

*Cnam*

**CONSERVATOIRE NATIONAL DES ARTS & MÉTIERS**  
**PARIS**

**MÉMOIRE**  
présenté en vue d'obtenir le  
**DIPLÔME D'INGÉNIEUR C.N.A.M.**

Spécialité : INFORMATIQUE  
Option : MULTIMÉDIA

par  
**JEAN FRÉOR**  
Septembre 2010

**CONCEPTION ET OPTIMISATION**  
**D'UN INTRANET**  
**POUR LE GROUPE LA POSTE**

**JURY**

PRÉSIDENT : Isabelle WATTIAU (Professeur, C.N.A.M.)

MEMBRES : Jacky AKOKA (Professeur, C.N.A.M.)

Tatiana AUBONNET (Maître de Conférences, C.N.A.M.)

Alain MICHEZ (Directeur, LA POSTE)

Virginie MEFFRE (Chef de projet, LA POSTE)

LA POSTE



## SOMMAIRE

<b>Remerciements .....</b>	<b>6</b>
<b>Introduction.....</b>	<b>7</b>
<b>Chapitre 1. Saturne, une brique pour le « Cap Qualité Courrier » .....</b>	<b>9</b>
1.1 LA POSTE, face à une concurrence plus qu'imminente.....	10
1.1.1 La direction du Courrier, une branche vitale pour le groupe LA POSTE.....	10
1.1.2 LA POSTE et ses principaux concurrents européens .....	13
1.1.3 Le tri du courrier .....	14
1.1.4 Le Cap Qualité Courrier.....	17
1.1.5 La DECLIC, cellule des développements informatiques .....	20
1.2 Le SI à LA POSTE .....	21
1.2.1 Le cycle de vie d'un projet informatique à LA POSTE.....	21
1.2.2 Les acteurs du SI.....	23
1.2.3 L'environnement réseau.....	23
1.2.4 L'environnement technologique général.....	25
1.2.5 Les impératifs liés à la modernisation de LA POSTE .....	25
1.3 GALAXIE, la grande la famille de l'application SATURNE .....	26
1.3.1 La famille GALAXIE .....	26
1.3.2 Planning du projet SATURNE.....	28
1.4 Cycle des projets : la théorie.....	29
1.4.1 Cycle de vie du projet .....	29
1.4.2 Cycles de développement du projet .....	30
1.4.3 Récapitulatif sur les livrables .....	33
1.5 Démarche appliquée à SATURNE : de la théorie à la pratique.....	34
1.5.1 J'ai rédigé le DCG d'une seule traite pour les deux lots avant de commencer le développement du lot 1 .....	34
1.5.2 Le DCD a été rédigé après le développement, et non avant.....	34
1.5.3 Je n'attends pas que soit validée la DFB avant de rédiger le DCG, ni que soit validé le DCG avant de commencer les développements. ....	35
1.5.4 La MOA refuse de valider le DCG, jugeant ce type de document « trop technique ». ....	35
1.5.5 J'annonce les délais avant la rédaction du DCG, et non après.....	36
1.5.6 Je livre souvent une version « beta » avant la date convenue. ....	37
1.5.7 Je n'ai pas reçu en entrée de Document d'Architecture Logicielle.....	37
1.5.8 Les réunions ne sont pas hebdomadaires. ....	38
1.6 L'équipe.....	38
1.6.1 Schéma d'ensemble .....	38
1.6.2 Mes niveaux d'interventions.....	40

<b>Chapitre 2. L'architecture de l'application Saturne.....</b>	<b>41</b>
2.1 Découpage en modules .....	42
2.2 Bases théoriques d'une architecture client-serveur multi-tiers.....	42
2.2.1 Les différents modèles client-serveur .....	42
2.2.2 Un modèle pour SATURNE ?.....	44
2.2.3 Description d'une architecture multi-tiers .....	44
2.2.4 Avantages et inconvénients d'une architecture multi-tiers .....	45
2.3 Schéma d'architecture technique de SATURNE .....	46
2.4 Perspective logicielle de l'architecture de SATURNE .....	50
2.5 Architecture fonctionnelle de SATURNE .....	51
2.5.1 Brique de consolidation des données .....	51
2.5.2 Brique d'agrégation des données .....	51
2.5.3 Brique de restitution aux utilisateurs .....	52
2.6 Les fruits du multi-tiers dans SATURNE .....	53
2.6.1 Zéro déploiement .....	53
2.6.2 Le web et les langages interprétés.....	53
2.6.3 Développement d'un composant métier.....	53
2.7 Inconvénients de l'architecture de SATURNE .....	55
2.7.1 Consommation mémoire .....	55
2.7.2 Consommation réseau .....	55
2.8 L'architecture de SATURNE : entre théorie et pratique... ..	57
2.8.1 Limite du découpage théorique de l'application .....	57
2.8.2 Limites de la problématique de déport des traitements SGBD .....	58
2.8.3 Limites du déport des traitements métier .....	58
<b>Chapitre 3. Saturne face aux nouveaux besoins de montée en charge .....</b>	<b>59</b>
3.1 Cadrage de notre étude d'optimisation.....	60
3.1.1 Évolution de la cible de performance.....	60
3.1.2 Baseline et nouvelles exigences.....	60
3.2 Démarche adoptée pour notre chantier d'optimisations .....	66
3.2.1 Les TMC (Tests de montée en charge) .....	67
3.2.2 La revue de code .....	67
3.2.3 Repenser l'architecture actuelle de SATURNE .....	67
3.3 Bilan de scalabilité : les TMC .....	69
3.3.1 Résultats des TMC sur SATURNE.....	69
3.3.2 L'utilité limitée des TMC... ..	70
3.4 Revue de code.....	71
3.5 Préconisations déjà respectées avant l'optimisation de SATURNE.....	72

3.6	Analyse des faiblesses du système existant.....	73
3.7	Scénarios d'amélioration de la scalabilité dans SATURNE.....	73
3.7.1	Évolution de l'architecture fonctionnelle.....	73
3.7.2	Évolution de l'architecture logicielle.....	79
3.7.3	Évolution de l'architecture technique logique.....	82
3.7.4	Évolution de l'architecture technique physique, ou « scaleup/scaleout ».....	85
3.8	Prototypage d'un composant de service pour les échanges de données entre SNT et SATURNE.....	87
3.8.1	Fonctionnement actuel du système d'échange de données.....	87
3.8.2	Refonte du système d'échange de données.....	88
3.8.3	Principe de fonctionnement.....	88
3.8.4	Comment planifier la sollicitation périodique des services SNT ?.....	89
3.8.5	Problème annexe.....	89
3.8.6	Comment distinguer les données non encore intégrées des autres données ?.....	90
3.8.7	Test du prototype.....	91
3.8.8	Bilan sur le prototype.....	91
3.9	Optimisations sortant du domaine de la montée en charge.....	92
3.9.1	Blindage de la partie SGBD.....	92
3.9.2	Blindage de la partie IHM.....	93
3.9.3	Consolidation de la gestion des erreurs.....	93
3.9.4	Mise en correspondance avec la revue de code.....	95
3.9.5	Des leçons à tirer pour le développement des projets ?.....	95
3.9.6	Des plateformes de tests comme solution complémentaire.....	96
3.9.7	Corriger les aberrations à la source.....	97
3.10	Résultat des tests d'optimisations.....	98
<b>Chapitre 4. Conjuguer performance et qualité de l'interface.....</b>		<b>100</b>
4.1	L'existant : un peu d'histoire.....	101
4.2	L'IHM de SATURNE, ou la proposition d'une meilleure interface.....	102
4.2.1	Constat de l'existant.....	102
4.2.2	Amélioration progressive de l'ergonomie générale.....	102
4.3	Le DHTML, pour servir ou desservir ?.....	105
4.3.1	DHTML : un savant mélange de HTML, CSS et Javascript.....	105
4.3.2	Le DHTML, une aide précieuse pour enrichir l'interface de SATURNE.....	107
4.4	Beau et performant : une utopie ?.....	116
4.4.1	La charte graphique, une base de l'ergonomie.....	117
4.4.2	Les fichiers inclus, le réflexe du bon développeur.....	117
4.4.3	L'importance du paramétrage du cache.....	118
4.4.4	Procédés d'optimisation des objets DHTML.....	119
4.4.5	Pendant ce temps-là, que fait le serveur ?.....	121
4.4.6	Conclusion sur la problématique ergonomique.....	122
4.5	Beau et performant, ça ne suffit pas.....	122

4.5.1	Facilité de maintenance et réutilisabilité.....	122
4.5.2	Aide en ligne.....	123
4.6	Et le « web 2.0 » ? .....	126
4.6.1	Qu'est-ce que le « web 2.0 » ?.....	126
4.6.2	SATURNE est-il un intranet 2.0 ? .....	127
4.6.3	Vous avez dit Ajax ?.....	129
4.6.4	Le « web 2.0 » est-il techniquement une révolution ? .....	130
<b>Chapitre 5. Une méthode comme UWE aurait-elle été utile ? .....</b>		<b>132</b>
5.1	Méthode de conception actuellement employée dans SATURNE .....	132
5.1.1	Modélisation de l'architecture .....	132
5.1.2	Modélisation des données .....	132
5.1.3	Modélisation des besoins fonctionnels .....	133
5.2	Mise en pratique de la méthode UWE avec le projet SATURNE .....	133
5.2.1	Qu'est-ce qu'UWE ?.....	133
5.2.2	Essais de modélisation du projet SATURNE en UWE .....	136
5.3	Méthode UWE vs démarche empirique.....	144
5.3.1	Avantages de la méthode UWE .....	144
5.3.2	Limites générales de la méthode UWE.....	144
5.3.3	Limites de la méthode UWE dans son application à SATURNE.....	146
5.3.4	Bilan sur la question de l'adaptation d'UWE à SATURNE.....	148
5.3.5	Alors, quid de la méthode actuellement employée ?.....	149
<b>Conclusion .....</b>		<b>150</b>
<b>Glossaire.....</b>		<b>155</b>
<b>Index.....</b>		<b>158</b>
<b>Références .....</b>		<b>159</b>
<b>Table des figures .....</b>		<b>160</b>
<b>Liste des tableaux.....</b>		<b>163</b>

## REMERCIEMENTS

Je remercie en premier lieu LA POSTE, pour laquelle j'ai été prestataire durant plusieurs années, d'avoir accordé son assentiment à mon sujet de mémoire. Cela a été possible grâce à l'intercession d'Alain MICHEZ, ancien responsable à LA POSTE du projet GALAXIE<sup>1</sup>, membre de mon jury de soutenance, et lui-même ingénieur C.N.A.M.

Ma gratitude envers LA POSTE ne se limite pas au cadre de mon mémoire C.N.A.M. En effet, je suis reconnaissant de la confiance que m'ont manifestée mes chefs de projets successifs au cours de cette mission : Frédéric COURAULT, Virginie MEFFRE (un grand merci à elle, ainsi qu'à Alain, d'avoir bien voulu faire partie des membres du Jury), et enfin Loïc BRUGGEMAN. Cette confiance m'a permis de travailler en autonomie et d'exploiter mon potentiel d'une manière qui s'est révélée satisfaisante pour LA POSTE et moi-même.

Je remercie également mon tuteur de mémoire, Jacky AKOKA<sup>2</sup>, pour s'être porté garant de mon projet jusqu'à la soutenance.

Enfin, je remercie ma famille de m'avoir encouragé sur les sentiers du C.N.A.M., ce qui m'a permis de surmonter les contraintes liées aux cours du soir et à la rédaction de ce mémoire. Celles-ci seront vite oubliées car, arriver à la fin de ce parcours m'offre une satisfaction fondée : je ne doute pas que ce cheminement m'assurera une meilleure reconnaissance professionnelle suite à l'obtention du diplôme d'ingénieur C.N.A.M., autant qu'il m'aura permis de consolider mes acquis.

---

<sup>1</sup> Jusqu'au 31 décembre 2007, Alain MICHEZ était responsable du domaine « DDP2 » à LA POSTE SI, et entre autre du vaste projet « Galaxie » (cf. 1.3.1 « La famille GALAXIE, page 26). Alain dirige à présent le service « Urbanisation & Méthodes ».

<sup>2</sup> Jacky AKOKA préside la chair « Informatique d'entreprise » au C.N.A.M. des Arts et Métiers de Paris.

## INTRODUCTION

Ce mémoire marque le point final de ma formation au C.N.A.M. des Arts & Métiers de Paris, où j'ai été auditeur depuis octobre 2003. Mon objectif est de compléter par un diplôme d'ingénieur C.N.A.M. mon profil « IUP MIAGE »<sup>3</sup>, en valorisant le travail que j'ai produit pour la branche Courrier de LA POSTE en tant que prestataire de service<sup>4</sup>. Il s'agit d'une mission à longue durée, qui a débuté en août 2002 pour s'achever en janvier 2009.

Je souhaite présenter ici le contexte, ainsi que certaines problématiques du projet qui m'a été confié en tant qu'analyste programmeur tout au long de cette mission. En l'occurrence, ce mémoire a pour but de démontrer mes capacités à être ingénieur. Tel est le cas quand on peut à la fois comprendre l'architecture d'un système informatique, analyser une problématique industrielle, faire des choix de conception et d'optimisation adaptés aux contraintes de performances, et savoir les argumenter. Aussi, dans le cadre de cette mission professionnelle, j'ai conçu, développé puis optimisé une application intranet<sup>5</sup> à haute volumétrie appelée SATURNE : cette dernière permet à plusieurs milliers d'utilisateurs de LA POSTE d'accéder quotidiennement aux données de production du tri postal. La réalisation de SATURNE s'est inscrite dans un projet plus vaste mettant directement en jeu l'avenir de LA POSTE : le Cap Qualité Courrier<sup>6</sup>.

Le travail qui m'a été confié sur mon lieu de mission présente un triple intérêt pour une personne prétendant au titre d'ingénieur. En premier lieu, ce projet se distingue par sa richesse, qu'il s'agisse des technologies utilisées (SGBD, web...) ou de son cycle de développement (conception, implémentation, puis livraison). En deuxième lieu, l'intérêt du projet vient de l'autonomie dont j'ai pu bénéficier. Celle-ci m'a permis d'effectuer ou de proposer des choix, aussi bien fonctionnels que techniques, puis de mettre en œuvre les solutions retenues et, enfin, optimiser l'existant. Le troisième et dernier intérêt de mon

---

<sup>3</sup> Suite à un DUT informatique (niveau bac+2), j'ai effectué à Nantes une maîtrise MIAGE (« Méthodes informatiques appliquées à la gestion des entreprises »), formation de niveau bac+4, avec comme spécialité, l'Ingénierie des systèmes d'information. Par ailleurs, la rédaction et la soutenance d'un mémoire facultatif m'a permis d'accoler à ma maîtrise MIAGE le « titre d'ingénieur maître ».

<sup>4</sup> Mon entreprise, Apside, est une SSII de la région parisienne, et me comptait parmi ses mille salariés depuis mai 2001.

<sup>5</sup> Un site web est dit « intranet » quand son accès se limite au réseau local de l'entreprise. Par ailleurs, on préfère pour SATURNE parler d'« application web » plutôt que de « site web », car SATURNE est un véritable logiciel, si ce n'est que son support est celui du web (HTML, Javascript...). En effet, les informations présentées par SATURNE ne sont pas statiques, mais dynamiques : celles-ci proviennent d'une base de données et découlent de l'interactivité qui s'opère entre l'utilisateur et les pages web qu'il manipule.

<sup>6</sup> Cf. §1.1.4 « Le Cap Qualité Courrier », page 17.

projet à LA POSTE tient à l'expérience d'encadrement et de gestion de projet qui m'a été demandée depuis que notre prestation de service a été forfaitisée vers un contrat de type TMA<sup>7</sup>.

Bien que la mission soit terminée aujourd'hui, je me permettrai d'écrire au présent dans les différentes parties de ce mémoire, qui sont les suivantes. La première partie s'attachera à la présentation du contexte, précisément celui du projet SATURNE : LA POSTE, le CQC, l'existant, le cycle de développement des projets, l'équipe dont je fais partie. Dans un deuxième temps sera présentée et analysée l'architecture, à la fois technique et logicielle, de l'application SATURNE. Quant à la troisième partie, elle sera consacrée à la problématique de montée en charge de l'application : il sera dressé une analyse des faiblesses de l'existant, ainsi qu'une élaboration des perspectives par couche d'architecture pour l'optimisation des performances et de la capacité à tenir la charge. Puis une quatrième partie, étroitement liée à la précédente, mettra en lumière une problématique intéressante : Les impératifs de performance d'une application cliente comme l'intranet SATURNE amènent souvent à sacrifier la qualité de l'interface graphique, or je n'ai pas voulu que ce soit le cas pour SATURNE. On peut ainsi se demander ce que doit être la teneur de ce compromis, ou encore les solutions permettant de limiter ou de contourner l'impact de la couche graphique sur les performances globales du système.

Enfin, une cinquième et dernière partie présentera un travail proposé et coordonné par Jacky AKOKA, que j'ai réalisé une fois ma mission à LA POSTE terminée. Ce travail consiste à revisiter une partie des spécifications de SATURNE en expérimentant UWE (méthode web exploitant UML). Cette partie du mémoire s'ouvrira par une analyse sur la question du bien fondé d'une telle méthode pour SATURNE.

---

<sup>7</sup> TMA, pour 'Tierce maintenance applicative' (Apside et LA POSTE sont liées par un contrat de TMA depuis mai 2007).

## Chapitre 1. SATURNE, UNE BRIQUE POUR LE « CAP QUALITÉ COURRIER »

L'enjeu de mon projet à LA POSTE revêt une importance certaine : il participe directement à la préparation de la branche Courrier à une concurrence imminente. En effet, le transport du courrier de moins de 50 grammes, dernier monopole de l'opérateur public, est sur le point de s'ouvrir à la concurrence européenne<sup>8</sup>. Cette perspective pousse naturellement LA POSTE à être compétitive et donc à moderniser sa branche consacrée au transport du courrier.

Une partie de cette modernisation est visible par le grand public, comme par exemple l'utilisation par les facteurs de nouveaux vélos à moteur électrique. Mais dans la partie cachée de l'iceberg se joue un vaste projet appelé « CQC » (*Cap Qualité Courrier*), auquel participe le développement de SATURNE. Le *Cap Qualité Courrier* vise pour l'horizon 2011 un objectif de 90% de courriers parvenus à leurs destinataires le lendemain de leur envoi. Ce niveau est atteint depuis longtemps par les concurrents allemands et néerlandais ; en 2008, le taux d'arrivée en « J+1 » était encore de 82,7% en France.

Ce chapitre s'attachera à présenter le contexte de mon projet : d'abord LA POSTE, ensuite le CQC, et enfin mon projet professionnel et son contexte, à savoir l'application SATURNE ainsi que le système dont elle fait partie. Il sera présenté le SI de LA POSTE ainsi que le cycle de développement en vigueur, et en particulier celui de SATURNE.

*Nota Bene :*

*Tout au long de ce chapitre, des vignettes bleues fléchées avec un libellé "je suis ici" ou encore "j'interviens ici" agrémentent les différents schémas d'ensemble, afin de mieux mettre en évidence ma localisation et mon rôle dans les différentes vues du contexte de mon projet à LA POSTE.*



<sup>8</sup> Ayant initialement été prévue pour le 1<sup>er</sup> janvier 2009, puis le 1<sup>er</sup> janvier 2010, l'arrivée de la concurrence est actuellement prévue pour courant 2011.

## 1.1 LA POSTE, face à une concurrence plus qu'imminente

Créée en 1576, LA POSTE est la plus ancienne entreprise de distribution de courrier au monde. Avec ses filiales, elle constitue le deuxième opérateur postal européen.

Devenue depuis le 1er mars 2010 une société anonyme, LA POSTE reste néanmoins 100% publique, car détenue en totalité par l'État et la Caisse des Dépôts et Consignations.

Principal opérateur postal en France, elle est d'abord constituée de la section « Courrier et Colis » : chaque jour, dans les bureaux de Poste, 3,5 millions de clients sont accueillis et 75 millions d'objets y sont traités.

Mais LA POSTE, c'est aussi le service grand public, avec la « Banque Postale »<sup>9</sup> : près de 30 millions de clients utilisent les services de finance et d'assurance.

Tout le monde le sait, notre premier service public français sera bientôt soumis à une difficile concurrence : LA POSTE se prépare depuis des années à cette situation inéluctable.

### 1.1.1 La direction du Courrier, une branche vitale pour le groupe LA POSTE

Le groupe *LA POSTE* se compose comme suit :



Figure 1 – Organisation du groupe LA POSTE

<sup>9</sup> Suite au vote de la loi européenne de régulation des activités postales, à compter du 1er janvier 2006, LA POSTE peut proposer à ses clients, par l'intermédiaire de « La Banque postale », créée à cet effet, l'ensemble des services bancaires : en particulier, proposer une grande partie de la panoplie des crédits, notamment immobiliers. (Les crédits à la consommation ne sont pas encore autorisés à ce jour.)

La direction pour laquelle je travaille est celle du Courrier. Cette dernière se décompose elle-même en plusieurs directions informatiques, qui sont les suivantes :



Je suis ici

- La DECLIC, ou « Direction de l'Étude et de la Conception de Logiciel Informatique du Courrier »<sup>10</sup>, pour laquelle je travaille, s'occupe de la mise en place du Système d'Information de LA POSTE (description au §1.1.5, en page 20) ;
- La DCI, ou « Direction du Courrier International »<sup>11</sup> gère, comme son nom l'indique, les courriers internationaux ;
- La DTC<sup>12</sup>, ou « Direction Technique du Courrier ». Elle s'occupe du déploiement et de la maintenance du matériel (des postes de travail aux machines de tri) ;
- Le SNTP, ou « Service National des Timbre-poste et de la Philatélie », gère l'édition et la distribution des timbres de collections.

#### 1.1.1.1 Combien de plis, envoyés par qui, et pour qui ?

Avec près de 100 millions de documents acheminés et distribués quotidiennement, le pôle Courrier occupe la 2ème place en Europe sur ce marché.

Nos envois de cartes postales sont bien peu de choses, puisque 90 % des expéditeurs sont des professionnels (entreprises, administrations, associations...).

2/3 des courriers sont destinés aux ménages.

#### 1.1.1.2 Parlons argent

Les statistiques financières officielles [COUR WEB] montrent que le chiffre d'affaire de la branche Courrier de LA POSTE a suivi une augmentation relativement régulière, comme le montre l'histogramme ci-dessous :

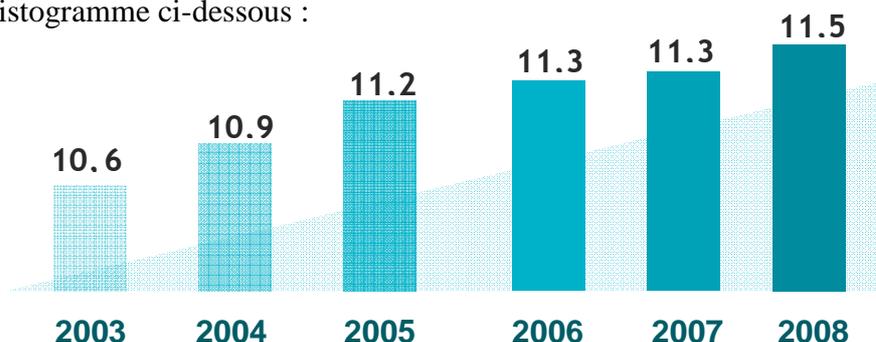


Figure 2 – Évolution du chiffre d'affaires du Courrier dans le groupe (en millions d'euros)

<sup>10</sup> Anciennement appelée Direction des Systèmes d'Informations et Informatique du Courrier (DSII-C)

<sup>11</sup> Désormais appelée "LA POSTE Courrier International"

<sup>12</sup> La DTC était anciennement divisée en deux branches : la DOMI (« Direction de l'Organisation de la Maintenance Industrielle ») et le SRTP (« Service de Recherche Technique de LA POSTE »).

Le chiffre d'affaire du Groupe LA POSTE, quant à lui, est de 20,8 milliards d'euros. La figure ci-dessous montre la répartition de ce chiffre d'affaires dans les 3 activités du groupe :

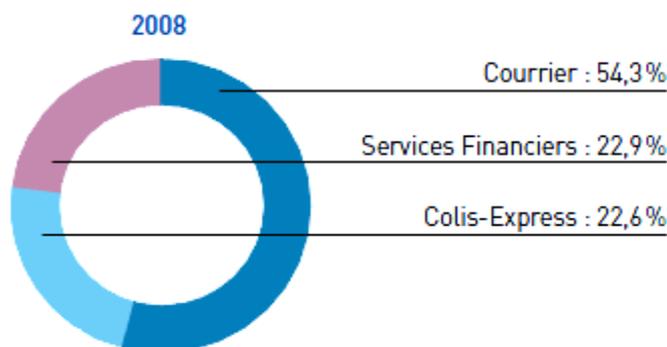


Figure 3 – Répartition du chiffre d'affaires du groupe LA POSTE

Les chiffres des ressources humaines font mesurer l'importance de la branche Courrier :

- 191.000 personnes travaillent au groupe Courrier : 174 000 à LA POSTE et 17 000 dans les filiales ;
- 100.000 d'entre elles sont des facteurs, dont 45,5 % de femmes ;
- 33.000 postiers<sup>13</sup> travaillent en centre de traitement du courrier, dont 10 % sont chargé de l'encadrement ;
- 1.000 chargés de comptes conseillent les entreprises ;
- 320 télévendeurs conseillent les clients à distance, sur 8 plateaux télévente installés en France ;
- 7.400 recrutements ont été effectués en 2006.

#### 1.1.1.3 La rentabilité de la branche Courrier

La marge dégagée par le Courrier<sup>14</sup> est encore inférieure à celle de ses concurrents étrangers<sup>15</sup>. Pourtant, il est essentiel de la développer : c'est ce qui permet à LA POSTE de financer sa modernisation et son indispensable croissance, en France comme à l'étranger, pour espérer passer un jour numéro 1 en Europe.

---

<sup>13</sup> Ce terme ne se restreint pas aux facteurs : on appelle « postier » toute personne employée par LA POSTE, tout comme on appelle « cheminot » tout employé de la SNCF.

<sup>14</sup> 6,8% de marge en 2006, contre 5,6% en 2005

<sup>15</sup> Le résultat d'exploitation du Courrier a progressé chaque année jusqu'en 2006 : il s'élève à 765 millions d'euros, alors que les 663 millions d'euros de 2005 constituaient déjà un progrès. La marge d'exploitation (résultat d'exploitation rapporté au chiffre d'affaires) s'est améliorée elle aussi et a atteint 6,8% (après des résultats de 5,7 en 2004 puis de 5,9% en 2005).

### 1.1.2 LA POSTE et ses principaux concurrents européens

En 2006, l'auteur Nicolas LECAUSSIN a produit un rapport intéressant, baptisé « *Deutsche Post et LA POSTE* ».

En France, le J+1 était de l'ordre de 75%. En Angleterre, il était de 93,4% avec la Royal Mail (soit le meilleur d'Europe). Enfin, pour la Deutsche Post allemande, il était de 90%.

Toutefois, il est à noter que le J+1 ne concerne que la lettre rapide (tarif normal), or de nombreuses entreprises préfèrent aujourd'hui adresser leur courrier au tarif économique... qui de fait, n'est pas soumis au J+1.

