.org/TR/html4/">http://www.w3.org/TR/html4/</a>
 NETSCAPE	Documentation sur JAVASCRIPT Site de référence : <a href="http://devedge.netscape.com/central/javascript/">http://devedge.netscape.com/central/javascript/</a>
	Documentation sur XML Site de référence : <a href="http://www.w3.org/XML/">http://www.w3.org/XML/</a>
	Documentation sur JAVA Site de référence : <a href="http://java.sun.com">http://java.sun.com</a>  <u>Autres sites :</u>  <a href="http://java.developpez.com/">http://java.developpez.com/</a> Partie consacrée à Java du site developpez.com. (FAQ, Forum, Tutoriaux).  <a href="http://penserinjava.free.fr/">http://penserinjava.free.fr/</a> Site du groupe de traduction du livre « <i>Thinking in Java</i> », le livre écrit en anglais par Bruce Eckel.

 <p><b>JAVASERVER PAGES™</b> DYNAMICALLY GENERATED WEB CONTENT</p>	<p><i>Documentation sur les JSP</i> Site de référence : <a href="http://java.sun.com/products/jsp/">http://java.sun.com/products/jsp/</a></p>
	<p><i>Documentation sur ORACLE</i> Site de référence : <a href="http://www.oracle.com/">http://www.oracle.com/</a></p>
	<p><i>Documentation sur CVS</i> Site de référence : <a href="http://www.cvshome.org">http://www.cvshome.org</a></p> <p><u>Autres sites :</u></p> <p><a href="http://www.cvsnt.org/wiki/">http://www.cvsnt.org/wiki/</a> Serveur CVS fonctionnant sous Windows. (Téléchargement, Documentation).</p> <p><a href="http://www.wincvs.org/">http://www.wincvs.org/</a> Une interface graphique (GUI) pour CVS sous Windows. (Téléchargement).</p>
	<p><i>Documentation sur Eclipse</i> Site de référence : <a href="http://www.eclipse.org">http://www.eclipse.org</a></p> <p><u>Autres sites :</u></p> <p><a href="http://www.eclipsetotale.com/index.html">http://www.eclipsetotale.com/index.html</a> Le site francophone consacré au projet Eclipse et aux outils WebSphere Studio d'IBM. (FAQ, Forum, Tutoriaux).</p> <p><a href="http://dev.eclipse.org:8080/help/help.jsp">http://dev.eclipse.org:8080/help/help.jsp</a> L'aide en ligne officielle de l'IDE Eclipse.</p> <p><a href="http://eclipse-plugins.2y.net/eclipse/plugins.jsp">http://eclipse- plugins.2y.net/eclipse/plugins.jsp</a> Site de téléchargement de plugins pour Eclipse, permettant d'ajouter des fonctionnalités à l'IDE.</p>
	<p><i>Documentation sur Tomcat</i> Site de référence : <a href="http://jakarta.apache.org/tomcat/index.html">http://jakarta.apache.org/tomcat/index.html</a></p>

 The JavaBeans mascot is a green bean-shaped character with a white body, arms, and legs. It has several small brown beans on its head and is holding a large green oval with the word "JavaBeans" written on it in white.	<p><i>Documentation sur les JavaBeans</i> Site de référence : <a href="http://java.sun.com/products/javabeans/">http://java.sun.com/products/javabeans/</a></p>
---	---

 The Sun Microsystems logo consists of a stylized sun icon to the left of the word "Sun" in a bold, sans-serif font, with "microsystems" in a smaller font below it.	<p><i>Documentation sur JDBC</i> Site de référence : <a href="http://java.sun.com/products/jdbc/">http://java.sun.com/products/jdbc/</a></p>
---	--

## GLOSSAIRE

Définitions sur les technologies utilisées dans le cadre du déploiement de GIMA WEB :



**Framework** : Infrastructure logicielle qui facilite la conception des applications par l'utilisation de bibliothèques de classes ou de générateurs de programmes, soit dit en quelques mots : un cadre de développement.