Le tableau ci-dessous, extrait du rapport de Nicolas LECAUSSIN, compare la Deutsche Post allemande et LA POSTE française :

	<b>Deutsche Post</b>	<b>LA POSTE</b>
<b>Effectifs moyens</b>	388 859	309 286
<b>Chiffre d'affaires/salarié</b>	124 156 euros	61 131 euros
<i>En milliards d'euros :</i>		
<b>Chiffre d'affaires</b>	48,279	19,329
<b>Résultat net</b>	2,448	0,377
<b>Capitaux propres</b>	12,540	4,204
<b>Dettes à long terme</b>	12,241	9,266
<b>Part du courrier dans le CA</b>	20 %	50 %
<b>Centres de tri</b>	80	130
<b>Qualité (J+1)</b>	90 %	75 %

*Tableau 1 – Deutsche Post versus LA POSTE*

D'après ce tableau, la Deutsche Post génère un chiffre d'affaires par salarié deux fois plus élevé qu'en France, trois fois plus de capitaux propres, un résultat net deux fois plus élevé, avec moins de centres de tri et moins de personnel rapporté au chiffre d'affaire...

Laissons-nous impressionner, mais avec modération : il est bon ici de rappeler qu'en Allemagne, le niveau de vie est lui aussi bien plus élevé qu'en France !

### 1.1.3 Le tri du courrier

Le cœur de métier de LA POSTE est le tri postal, et il en est de même pour les données que traite l'application SATURNE : aussi, il est important de comprendre les différentes étapes du principe d'acheminement du courrier depuis la boîte aux lettres de LA POSTE jusqu'à la notre.

#### 1.1.3.1 Cycle d'acheminement et de distribution du courrier

Comme le montre le schéma ci-dessous, le courrier collecté par LA POSTE est d'abord rassemblé au niveau des centres de distribution (CDIS), puis là commence son acheminement : il est envoyé en centre de tri (CTC) et le courrier circule d'un CTC à l'autre dans le circuit interdépartemental. La chaîne d'acheminement aboutit dans tous les cas au niveau d'un CDIS : là, sa distribution peut débuter.

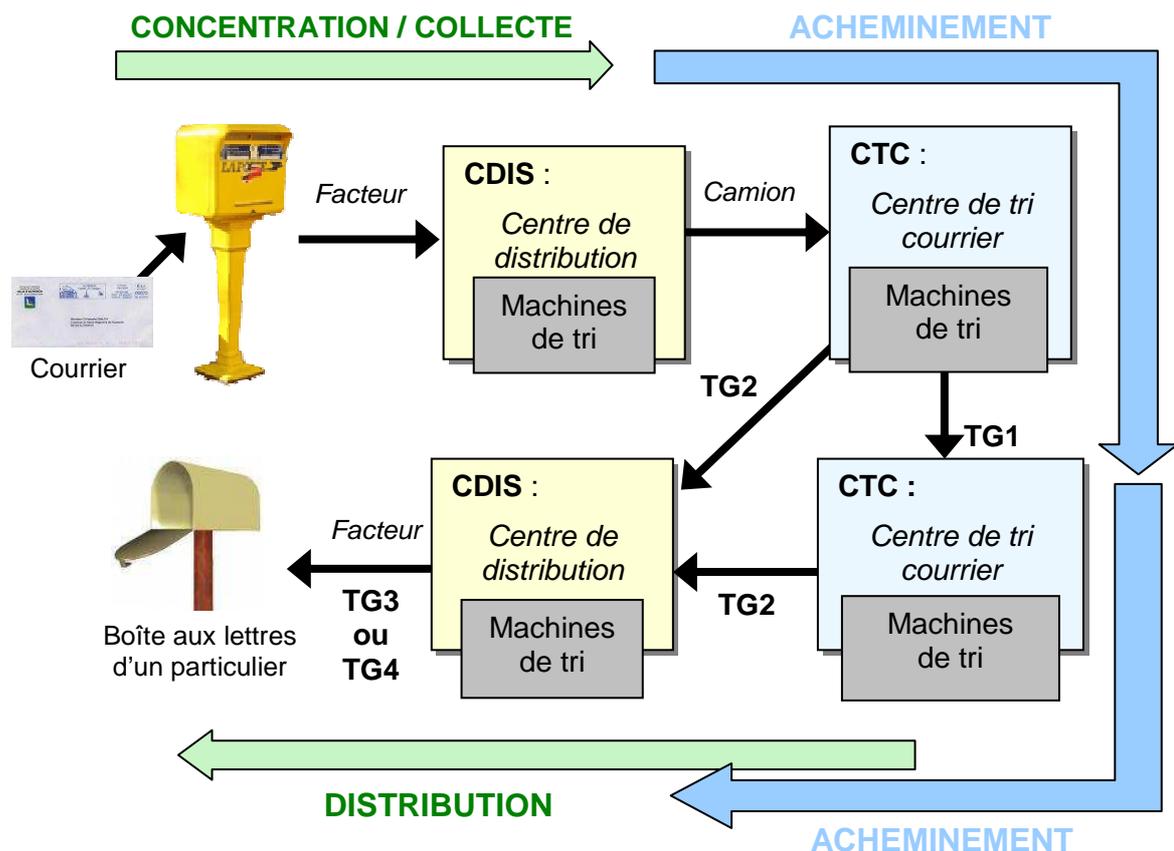


Figure 4 – Cycle d'acheminement et de distribution du courrier



- « À retrier » : Un lot de courrier est qualifié « à retrier » lorsque, à l'issue d'un premier passage en machine, les plis doivent repasser dans une autre machine, avec un programme de tri différent ;
- Tri Définitif : Les plis sont dits "triés définitivement" lorsqu'ils ne sont plus retriés dans le centre, c'est à dire qu'ils ne sont pas repris sur une autre machine ou sur un autre chantier de ce centre.

Plusieurs visites en centre de tri (*Yvelines CTC* et *Melun CTC*) m'ont permis de mieux saisir toute la dimension industrielle qui se cache derrière SATURNE, l'application dont j'ai eu la charge et qui a pour vocation de fournir quotidiennement à des milliers d'utilisateurs les statistiques du jour en termes de tri courrier.

À cette occasion j'ai pris quelques photos dont celles ci-dessous, qui mettent respectivement en évidence les tasseurs<sup>20</sup> de la machine de tri, puis une unité de reconnaissance d'adresse<sup>21</sup> (*URA*).



Figure 6 – Vues intérieure et extérieure d'une machine de tri

---

<sup>20</sup> Réceptacle se trouvant au début du circuit de la machine sur les machines TPF (Tri Petit Format). Un tasseur contient environ 400 lettres (contre 80 pour une case). Le courrier y est présenté sur le champ. Un tasseur correspond à une séparation productive et permet des sorties groupées avant affinage du tri.

<sup>21</sup> La reconnaissance automatique d'adresse se fait par le biais d'un lecteur optique.

## 1.1.4 Le Cap Qualité Courrier

### 1.1.4.1 Présentation du CQC

L'objectif que LA POSTE se fixe pour 2011 est le suivant : 90% de « J+1 » pour l'envoi d'une lettre. Cet objectif a un nom : « Cap Qualité Courrier », ou autrement dit, « CQC ».

Atteindre cet objectif revient à rejoindre les meilleurs standards européens. Pour ce faire, la stratégie principale de LA POSTE est la suivante :

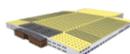
- Adapter son organisation, grâce à la construction de :



**PDC** : Plates-formes de Distribution du Courrier (plusieurs milliers) ;



**PPDC** : Plates-formes de Préparation et de Distribution du Courrier (plusieurs centaines) ;



**PIC** : Plates-formes Industrielles du Courrier (plusieurs dizaines).

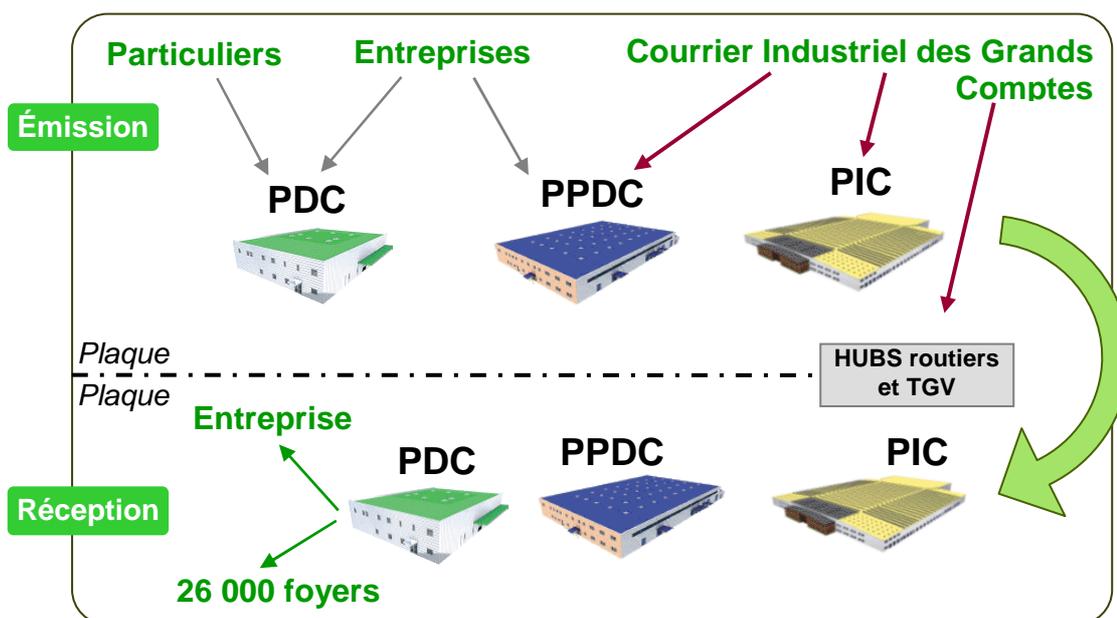


Figure 7 – Le process industriel du Cap Qualité Courrier

- Moderniser son système d'information ;
- Mettre la DECLIC (cf. page 11) au cœur de ces profonds changements ;
- Favoriser le recours à la sous-traitance (prestation de service en régie, ou en TMA). Exemple : en janvier 2006, Bull reprend l'exploitation des serveurs de production. Exemple encore plus récent : GALAXIE, l'ensemble d'applications dont fait partie SATURNE a été développé en grande majorité par des prestataires de service (dont

moi-même). Dans les derniers mois de ma mission (début 2009), on ne comptait quasiment plus aucun développement réalisé en interne : tout est confié aux quelques grosses SSII référencées par LA POSTE (Cap Gemini, SOPRA, etc) ;

- La mise en œuvre de nouveaux outils pour les nouveaux projets : on peut citer l'utilisation d'Oracle à la place de Sybase, ou encore l'adoption des technologies J2EE (surtout à St Quentin en Yvelines... mais pas pour SATURNE) ou .NET (surtout à Montpellier) ;
- Une évolution perpétuelle du périmètre des applications. On pense particulièrement aux nouvelles machines de tri (notamment les MTI – Machine de tri industriel – Cf. représentation ci-après) qui impliquent des changements dans les flux de données.

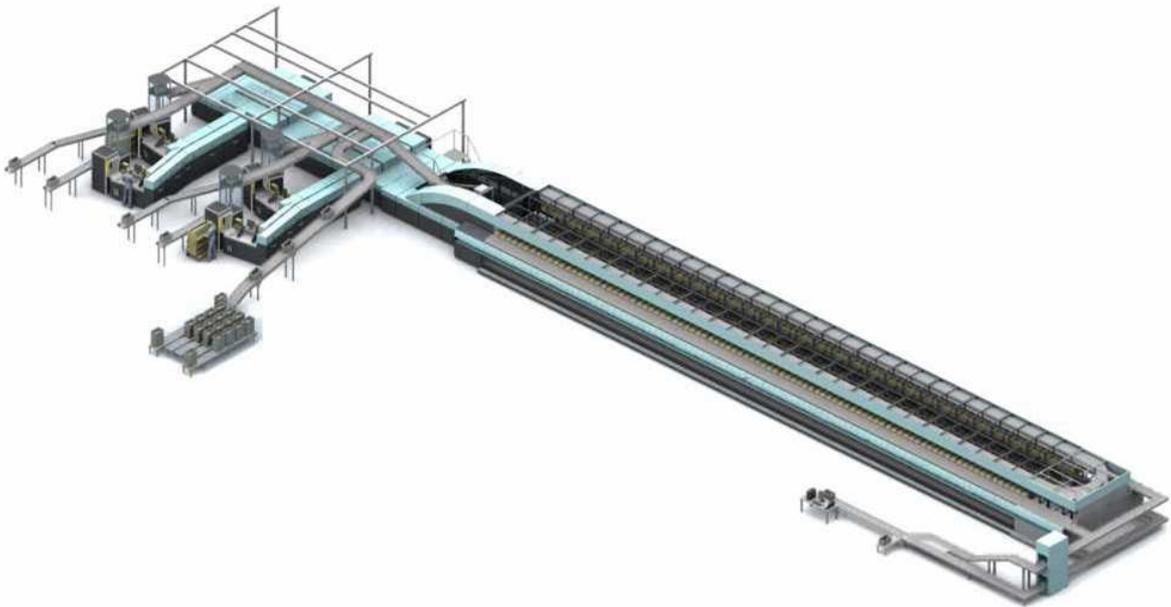


Figure 8 – Machine de tri industriel Grand Format (MTI GF)

La branche courrier de LA POSTE vit probablement en ce moment la plus grande transformation de son histoire : 3,4 milliards d'euros sont dédiés à cette modernisation. Le projet « Cap Qualité Courrier » s'opère à l'instant même, en mettant en œuvre quatre types de révolution pour être le numéro 1 du courrier en Europe en 2011. Elles sont les suivantes.

#### 1.1.4.2 Révolution industrielle

LA POSTE prétend, au travers du CQC, se munir de l'outil industriel le plus moderne d'Europe. En effet, plus d'un milliard d'euros sont investis dans la modernisation industrielle :

- 860 millions d'euros pour 650 machines de traitement, les plus modernes au monde ;

- 150 millions d'euros pour le « système d'interprétation de l'enveloppe », un système de lecture optique révolutionnaire capable de lire toutes les informations figurant sur la face du pli ;
- 110 millions d'euros pour équiper la plupart des plates-formes industrielles courrier de « transitiques » : des tapis roulants intelligents pour un premier tri automatisé des bacs de courrier.

#### *1.1.4.3 Révolution commerciale*

30 000 clients ont été interrogés sur leur perception et leurs attentes vis-à-vis du programme Cap Qualité Courrier. LA POSTE française est la seule à s'être positionnée sur l'ensemble de la chaîne de valeur du courrier. Elle peut ainsi proposer à la fois des solutions globales pour les entreprises, et un service Courrier simple et accessible pour les particuliers, et dans l'idéal, à un haut niveau de qualité.

#### *1.1.4.4 Révolution sociale*

800 millions d'euros sont destinés aux nouveaux métiers, aux nouvelles qualifications, à la formation et à l'amélioration des parcours professionnels et des conditions de vie au travail des postiers.

#### *1.1.4.5 Révolution de la distribution*

Le CQC implique aussi de nouveaux métiers, de nouveaux services aux clients (exemple : envoi de courrier via internet), de nouveaux matériels (exemple : vélos électriques, machines de préparation automatique de la tournée du facteur), ainsi qu'une amélioration des conditions de travail : les 100.000 facteurs de LA POSTE sont au cœur du CQC, car ils sont considérés comme de véritables ambassadeurs du Courrier et des intermédiaires de confiance.



### 1.1.5 La DECLIC, cellule des développements informatiques

La DECLIC (Direction de l'étude et la conception de logiciel informatique du courrier) est une direction à compétence nationale de LA POSTE et rattachée à la direction du Courrier. En 2006, La DECLIC représentait un effectif de 456 personnes, dont 207 contractuels et 249 fonctionnaires, réparti entre les sites de Saint Quentin en Yvelines (224 personnes) et de Montpellier (232 personnes).

À la fin de l'année 2003, la DECLIC a été mandatée par la Direction du Courrier pour impulser une démarche de modernisation du système d'information du courrier associant les maîtrises d'ouvrage (MOA), les maîtrises d'œuvre (MOE) et les prescripteurs métiers. Un total de 173 applications informatiques est développé et maintenu par la DECLIC.

La DECLIC intervient dans quatre domaines complémentaires :

- Les systèmes d'information proprement dits, où il s'agit de garantir un SI cohérent, évolutif et adapté à la stratégie du Courrier ;
- L'organisation du processus informatique, dont vont dépendre la qualité du service rendu, la rapidité de mise en œuvre des besoins au moindre coût et le bon fonctionnement des systèmes existants ;
- Le pilotage des projets d'évolution, ce qui inclut les outils, processus et comportements qui permettent d'atteindre les objectifs de transformation dans des conditions optimales de qualité, de coût et de délai ;
- La technologie elle-même, c'est-à-dire l'infrastructure et les services liés qui vont permettre aux utilisateurs de travailler dans de meilleures conditions avec des outils adaptés et d'accéder aux applications définies par le maître d'œuvre.

La DECLIC est composée de 10 services, qui sont les suivants :

- La Direction du Système d'Information du Courrier ;
- La Direction de l'Urbanisme et de l'Architecture Fonctionnelle ;
- La Direction de l'Organisation et de la Planification, la Direction Technique ;
- La Direction de la Veille et Innovations Technologiques ;
- La Direction du Domaine « Processus Transverse » ;
- La Direction du Domaine « Commercial » ;
- **La Direction du Domaine « Production » ;**
- La Direction de la Qualité et des Services aux Clients ;
- La Direction des Ressources.

Je dépends de la DDP : *Direction du Domaine « Production »*.

Ses missions sont les suivantes :

- La gestion des référentiels de la distribution<sup>22</sup>. Ces référentiels décrivent l'organisation des tournées de distribution du « Courrier Ménage ».
- La gestion du référentiel « Adresse ». Les données d'acheminement sont complétées par :
  - les informations sur le niveau de tri (cf. § 1.1.3.1 « Cycle d'acheminement et de distribution du courrier », page 14)
  - les techniques de codification des adresses pour l'interprétation d'adresse (IA).
- La gestion des plans de tri. Les données d'organisation postales sont traduites en « séparations » (encore appelées « plan de tri »). Puis elles sont transmises aux différents automates des centres de production (machine de tri, avec URA, vidéocodage...) par le logiciel IGP.
- Le pilotage de l'activité en CTC. Les machines de tri fournissent des données chiffrées sur le trafic écoulé par direction de tri. Ces données permettent un pilotage opérationnel ou décisionnel de l'activité.
- Le pilotage de la distribution. Des comptages descriptifs de la distribution au regard des ressources engagées permettent le pilotage opérationnel de l'activité.
- Le pilotage du transport. Ce projet répond à une problématique de maîtrise des flux de transports postaux. Ces applications sont basées sur une acquisition des données sous forme de « code à barres ». Elles sont restituées aux acteurs pour assurer un pilotage destiné à améliorer la qualité de service de l'acheminement du courrier.

## 1.2 Le SI à LA POSTE

### 1.2.1 Le cycle de vie d'un projet informatique à LA POSTE

Tout projet informatique qui est conduit à LA POSTE suit le cycle de vie indiqué dans le schéma qui suit.

---

<sup>22</sup> Base de données regroupant l'ensemble des tournées de facteur.

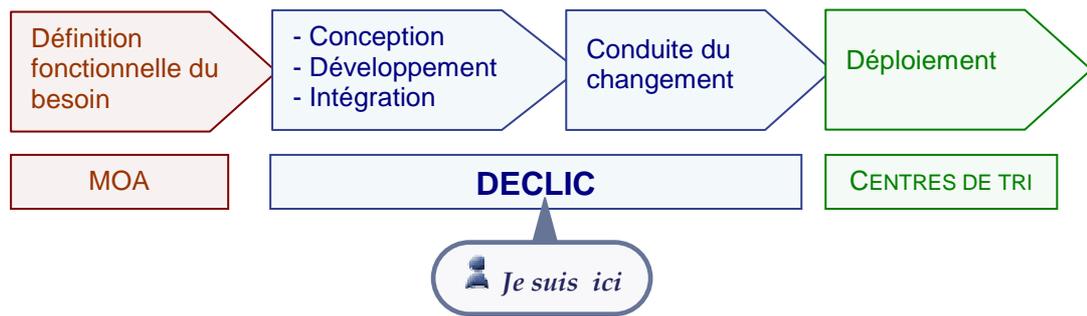


Figure 9 – Cycle de vie d'un projet à LA POSTE : les acteurs

La DECLIC s'appuie, pour l'ensemble de ses projets (chaque direction peut avoir des phases différentes), sur la Méthode de Conduite de Projets Système d'Information (MCPSI) de LA POSTE. Pour tout nouveau projet, les phases officielles à respecter sont les suivantes :

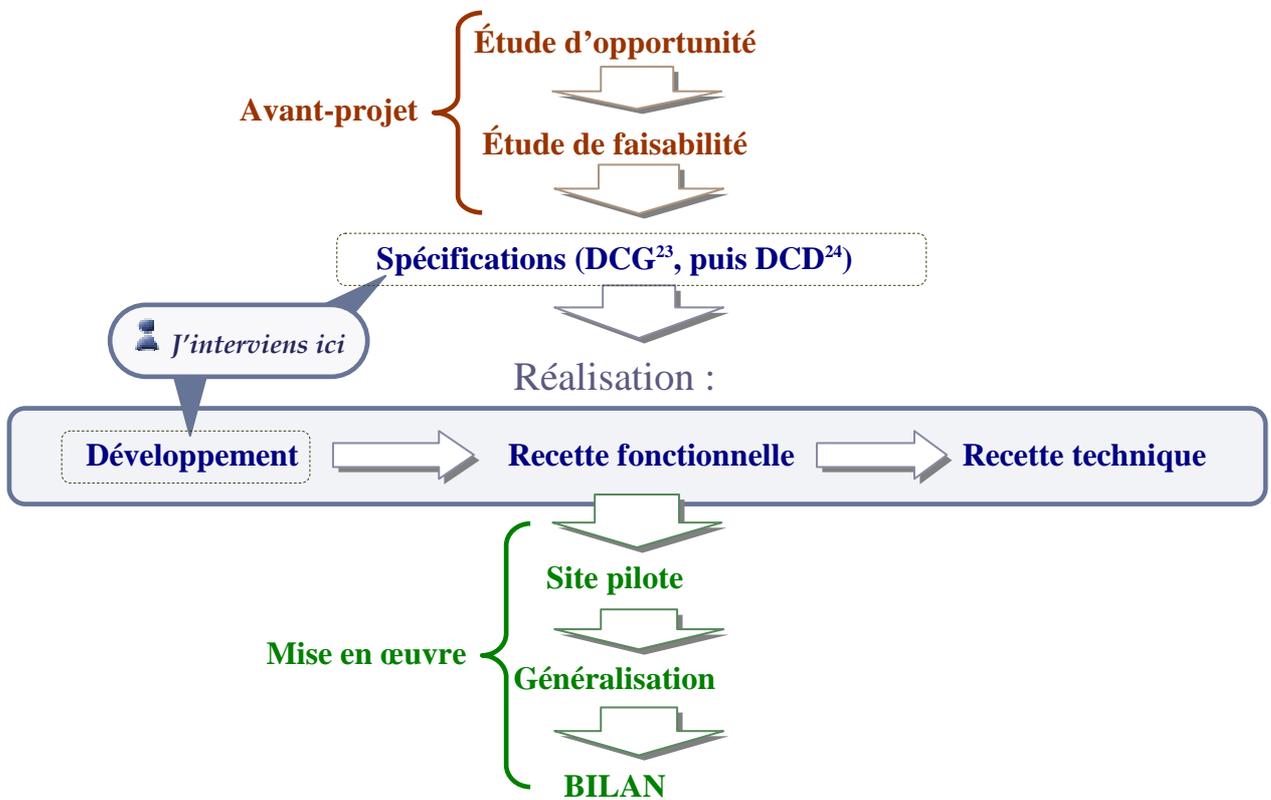


Figure 10 – Planification des phases de développement à LA POSTE

Bien sûr, ce cycle devient itératif avec les versions successives des applications : voir §1.4.1 « Cycle de vie du projet » (page 29).

<sup>23</sup> Dossier de conception générale

<sup>24</sup> Dossier de conception détaillée

La DECLIC s'est engagée dans un projet de rénovation de son environnement de conception pour promouvoir l'utilisation d'UML dans ses nouveaux projets. Toutefois, jusqu'à maintenant, cette réforme ne s'est pas encore imposée dans mon équipe<sup>25</sup>.

### 1.2.2 Les acteurs du SI

Les MOA (maîtres d'ouvrage) dont la mission est, dans le cadre de leurs activités de responsables de processus (gestion de l'organisation et des procédures), de définir leurs besoins en matière de système d'information, de réaliser les recettes fonctionnelles des applications qui leur sont livrées et de maîtriser la conduite du changement.

La MOA joue le rôle de client, et garantit l'exécution quotidienne des processus et leurs évolutions. Il existe deux types de processus :

- Les processus de production qui sont au centre des métiers ;
- Les processus de pilotage qui permettent de gérer la chaîne de valeur des processus de production.

La MOE (maîtrise d'œuvre) a pour mission d'assurer la conception informatique, le développement, l'intégration, l'exploitation et le support. La MOE joue le rôle de fournisseur.

Les GSI (gestionnaires du système d'information) a pour rôle de concevoir les architectures des domaines, d'assurer la cohérence fonctionnelle, applicative et technique de leur domaine, d'assurer la veille technologique et de piloter les projets.

La MOA et la MOE ont des obligations réciproques d'information et de production de livrables à respecter à des échéances précises du projet.

### 1.2.3 L'environnement réseau

Dans un souci d'économie et de rationalisation des différentes plateformes (développement, intégration, recette, production...), il a été décidé au cours de l'année 2008 de les consolider en utilisant la virtualisation<sup>26</sup>, et de les centraliser dans un même lieu, à St Ouen (région parisienne).

---

<sup>25</sup> Le manque d'UML à LA POSTE a suscité un travail expérimental et complémentaire à ma mission, exposé dans le Chapitre 5 (page 123 de ce mémoire).

<sup>26</sup> La virtualisation de serveurs consiste à en héberger plusieurs sur une même machine. Cette solution apporte d'importants avantages, à savoir la réduction de l'infrastructure technique, des contraintes d'administration, des coûts (exploitation, licences serveurs achats de matériels). D'autres avantages majeurs de la virtualisation sont la rapidité de mise en œuvre, l'extension facile du parc informatique ou encore la facilité d'administration (en un clic, il est possible de déplacer plusieurs serveurs virtuels sur une autre machine physique).

Les deux sites de développement de Montpellier et de Saint Quentin en Yvelines ont donc été reliés récemment au site de consolidation, par une ligne à 50 Mbps<sup>27</sup>. Aussi, les postes de développement sont reliés aux serveurs par une ligne à moyen débit en WAN<sup>28</sup>, et ne sont plus reliés aux serveurs sur un réseau local.

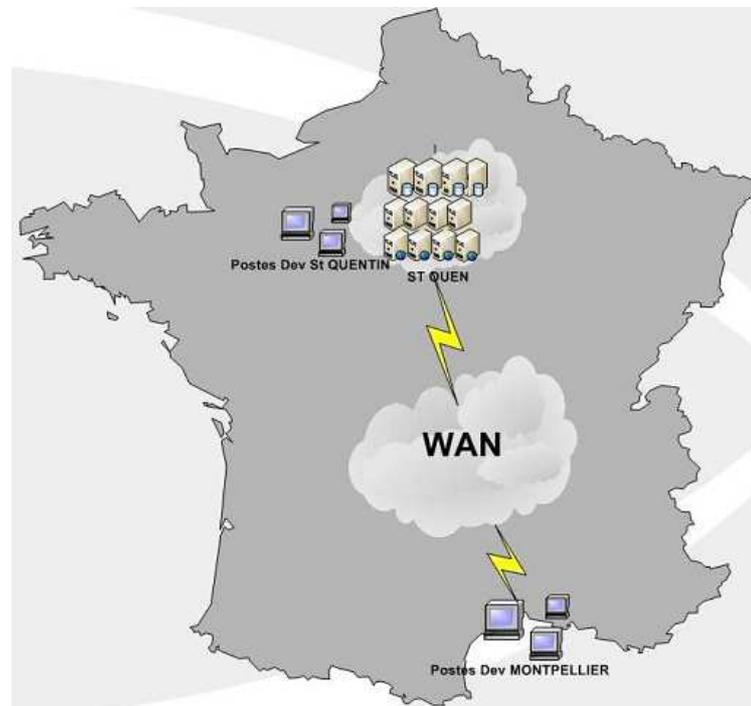


Figure 11 – Typologie du réseau informatique national à la DECLIC

Le risque de contention est plus important sur un réseau à débit moins élevé, et avec une plus grande distance entre les postes de développement et les serveurs consolidés. Ainsi, ce changement de typologie en faveur de la virtualisation était sensé entraîner des effets indésirables dans nos tâches de développeurs au quotidien : temps de réponses plus élevés, corruptions potentielles des données qui transitent sur le réseau, impossibilité d'envoyer de gros volumes de données, coupures intempestives.

Dans la réalité, ces cassandres ne se sont pas abattues sur nous... bien au contraire, j'ai même pu constater des temps de réponses moins élevés qu'avant !

Conclusion : virtualiser, c'est bien.

---

<sup>27</sup> Mbps = Megabits par seconde. Pour avoir l'équivalent en Mo/s (megaoctets par seconde), il suffit de diviser la valeur par 8. Aussi, 50 Mbps = 6,25 Mo/s.

<sup>28</sup> WAN, pour « Wide Area Network », désigne un réseau informatique couvrant une grande zone géographique, par contraste avec le LAN (Local Area Network).

#### 1.2.4 L'environnement technologique général

L'environnement technologique du SI le cas du projet SATURNE est composé de :

- Sybase 12.0 comme SGBD ;
- IBM WTX comme outil d'EAI ;
- Microsoft MSMQ comme file de messages ;
- IIS 5 comme serveur web ;
- Power AMC Designer pour modéliser les données ;
- Visual Interdev 6.0 pour le développement des fichiers web
- Visual Basic 6.0 pour le développement des composants métier
- Visual SourceSafe pour le contrôle de code source ;
- Terminal server pour le paramétrage et installation des serveurs à distance.

En réalité, l'environnement technologique du SI LA POSTE est considérablement plus étendu. Pour les nouveaux projets, on peut en effet exploiter la version .NET de Visual Studio, voire même du J2EE, ainsi qu'Oracle en termes de SGBD.

Pour tout problème d'ordre technique, nous faisons appel à la DT (Direction Technique) en utilisant une application intranet de gestion de nos diverses demandes (« Ges'Dem »).

La gestion documentaire est assurée par l'outil « Qualipso » (produit Livelink), et est accessible, elle aussi, via la page d'accueil de l'intranet postal ; mais elle s'intègre tel un plugin dans toutes nos applications office (Word, Excel, PowerPoint) afin de faciliter la publication de nos documents de projet. Cependant, notre équipe ne fait l'objet d'aucune sollicitation pour faire vivre la rubrique documentaire propre au projet SATURNE : aussi, je n'utilise jamais Qualipso en tant que publicateur, mais pour des recherches ponctuelles<sup>29</sup>.

#### 1.2.5 Les impératifs liés à la modernisation de LA POSTE

Nous ressentons bien la pression que met le CQC sur la qualité des applications :

- Celles-ci sont (et seront encore) beaucoup plus sollicitées (extension du parc des machines de tri et de la population utilisateurs)
- Les enjeux du CQC (directement liés au caractère compétitif dont devront faire preuve les services de LA POSTE en 2011) deviennent naturellement les enjeux des applications informatiques postales : elles n'auront plus droit à l'erreur<sup>30</sup>.

---

<sup>29</sup> Nous pouvons être amenés à rechercher certaines normes de développement, une description de socles ou tout simplement consulter un organigramme ou le trombinoscope.

<sup>30</sup> SATURNE non plus n'a pas droit à l'erreur, d'autant que les statistiques qu'il délivre conditionnent de très près la facturation des clients : Les gros clients de LA POSTE (que l'on appelle dans SATURNE les « usagers »),

## 1.3 GALAXIE, la grande la famille de l'application SATURNE

### 1.3.1 La famille GALAXIE

La Direction du Domaine « Production » (cf. page 20) est chargée de la mise en place d'un projet nommé « GALAXIE ». Ce projet est crucial pour atteindre l'objectif de la distribution du courrier des ménages à 'J+1'... celui-ci même que l'on appelle le « Cap Qualité Courrier », ou encore « CQC ». Aussi, le projet GALAXIE consiste en une refonte de l'informatique du tri. Cette refonte des outils informatiques implique les sept applications suivantes :

#### ■ OMEGA

OMEGA est le système de codification, de personnalisation et de diffusion des données référentielles d'adresses et d'organisation de la distribution vers les systèmes de traitement automatisé du courrier (interprétation d'adresses machines de tri en CTC et centre courrier, outil de plan de tri)

#### ■ HUBBLE

Paramétrage centralisé et commun des machines, avec la description des équipements installés sur site : machines, URA et périphériques.

#### ■ GENESE

GENESE est l'outil d'intégration des consignes nationales, régionales et locales, au travers des règles d'acheminement, pour constituer les plans de séparations correspondant aux contrats entre établissements du réseau.

#### ■ SPOT

Outil de conception des plans de tri (remplace GIPSI). SPOT donne une vision globale du chantier de tri mécanisé. Il permet des optimisations des ressources machines, l'anticipation des impacts d'évolutions matérielles et organisationnelles, au travers de simulations et de l'intégration de la volumétrie prévisionnelle des flux. Il facilite l'entraide.

#### ■ IGP

IGP est l'interface entre le système d'informations et les machines/équipements.

IGP trace tous les événements du tri automatique. Il reçoit :

- Les programmes de tri (il les transfère au x machines) ;
- Les fichiers de sessions de tri ;

---

se voient attribuer des remises tarifaires non négligeables si le courrier qu'ils distribuent fait l'objet d'un fort taux de reconnaissance optique de l'adresse.

- Les données machines (il les transfère à SNT).

## ■ SNT

SNT collecte les données de production, les centralise (il n'y a donc plus d'administration locale des anciennes bases GIPSI) puis les présente de façon homogène (via un « catalogue de données ») aux applications de pilotage du traitement.

## ■ SATURNE

*C'est mon projet*

SATURNE est l'unique point d'observation de toutes les données machines relatives au tri du courrier (compteurs, événements, ...). Il remplace le module statistique des anciennes applications GIPSI et GIPSI STATS.

SATURNE est doté d'une interface graphique riche, et permet entre autres de visualiser les données de production au format Excel ainsi que sous différentes formes graphiques (chronogrammes, histogrammes, ...).

Le schéma ci-dessous consiste en une vue macroscopique de l'acheminement des données de production depuis leur envoi par les machines de tri jusqu'à leur consultation en établissement :

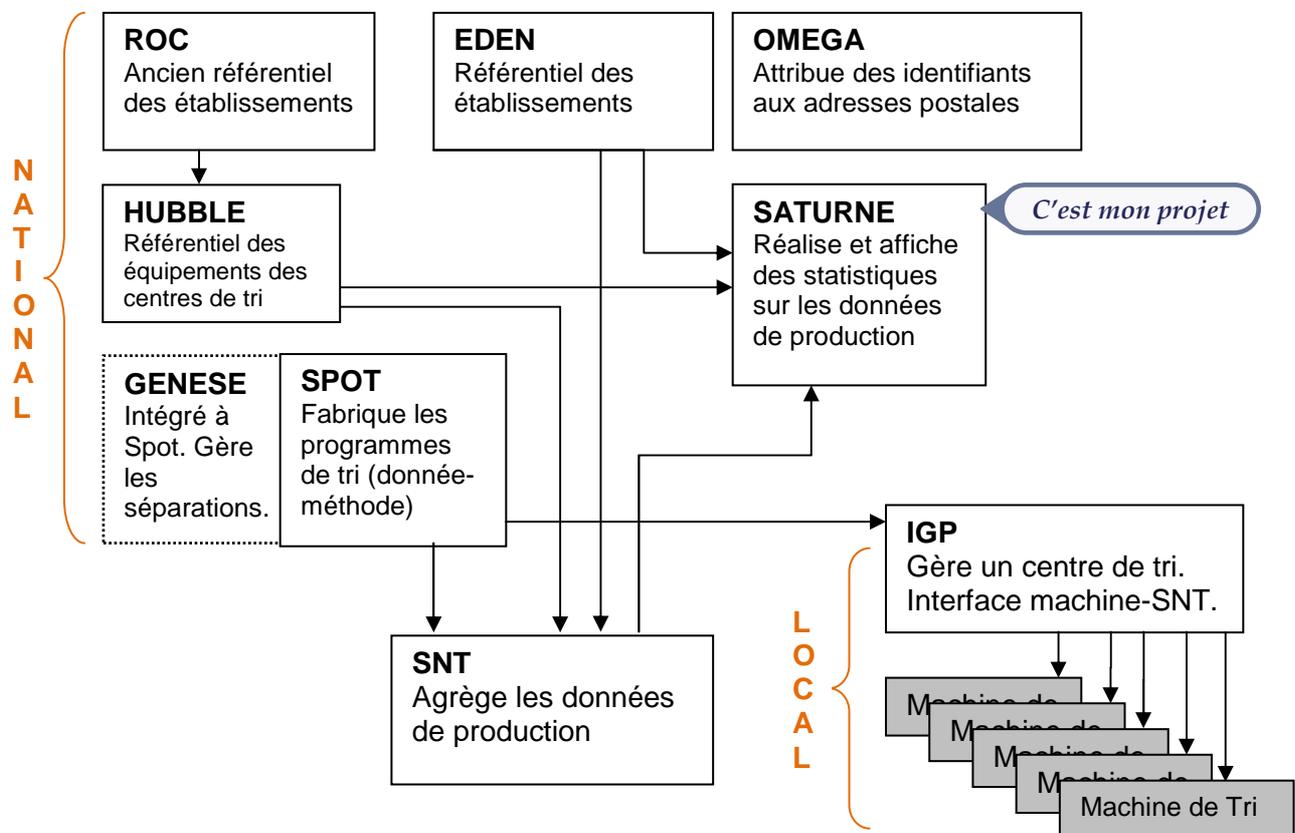


Figure 12 – Imbrications des différentes applications du projet GALAXIE

La mise en place de GALAXIE s'est faite en trois étapes :

**Étape 1** : Mise en œuvre de SNT V1, qui collecte les données de production et les adresses aux applications de pilotage. Cette migration doit être totalement transparente pour les CTC.

Cette étape s'est achevée en 2002.

**Étape 2** : Essentiellement un projet de refonte de l'infrastructure informatique des CTC : pour l'utilisateur, le principal changement est l'arrivée de SATURNE qui remplace GIPSI STATS.

Cette étape s'est achevée en 2005.

**Étape 3** : Essentiellement une refonte des outils métiers du tri. Le principal changement est l'arrivée de SPOT en remplacement de GIPSI, la mise en œuvre d'une nouvelle version de GENESE et une réorganisation complète des processus entre les différents outils.

Cette étape s'est achevée en 2006.

### 1.3.2 Planning du projet SATURNE

Le planning ci-dessous présente l'état d'avancement macroscopique du projet SATURNE au sein de la DECLIC (ce planning n'était pas prévisionnel, mais régularisé en fonction des avenants).

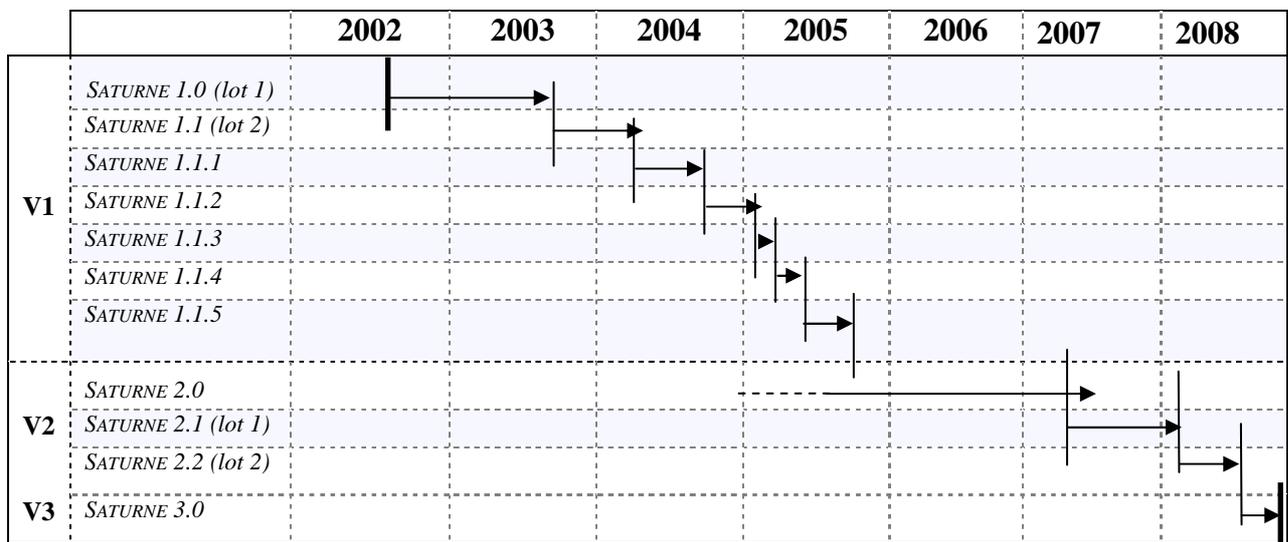


Figure 13 – Planning du projet SATURNE

La partie en pointillés dans la ligne « SATURNE 2.0 » signifie que j'ai progressivement commencé à travailler sur SATURNE 2.0 dès la fin 2004. Pourtant, j'ai continué à maintenir la version 1 de SATURNE en fonction des besoins prioritaires, et ce, parallèlement à la conception et au développement de SATURNE 2.0.

## 1.4 Cycle des projets : la théorie

Cette partie présente les recommandations qui s'appliquent pour tout projet informatique mené à LA POSTE en ce qui concerne son cycle de vie, et surtout son cycle de développement.

Mais pour prendre le recul qui se doit, une partie ultérieure de ce mémoire (partie 1.5, page 34) montrera les limites de cette théorie telle qu'elle est applicable dans le projet SATURNE.

### 1.4.1 Cycle de vie du projet

Le cycle de vie du projet SATURNE, tel qu'il était prévu dès le début en mode TMA<sup>31</sup> pour les deux lots de SATURNE V2, est, comme toujours, basé sur des cycles de développement répondant chacun au nom de « cycle en V » (cf. explication dans le prochain paragraphe).

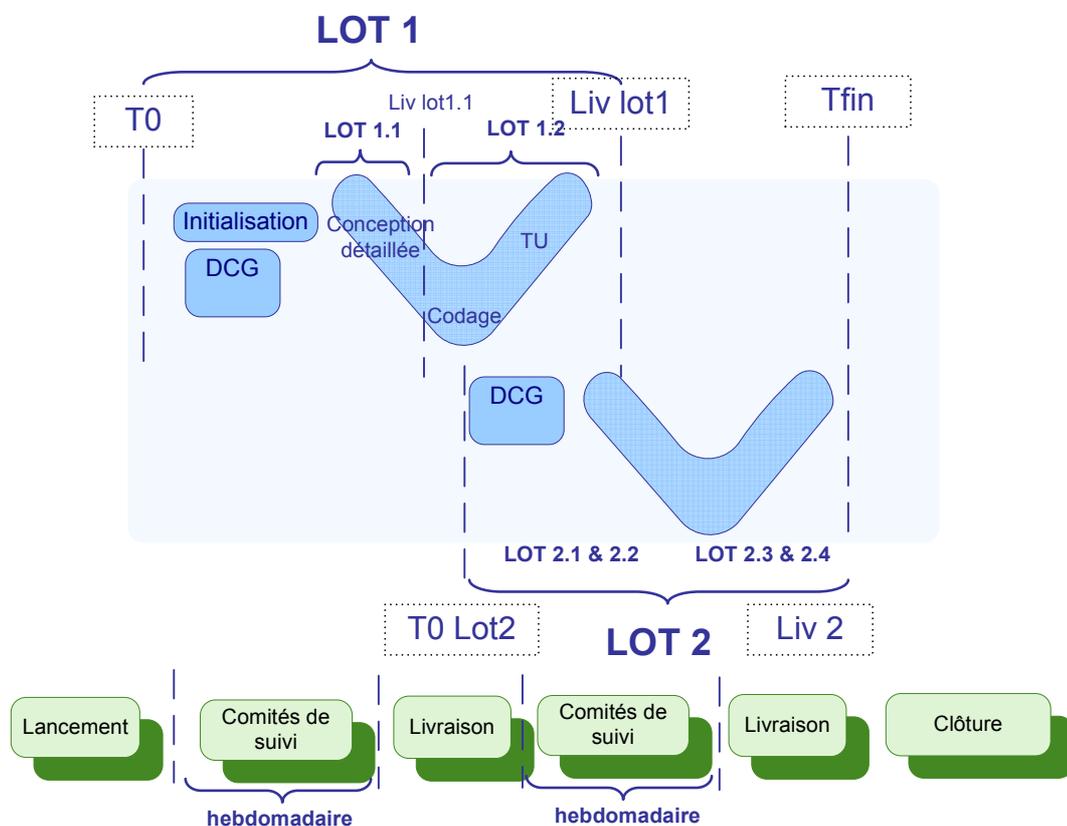


Figure 14 – Cycle de vie type du projet SATURNE

<sup>31</sup> Le contexte de la TMA est expliqué en page 123.

## 1.4.2 Cycles de développement du projet

Les cycles de développement mis en œuvre pour chacun des lots du projet SATURNE sont donc des « cycles en V ». Ce dernier, vu et revu dans le monde du projet informatique, consiste à dérouler les différentes étapes de développement en s'appuyant à chaque phase sur un contrôle de conformité par rapport à la phase amont. En d'autres termes, lorsqu'une étape est validée et que le passage à l'étape suivante est accepté, il est impossible de modifier les éléments de sortie de la phase amont sans reprendre le cycle du projet suivant :

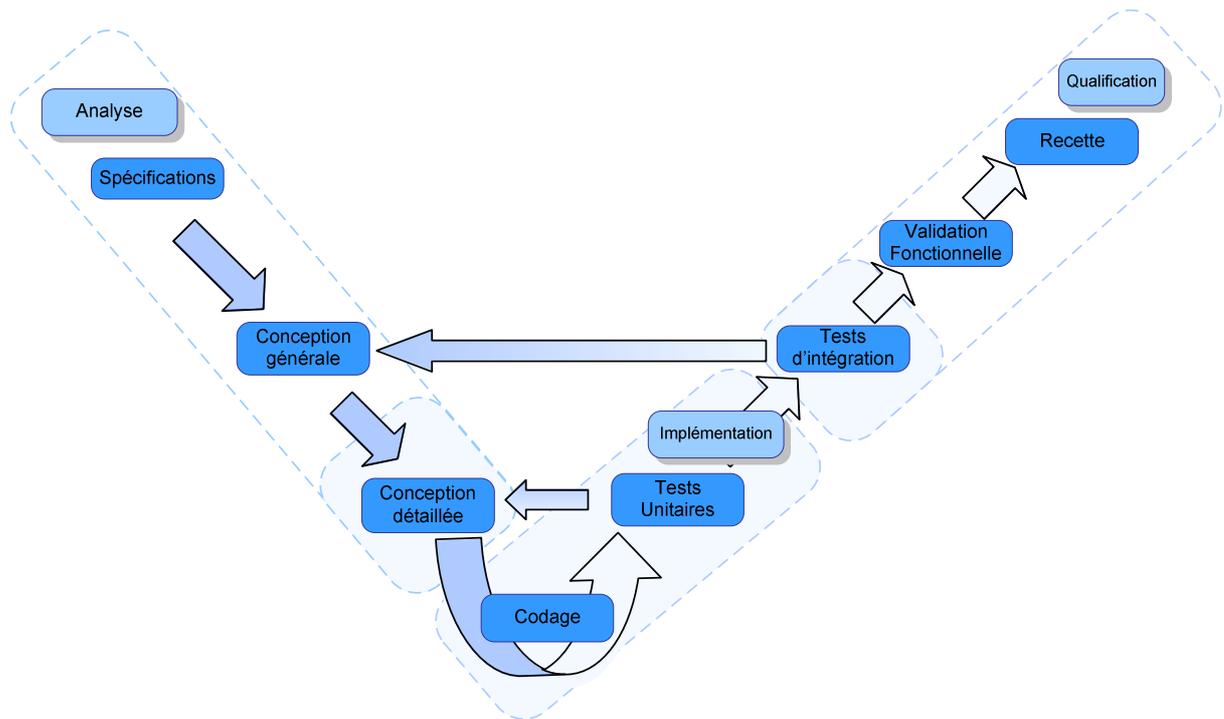


Figure 15 – Schématisation du cycle en V d'un projet informatique

### 1.4.2.1 Phase d'analyse

La phase d'analyse est constituée pour ce projet d'une étape de conception générale et d'une étape de conception détaillée.

#### 1.4.2.1.1 Étape de conception générale

L'étape de conception générale comprend l'ensemble des activités consistant à l'étude de la DFB (Définitions des besoins<sup>32</sup>).

La DFB a été rédigée par la MOA, et c'est l'équipe projet (MOE, c'est-à-dire moi-même) qui réalise le DCG (Dossier de conception générale) sur la base de la DFB fournie par LA POSTE DECLIC.

<sup>32</sup> Terminologie LA POSTE (synonyme de « cahier des charges »)

Éléments en entrée :

- DFB, avec date de livraison envisagée pour les éléments en sortie
- Document d'Architecture Technique
- Document d'Architecture Logicielle
- Documents de Questions Réponses

Éléments en sortie :

- Dossiers de Conception Générale (DCG)
- Dossier de tests et planning de réalisation

Condition de passage à la phase suivante :

- Acceptation du DCG par le client, avec date de validation.

*1.4.2.1.2 Étape de conception détaillée*

L'étape de conception détaillée comprend l'ensemble des activités consistant à l'étude de l'architecture technique et à la définition plus précise des fonctionnalités logicielles. À ce niveau, la qualité de la description doit être suffisamment fine et précise pour permettre la programmation.

L'équipe SATURNE réalise les dossiers de conception détaillée sur la base des dossiers de conception générale fournis par LA POSTE DECLIC.

Éléments en entrée :

- Dossiers de conception Générale (DCG)
- Fiches de Questions Réponses (FQR)

Éléments en sortie :

- Dossiers de Conception Détaillée (DCD)
- Plan de tests unitaires

Condition de passage à la phase suivante :

- Acceptation du DCD par le client, avec date de validation.

#### 1.4.2.2 Phase d'implémentation

La phase d'implémentation est constituée d'une étape de codage, d'une étape de tests unitaires et d'une étape de tests d'assemblage.

##### 1.4.2.2.1 Codage

Le codage des programmes consiste à implémenter la conception détaillée validée par LA POSTE DECLIC.

Chaque membre de l'équipe traduit en code la ou les fonctionnalités dont il a la charge.

##### Éléments en entrée :

- Dossiers de conception détaillée validés par LA POSTE DECLIC.

##### Éléments en sortie :

- Code généré avec date de livrable estimée.

##### Condition de passage à la phase suivante :

- Une revue de code est effectuée par LA POSTE (uniquement sur livraison d'une version majeure, ou à la demande) avec des indicateurs qualité mis à la connaissance de l'équipe APSIDE afin de remédier à d'éventuels dysfonctionnements.

##### 1.4.2.2.2 Tests unitaires

L'étape de test unitaire est le premier niveau de test. Chaque membre de l'équipe teste unitairement les parties de code dont il a la charge. Pour cela, il doit construire les tests en prenant en compte les exigences suivantes :

- Tester la fonctionnalité dans les cas passants et non passants ;
- Contrôler la validité des algorithmes de calculs ;
- Tester toutes les boucles afin de détecter le code mort<sup>33</sup> ou les boucles infinies ;
- Des tests unitaires sont réalisés pour chaque composant afin de vérifier leur conformité aux spécifications.

##### Éléments en entrée :

- Code généré
- Plan de test suivant normes et préconisations

##### Éléments en sortie :

---

<sup>33</sup> code jamais exécuté

- Code unitairement testé.
- Dossiers de tests unitaires.

Condition de passage à la phase suivante :

- Résultats de test sont conformes aux résultats attendus.

*1.4.2.2.3 Tests d'assemblages*

L'étape de test d'assemblage permet de s'assurer de la bonne intégration des différents composants développés ou modifiés et de la validité de l'ensemble.

*1.4.2.3 Dossiers de mise en production*

Nous devons réaliser les dossiers de mise en production afin d'accompagner l'équipe de production de la DECLIC lors de la mise en production des différents composants ou chaînes.

Les dossiers d'installation et d'exploitation seront réalisés et fournis à chaque mise à jour de module et pour chaque lot.

*1.4.2.4 Recette provisoire et définitive*

Après livraison du lot et selon les jalons définis, le PV de recette sera émis selon les dates définies lors des points d'avancement.

### 1.4.3 Récapitulatif sur les livrables

Dans le cadre du contrat de service liant APSIDE à LA POSTE, nous avons le devoir de tenir à jour les documents suivants pour maintenir constamment la possibilité de reprise des applications dont on a la charge, ainsi que la facilité de maintenance :

Livrables (documents)	Étapes de réalisation	Disponibilité Client
Dossiers de conception générale	Conception générale	Livraisons intermédiaires
Dossiers de conception détaillée	Conception détaillée	Livraisons intermédiaires
Plan de test	Conception détaillée	Livraison intermédiaire
Dossiers de tests unitaires	Test unitaire	A la livraison du lot
Programmes	Codage	A la livraison du lot
Dossiers de mise en production	-	A la livraison du lot
Tableau de bord	Toutes	Hebdomadaire
Compte-rendu de pilotage	Toutes	Ponctuel
Compte rendu d'avancement	Toutes	Hebdomadaire

Tableau 2 – Ensemble des documents à livrer dans le cadre de notre prestation de service

## **1.5 Démarche appliquée à SATURNE : de la théorie à la pratique**

Il est intéressant de décrire chacun des cas de « dérive » entre la théorie, exposée dans la partie précédente, et la manière effective dont LA POSTE et l'équipe dont je fais partie avons mené le projet SATURNE.

En effet, les impératifs du terrain nous ont souvent imposé de faire preuve de flexibilité par rapport aux recommandations théoriques en termes de gestion de projet. C'est pour cela qu'à mon sens, l'esprit d'initiative attendu pour un ingénieur implique entre autres une capacité à prendre un juste recul par rapport aux normes théoriques de gestion de projet.

Énumérons donc quelques unes de ces différences, en précisant pour chacune l'avantage et l'inconvénient :

### **1.5.1 J'ai rédigé le DCG d'une seule traite pour les deux lots avant de commencer le développement du lot 1.**

Il était pourtant prévu deux étapes dans la rédaction du DCG de SATURNE 2.1 : une fois avant le développement du lot 1, une autre fois avant le développement du lot 2.

#### ***Avantage :***

Cela m'a permis d'obtenir à l'avance une vue complète des spécifications du projet, ce qui a fait anticiper quelques contraintes liées aux spécifications du lot 2, et donc en prendre quelques unes en considérations dès le lot 1.

#### ***Inconvénient :***

Le lot 1 a été livré moins rapidement (mais la MOA était consentante).

### **1.5.2 Le DCD a été rédigé après le développement, et non avant.**

#### ***Avantage :***

J'ai pu livrer plus rapidement, ce qui m'a permis à la fois de satisfaire les exigences de délai de la maîtrise d'ouvrage, et de rattraper une perte de temps déjà engendrée dans les premières phases du projet (cette perte de temps est typiquement due à une DFB finalisée avec beaucoup de retard par la MOA, ou encore un DCG dont la rédaction a demandé plus de temps que prévu).