**Hibernate** : Framework qui permet de gérer l'accès aux données.



**JSP** : Le langage JSP (Java Server Page) associe l'HTML et le JAVA; c'est une description de la page avec des tags Struts qui permettent de l'enrichir.

**J2EE** : Environnement JAVA pour le développement d'applications Entreprise.

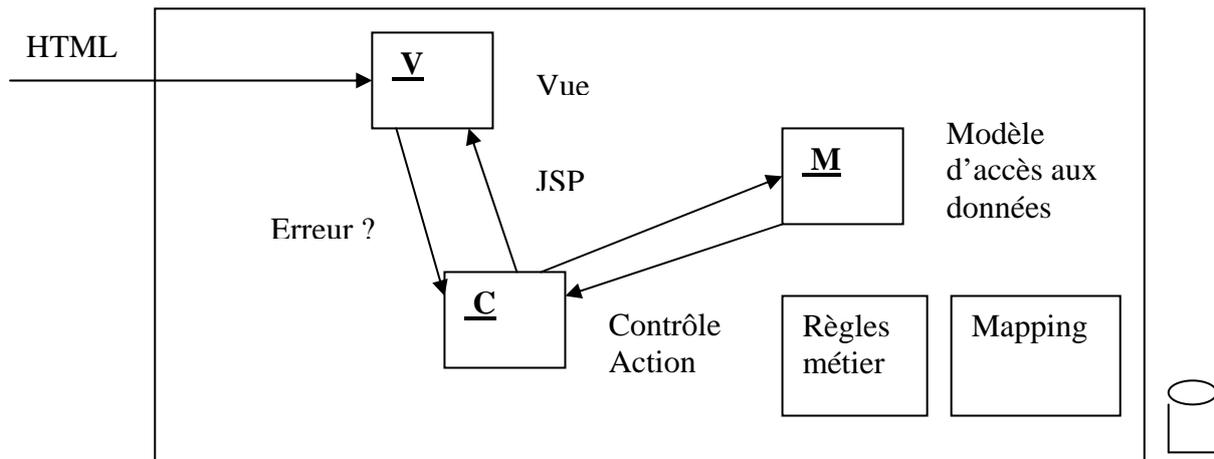
**Javadoc** : Permet au développeur de générer la documentation technique en utilisant certains mots - clé.



**Patrimoine (Observatoire du)** : Module qui permet de visualiser globalement les dépenses et les recettes gérées dans GIMA Client/Serveur.

**Patterns** : Ce sont des modélisations de développement, c'est à dire des fonctionnalités-type (ex: organigramme) Les patterns permettent de découper le développement et de spécialiser les acteurs du projet. Ces comportements standard correspondent à 90% des applications des entreprises.

Exemple de pattern : le modèle MVC (ou Modèle - Vue - Contrôleur)



Développeurs premier niveau

Règles de gestion métier

**Progiciel** : Logiciel paramétrable en fonction des besoins du client.



**Recettage** : Déroulement de jeux de tests sur une application afin de recenser puis corriger les cas pouvant générer des exceptions.



**Struts** : Framework permettant de gérer les informations JAVA avec les tags. STRUTS permet entre autre de faire l'affectation entre JAVA et le formulaire HTML et permet donc de simplifier la cinématique de l'application et l'interface.



**Tierce Maintenance Applicative** : Maintenance effectuée par une entité qui n'est ni le fournisseur ni le propriétaire de l'équipement ou du logiciel.

**Tuxedo** : Moniteur transactionnel utilisé dans les architectures Client/Serveur de 2<sup>ème</sup> génération, il gère la connexion entre les demandes des clients et optimise les requêtes à la base de données, compression de données, permet une diminution du temps de réponse ainsi que de dispatcher les requêtes. Ce module est nécessaire lorsque il y a beaucoup de clients qui se connectent à une base de données (>500 utilisateurs).

**Tomcat** : Serveur d'application Open Source. TOMCAT est utilisé pour exécuter l'application développée.