#### ***Inconvénient :***

Le DCD ayant constitué la dernière tâche du planning effectif avant que ma mission à LA POSTE ne prenne fin, il n'a donc pas été validé par le client : cela contredit un des principes de base de notre contrat de service, selon lequel tout DCD doit être validé (au même titre que le DCG).

Précisons au passage que le simple fait de livrer un DCG est un luxe que ne se permettent pas toutes les entreprises. Ce fait s'explique généralement soit par une restriction budgétaire, soit par une restriction de délai très forte<sup>34</sup>. Et malheureusement, il est aisé de constater que l'une ou l'autre de ces deux restrictions entraîne des sacrifices facilement identifiables :

- dans la phase de documentation : ce qui pénalise la maintenance ;
- dans la qualité de l'interface : ce qui pénalise la convivialité de l'application ;
- dans la phase de tests : ce qui pénalise la fiabilité de l'application.

Mais ne nous leurrions pas : en ces temps de crise, alors que ma mission a pris fin début 2009, même LA POSTE est concernée par les restrictions budgétaires.

### **1.5.3 Je n'attends pas que soit validée la DFB avant de rédiger le DCG, ni que soit validé le DCG avant de commencer les développements.**

#### ***Avantage :***

Il n'y a aucun temps mort entre chacune des phases du projet.

#### ***Inconvénient :***

Des pertes de temps sont occasionnées par une éventuelle remise en cause de certaines implémentations dont le développement a déjà été réalisé : ce qui impose donc un retour arrière<sup>35</sup>.

### **1.5.4 La MOA refuse de valider le DCG, jugeant ce type de document « trop technique ».**

La raison principale est la présence du MCD (pourtant cruciale dans un DCG), et ce, malgré l'AMOA<sup>36</sup> qui aurait pourtant prouvé ici sa valeur ajoutée si elle avait été consultée. La validation du DCG est donc assurée par le chef de projet MOE (Loïc Bruggeman), bien qu'il ne s'agisse pas là du protocole officiel.

#### ***Avantage :***

Un gain de temps et d'énergie pour la MOA, qui économise ainsi la lecture fastidieuse d'un document de 466 pages jugé trop technique.

---

<sup>34</sup> « – C'est à faire pour quand ? – Pour hier ! ». Ce dialogue-type bien connu que j'entends régulièrement illustre la pression frappant la plupart des entreprises dans un monde concurrentiel où le rythme de production déjà fortement soutenu est boosté par les possibilités à double tranchant offertes par les nouvelles technologies : tout *peut* aller très vite, et la concurrence est à notre porte, donc... tout *va* très vite.

<sup>35</sup> Mais les retours arrière sur développements sont peu fréquents : ces pertes de temps sont donc largement compensées par les gains de temps apportés par l'avancement des développements.

<sup>36</sup> AMOA signifie « Assistance à maîtrise d'ouvrage ».

***Inconvénient :***

Un nouveau document, compréhensible par la MOA, devient obligatoire à chaque livraison (quelle chance, un document de plus). Il s'agit du « Document de spécification fonctionnelle ». Ce document revient à une sorte de réécriture de la DFB avec nos yeux d'informaticiens, et doit dès lors être mis à jour pour chaque correctif, au même titre que le DCG. La rédaction de ce document, conjointe à celle du DCG, représente donc un alourdissement conséquent de notre budget temps dans chaque lot.

**1.5.5 J'annonce les délais avant la rédaction du DCG, et non après.**

***Avantages :***

Ce geste de souplesse vis-à-vis de la société cliente (en l'occurrence LA POSTE) lui permet de s'organiser au plus vite pour définir les créneaux de livraison sur son planning général. Cette livraison se fait successivement :

- à l'équipe de tests internes (deux prestataires d'une autre société externe) ;
- puis à l'équipe TIT (deux prestataires de la société ST Group) ;
- puis à Nantes (DTC<sup>37</sup>) : plateforme de recette fonctionnelle ;
- puis à Chalons en Champagne (SPT<sup>38</sup>) : plateforme de pré-production pour les tests de montée en charge ;
- et enfin, à Limoges : plateforme de production.

***Inconvénient :***

La rédaction du DCG peut mettre en lumière des difficultés imprévues dans l'implémentation, ce qui augmente le temps de développement qui pourtant a déjà été communiqué à LA POSTE.

Pour nous protéger au mieux des risques de débordement de planning dus aux imprévus, nous avons convenu en accord avec Loïc Bruggeman (mon chef de projet à LA POSTE) d'appliquer le récent principe enseigné par les formations « MMI » en conduite de projet, consistant à rajouter d'emblée un tiers de temps au chiffrage initial. Ce « bonus de temps » a pour fonction de supporter tout imprévu dans l'expression du besoin, sans pour autant remettre en cause la date de livraison qui a été entendue.

D'ailleurs, dès le début du 20<sup>e</sup> siècle, l'industriel Auguste Detœuf affirmait « *Il ne faut jamais oublier de prévoir l'imprévu.* »

---

<sup>37</sup> Direction Technique du Courrier

<sup>38</sup> Site Pilote Technique

### 1.5.6 Je livre souvent une version « beta » avant la date convenue.

#### *Avantages :*

L'équipe de tests gagne du temps, tandis que je reçois les premières anomalies en avance. Cela me permet de :

- Livrer une version définitive plus stable que si on n'avait pas livré de version beta
- Comptabiliser moins d'anomalies officielles... ce qui améliore nos « statistiques qualité » en tant qu'équipe TMA.

#### *Inconvénients :*

- Souvent, tout ne peut pas être testé dans la version beta, car le fait de livrer une version plus tôt que prévu fait que certaines fonctions peuvent ne pas être complètement finalisées. Mais souvent, on essaye de laisser pour la fin les développements les moins déterminants (notamment la mise à jour de l'aide en ligne).
- Augmentation du nombre de livraisons, donc plus de temps pour préparer les paquets.

### 1.5.7 Je n'ai pas reçu en entrée de Document d'Architecture Logicielle.

Cela s'explique par le fait que lorsque je suis arrivé dans cette mission, le projet n'en était qu'à ses prémices.

*Avantage :* aucun

#### *Inconvénient :*

C'est à moi de rédiger le *Document d'Architecture Logicielle*<sup>39</sup> de SATURNE, afin de mettre mon projet aux normes, mais surtout de faciliter un éventuel passage de compétence.

---

<sup>39</sup> En raison des priorités, en accord avec notre client, nous faisons l'impasse sur le *Document d'Architecture Technique*.

### **1.5.8 Les réunions ne sont pas hebdomadaires.**

Un principe généralement valable dans tout projet informatique est de tenir une réunion au moins une fois par semaine, en vue d'un point régulier sur les tâches en cours. Il se trouve qu'en raison d'une trop faible disponibilité de mes différents chefs de projets, ce principe pourtant retenu en théorie a rarement été respecté durant mon projet à La Poste.

#### ***Avantages :***

Certaines parties des réunions ne concernent nullement mon travail : cela occasionne d'une part une perte de temps, et d'autre part un potentiel sentiment d'ennui. Aussi, la raréfaction des réunions me permet d'accorder plus de temps pour le reste de mon travail, autant qu'il optimise ma stimulation quotidienne. Précisons au passage qu'une meilleure disponibilité entraîne une meilleure réactivité dans l'analyse des problèmes rencontrés en production.

#### ***Inconvénients :***

En échappant à une réunion, on échappe aussi aux bienfaits d'une mise en commun sur le suivi des tâches en cours. Cela peut avoir comme effets la perte de temps précédant une réunion qui peut nous « remettre sur les rails » (si jamais, par exemple, une initiative personnelle a dépassé les limites du cahier des charges).

*Pour contrebalancer cette partie « Inconvénients », précisons toutefois qu'une réunion en bonne et due forme (programmée à l'avance dans une salle de réunion) ne peut constituer la seule manière de faire le point avec son équipe : d'une part, les membres de mon équipe et moi-même travaillant dans le même bureau, il arrive que nous fassions le point de manière spontanée et adaptée jusqu'à deux ou trois fois par jour. D'autre part, le bureau de mon chef de projet étant accessible en quinze secondes, il n'est pas rare que nous fassions spontanément une mise au point avec lui sur la manière de traiter tel incident, tel manque de précision dans la DFB ou encore tel problème logistique.*

## **1.6 L'équipe**

### **1.6.1 Schéma d'ensemble**

Comme le montre le schéma suivant, les interactions humaines entre les membres de mon équipe sont multiples. De même, il existe une forte imbrication des deux projets SNT et SATURNE. Cela s'explique par le fait que SNT transmet à SATURNE une partie des données

de production, et qu'un bon nombre de règles de gestion exprimées par la MOA pour SNT sont en réalité appliquées dans SATURNE, ou inversement<sup>40</sup>.

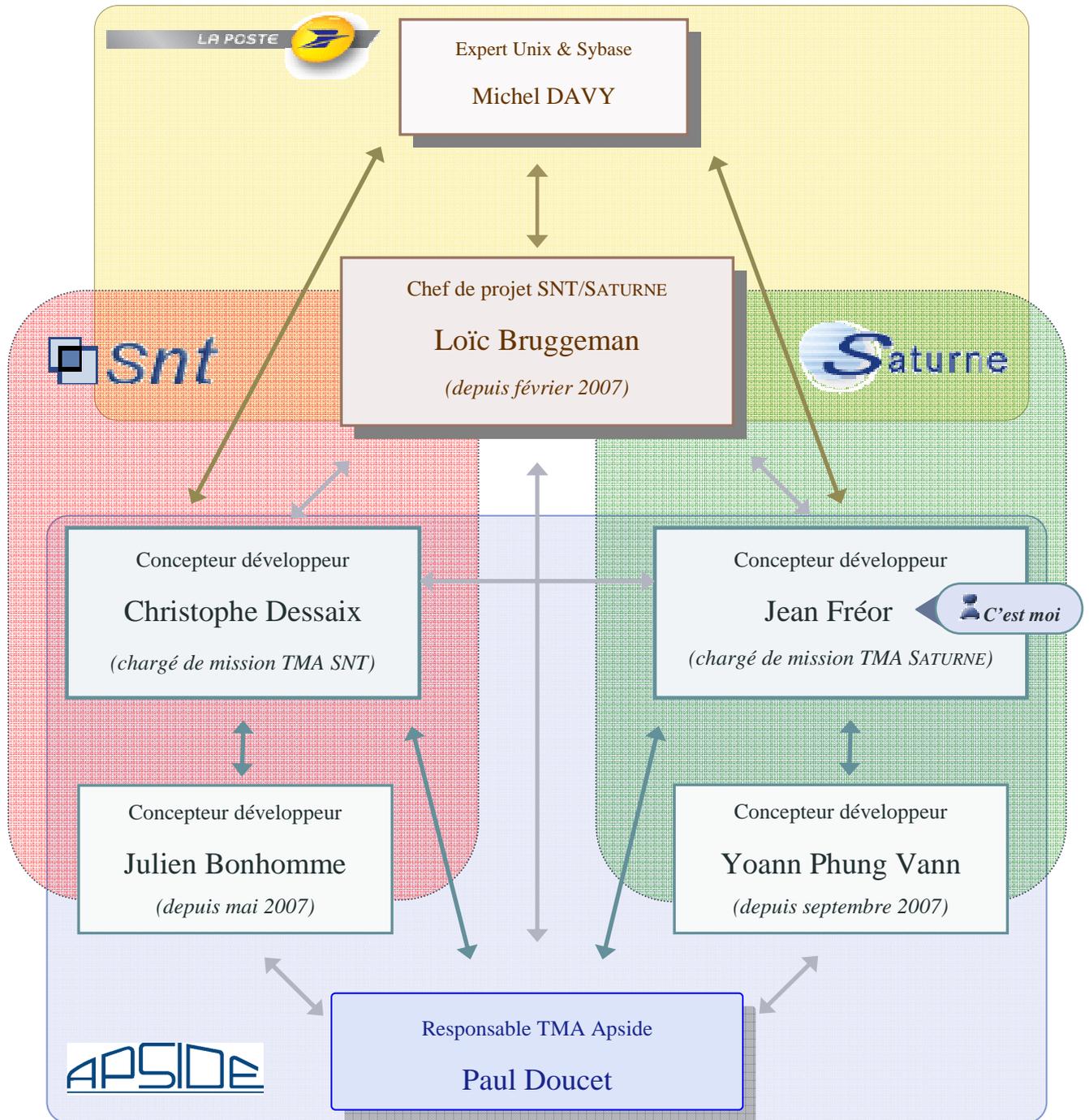


Figure 16 – Interactions entre les acteurs LA POSTE et APSIDE pour les projets SNT et SATURNE

<sup>40</sup> Par exemple, certains indicateurs sont calculés par SNT mais visibles dans SATURNE.

### 1.6.2 Mes niveaux d'interventions

Je couvre personnellement la plupart des étapes de conception et de développement du projet SATURNE :

- Spécifications des IHM ;
- Spécifications du modèle de données (pour les évolutions demandées) ;
- Force de proposition pour intégrer de nouvelles fonctionnalités ou améliorer des fonctionnalités existantes, en particulier la qualité des interfaces graphiques ;
- Rédaction du DCG (Document de conception générale – spécification précise du système, en réponse au cahier des charges, à valider par la maîtrise d'ouvrage –) ;
- Le développement du module web ;
- Le développement des procédures stockées Sybase ;
- Rédaction du DCD (Document de conception détaillée) ;
- Les tests ;
- Le packaging et la livraison de chaque version ;
- Traitement des anomalies.

En revanche, je ne prends pas en charge :

- Les choix technologiques ;
- Le cahier des charges (maîtrise d'ouvrage) ;
- Le dialogue avec les utilisateurs et avec la maîtrise d'ouvrage ;
- Le transfert de données entre SATURNE et ses référentiels de données (LA POSTE utilise un produit d'EAI : WTX 8.1<sup>41</sup>) ;
- La validation du DCG (Document de conception générale) ;
- Les tests de montée en charge (délégués à l'équipe de métrologie) ;
- L'installation de SATURNE (déléguée à l'équipe de production) ;
- La maintenance du serveur web ;
- La maintenance de la base de données ;
- La répartition de la charge par clonage multiple du serveur web (système Altéon).

---

<sup>41</sup> WTX (« Websphere Transformation Extender »), était appelé « Mercator » avant de se faire racheter par IBM en 2005. Ce produit complète désormais la suite logicielle « Websphere » d'IBM.

## Chapitre 2. L'ARCHITECTURE DE L'APPLICATION SATURNE

Avant de poursuivre, précisons pourquoi il est bon de se soucier de l'architecture dans un projet comme SATURNE : La conception d'une application informatique n'est pas faite de créativité artistique... malheureusement pour moi, aurais-je pu croire, mais réfléchir à l'architecture d'un système informatique représente deux avantages de taille. D'une part, cela implique de prendre du recul sur un projet, ce qui permet de mieux appréhender ou de remettre en question la pertinence des choix technologiques. D'autre part, s'occuper d'architecture des systèmes peut apporter une garantie qui peut se révéler bénéfique : celle d'un revenu confortable<sup>42</sup>.

Pour reprendre le fil de ce mémoire, il s'agit ici de présenter la façon dont l'application intranet SATURNE « se découpe ». Or, il y a toujours plusieurs manières de découper une application en différentes parties.

Aussi, dans ce chapitre, il sera d'abord montré de quelle manière SATURNE répond au modèle d'architecture dite « client-serveur », et ce en quoi consiste cette dernière. Puis seront exposées les différentes vues de l'architecture de SATURNE : logique, physique, etc.



Plan d'une maison par Ferdinand von Arnim (1852)

<sup>42</sup> Dans les métiers de l'informatique, à moins d'avoir le statut d'expert technique, la rémunération est en général proportionnelle à la distance qui relie l'informaticien et le code source. Et de la même manière, un webdesigner ne peut pas prétendre au même salaire qu'un ingénieur en informatique. À titre d'exemple, le site internet *CadresOnline* [COL WEB] annonce une rémunération moyenne de 58k€ pour le métier d'Architecte des systèmes d'information, contre 40k€ pour un ingénieur d'étude.

## 2.1 Découpage en modules

Considérons d'abord le découpage en modules. Il va sans dire que la notion de « module » n'a pas toujours le même sens selon la personne qui emploie ce terme : en ce qui concerne la terminologie LA POSTE, un module désigne un « morceau logiciel » de l'application, répondant à tel type de plateforme, et donc à tel « socle »<sup>43</sup>. L'application SATURNE est donc composée de trois « modules », chacun pouvant être livré séparément, et chacun devant être installé sur un serveur spécifique :

- Le module « web » ;
- Le module « base de données » ;
- Le module « intégration des données ».

## 2.2 Bases théoriques d'une architecture client-serveur multi-tiers

Pour rappel, on appelle traditionnellement « client-serveur » une architecture où un poste client exécute une application se trouvant sur un serveur, c'est-à-dire un poste généralement distant, et avec lequel l'application cliente interagit via des requêtes.

Le client-serveur est apparu pour répondre aux problématiques que posaient les systèmes centraux de type Mainframe : ces systèmes, qui pourtant à l'heure actuelle ont toujours leur place dans les SI, sont fermés<sup>44</sup>.

### 2.2.1 Les différents modèles client-serveur

La première des questions à se poser en architecture des systèmes est en effet de savoir de quelle manière on sépare les trois champs d'activité de l'application, qui sont généralement les suivants :

- La présentation (récupération des données, affichage avec mise en forme, envoi vers le serveur de paramètres et de données saisies) ;
- Les traitements applicatifs<sup>45</sup> (vérification, application des règles de gestion métier, transformation et transfert des données) ;
- Le SGBD (application règles de gestion technique, fourniture de données et réception de nouvelles données).

---

<sup>43</sup> Un socle correspond à un ensemble de standards et de versions d'OS et logiciels propres à la dite plateforme.

<sup>44</sup> On les dit « propriétaires » : autrement dit, il n'y a pas de possibilité de les adapter sur des produits de fournisseurs différents, car ils sont à l'opposé des « standards ».

<sup>45</sup> Encore appelé « tiers métier »

Le schéma suivant montre la diversité des procédés traditionnels pour répartir ces trois champs d'activité :

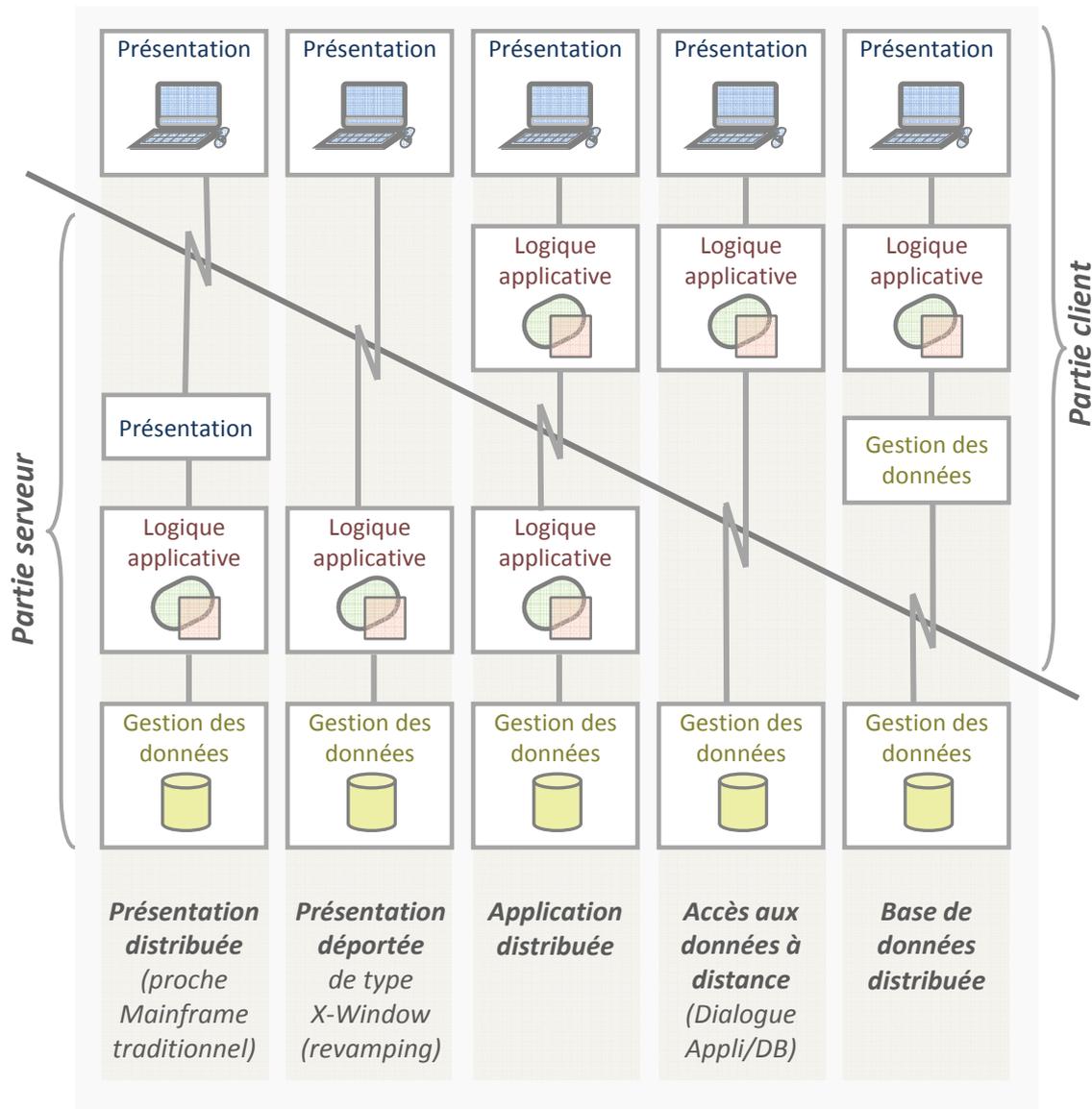


Figure 17 – Les différents modèles client-serveur

Ce schéma, qui s'inspire d'une partie du livre « Client/serveur, guide de survie » [CS 99] montre des architectures fondées sur le fait qu'il y a une partie cliente et une partie serveur, et que chacune d'entre elles reposent sur des machines séparées. Mais ce que ce schéma met moins en évidence, est la notion d'architecture à plusieurs niveaux : pas seulement une machine pour le client et une pour le serveur, mais souvent une machine pour chacune des trois couches décrites précédemment (présentation, traitements et données). C'est ce que l'on appelle une architecture « 3-tiers<sup>46</sup> », ou « tripartite ».

<sup>46</sup> Rien à voir avec une notion de division en trois : tiers est ici synonyme de « partie ».

Enfin, on peut même aller plus loin, en considérant qu'une ou plusieurs des fameuses couches présentation, traitement et données puisse être elle-même décomposée en plusieurs parties, chacune résidant sur un serveur à part. C'est ce que l'on appelle une architecture « n-tiers » (ou « multi-tiers » : par exemple, on aura le client, puis un serveur pour la partie présentation à envoyer au client (serveur web par exemple), ensuite un voire plusieurs serveurs applicatifs pour la partie assurant les règles de gestion, et enfin un serveur de base de données.

### **2.2.2 Un modèle pour SATURNE ?**

En rapport avec le schéma précédent, j'ai d'abord identifié l'architecture de SATURNE comme celle d'une « application distribuée ». En effet, la présentation et les données reposent respectivement sur les parties client et serveur, et la couche applicative prend place à la fois côté client (traitements Javascript) et côté serveur (traitement dans les pages ASP, les composants Active X et les procédures stockées).

Mais en réfléchissant, l'architecture de SATURNE ne se rapproche-t-elle pas plutôt du modèle « présentation déportée » ? En effet, le fait de faire une simple distinction « client / serveur » correspond à une architecture « bipartite », et ne suffit plus de nos jours : au-delà de celle-ci, et même de l'architecture « trois tiers », le concept de client "universel" (comprendre navigateur web) a émergé. Ce concept repose sur la technologie internet : aussi, les applications internet/intranet proposent une architecture client-serveur appelée « multi-tiers » : c'est-à-dire à trois, quatre, cinq niveaux ou plus encore. Cette classe d'architecture permet de répondre aux problèmes posés par l'hétérogénéité des systèmes, et notamment leur administration.

D'ailleurs, pour une personne comme moi issue de la « génération web », considérer qu'il y a différentes approches client/serveur possibles pour développer une application n'est pas si évident : en effet, développer une application de gestion de données multi-utilisateur en n-tiers ou pas, est une question que l'on ne se pose même plus souvent dans le monde des nouvelles technologies. Tout simplement parce qu'avec une application comme SATURNE, le choix d'une architecture internet/intranet est devenu un réflexe... ce qui induira à coup sûr du « multi-tiers » !

### **2.2.3 Description d'une architecture multi-tiers**

Dans une architecture multi-tiers, on sépare normalement toute la partie d'interfaçage et la gestion des données des diverses règles de gestion, afin de pouvoir servir des clients hétérogènes et utiliser potentiellement divers moyens de stockage des données.

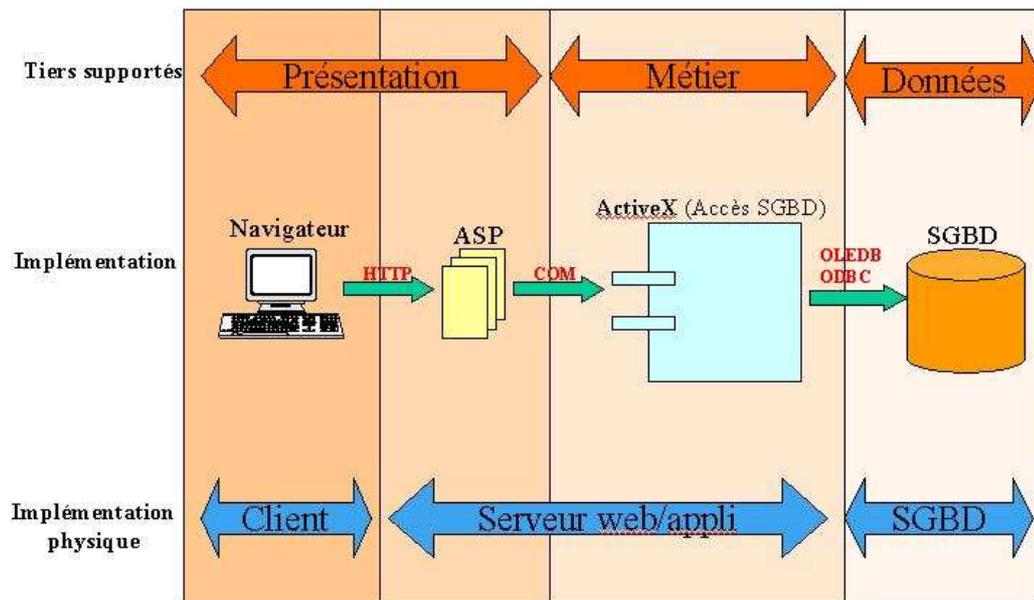


Figure 18 – Découpage d'une application client-serveur 3 tiers avec un serveur web Microsoft

Cette architecture vise aussi à ramener les règles de gestion au niveau des composants plutôt qu'au cœur du SGBD. Ceci offre 2 avantages :

- L'utilisation de langages évolués et d'ateliers de développement pour la réalisation des différents composants ;
- La possibilité d'augmenter la résistance à la montée en charge de la solution développée en constituant des fermes de serveurs web, plutôt que par augmentation des capacités du serveur portant le SGBD.

#### 2.2.4 Avantages et inconvénients d'une architecture multi-tiers

Cette architecture n'est pas forcément valable pour les petites applications (sauf si c'est une application web, multi-tiers par essence), alors qu'elle est forcément valable pour les moyennes, et encore plus pour les grosses : en d'autres termes, le surcoût de développement engendré par cette architecture la rend moins intéressante pour les petits projets. Finalement, la notion vraiment déterminante pour savoir si une architecture n-tiers vaut le coup ou pas, est souvent celle de la rentabilité des développements.

Les avantages d'une application multi-tiers sont nombreux :

- La scalabilité (résistance à la montée en charge) ;
- Le coût et les contraintes de maintenance réduits (pas de déploiement sur tous les postes clients) ;
- La performance.

### 2.3 Schéma d'architecture technique de SATURNE

Pour faire comprendre l'architecture technique de l'application SATURNE, le schéma le plus lisible sera certainement une illustration mettant à la fois en évidence la vue « technique logique »<sup>47</sup> et la vue « technique physique »<sup>48</sup>.

Celui-ci figure en page suivante.

---

<sup>47</sup> L'architecture *logique* d'un système est divisée en trois couches : présentation, métier et accès aux données ; Les différents modes de répartition de ces trois couches sont décrits au paragraphe 2.2.1 (page 43).

<sup>48</sup> L'architecture *physique* d'un système décrit l'ensemble des composants matériels supportant l'application (postes de travail, serveurs web, base de données, modules divers...).

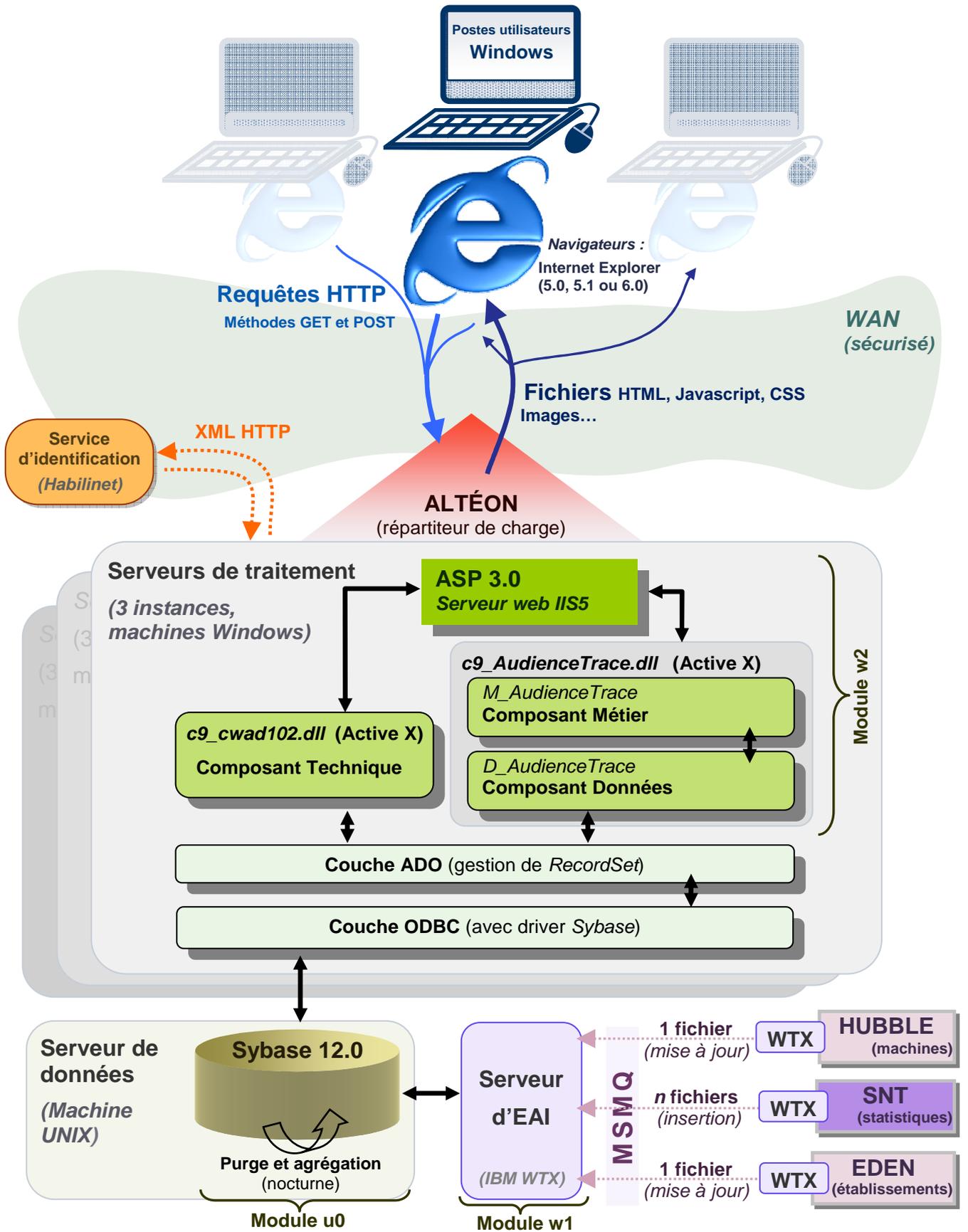


Figure 19 – Schéma d'architecture technique de l'application SATURNE

Ce schéma illustre les principes architecturaux suivants :

- Comme dit précédemment, l'architecture de SATURNE est une architecture client-serveur de nouvelle génération, dite multi-tiers. En effet, la partie serveur compte de nombreuses couches logicielles.
- Les utilisateurs se situent à distance, dans les établissements de LA POSTE : centres de tri, DOTC, PIC et autres, répartis sur tout le territoire français :
  - Le réseau est de type WAN (cf. partie 1.2.3 « L'environnement réseau », page 23)
  - Les postes client sont des instances du socle "WebAppli" :
    - Système d'exploitation : Windows 2000
    - Leur navigateur web est Internet Explorer, en version 5 ou 6

**Remarque**

Le module d'audiences utilisateurs dans SATURNE a révélé qu'en 2008, moins de 0.05% des utilisateurs utilisent encore la version 5 d'Internet Explorer : *a priori*, il n'est donc plus aussi impératif qu'en 2003 (50% d'IE5) de rendre 'compatible IE5' le code source du module web de SATURNE. Cependant, cet impératif n'a pas encore été levé officiellement.

- La résolution d'écran des postes utilisateurs peut être du 800x600 (bien que la grande majorité soit du 1024x768).
- Les utilisateurs se connectent à SATURNE via Internet Explorer. Les étapes de connexion sont les suivantes :
  - Connexion à l'intranet de LA POSTE appelé WAC<sup>49</sup> (Web d'accès au Courrier)
  - Clic sur « Connexion Habilinet » (l'application intranet de gestion des habilitations d'accès aux applications de LA POSTE)
  - Identification (saisie code Samba et mot de passe Habilinet) : le serveur web communique les données saisies au serveur Habilinet, via une communication « XML HTTP » (fonctionnement similaire à celui des web services).
  - Si l'identification a réussi, Habilinet ouvre SATURNE dans une fenêtre web séparée (sans barre d'adresse), en lui transmettant le nom, prénom et profil de l'utilisateur. (La dimension de la fenêtre web SATURNE s'adapte automatiquement à la résolution d'écran de l'utilisateur).
- En réalité, les requêtes HTTP passent par l'intermédiaire d'un serveur ALTÉON, dont le rôle est de distribuer la charge à 3 instances physiques de serveur HTTP SATURNE.

---

<sup>49</sup> <http://www.wac.courrier.intra.laposte.fr>

- Chacun de ces 3 serveurs d'application sont sur socle « PACC-WINDOWS-1.7 ». Ils sont installés<sup>50</sup> avec les composants suivants :
  - un serveur web Microsoft IIS5
  - un composants métier : *M\_AudienceTrace* (intégré au contrôle Active X *c9\_AudienceTrace.dll*)
  - un composant d'accès aux données : *D\_AudienceTrace* (intégré lui aussi à *c9\_AudienceTrace.dll*)
  - un composant technique : CWAD<sup>51</sup> 1.02 (*c9\_cwad102.dll*)

Le tout communiquant via le bus d'objets DCOM.

- Les pages ASP et le composant d'accès aux données communiquent avec la base de données (via ADO), installée sur un serveur UNIX non instancié (et non instanciable dans le cadre d'architecture actuelle). Sur ce serveur SGBD (socle "SGBD ACE 12"), est installée la base SATURNE, au format « Sybase 12.0 ».
- Cette base de données est alimentée par le module d'intégration des données en technologie WTX (*IBM Websphere Transformation Extender*), qui traite les données en provenance des 3 référentiels : HUBBLE pour les machines de tri et EDEN pour les établissements, ainsi que SNT (*Serveur National de Traitement*, qui met à disposition auprès des applications SATURNE et SYSPEO les données de production des machines de tri et des préparatrices du courrier).

Le principe technologique de la communication entre SATURNE et ses 3 référentiels de données est le suivant :

- Le transport des données d'un serveur à l'autre se fait sous forme de fichiers texte transitant via une file de message au format MSMQ (*Microsoft*) ;
- Les fichiers texte sont transmis et reçus par des modules WTX de chaque côté, dont chacun est connecté à sa base de données propre via un 'adaptateur Sybase' ;
- Le déclenchement des imports de données n'est pas ordonnancé, mais répond au principe de détection de fichier dans le répertoire d'entrée du serveur WTX côté SATURNE. Les imports de données se font donc au fil de l'eau.

#### **Remarque**

Contrairement à l'application SYSPEO, SATURNE ne communique pas avec l'application SNT via un composant de service DCOM. En effet, les normes techniques en vigueur à LA POSTE n'ont pas permis l'adoption de ce choix technologique (qui aurait pourtant assuré de la fiabilité, un coût réduit à la fois en terme d'argent et de temps de développement, ainsi qu'une simplicité de maintenance).

---

<sup>50</sup> L'installation du module web de SATURNE se fait suivant la doc 'c9\_w2mis\_<version\_SATURNE>.doc'

<sup>51</sup> CWAD, pour « Composant web d'accès aux données », est un composant technique développé par la DT.

## 2.4 Perspective logicielle de l'architecture de SATURNE

Le schéma ci-dessous présente une perspective logicielle de SATURNE :

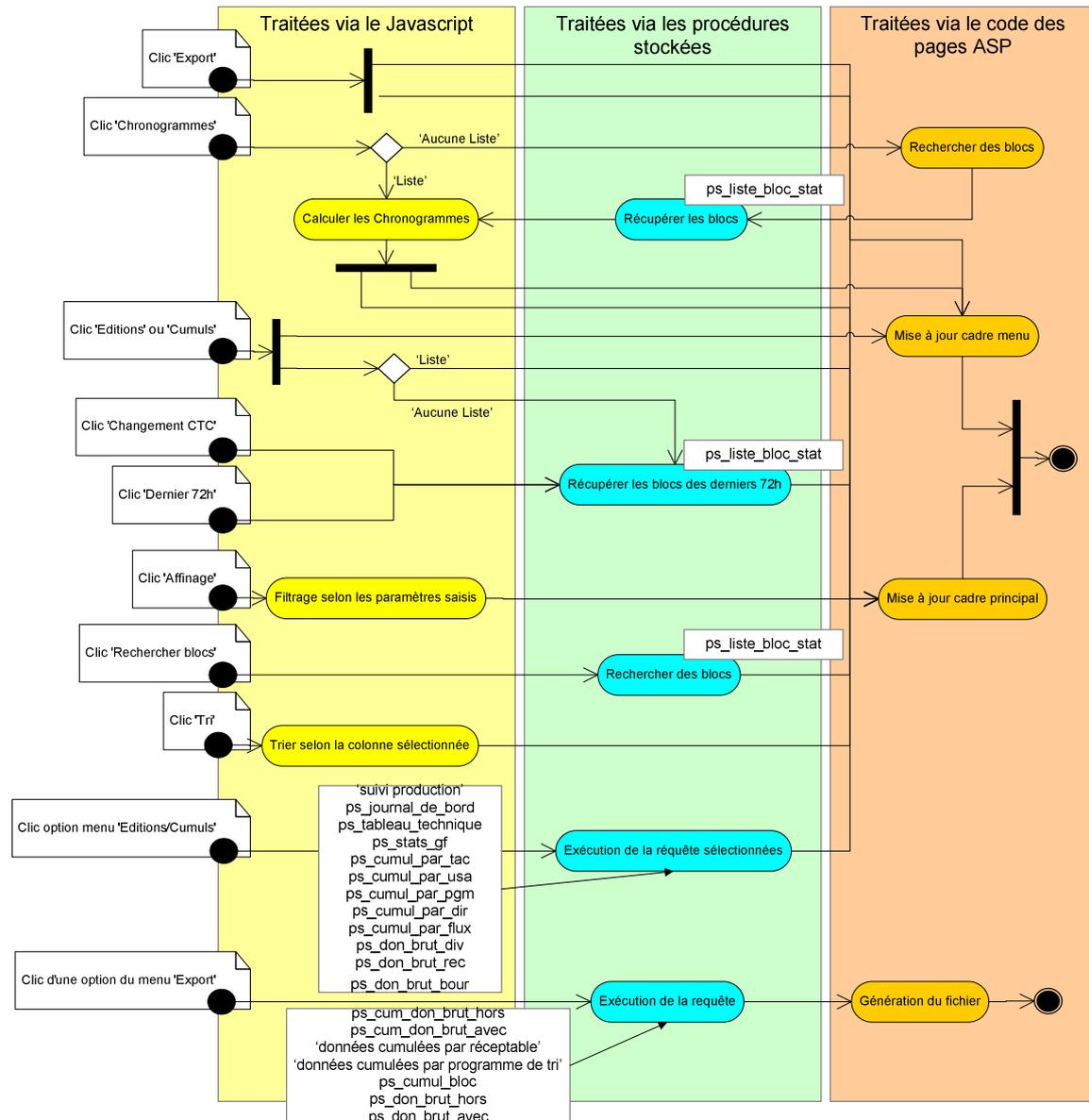


Figure 20 – Schéma de la perspective logicielle de SATURNE

Ce schéma permet de constater et de comprendre qu'il est délicat de libérer du traitement sur le tiers données (SGBD) au profit du tiers traitement (composants applicatifs).

En effet, le tiers données comporte un grand nombre de procédures stockées, mais chacune d'entre elles n'effectue quasiment que de l'extraction de données : le gros des traitements est déjà massivement assuré par le serveur web et le client.

## 2.5 Architecture fonctionnelle de SATURNE

Le schéma suivant expose l'architecture fonctionnelle, c'est-à-dire décomposée en « briques logiques » (acteurs) l'application SATURNE au sein du système d'échanges GALAXIE :

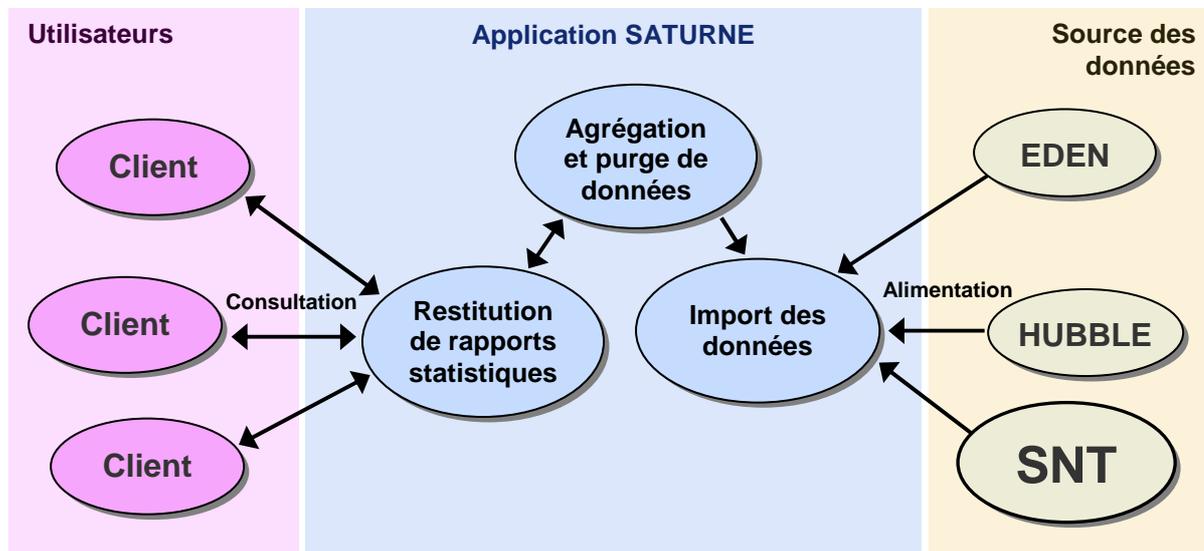


Figure 21 – Architecture de SATURNE : la décomposition en briques logiques

SATURNE se décompose en trois briques logiques :

- Une brique dédiée à la consolidation des données, remontant d'une part des deux référentiels EDEN et HUBBLE et d'autre part de l'application SNT ;
- Une brique dédiée à l'agrégation des données et des calculs statistiques ;
- Une brique dédiée à la restitution de rapports statistiques aux utilisateurs.

### 2.5.1 Brique de consolidation des données

Plus précisément, la première brique consolide les données provenant de trois sources de données différentes :

1. Le référentiel EDEN, qui alimente l'application SATURNE des données sur les établissements ;
2. Le référentiel HUBBLE, qui alimente l'application SATURNE des données sur les équipements (machines de tri et préparatrices de courrier) ;
3. L'application SNT, qui alimente au fil de l'eau l'application SATURNE des données de production.

Les données sont sauvegardées dans la base de données de l'application SATURNE telles qu'elles ont été reçues par le biais de l'une des trois sources de données. La brique de consolidation des données n'effectue donc aucun traitement sur celles-ci.

### 2.5.2 Brique d'agrégation des données

La deuxième brique agrège les données en fonction des demandes des utilisateurs. Un certain nombre de requêtes sont proposées à l'utilisateur, qui peut les affiner au besoin en

précisant un certain nombre de paramètres (type de machine de tri, date et heure, etc.). Cette brique s'occupe aussi de générer un certain nombre de compteurs et durées calculées. Certaines de ces données calculées devaient être initialement fournies par SNT.

### **2.5.3 Brique de restitution aux utilisateurs**

Finalement, la troisième brique gère les interactions avec les utilisateurs. À leurs demandes, elle envoie les requêtes des utilisateurs à la deuxième brique (agrégation et calculs) afin d'obtenir les sous-ensembles de données de production, et autres compteurs ainsi que les valeurs calculées à restituer aux utilisateurs.

Elle a donc en charge la mise en page des données et la restitution des rapports statistiques (que l'on appelle « éditions » dans SATURNE) que tableaux HTML et fichiers Excel générés à la volée.

## 2.6 Les fruits du multi-tiers dans SATURNE

### 2.6.1 Zéro déploiement

Pour SATURNE, le choix d'une application multi-tiers s'impose très clairement, et principalement en raison du grand nombre d'utilisateurs et de leur répartition géographique : on échappe ainsi à un coût et à des contraintes de déploiement conséquents.

### 2.6.2 Le web et les langages interprétés

De même, il n'est peut-être pas nécessaire de préciser qu'en tant que développeur d'une application web, je profite d'un incontestable avantage, en termes de coût de développement : pouvoir utiliser un langage interprété (proche de la notion de compilation *Just In Time*<sup>52</sup>). Le gain de temps vient du fait que visualiser le résultat du développement en cours se fait de manière instantanée, sans la moindre compilation manuelle.

En revanche, si l'utilisation de langage interprété a représenté un atout pour moi dans SATURNE en termes de confort de programmation, cela n'a été le cas que pour le développement des pages ASP : le code du composant métier (VB) doit être compilé manuellement (en un fichier .dll en l'occurrence<sup>53</sup>). Mais on sait qu'augmenter le confort d'un développeur en favorisant sa production de code à compilation *Just In Time* se fera au détriment des performances : c'est pourquoi il a été nécessaire dans SATURNE de créer ou migrer dans un composant métier les parties les plus gourmandes de la couche applicative. En effet, les composants métiers sont réellement plus rapides d'exécution que les pages ASP.

### 2.6.3 Développement d'un composant métier

J'ai donc été amené au cours de mon projet à LA POSTE à déporter certains traitements de SATURNE vers un composant métier, c'est-à-dire un module codé en VB.

---

<sup>52</sup> Ou encore « *JIT* » : la compilation du code se fait automatiquement, et juste avant son exécution (cas des servlets Java ou encore des pages.ASPX).

<sup>53</sup> Compiler le code d'un composant métier implémenté en tant que composant Active X consiste à générer sous Visual Studio un fichier '.dll', qui sera chargé en mémoire par une page ASP. Dans le développement de SATURNE, cela a représenté une contrainte pour moi, étant donné que généralement, à chaque nouvelle compilation, il faut libérer le '.dll' existant, donc arrêter le lot COM+ référencant cet Active X, puis écraser l'ancienne DLL par la nouvelle, et enfin redémarrer les services COM+. Plus encore : bien souvent, cette manipulation ne suffit pas et un bug système se produit sous Windows : il faut alors refaire le référencement des composants dans le lot COM+, c'est à dire les enregistrer de nouveau dans le système d'exploitation... C'est dans ce genre de cas d'instabilité du système que l'on regrette les langages interprétés !

L'intérêt du déport du métier des pages ASP et des procédures stockées vers un composant métier est multiple :

- En termes de scalabilité, il est plus simple, et (aussi et surtout) quasiment illimité d'ajouter des serveurs dans la ferme que de faire un upgrade de la machine SGBD ;
- En termes de réutilisation, l'indépendance vis à vis des autres tiers favorise la récupération des développements et la multiplicité des clients<sup>54</sup> ;
- En termes de maintenance, les composants sont écrits en VB6 ou VC++ (dans le cas de SATURNE, c'est du VB6), en général plus simples à maintenir que du code T-SQL (même si en revanche, pour ce qui est du code ASP, je le trouve plus facile à maintenir que le code VB6 : cf. partie 2.6.2 « Le web et I », page 53).

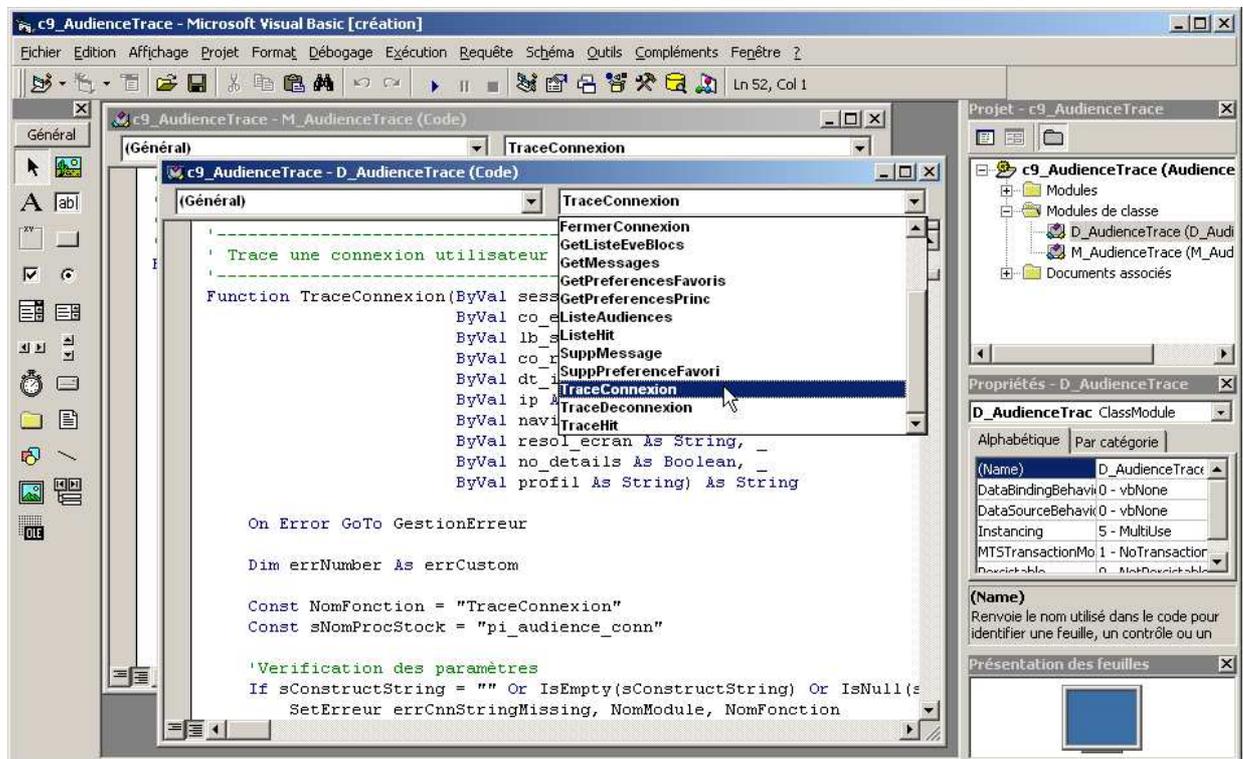


Figure 22 – Environnement de développement VB6 d'un composant métier de SATURNE

<sup>54</sup> Il s'agit là des notions d'interface et de contrat.

## 2.7 Inconvénients de l'architecture de SATURNE

### 2.7.1 Consommation mémoire

L'interface de présentation de l'application SATURNE est très riche, elle est fortement orientée Javascript (menus contextuels, possibilité de trier ou filtrer les listes, etc.) et l'application fonctionne exclusivement avec Internet Explorer, en version 5 ou 6. En même temps, la forte utilisation des fonctionnalités offertes par DHTML<sup>55</sup> et ActiveX empêche, sans une refonte complète de l'application, le portage vers d'autres navigateurs comme Mozilla Firefox ou encore Google Chrome. De plus, des questions de compatibilité pourront se poser avec les versions successives du navigateur de Microsoft.

À l'aide de l'outil de surveillance des ressources système intégré à Windows NT, j'ai procédé à une rapide analyse qui a montré que la consommation mémoire est élevée sur le poste client. Ceci, en raison de la construction des écrans en Javascript : les écrans conservent les données en mémoire vive dans des tableaux Javascript.

L'implémentation de la plupart des constructions d'écrans est donc effectuée coté client en DHTML, et l'application nécessite de ce fait des postes de travail d'une puissance suffisante et disposant de suffisamment de mémoire. Cette implémentation peut ainsi se révéler être une contrainte en fonction du périmètre de déploiement de l'application : les centres de tri et les bureaux de Poste ne disposent pas tous de postes récents offrant des caractéristiques suffisantes pour assurer des temps de réponse optimaux.

Toutefois, personne ne s'en est plaint jusqu'ici. Et les éventuels problèmes de performance côté client risquent de se manifester de moins en moins, puisque les postes de travail des utilisateurs sont upgradés au fil du temps, suivant la fameuse loi de Moore<sup>56</sup>...

### 2.7.2 Consommation réseau

Le volume du code Javascript, modularisé dans des fichiers inclus pour tirer parti du cache navigateur, représente quasiment 1 Mo à transférer sur le réseau.

Cependant, le code est compacté par un outil que j'ai développé à mon initiative<sup>57</sup>, et que j'utilise avant chaque livraison : celui-ci me permet de diviser environ de moitié la taille

---

<sup>55</sup> Cf. partie 4.3.1 : « DHTML : un savant mélange de HTML, CSS et Javascript », page 101

<sup>56</sup> Selon la loi de Moore, le nombre de transistors que contient une puce double tous les deux ans en moyenne. Et à peu de choses près, cette loi pourrait probablement s'appliquer de la même façon à la taille des disques durs ou encore au rapport quantité/prix des barrettes de mémoire vive.

<sup>57</sup> Il s'agit d'une page web dynamique utilisant le composant générique *FileSystemObject* ainsi qu'une multitude d'expressions régulières.

d'un fichier Javascript de la solution SATURNE (ce qui ramène donc à environ 500 Ko la taille totale des fichiers Javascript sur la plateforme de production).

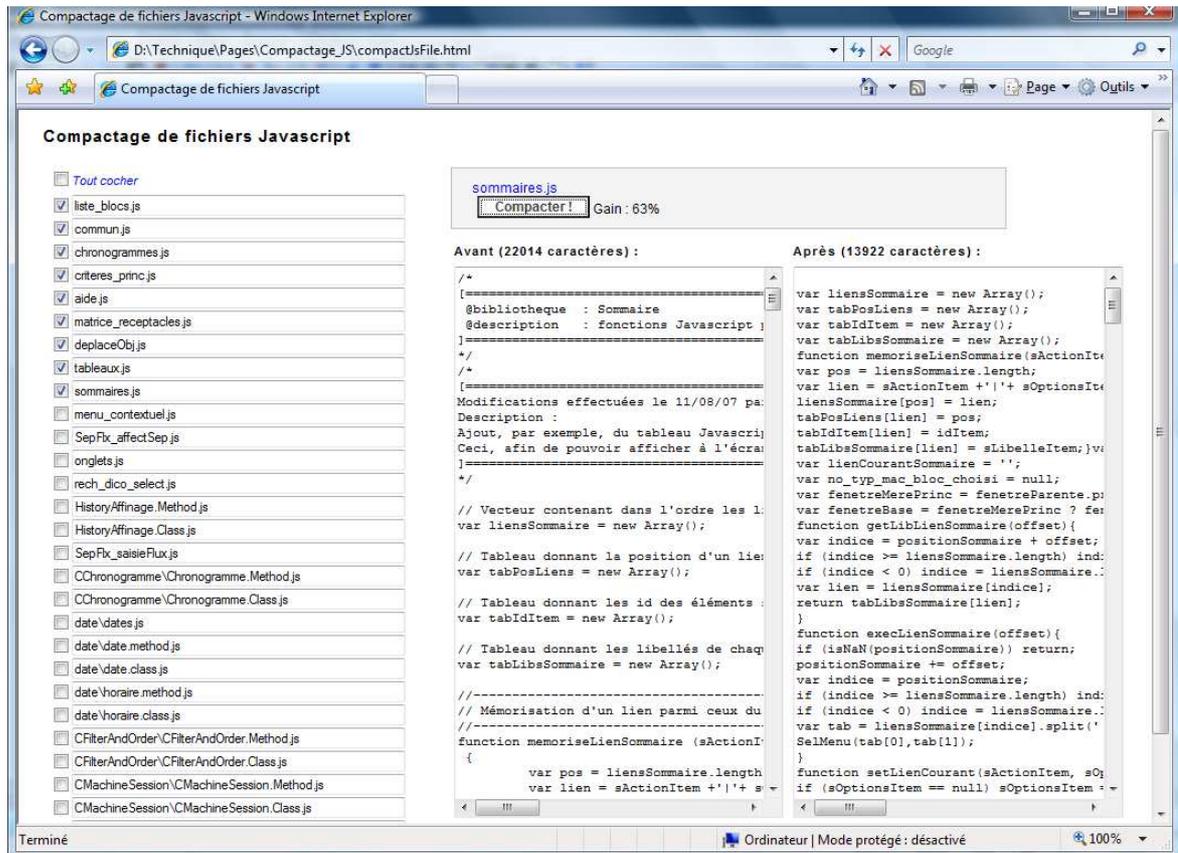


Figure 23 – Outil de compactage des fichiers Javascript de la solution SATURNE

## 2.8 L'architecture de SATURNE : entre théorie et pratique...

### 2.8.1 Limite du découpage théorique de l'application

Avec une application comme SATURNE, toutes les règles théoriques d'une application multi-tiers ne me semblent pas respectées à la lettre. La raison est que le tiers métier « déborde » à la fois sur la couche du client et sur celle du serveur de données. En effet :

- En ce qui concerne le client, certaines règles de gestion sont assurées par le Javascript (comme par exemple le contrôle de la validité des dates saisies, ou encore la construction dynamique du tableau de blocs stats ou des chronogrammes côté client).

***Inconvénient :** Si un autre type de client se connecte un jour au serveur web (ex. application PDA), alors il ne sera pas assuré que les données envoyées en paramètres pour les requêtes auprès du SGBD soient valides. En d'autres termes, le contrôle d'erreurs ne devrait être assuré, dans l'idéal, que par le serveur web (bien que celui-ci en assure de toute façon un minimum à ce jour).*

Pour nuancer cet aspect, précisons toutefois que le code Javascript, au même titre que tous les autres fichiers web, est automatiquement téléchargé par le navigateur client dès qu'une modification du fichier '.js' est détectée. Alors ne pourrait-on pas dire que d'un point de vue purement physique, la partie du traitement assurée par le Javascript réside sur le serveur plus que sur le client, étant donné que ce dernier ne se contente que d'en faire une copie ?

- En ce qui concerne le serveur de données, certaines règles de gestion sont assurées par les procédures stockées, comme par exemple les différentes requêtes SQL dont l'application a besoin, ou encore les règles de gestion à l'import des données (exemple : pour les machines de tri MTP (Machines de tri préparatoire), certains champs dans la table des blocs statistiques sont passés à NULL).

***Inconvénient :** Dans la pratique, le SGBD n'est pas instanciable à ce jour (contrairement au serveur web). Aussi, pour assurer les meilleures performances, on ne devrait pas lui demander autre chose que de stocker et de restituer des données. En d'autres termes, une partie du traitement actuellement assurée par les procédures stockées de SATURNE devrait être déportée dans un composant métier du serveur web<sup>58</sup>.*

---

<sup>58</sup> Cela a justement été fait en partie à l'occasion de la phase d'optimisation de SATURNE : cf. page 53, §2.6.3.

*En même temps, comme il a été montré dans la partie 2.4 page 50 (« Perspective logicielle de l'architecture de SATURNE »), il n'est pas si aisé de libérer du traitement sur le tiers données dans l'état actuel de l'application SATURNE.*

### **2.8.2 Limites de la problématique de déport des traitements SGBD**

La réflexion sur la limitation du rôle du SGBD n'est pas simple : à partir de quel moment considère-t-on que les requêtes SQL ne comportent aucun traitement ?

Certes, on pourrait proposer de retirer les agrégats tels que les GROUP BY, SUM, ainsi que les ORDER BY afin d'assurer respectivement le groupement, le calcul et le tri des données dans un composant métier hébergé sur les différentes instances du serveur web.

Mais cette hypothèse implique aussi d'envoyer sur le réseau beaucoup plus de données : le réseau deviendra-t-il alors le goulet d'étranglement du système ? Pour le savoir, il est toujours envisageable de faire des calculs théoriques, mais le mieux resterait à mon avis de se contenter de la pratique en réalisant un prototype puis en observant le résultat avec un outil d'analyse de l'occupation des ressources, afin de comparer les performances avec la méthode actuelle de répartition des rôles dans les couches client-serveur (cette réflexion n'est pas encore mise à l'ordre du jour à LA POSTE).

En revanche, n'oublions pas une chose : les SGBD n'ont pas été écrits seulement pour stocker des données, mais aussi pour effectuer certaines tâches (jointures, unions...). Et ces tâches, ils les exécutent avec une efficacité certaine.

La conclusion est simple : dans les modules métiers, il ne me semble pas légitime « d'imiter » les fonctions de base d'un SGBD au prétexte de vouloir le soulager.

### **2.8.3 Limites du déport des traitements métier**

Le métier ne doit en aucun cas préjuger de la partie mise en forme pour ne pas devenir inutilisable. (Par exemple, les données que le composant métier envoie au serveur web ne doivent en aucun cas être pré-formatées en HTML).

J'y ai particulièrement veillé dans le développement des composants métier de SATURNE.

### Chapitre 3. SATURNE FACE AUX NOUVEAUX BESOINS DE MONTÉE EN CHARGE

Dans le monde de l'informatique, les néologismes sont à la fête. Alors parlons de la « scalabilité<sup>59</sup> » de l'application SATURNE : dans ce chapitre, ce mot nous sera utile, car il évitera d'employer une expression trop longue, comme « capacité du système à résister à la montée en charge ».



Avant d'aller plus loin, il est essentiel de mettre au jour la raison pour laquelle SATURNE et SNT n'ont pas le droit à l'erreur en ce qui concerne leur résistance à la montée en charge : si tel n'est pas le cas pour l'une ou l'autre application, alors la file d'attente de messages les reliant entre elles<sup>60</sup> risquent un engorgement lors des pics d'utilisation du système d'intégration de fichiers<sup>61</sup>. Un tel problème

d'engorgement pourrait avoir deux conséquences possibles : soit une perte irrémédiable de données dans le cas d'un véritable engorgement, soit un délai non garanti pour le traitement des données dans le cas d'une attente d'intégration qui s'éternise (en d'autres termes, il faudrait attendre les heures creuses pour que soient réalisées les intégrations en attente, sans même être certain que cela suffirait).

Une question cruciale est donc d'assurer qu'une application massivement utilisée, comme SATURNE, est capable de « tenir » la charge prévue. Mais le phénomène de montée en charge s'observe souvent à différents niveaux. Pour SATURNE, j'ai pu identifier deux besoins distincts : d'abord l'intégration des données, ensuite leur consultation. C'est de ce dernier besoin que traitera le présent chapitre<sup>62</sup>. En l'occurrence, il s'agit que l'application SATURNE, dédiée à la consultation des statistiques, accueille plusieurs milliers d'utilisateurs par jour, et avec un strict impératif d'infailibilité étant donné que cette application est l'unique observatoire des données du tri postal.

---

<sup>59</sup> Vient de l'anglais « scalability », ce qui pourrait se traduire par le mot français « extensibilité ».

<sup>60</sup> LA POSTE a choisi MSMQ pour gérer les files d'attentes de messages, à savoir la solution Microsoft.

<sup>61</sup> En l'occurrence, il s'agit de l'outil d'EAI « IBM WTX » (Websphere Transformation eXtender).

<sup>62</sup> L'intégration des données n'est pas de mon ressort, mais celui de mon collègue, Christophe DESSAIX.

## 3.1 Cadrage de notre étude d'optimisation

### 3.1.1 Évolution de la cible de performance

En raison du CQC, les exigences et contraintes sur le système d'informations vont, dans les années à venir, changer significativement sur les plans :

- Techniques : montée en charge prévue lors de l'ouverture des nouvelles plaques de distribution (problématique de consultation de données) et la mise en service des nouveaux équipements (problématique d'intégration de données) ;
- Fonctionnels : nouvelles fonctionnalités et interfaçage à prévoir avec d'autres systèmes du SI LA POSTE (les échanges seront au cœur des futures évolutions de GALAXIE).

Dans cette perspective, il est inévitable de remettre en cause la capacité actuelle de SATURNE à résister à la montée en charge, en anticipant dès maintenant des besoins encore plus exigeants en termes de performance.

### 3.1.2 Baseline et nouvelles exigences

#### 3.1.2.1 Caractéristiques de l'existant

À mon arrivée, les données de base pour SATURNE v1 étaient les suivantes :

- Un seul profil applicatif semblait se dégager de l'ensemble des utilisateurs, indépendamment de leurs établissements de rattachement (CTC, CDIS, DOTC...) ;
- Un pic d'utilisation semblait se produire entre 7h00 et 9h00<sup>63</sup> ;
- Une augmentation possible du nombre d'utilisateurs se justifierait par le passage d'une douzaine de CTC sous IGP (il reste encore 4 autres CTC à passer sous IGP).

Les TMC<sup>64</sup> effectués sur l'application SATURNE ont porté sur un nombre potentiel d'utilisateurs de 2500 utilisateurs par jour pour un maximum de 200 utilisateurs connectés simultanément. Le nombre moyen d'utilisateurs simultanément connectés est de 40.

Par rapport aux résultats obtenus, il est à noter que la plate-forme de tests ne correspondait pas en termes de configuration matérielle à la plate-forme en exploitation. Toutefois, les TMC ont montré qu'il y avait un fort risque de contention au niveau du serveur SGBD de l'application SATURNE.

---

<sup>63</sup> Le module de consultation des audiences que j'ai développé depuis a révélé que c'est bien le cas.

<sup>64</sup> « TMC », dans ce mémoire, signifie « Tests de montée en charge ». (Rien à voir avec la chaîne TV franco-monégasque.)

### 3.1.2.2 Besoin de profilage

L'évolution progressive des besoins fonctionnels sur SATURNE m'a amené à prévoir plusieurs profils utilisateurs, afin de ne pas afficher les mêmes choses selon la nature du profil de la personne connectée à SATURNE<sup>65</sup> :

- Le profil « consultation » : concerne la très grande majorité des utilisateurs ;
- Le profil « CDIS » (centres de distribution, accédant seulement aux flux cumulés) ;
- Le profil « administrateur » (les administrateurs accèdent, en plus du profil 'consultation', à une console d'administration offrant des possibilités telles que la visualisation des audiences ou encore la saisie des formats des différents types de machines de tri) ;
- La « saisie des flux » (peuvent, en plus du profil 'consultation', saisir les informations de flux et de séparations sur l'établissement auquel ils sont affectés)

Le schéma suivant, que j'ai emprunté à la méthode UML appelée « diagramme des cas d'utilisations » met en avant la correspondance fonctionnalités / profils dans SATURNE :

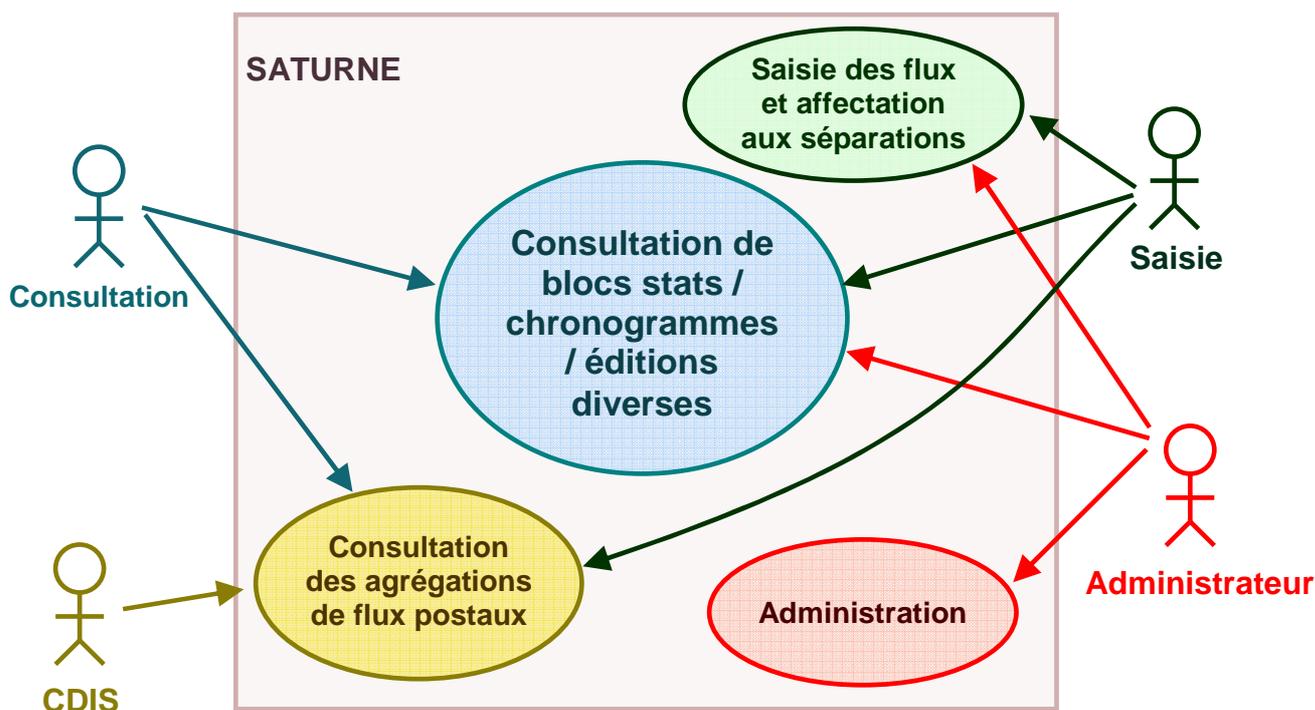


Figure 24 – Imbrications fonctionnelles des différents profils dans SATURNE

<sup>65</sup> On peut parler de *profilage*, ou même employer l'expression anglaise, plus smart : *profiling*.

L'état de fait quantitatif est le suivant :

- Les mesures d'audiences montrent qu'il y a en moyenne 40 utilisateurs connectés simultanément et que le nombre total d'utilisateurs par jour pouvait atteindre 1500 ;
- Un pic d'utilisation semble se produire entre 7h00 et 9h00 ;
- Il faut prévoir une augmentation du nombre d'utilisateurs en raison du passage d'une douzaine de CTC sous IGP (il reste encore 4 autres CTC à passer sous IGP).

### 3.1.2.3 Besoins généraux de la MOA pouvant influencer sur la montée en charge

La MOA a exprimé les exigences suivantes pour la version 2.0 de SATURNE :

- Sur la base d'informations fournies sur le comportement des utilisateurs de PILD (application de LA POSTE mettant à disposition les mêmes informations, propres au CDIS) :
  - Il faut prévoir 2000 nouveaux utilisateurs, ce qui correspond à une moyenne de 1 utilisateur par bureau de distribution sachant qu'on compte aujourd'hui 2000 sites.
  - Le pic d'utilisation est à prévoir entre 8h00 et 9h00 pour un maximum de 1400 connexions utilisateurs.
  - Sur la base de 2 connexions à PILD par jour en saisie, 4000 connexions par jour sont à prévoir.
- SATURNE doit s'adapter à de nouveaux équipements tels les MTB (machine de tri bureau) ou les MTP (Machine de tri préparatoire).

### 3.1.2.4 Précisions de la MOE sur le cadre d'évolution fonctionnelle

Des précisions sur le cadre d'évolution fonctionnelle de l'application SATURNE ont été fournies par la MOE sur la base d'une extrapolation :

- Plusieurs vues des données de production (appelée « éditions ») existent aujourd'hui dans l'application SATURNE. Parmi celles-ci, deux sont particulièrement pertinentes pour les utilisateurs des CDIS. Il s'agit des éditions « Cumul par séparation » et « Cumul par flux de séparation »<sup>66</sup> sur une période correspondant à une journée postale. Par rapport aux paramètres d'affinage des requêtes de recherche de blocs stats, une recherche par « petit courrier » et « grand courrier » pourrait potentiellement intéresser les utilisateurs des CDIS.
- Les utilisateurs des CDIS devront porter une attention particulière aux libellés des flux de séparation, puisque ces derniers sont actuellement saisis manuellement lors

---

<sup>66</sup> Ces éditions répondent au type de question : « Combien de plis ont été acheminés vers telle direction ? » et prennent en des paramètres comme notamment le lot en sortie du courrier.

de l'établissement des plans de séparations et ne suivent donc pas nécessairement un formalisme précis. Ceci devrait toutefois être temporaire, puisque la normalisation de ces libellés sera réalisée par leur intégration prochaine dans l'application SPOT.

D'autres précisions ont été apportées par le responsable de l'application Pildi, assurant le pilotage des CDIS. L'utilisation de SATURNE par les utilisateurs des CDIS devrait suivre le schéma suivant :

- Avec une forte probabilité : un pic entre 8h30 et 9h30 alors que le tri est terminé et que les tournées sont complétées. Les utilisateurs CDIS vont utiliser l'application SATURNE comme aide pour saisir les données, soit les volumes constatés, dans l'application PILDI ;
- Avec une probabilité moyenne : un pic entre 16h30 et 17h30 alors que les responsables des CDIS consultent l'application SATURNE pour connaître le volume approximatif des flux qui vont arriver le lendemain matin. Il s'agit d'une démarche de prévision de l'organisation des équipes de travail ;
- Avec une faible probabilité : un pic éventuel entre 6h00 et 6h30 alors que les responsables des CDIS consultent l'application SATURNE pour connaître le volume de chacun des flux qui vont arriver à leur centre de distribution. Il s'agit simplement d'une démarche de sécurisation des équipes par rapport à la force de travail qui devra être déployée.

#### 3.1.2.5 Caractérisation d'un niveau de scalabilité

Au regard de ces informations, nous avons estimé que la charge de l'application SATURNE pouvait être décrite comme suit :

- Un nombre total approximatif de 2800 utilisateurs tous établissements confondus (CTC, CDIS, DOTC, etc.), soit 2520 utilisateurs au total pour les établissements CTC et CDIS.
- Plusieurs créneaux de forte utilisation sont à prévoir : 6h00 à 6h30, 7h00 à 9h30 et 16h30 à 17h30.
- Il faut considérer 1400 nouveaux utilisateurs dans ces créneaux à forte utilisation de l'application.
- L'activité habituelle d'un utilisateur sur SATURNE dure en moyenne 5 minutes.

Le modèle suivant est une estimation de la charge à venir pour l'application SATURNE. Elle est faite sur la base des hypothèses suivantes :

- La population des utilisateurs des CDIS sera d'environ 2000 utilisateurs et la population des utilisateurs des CTC est d'environ 520 utilisateurs ;

- Le nombre d'utilisateurs des CTC connectés simultanément à l'application SATURNE est de 40 utilisateurs en moyenne ;
- Le nombre total de connexions à l'application SATURNE provenant des utilisateurs des CTC est d'environ 1500 et celle des utilisateurs des CDIS est estimée à 4000 (soit en moyenne deux connexions par utilisateurs au cours d'une journée) ;
- Pour calculer les moyennes de connexions simultanées, on considère un créneau de 3h30 d'utilisation de l'application SATURNE pour les utilisateurs des CTC et de 6h30 pour les utilisateurs des CDIS.

En considérant les créneaux probables de forte utilisation de l'application SATURNE, on peut répartir les utilisateurs de chacune des populations tout en respectant les hypothèses selon le modèle empirique suivant :

Créneaux	CDIS	CTC	Global
6h00-6h30	250	0	250
6h30-7h00	100	42	142
7h00-7h30	100	114	214
7h30-8h00	250	343	593
8h00-8h30	500	510	1010
8h30-9h00	1400	343	1743
9h00-9h30	500	114	614
9H30-10h00	250	42	292
10h00-10h30	100	0	100
10h30-11h00	100		100
11h00-11h30	0		0
11h30-12h00			
12h00-12h30			
12h30-13h00			
13h00-13h30			
13h30-14h00			
14h00-14h30			
14h30-15h00			
15h00-15h30			
15h30-16h00	0		0
16h00-16h30	100		100
16h30-17h00	250		250
17h00-17h30	100		100
17h30-18h00	0		0
<b>Total de connexions</b>	4000	1508	5508
<b>Moyenne</b>	51	36	71

Tableau 3 – Répartition de la montée en charge prévisionnelle dans SATURNE

Il s'en dégage une courbe visible sur la page suivante (la violette est celle qui totalise les différents profils).

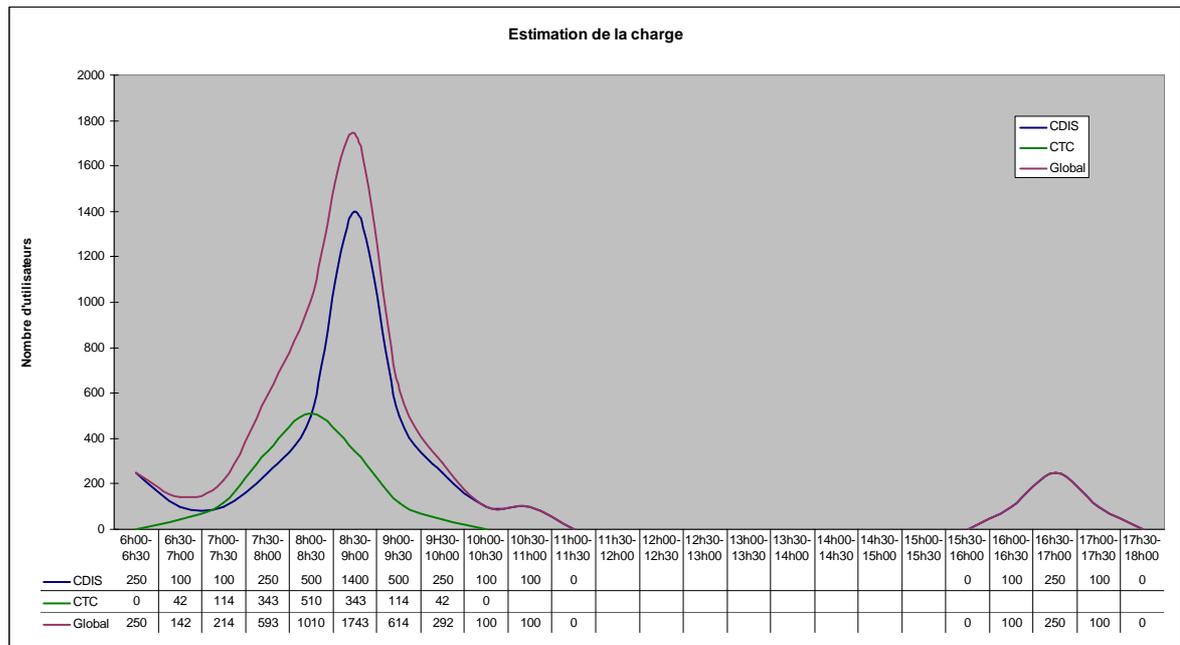


Figure 25 – Répartition de la montée en charge prévisionnelle dans SATURNE

On observe que le pic le plus important se trouve entre 8h30 et 9h00, où environ 1750 utilisateurs sont connectés simultanément : soit environ 70% de la population totale des utilisateurs des CTC et des CDIS.

Il sera important de tenir compte de cette estimation de la charge dans l'évaluation des scénarios proposés dans cette étude. On tiendra compte de la modélisation de la charge pour identifier les meilleurs axes de scalabilité.

### 3.2 Démarche adoptée pour notre chantier d'optimisations

Nous avons donc pu, en collectant statistiques réelles et statistiques prévisionnelles, caractériser un « niveau de scalabilité » à atteindre pour l'application SATURNE.

Exposons à présent les différentes solutions permettant de satisfaire ce besoin de scalabilité. Avant de commencer, précisons que notre chantier d'optimisation devra s'effectuer dans un périmètre iso-technologique<sup>67</sup>.

La méthode utilisée consiste à boucler sur le cycle suivant tant que la scalabilité n'est pas acceptable :

1. Tests de montée en charge (pour connaître la scalabilité de SATURNE à un temps  $t$ ) ;
2. Revue de code (pour connaître les pistes d'optimisation du code) ;
3. Et, par voie de conséquence, la remise en question de SATURNE à tous niveaux.

Chacune de ces méthodes complémentaires sera exposée brièvement ci-après (parties 3.2.1, 3.2.2 et 3.2.3), avant de revenir dessus de manière plus approfondie (parties 3.3, 3.4 et 3.7). Enfin, la partie 3.9 est plus personnelle, car elle expliquera les solutions que j'ai trouvées pour chacun des problèmes techniques rencontrés dans une démarche complémentaires d'adaptation de SATURNE à la montée en charge.

---

<sup>67</sup> C'est-à-dire sans remettre en cause les choix technologiques ayant été effectués jusque là dans SATURNE.

### **3.2.1 Les TMC (Tests de montée en charge)**

Les TMC sont menés par une équipe distincte. Toutefois, une collaboration est nécessaire : nous les aidons à installer les applications sur leurs serveurs de développement. En contrepartie, l'équipe TMC rédige un rapport complet, visant d'une part à identifier les goulets d'étranglement de l'application, et d'autre part à proposer des optimisations pour pallier les limites du système (par exemple, éviter l'utilisation de tables temporaires sur les procédures stockées fortement soumises à la montée en charge).

### **3.2.2 La revue de code**

Une revue de code a été commandée par la Direction Informatique du Courrier, dans le souci de maintenir les objectifs du Cap Qualité Courrier. Cela a permis de dégager une série de recommandations portant sur la façon d'optimiser le code source, dans l'objectif d'une meilleure performance générale de l'application SATURNE, et notamment une amélioration de sa scalabilité.

### **3.2.3 Repenser l'architecture actuelle de SATURNE**

Il s'agit d'identifier différents scénarios sous différents angles (architecture fonctionnelle, architecture logicielle, architecture technique...) et permettant par leur assemblage (solutions) d'atteindre des objectifs de scalabilité complémentaires aux optimisations préconisées par les TMC et la revue de code. Cette étude se voue à cerner ce que l'on appelle en jargon d'ingénieur des « axes de scalabilité ». Toutefois, nous ne sommes pas allés jusqu'à l'identification précise des niveaux de performance cible, étant donné que la baseline était insuffisamment caractérisée.

Aussi, nous avons procédé en équipe à une analyse visant à cerner les axes de scalabilité dans SATURNE, afin de savoir à quel niveau d'exigence répondre. On appelle « facteur de scalabilité » toute solution participant à cet objectif.

Pour cela, une étude technique nous a été nécessaire afin d'identifier les meilleurs leviers de scalabilité en procédant par couche, comme le décrit le schéma suivant.

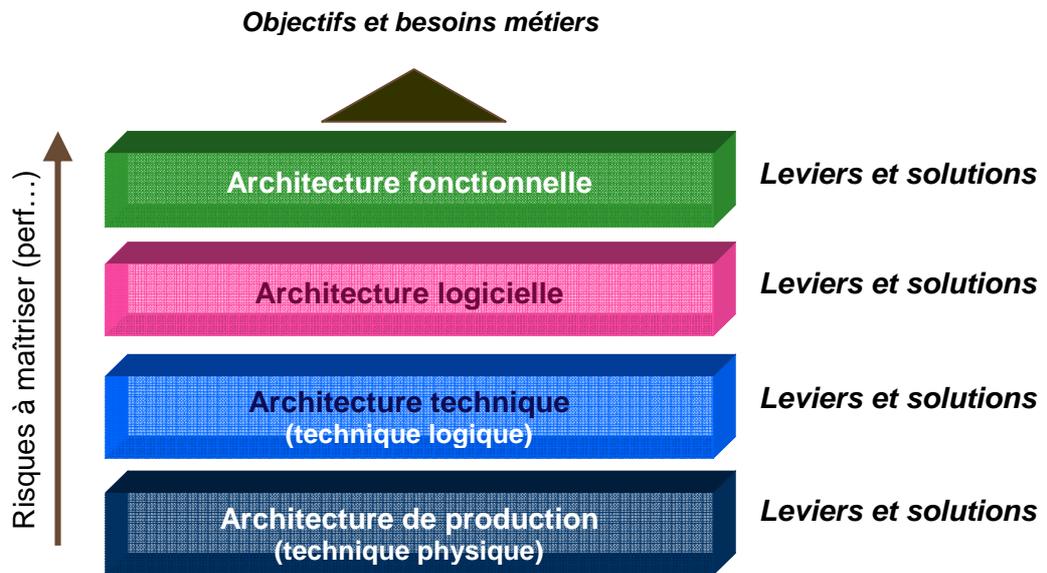


Figure 26 – Schéma des différentes facettes architecturales d'un système

Les scénarios identifiés sont catégorisés par perspective d'architecture :

- A : Évolutions de nature fonctionnelle
- B : Évolutions de nature logicielle
- C : Évolutions de nature technique logique
- D : Évolutions de nature technique physique

**Remarque :**

*Les architectures logicielles et fonctionnelles sont toutes les deux des architectures dites « applicatives » : l'architecture logicielle est vue comme une instanciation de l'architecture fonctionnelle dans un environnement technique et d'exploitation donné.*

*Enfin, la différence entre une architecture logicielle et architecture fonctionnelle n'est-elle pas similaire à celle qui sépare une architecture technique physique (de production) et une architecture technique logique : l'architecture logicielle se situant à un niveau software et l'architecture technique physique à un niveau hardware.*

### 3.3 Bilan de scalabilité : les TMC

Des TMC (tests de montée en charge) ont été menés par l'équipe de métrologie de la Direction Technique, avant et après la mise en place de nos optimisations sur l'application SATURNE. Il s'agit d'une mise à l'épreuve de la capacité de l'application à supporter un grand nombre de clients simultanés.

Les TMC ont donné lieu à la rédaction d'un dossier de synthèse indiquant par exemple le seuil de résistance du système en termes de nombre de connexions simultanées, ou encore en identifiant le goulet d'étranglement (couche métier ? couche SGBD ?). Mais, plus intéressant, les TMC donnent aussi une série de préconisations techniques (complémentaires aux nôtres, et plus générales) sur la façon d'optimiser chaque couche de l'application.

#### 3.3.1 Résultats des TMC sur SATURNE

En l'occurrence, pour SATURNE, ces tests ont porté sur un nombre potentiel d'utilisateurs de 2500 par jour, pour un maximum de 200 utilisateurs connectés simultanément, et une moyenne d'environ 40 utilisateurs connectés simultanément.

Les premiers TMC menés sur SATURNE ont montré que l'application pouvait répondre à un maximum de 400 utilisateurs connectés simultanément. Par rapport aux résultats obtenus, il est à noter que la plate-forme de tests ne correspondait pas en termes de configuration matérielle à la plate-forme en exploitation. Toutefois, les TMC ont montré qu'il y avait un fort risque de contention au niveau du serveur SGBD de l'application SATURNE.

Ce constat a servi de point de départ pour notre étude de scalabilité, ce qui est d'autant nécessaire que l'ouverture vers les utilisateurs des CDIS (centres de distribution) était déjà envisagée.

##### 3.3.1.1 Pistes d'optimisation du module SGBD énoncées dans le rapport TMC

L'équipe des TMC (Tests de montée en charge) a rapporté que dans Sybase, un certain nombre de procédures stockées dans SATURNE comportaient des défauts d'optimisation. La manière dont ces procédures ont été développées allait en effet à l'encontre des recommandations suivantes :

Recommandation	Raison	Solution de contournement
Éviter les tables temporaires	Cela met à contribution l'espace mémoire	Utiliser des requêtes plus complexes
WITH RECOMPILE	Temps de traitement inutile	Suppression de cette option
SET NOCOUNT ON	Temps de traitement inutile	Suppression de cette option

Éviter les OR dans les paramètres de recherche	Les OR sont coûteux	Dédoublage de la requête d'extraction des données en fonction des paramètres de recherche
Éviter le SQL dynamique quand c'est possible	Le code précompilé s'exécute plus vite	Passer le code SQL dans des procédures stockées
Dans les requêtes de consultation, mettre le niveau d'isolation à READ UNCOMMITTED	Sinon, un verrou est posé même pour une simple lecture.	Mettre un SET TRANSACTION ISOLATION LEVEL 0 au début de chaque proc de consultation (et SET TRANSACTION ISOLATION LEVEL 1 à la fin)
Préférer le 'IF EXISTS' au 'IF NOT EXISTS'	Bizarrement, la version 12.0 de Sybase optimise l'un, mais pas l'autre.	Mettre le tout en variable booléenne, puis prendre l'inverse de sa valeur.

Tableau 4 – Pistes d'optimisation du module SGBD

### 3.3.1.2 Bilan des optimisations du module SGBD suite aux TMC

Il s'avère que le respect de ces recommandations a été un véritable facteur de scalabilité pour la partie SGBD des traitements du projet SATURNE.

Pour atteindre une meilleure scalabilité, nous avons trouvé les astuces suivantes :

- Retrait des verrous en lecture (via l'instruction 'SET TRANSACTION ISOLATION LEVEL 0') ;
- Forçage de certains plans d'exécution (via l'instruction 'SET FORCEPLAN ON') ;
- Découpage en sous-procédures stockées (via l'instruction 'EXECUTE <proc>') ;
- Forçage de l'utilisation de certains index (via l'instruction ' INDEX <nom index>').

### 3.3.2 L'utilité limitée des TMC...

Dans la pratique, je n'ai que rarement bénéficié du fruit des TMC, en raison du manque de réactivité des personnes qui en ont la charge : souvent, les rapports nous sont livrés trop tard<sup>68</sup> obligeant à appliquer les bénéfices de ces tests au mieux dans une version ultérieure de SATURNE.

---

<sup>68</sup> Voire même après la date de mise en production... ce qui est un comble, sachant que le plus souvent, trois mois s'écoulent entre la livraison d'une application et sa date de mise en production

### 3.4 Revue de code

La revue de code a été réalisée sur SATURNE par une société d'audit, au même titre que toutes les autres applications de GALAXIE dans le cadre du CQC. Cette revue a été menée en deux temps : d'une part une première passe rapide, qui a permis de dégager des recommandations globales à partir de la manière générale dont l'application était codée. D'autre part, une analyse plus détaillée a abouti sur une série de recommandations précises : celles-ci sont visibles dans les annexes de ce mémoire, dont une synthèse est présentée page 95, en partie 3.9.4 de ce document.

#### 3.4.1.1 Bilan général pour la base de données

L'analyse du code a mis en évidence les points d'attention suivants :

- Utilisation fréquente de tables temporaires ;

*La mise en œuvre de tables temporaires utilise de nombreuses opérations d'entrée-sortie avec la mémoire et avec le disque dur le cas échéant. Leur utilisation doit donc être limitée et il vaut mieux privilégier des requêtes plus complexes.*

- Deux requêtes dynamiques (or le code précompilé s'exécute plus vite) ;
- Usage fréquent de l'instruction FORCEPLAN, qui permet de forcer l'ordre dans lequel le plan de lecture des différentes tables d'une requête SQL est lu. En cas d'évolution de la base de données, l'adhérence du code métier au plan d'indexation peut rendre certaines tâches de maintenance complexe et peuvent dégrader les performances.

#### 3.4.1.2 Bilan général pour le web

La première phase de la revue de code a permis de dresser la description suivante de l'état général de SATURNE concernant la partie web.

L'application est orientée Javascript. Lors de la démonstration, les temps de réponses n'ont pas été jugés satisfaisants sur la première page. Une analyse a permis de montrer que deux facteurs étaient à prendre en compte:

- Utilisation non recommandée de tables temporaires dans la procédure stockée *ps\_liste\_bloc\_stat* ;
- Utilisation d'événements DHTML au niveau de chaque ligne de tableau, au lieu d'une seule fois au niveau du tableau lui-même (ce qui allonge le temps de traitement côté client, ainsi que l'allocation de mémoire vive sur le poste client).

Ces remarques ont été prises en compte et les temps de réponse sont désormais meilleurs.

*Remarque* : La revue de code a pris en compte le fait que j'utilise un outil de compactage du code Javascript avant livraison pour intégration. Je n'ai donc pas eu à pâtir de remarques du type « trop de commentaires dans le code source ».

### **3.5 Préconisations déjà respectées avant l'optimisation de SATURNE**

Il est à noter que SATURNE respecte déjà certaines préconisations, lui permettant de rendre acceptable sa scalabilité. On citera en exemple l'utilisation du pooling de connexion ODBC pour Sybase. Voyons exactement de quoi il en retourne.

La création de la connexion de base de données prend fréquemment presque autant de temps que l'accès aux données. Dans la pratique, les applications qui ne mettent pas en œuvre un « pooling » de connexions sont pratiquement inexploitable à cause de leur faible performance et leur limitation de montée en charge. Cela vaut particulièrement pour des applications web, pour lesquelles la nature de l'environnement aggrave le problème et rend la charge du serveur pratiquement impossible à prévoir. Sans gestion efficace des connexions de base de données, les ressources du serveur sont trop rapidement consommées, ce qui nuit à la disponibilité des applications. Le principe de fonctionnement du pooling de connexions est relativement simple. Voici deux scénarios qui décrivent sommairement ce qui se passe lorsqu'un utilisateur accède à la base de données.

#### **■ Connexion sans pooling :**

Une connexion sur la base de données est créée quand l'utilisateur ouvre la connexion. Cette connexion lui reste attribuée pendant tout le traitement. Lorsqu'elle est fermée, la connexion à la base de données est alors détruite.

#### **■ Connexion avec pooling :**

Une connexion sur la base de données est créée quand l'utilisateur ouvre la connexion. Cette connexion lui reste attribuée pendant tout le traitement. Quand il ferme la connexion, la connexion est placée dans un pool de connexion et elle est mise à disposition pour une autre utilisation. Elle est conservée dans ce pool pendant un certain temps puis elle est détruite. Si un utilisateur ouvre une même connexion et qu'il y en a dans le pooling de connexions, une d'entre elles lui sera attribuée (ce qui occasionne un gain de temps non négligeable).

### 3.6 Analyse des faiblesses du système existant

Au sein de l'équipe GALAXIE, nous avons pu déterminer les facteurs de scalabilité qui devaient être mis en œuvre afin d'atteindre nos axes de scalabilité :

- Optimiser les procédures stockées et la taille des pages ;
- Mettre en place des composants métiers, et migrer du code SGBD (autrement dit du code SQL) dans ce nouveau tiers traitement, ce qui allégera la base de données, actuel goulet d'étranglement de l'application (cf. 2.6.3 « Développement d'un composant métier », page 53) ;
- Mettre en place une gestion de profils pour s'adapter à différentes classes d'utilisateurs, donc différentes façons d'accéder aux statistiques de production (cf. 3.1.2.2 « Besoin de profilage », page 61) ;
- Répartir la charge en multiplexant les données vers plusieurs instances de serveur web, et en séparant physiquement la couche du serveur web et celle des composants ;
- Re-paramétrer le serveur physique SGBD (notamment pour que la mémoire vive et le processeur soient exploités en totalité) ce qui ne s'est pas avéré être le cas.

### 3.7 Scénarios d'amélioration de la scalabilité dans SATURNE

Seul ou en équipe, de nombreuses idées d'amélioration de SATURNE ont germé. Chacune d'entre elles ne concerne pas la même couche de l'application : j'ai jugé pertinent de les répartir par types d'architecture (cf. synthèse page 66, partie 3.2 « Démarche adoptée »).

Ainsi, les parties suivantes présentent les différents scénarios que nous avons envisagés, répartis en quatre couches : *fonctionnelle, logicielle, technique logique et technique physique* (les solutions pour améliorer la scalabilité de l'application SATURNE peuvent être des combinaisons des différents scénarios envisagés).

#### 3.7.1 Évolution de l'architecture fonctionnelle

Les scénarios d'évolution au niveau de l'architecture fonctionnelle qui sont considérés pour l'application SATURNE se concentrent sur la pertinence des fonctionnalités actuellement proposées au regard des besoins des nouveaux utilisateurs.

L'ouverture vers de nouveaux établissements comme les centres de distribution (CDIS) apporte une population d'utilisateurs différents ayant des besoins précis en matière de consultation des données de production (usage différent de SATURNE).

Par ailleurs, la population actuelle des utilisateurs de l'application SATURNE comporte un certain nombre d'utilisateurs provenant des DOTC, qui ont besoin des données de production à des fins d'analyse sur le moyen et le long terme.

La population d'utilisateurs de l'application SATURNE est donc diversifiée et on identifie facilement deux grands objectifs par rapport à la mise à disposition des données de production via l'application SATURNE :

- « Data mining » (opérations de surveillance journalières) : implique une consultation de données récentes et disponible rapidement après leur remontée dans SATURNE ;
- Pilotage (opérations d'analyse et de prise de décision) : implique une consultation de rapports statistiques portant sur une longue période, et dont la fréquence est souvent mensuelle.

Face à ces constats, les scénarios à envisager sont présentés dans les paragraphes qui suivent.

#### 3.7.1.1 Scénario A1 : « Dissociation des grandes fonctions »

Les deux grandes fonctions identifiées étant le data mining et le pilotage, deux fonctions qui ont en commun les données mais qui sont différentes dans leurs modes de restitution (le pilotage reposant sur l'historisation et le data mining sur la consolidation des données), nous avons pensé à revoir l'architecture fonctionnelle de l'application SATURNE afin de dissocier ces fonctions.

3.7.1.2 Scénario A2 : « Prise en compte des préférences utilisateurs »

Les utilisateurs qui consultent des données à des fins de suivi au jour le jour du bon fonctionnement des machines de tri (volumes par centre de tri, nombres de bourrages ou autres) exécuteront nécessairement les mêmes requêtes de récupération de données tous les jours, et le plus souvent pour la dernière journée postale (de 6h00 la veille à la même heure le jour-même). Il est donc envisageable de sauvegarder par utilisateurs, les requêtes qu'ils effectuent de manière récurrente, afin de pré-générer dans les heures creuses les rapports statistiques correspondant à leurs besoins.

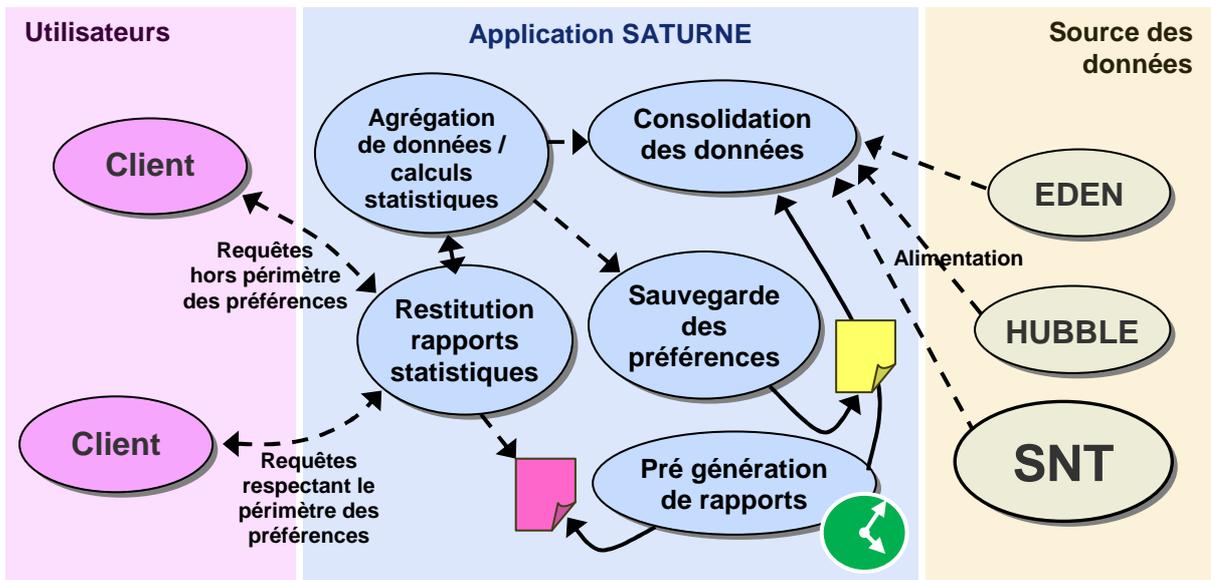


Figure 27 – Architecture fonctionnelle du scénario « Prise en compte des préférences utilisateurs »

3.7.1.3 Scénario A3 : « Création d'une application dédiée CDIS »

Les utilisateurs des CDIS sont concernés essentiellement par les volumes que leurs centres de distribution vont recevoir des centres de tri au début d'une journée. Ce besoin étant simple par rapport à l'interface de l'application SATURNE, qui ne rend pas de manière directe les informations attendues par les utilisateurs des CDIS, il fut envisageable de faire une application complètement dédiée aux utilisateurs CDIS. Cette application n'aurait eu en commun avec l'application SATURNE que les données de production, mais aurait éventuellement pu s'inspirer de la même ergonomie.

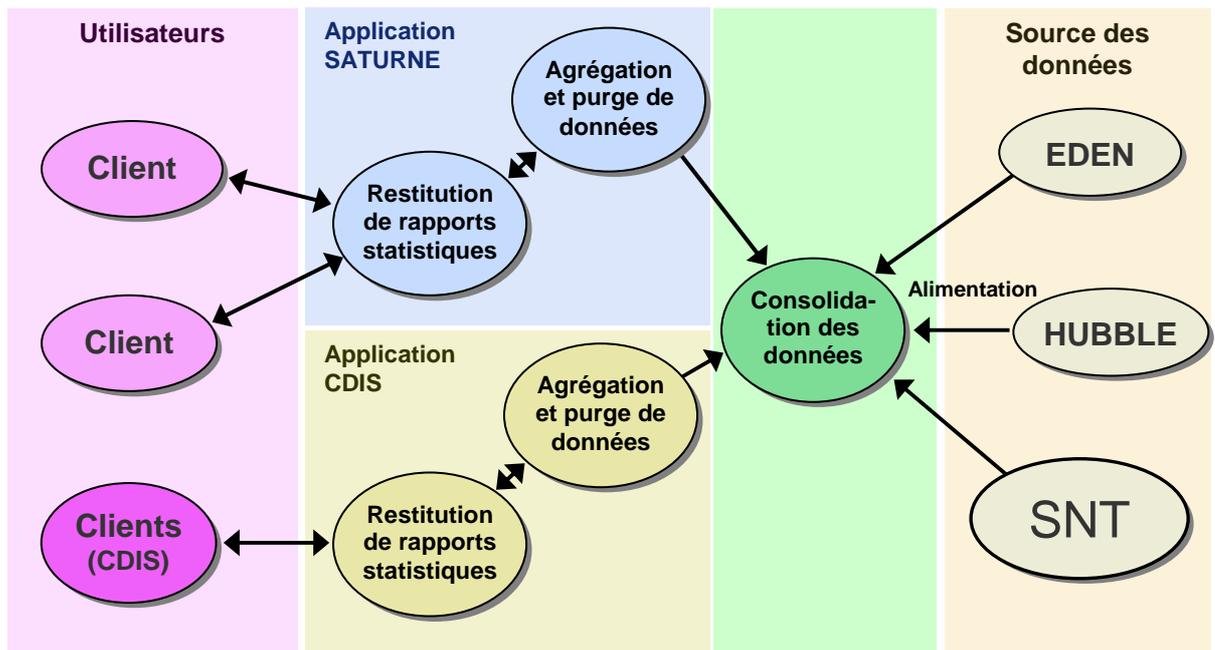


Figure 28 – Architecture fonctionnelle du scénario « Création d'une application CDIS »

### 3.7.1.4 Scénario A4 : « Intégration du profil CDIS »

Dans le même ordre d'idées que le scénario 3, une idée est de créer des profils utilisateurs dans l'application SATURNE afin de répondre au mieux aux besoins des utilisateurs. En fonction du profil de l'utilisateur connecté (déterminé par son établissement de rattachement qui est remonté par Habilinet<sup>69</sup>), les menus et requêtes de consultations seront différents. Ils proposeront à l'utilisateur les fonctions de consultation dont il a directement besoin en termes de pertinence, notamment par rapport à la visualisation des données de production. Cela empêchera les utilisateurs en CDIS d'accéder par curiosité ou par inexpérience à d'autres fonctions que celles qui leur sont utiles, ce qui participe de fait à une meilleure montée en charge puisque le serveur ne sera pas inutilement sollicité.

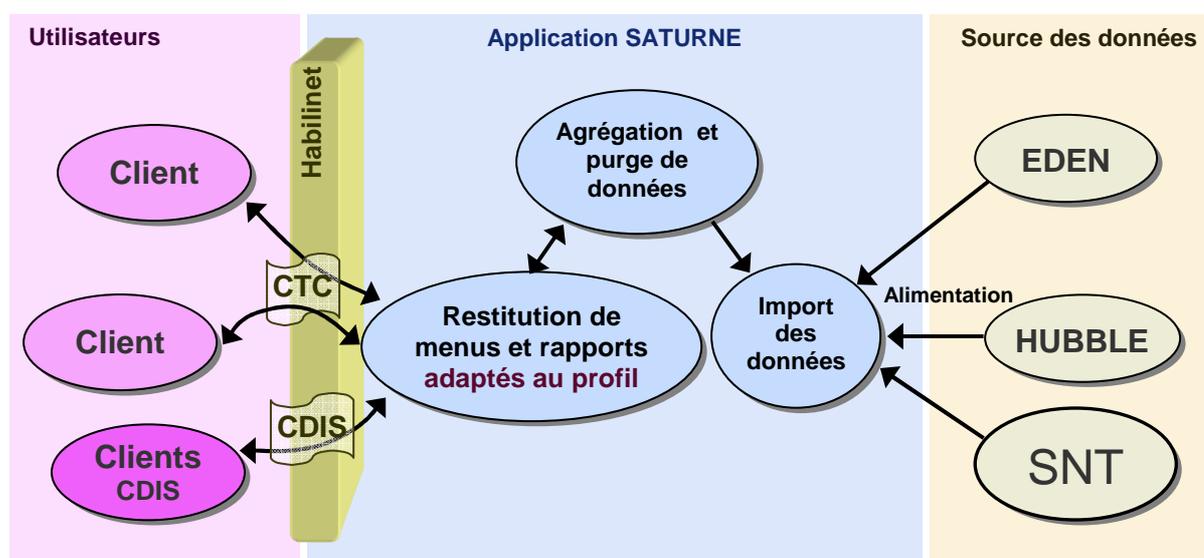


Figure 29 – Architecture fonctionnelle du scénario « Profil CDIS »

### 3.7.1.5 Tableau comparatif des scénarios d'évolution de l'architecture fonctionnelle

Établissons maintenant une comparaison point à point des différents scénarios imaginés pour faire évoluer l'architecture fonctionnelle :

	Scénario A1 « Dissociation des grandes fonctions »	Scénario A2 « Prise en compte des préférences utilisateurs »	Scénario A3 « Création d'une application dédiée CDIS »	Scénario A4 « Intégration du profil CDIS »
Caractéristiques majeures	Apporte le découpage de l'application de manière à offrir aux utilisateurs deux grandes	Apporte des possibilités de pré-génération ciblée des rapports statistiques.	Apporte les fonctionnalités répondant directement et de manière pertinentes	Apporte un moyen de différenciation des utilisateurs afin de répondre au mieux à leurs besoins en

<sup>69</sup> Habilinet est l'application contrôlant les habilitations d'accès aux applications dans l'intranet LA POSTE.

	<p>fonctionnalités : le pilotage et le «data mining» au jour le jour.</p> <p>Apporte un mécanisme de séparation des données : données dédiées aux opérations de surveillance journalières versus les données dédiées aux opérations d'analyse et de prise de décision.</p>		aux besoins des utilisateurs des CDIS.	termes de consultation des données de production.
Principaux avantages	<p>Dissociation des typologies de traitement.</p> <p>Isolation des traitements lourds (décisionnel).</p> <p>Définition de « data mining » très opérationnels pouvant être optimisés.</p>	<p>Les préférences fournissent un bon moyen d'étendre de manière externe le contenu d'une application .</p> <p>Possibilité de pré-générer certains rapports dès la connexion de l'utilisateur.</p> <p>Possibilité de créer un « portail » par typologie d'utilisateur (CTC, CDIS, ...)</p>	<p>Dissocie les besoins CDIS des besoins CTC.</p> <p>Permet de présenter de manière synthétique les informations nécessaires à PILDI par exemple.</p> <p>N'est pas intrusif dans l'existant (partie IHM).</p>	<p>Une unique application utilisateur pour la gestion des résultats de tri.</p> <p>Mutualisation de certaines fonctionnalités SATURNE entre les profils CTC et CDIS.</p>
Principaux inconvénients	<p>Refonte de l'architecture existante.</p> <p>Modification éventuelle du modèle de données .</p> <p>Difficulté à établir fonctionnellement le contenu des « data mining » journaliers.</p> <p>Délai difficile à tenir.</p>	<p>Les pages ASP/javascript doivent pouvoir assurer les fonctionnalités de tri, de filtrage par date, etc.</p> <p>Perte de la remontée dynamique en cas de pré-génération (délai de 5 minutes) des données de production.</p> <p>Intégration de cette fonctionnalité dans la version 1.2 de SATURNE (en production en mars 2006).</p> <p>Ne spécialise pas les</p>	<p>Difficulté d'établir dans la DFB l'expression de besoins autour de la synthèse présentée à l'utilisateur CDIS.</p>	<p>Délai de mise en œuvre et de mise ne production (complexité de l'intégration).</p>

		types d'utilisateurs Nécessite de modifier de manière importante le code existant pour intégrer les aspects de recherche.		
Gains	Optimisation de la partie «data mining» journalier  Précalcul possible pour les données décisionnelles (cubes).  4 / 4	Pré-génération de certains rapports standards  2 / 4	Optimisation de la partie CDIS  Dissociation tiers IHM et traitement de l'architecture existant  3 / 4	Optimisation et spécialisation du profil CDIS  3 / 4
Risques	Refonte de l'application  4 / 4	Mode d'usage de l'application à confirmer  2 / 4	Mise en production d'une nouvelle application  1 / 4	Intégration délicate au sein de l'application  2 / 4
Niveau de charge de travail en réalisation	4 / 4	2 / 4	2 / 4	3 / 4
Acteurs impliqués	MOA MOE DPI	MOE (MOA)	MOE MOA (DPI)	MOE (MOA)

Tableau 5 – Évolution de l'architecture fonctionnelle

### 3.7.2 Évolution de l'architecture logicielle

Les scénarios d'évolution de l'architecture logicielle portent essentiellement sur des leviers d'amélioration de la scalabilité pouvant être mis en œuvre rapidement. Aucun de ces scénarios n'implique de refonte intégrale de l'architecture logicielle. En effet, il s'agit ici d'identifier des scénarios susceptibles d'amener une diminution directe des risques de contention au niveau du serveur SGBD de l'application SATURNE, tout en minimisant les impacts trop conséquents sur les développements déjà réalisés.

Les scénarios envisageables sont donc les suivants.

#### 3.7.2.1 Scénario B1 : « Optimisations de l'implémentation SQL »

Les TMC ont montré que certaines des procédures stockées (par exemple celle permettant de récupérer une liste des blocs stats) utilisées actuellement pour agréger les données, supportent moins bien la montée en charge. L'objectif de ce scénario est de considérer des moyens d'optimiser ces procédures stockées.

### 3.7.2.2 Scénario B2 : « Remontée de traitements vers le tiers traitement »

Dans le même ordre d'idée que le premier scénario, le scénario 2 propose de diminuer les risques de contention sur le serveur SGBD de l'application SATURNE en allégeant cette fois les procédures stockées les moins scalables. Le moyen proposé pour y arriver consiste à déporter une partie des traitements effectués actuellement au sein des procédures stockées, dans des composants coté tiers traitement<sup>70</sup>.

### 3.7.2.3 Scénario B3 : « Pré-génération de la dernière journée postale »

Pour ce scénario, normalement aucune modification n'est à apporter aux procédures stockées. La scalabilité est assurée par l'ajout d'un mécanisme de pré-génération des tableaux de bloc stats<sup>71</sup> sur une période fixe correspondante à une journée postale et selon certains modèles types, par exemple par CTC. Les calculs nécessaires pour pré-générer les tableaux pouvant être effectués en période de basse utilisation de l'application SATURNE, la charge de travail est davantage lissée dans le temps.

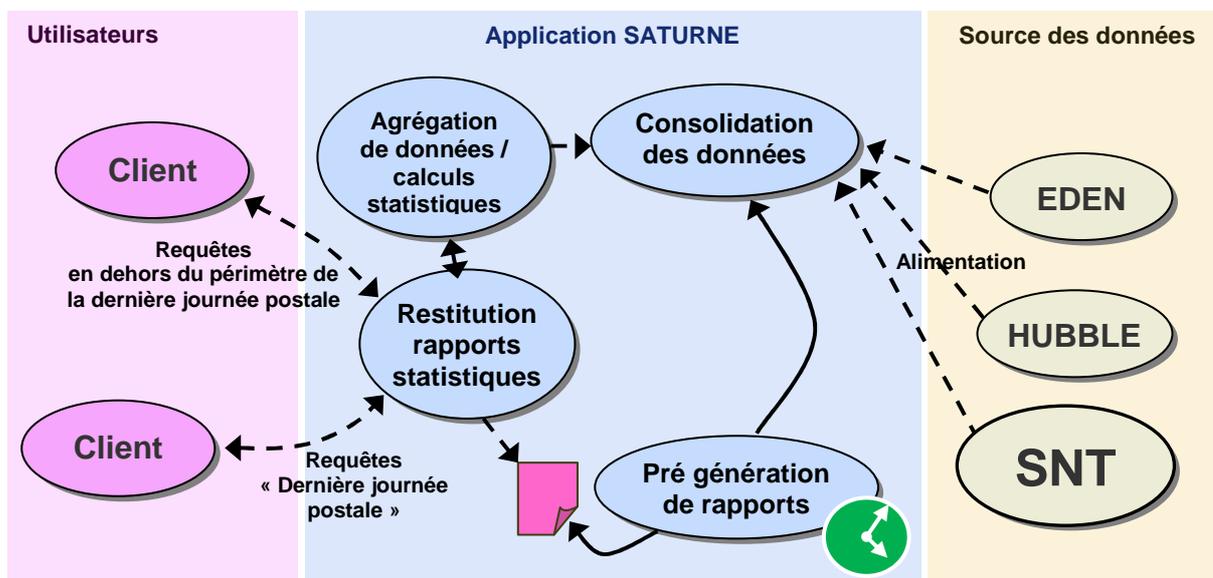


Figure 30 – Architecture fonctionnelle du scénario « Pré-génération de la dernière journée postale »

#### **Recul sur ce scénario :**

En théorie, cette solution peut être séduisante, mais il s'est avéré que nous avons oublié un détail : les heures de début et de fin d'une journée postale varient selon les centres de tri, ce qui alourdit et complexifie la solution technique, voire même l'invalide. De plus, il faut ajouter que nous avons réussi à obtenir un temps d'exécution de la recherche de blocs qui soit « immédiat » en s'adonnant à de simples optimisations SQL !

<sup>70</sup> En partie 2.6.3, page 54, il est question de la réalisation de composants métier.

<sup>71</sup> Un « bloc stats », dans SATURNE, désigne l'ensemble des informations statistiques d'une session de tri.

3.7.2.4 *Tableau comparatif des scénarios d'évolution de l'architecture logicielle*

	<b>Scénario B1</b> <b>« Optimisations de l'implémentation SQL »</b>	<b>Scénario B2</b> <b>« Remontée de traitements vers le tiers traitement »</b>	<b>Scénario B3</b> <b>« Pré-génération de la dernière journée postale »</b>
Caractéristiques majeures	Apporte des moyens d'optimiser les procédures stockées.  Apporte des moyens d'optimiser le schéma de la base de données.	Apporte un moyen d'alléger les procédures stockées	Apporte un mécanisme de pré-génération des rapports statistiques durant la faible charge du serveur SGBDR.
Principaux avantages	Minimisation des impacts sur l'existant (la couche « données » uniquement)  Un seul acteur impliqué	Soulager la base de données des ordres de tri  Un seul acteur impliqué	Pré-calcul de cumuls standards
Principaux inconvénients	Régression possible sur certains cas d'usage	Peu de requêtes sont sujettes à un gain effectif sur ces axes de remontée des fonctions de tri.  Nécessité de modifier deux couches de l'architecture	Les pages ASP/javascript doivent pouvoir assurer les fonctionnalités de tri, de filtrage par date, etc.  Perte de la remontée dynamique (délai de 5 minutes) des données de production.
Gains	Les performances de certaines requêtes peuvent être doublées voire nettement plus. De manière globale les gains de performance peuvent être notables.  3 / 4	Peu de requêtes manipulent suffisamment de données pour que Sybase rencontre des problèmes de tri  1 / 4	Dépend de l'utilisation des cumuls standard actuellement  2 / 4
Risques	Perte de performance sur certains cas d'usage de certaines procédures stockées.  Optimisation technique sur l'hypothèse des usages utilisateurs  1 / 4	Difficulté à remonter les traitements de tri dans les traitements. Adéquation CWAD ?  Deux couches sont touchées  2 / 4	Difficulté de définir ces cas standard dans des délais courts  Nouvelle architecture à mettre en œuvre  3 / 4
Niveau de charge de travail en réalisation	1 / 4 (a nécessité l'expertise de M. DAVY, notre expert Sybase)	2 / 4	3 / 4
Acteurs impliqués	MOE  Expert en optimisations DBA	MOE	MOE  MOA

Tableau 6 – Évolution de l'architecture logicielle

### 3.7.3 Évolution de l'architecture technique logique

Les scénarios d'évolution de l'architecture technique logique portent sur des options de découpage de l'application SATURNE en briques logiques, leurs interactions et l'optimisation des processus sous-jacents.

Les scénarios envisageables sont donc les suivants :

#### 3.7.3.1 Scénario C1 : « Passage complet en architecture 3-tiers »

Dans ce scénario, l'architecture technique logique est remise en cause. Il est proposé en effet de revoir complètement la décomposition de l'application SATURNE de sorte que les composants du côté tiers données n'effectuent plus aucun traitement sur les données. L'objectif est que les composants du tiers données ne soient plus qu'en charge de la remontée des données depuis la base de données où sont consolidées les données reçues de l'application SNT jusqu'aux composants du tiers traitements. Il est envisagé de faire effectuer les traitements de type « filtrage » (par type de machine, par machine, etc.) par les composants du tiers traitement plutôt que par ceux du tiers données. Les composants du tiers données n'auraient donc plus que des requêtes dont l'agrégation est simplifiée.

#### 3.7.3.2 Scénario C2 : « Contrôle des traitements parallèles Sybase »

L'optimisateur de Sybase exécute les procédures stockées selon un découpage en processus parallèles qui est établi en fonction d'un certain nombre de règles fournies par défaut. Le scénario 2 propose de modifier ces règles afin de gérer plus finement le découpage en traitements parallèles au sein du moteur Sybase.

#### 3.7.3.3 Scénario C3 : « Base d'historisation / Base à J+1 »

Ce scénario propose la création de deux bases de données qui viennent s'ajouter à celle existante au sein de laquelle sont consolidées les données remontées depuis l'application SNT. L'objectif est de dissocier les données journalières des données conservées à des fins d'historisation. Du côté de l'utilisation courante de l'application SATURNE, ceci permet d'avoir une base de données de petite taille et donc d'améliorer les temps de recherche « tablescan » au sein de la base lors de l'agrégation des données. Par ailleurs la dissociation de la base de consultation de la base de consolidation permet de diminuer les risques de contention pouvant être dus à l'alimentation au fil de l'eau par l'application SNT. La base d'historisation poursuit aussi cet objectif mais vise la conservation des données sur une période plus large afin de continuer de permettre au petit nombre d'utilisateurs concernés, de faire les analyses moyen/long termes à partir des données de production.

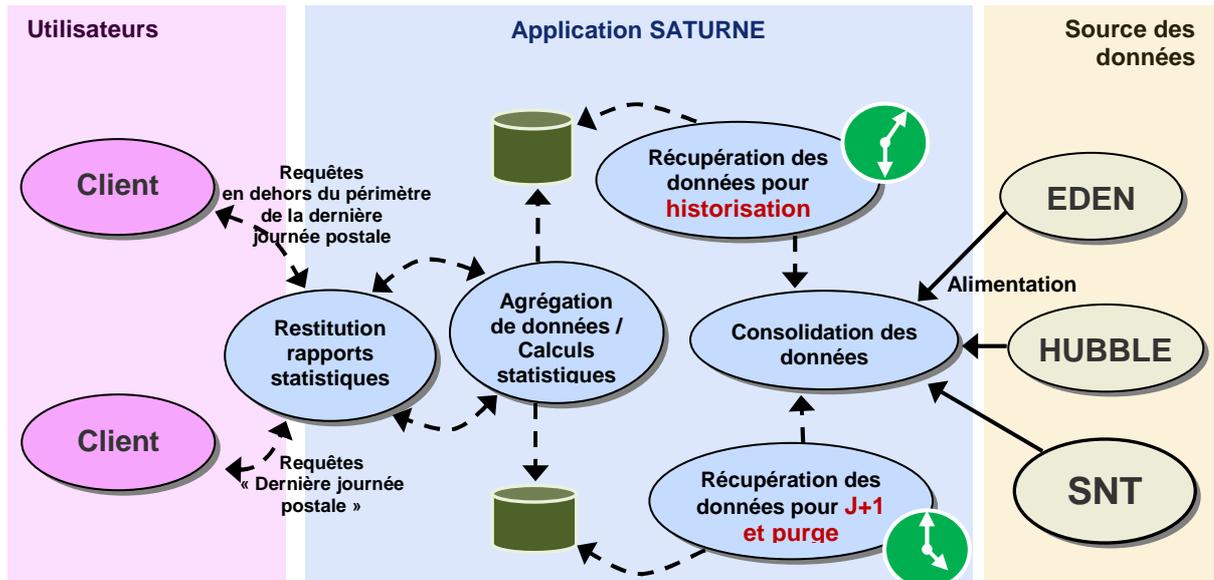


Figure 31 – Architecture fonctionnelle du scénario « Base d'historisation / Base à J+1 »

### 3.7.3.4 Tableau comparatif des scénarios d'évolution de l'architecture logique

	Scénario C1 « Passage complet en architecture 3-tiers »	Scénario C2 « Contrôle des traitements parallèles de Sybase »	Scénario C3 « Base d'historisation / Base à J+1 »
Caractéristiques majeures	Apporte une séparation logique complète entre le tiers traitement et le tiers données.	Apporte des moyens de contrôler plus finement les traitements parallèles par rapport à ceux que choisit de faire par défaut l'optimiseur de Sybase.	Apporte un mécanisme d'archivage des données.  Apporte un allègement de la base de données dédiée aux rapports statistiques journaliers.
Principaux avantages	Mise en place d'une architecture scalable par augmentation du nombre de machine.	Possibilité d'améliorer les performances du serveur ASE de Sybase.	Allègement de la base de données la plus sollicitée par les utilisateurs.  Suppression des contentions sur la base de données dues aux remontées au fil de l'eau des données de production par l'application SNT.
Principaux inconvénients	Difficulté de transformer une application orientée « Données » en une application orientée « Traitement ».	L'optimiseur ne fait pas d'exécution en parallèle de requêtes faites sur des tables temporaires ou sur des tables système.  Si un nombre insuffisant de	Perte de la remontée dynamique (délai de 5 minutes) des données de production.

		<p>processus est spécifié dans la configuration par rapport à ce que requiert un plan d'exécution, alors l'optimisateur effectue un ajustement dynamique du plan pour l'exécuter avec un nombre moindre de processus. Ceci peut entraîner de sérieuses dégradations des performances.</p> <p>Requiert le partitionnement des tables sur plusieurs disques physiques (sauf les tables temporaires et les tables de travail, qui ne peuvent pas être partitionnées).</p>	
Gains	<p>Les traitements effectués aujourd'hui par le SGBD sont difficilement externalisables (group by, jointure)</p> <p>2 / 4 (si l'ambition de traiter les group by côté traitement est effective)</p> <p>1 / 4 sinon</p>	<p>2 / 4 si l'on considère que les requêtes sont assez parallélisables.</p>	<p>2 / 4 (au moins, sur la base de profils de requêtes, et d'un paramétrage des SGBD adéquats)</p>
Risques	<p>Deux couches sont touchées</p> <p>3 / 4</p>	<p>L'utilisation du parallélisme dans un contexte très ouvert comme SATURNE peut être aléatoire (d'autant plus que le mécanisme nécessite d'être maîtrisé)</p> <p>4 / 4</p>	<p>Trouver un mécanisme de réplication fiable (expérience WTX et technologie Sybase – la réplication par bcp ne semble pas adéquat)</p> <p>3 / 4</p>
Niveau de charge de travail en réalisation	4 / 4	<p>Mise en place du parallélisme et série de tests</p> <p>2 / 4</p>	<p>3 / 4 suivant la solution de réplication utilisée, et sur la base du découpage du modèle de données</p>
Acteurs	MOE	<p>MOE</p> <p>(DPI)</p> <p>Expertise Sybase (éditeur)</p>	<p>MOE</p> <p>DPI</p> <p>MOA</p>

Tableau 7 – Évolution de l'architecture technique logique

### 3.7.4 Évolution de l'architecture technique physique, ou « scaleup/scaleout »

Les scénarios d'évolution de l'architecture technique physique sont considérés selon deux aspects : les solutions de scaleup (augmentation des performances physiques du serveur) et de celles de scaleout (augmentation du nombre de serveurs).

#### 3.7.4.1 Solutions de type « scaleup »

Ce scénario propose deux solutions. L'une consiste en l'augmentation de la puissance intrinsèque de la configuration matérielle de la plate-forme d'exploitation, l'autre porte plutôt sur une configuration plus fine du serveur ASE de Sybase afin d'utiliser au mieux la plate-forme actuelle, à savoir :

- Optimiser le cache mémoire du SGBD (seuil maxi de *locks*, de connexions, etc)
- Vérifier la répartition de charge effective entre les différents CPU du système<sup>72</sup>
- Vérifier la quantité de mémoire vive installée sur chaque serveur<sup>73</sup>
- Augmenter le nombre de moteurs Sybase si besoin

#### 3.7.4.2 Solutions de type « scaleout »

Ce scénario porte sur l'apport de scalabilité par la mise en place d'un système de répartition de charge entre plusieurs serveurs physiques pour la base de données de SATURNE. Cette dernière serait alors actualisée sur chacun de ces serveurs par une réplication des données.

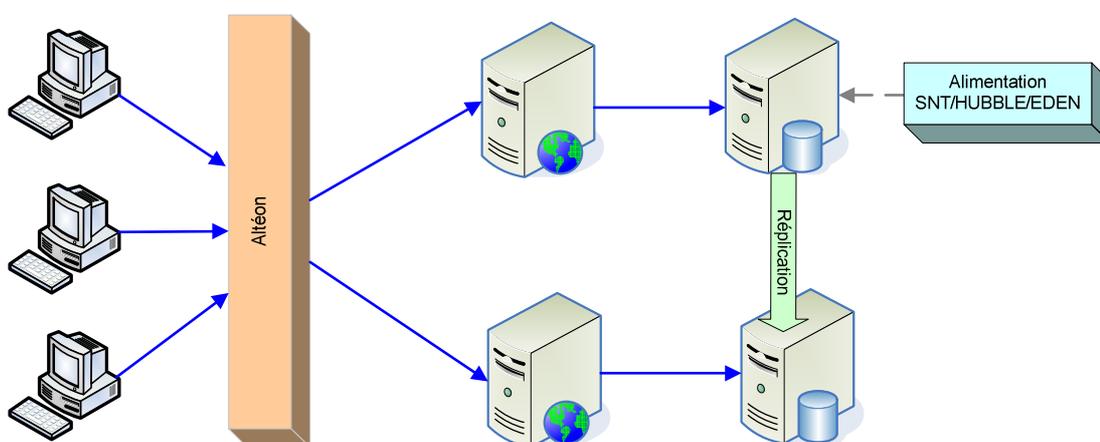


Figure 32 – Scénario prévisionnel d'une solution complète de type « scale-out » dans SATURNE

<sup>72</sup> Les données fournies par l'outil d'analyse « OmniVision » ont montré que le serveur SGBDR utilise un seul processeur (à 15% en moyenne) alors que six sont disponibles... Cette information s'est révélée vraie : un seul moteur Sybase était bien paramétré au sein de la configuration du serveur !

<sup>73</sup> Le calcul de la mémoire requise peut montrer que l'application SATURNE requiert en théorie plus de mémoire que le maximum possible adressable par le serveur ASE de Sybase. Auquel cas, l'ajout de mémoire ne serait finalement pas un levier de scalabilité...

3.7.4.3 Tableau comparatif des scénarios d'évolution de l'architecture physique

	Solutions de scaleup	Solutions de scaleout
Caractéristiques majeures	<p>Apporte des moyens physiques de garantir directement une certaine scalabilité en fonction de l'architecture de principe existante.</p> <p>Apporte des moyens de contrôler la configuration du serveur ASE.</p>	<p>Apporte un mécanisme de répartition de charge sur plusieurs serveurs SGBDR.</p>
Principaux avantages	Aucune modification des développements existants	Aucune modification des développements existants
Principaux inconvénients	Identifier la ressource à étendre (contentions actuelles)	Difficulté de monter une telle solution
Gains	<p>2 / 4</p> <p>L'amélioration des performances n'est pas garantie. L'augmentation du nombre de moteurs Sybase ne résoudra pas les problèmes de contentions au niveau des ressources : contentions au niveau des locks logiques et d'adresses, pages splits, I/O retardés, contention sur le cache, etc.</p> <p>Le calcul de la mémoire requise fait en tenant compte de tous les paramètres a montré que l'application SATURNE requiert en théorie plus de mémoire que le maximum possible adressable par Sybase.</p>	<p>2 / 4</p> <p>Multiplicité des instances physiques de la base de données mais nécessitant de maîtriser les répliquions</p>
Risques	1 / 4 (hormis le risque d'un investissement peu rentable)	<p>2 / 4 (lié à la maîtrise)</p> <p>Le coût peut être également élevé</p>
Niveau de charge de travail en réalisation	1 / 4	<p>2 / 4</p> <p>Si la solution est purement externe ; des tests de charges sont à réaliser.</p>
Acteurs impliqués	<p>DPI</p> <p>DT</p> <p>(MOE)</p>	<p>DPI</p> <p>DT</p> <p>(MOE)</p>

Tableau 8 – Évolution de l'architecture technique physique, ou « scaleup/scaleout »

Il est à préciser que les choix techniques de type scaleup ou scaleout sont à prendre en fonction des caractéristiques architecturales (système, réseau, applicatif...) et pas uniquement technologiques.

### 3.8 Prototypage d'un composant de service pour les échanges de données entre SNT et SATURNE

Un autre travail s'est inscrit dans le cadre de nos efforts d'optimisation visant à offrir à SATURNE une meilleure scalabilité. En revanche cette fois-ci, il ne s'agissait pas de remettre en cause l'architecture de l'application elle-même, mais plutôt celle du système d'échange de données qui permet à SNT<sup>74</sup> de transmettre les données de production à SATURNE.

Aussi, en collaboration avec Christophe DESSAIX (développeur SNT, également prestataire APSIDE), j'ai entrepris de bâtir un prototype de solution DCOM dans l'idée de démontrer qu'il était à la fois simple et avantageux<sup>75</sup> de remplacer la technologie WTX par un composant de service développé en VB6.

L'idée est que les flux de données de SNT vers SATURNE se fassent désormais par le biais d'un composant DCOM hébergé par SNT et appelé par un composant SATURNE. Le composant SNT fournirait ce que l'on appelle un service aux différentes applications susceptibles de se faire fournir des données qu'il met à disposition (SATURNE, SPOT, Genève, etc).

#### 3.8.1 Fonctionnement actuel du système d'échange de données

L'actuel système d'échange de données est le suivant.



Figure 33 – Fonctionnement actuel de l'échange de données de SNT vers SATURNE

Le principe est que SNT se comporte comme un référentiel actif, c'est-à-dire que lui-même prend l'initiative des exports<sup>76</sup>.

<sup>74</sup> Une telle migration ne concernerait pour l'instant pas les référentiels HUBBLE et EDEN, car ceux-là ont un poids négligeable dans les risques liés à la montée en charge.

<sup>75</sup> À la fois en termes de performance et d'économie financière (une licence WTX coûte excessivement cher)

<sup>76</sup> Ce mode d'export est expliqué dans les commentaires succédant au schéma d'architecture page 46.

### 3.8.2 Refonte du système d'échange de données

Le scénario que nous avons retenu implique un changement à la fois de principe et de technologie.

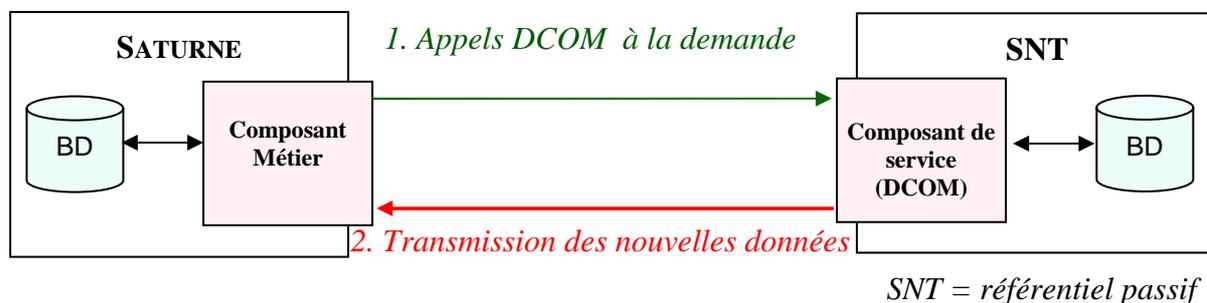


Figure 34 – Scénario d'évolution de l'échange de données vers un composant de service

Le schéma ci-dessus modélise le scénario dans lequel le module WTX serait abandonné au profit de deux composants « faits maison », et communiquant via COM+<sup>77</sup>.

Le changement de principe que l'on a proposé est que le composant hébergé sur SATURNE prend lui-même l'initiative de demander régulièrement à SNT de lui envoyer une mise à jour des données. Ainsi, peu importe le nombre et la nature des clients que fournit SNT ! Ce dernier n'aurait plus besoin de connaître ses clients, ni de subir une évolution pour chaque nouveau client.

### 3.8.3 Principe de fonctionnement

Avec cette solution, SATURNE serait donc capable de mettre à jour des données partiellement reçues. Les mises à jour possibles seraient les suivantes :

- Dans un bloc stats, les quantités (de la forme  $qt_{<x>}$ ) et le code de validité sont calculés après insertion du bloc dans SNT. Il en résulte donc une mise à jour nécessaire de ces champs dans ce cas-là.
- Un bloc stats peut aussi changer de données méthode<sup>78</sup> à n'importe quel moment. Il en résulte donc une mise à jour du *no\_don\_met* dans ce cas.
- Enfin, il sera peut-être souhaitable d'enregistrer dans SATURNE une session de tri non encore terminée (afin de visualiser dans les chronogrammes ou la liste de blocs les sessions de tri en cours). Il en découle donc une mise à jour de la date de fin de

<sup>77</sup> COM+ est dans le monde Windows l'équivalent de CORBA dans le monde Unix, c'est un bus de communication entre objets.

<sup>78</sup> Dans SATURNE, une donnée-méthode désigne la famille de plans de tri à laquelle est relié tel bloc stats.

session et de toutes les données rattachées. Cette mise à jour serait déclenchée par la réception par SNT d'un fichier de fin de session de tri.

Le fonctionnement technique de cette solution est le suivant. Cela revient à des appels distants d'un composant de service SNT, qui communique via ODBC avec la base de données SNT. Si le composant de service sur le serveur SNT s'appelle 'ak\_w3.SV\_Bloc\_Stat', alors le client, pour rapatrier les données de SNT, devra instancier le composant de service par un 'Set obj = CreateObject("ak\_w3.SV\_Bloc\_Stat")' suivi d'une requête du type 'obj.OP\_ListeBlocs(...)' qui récupère un RecordSet<sup>79</sup>.

#### 3.8.4 Comment planifier la sollicitation périodique des services SNT ?

Les imports de SNT vers SATURNE ne se faisant plus au fil de l'eau mais sur l'initiative de SATURNE, ce dernier doit donc interroger SNT régulièrement pour « télécharger » des nouvelles données à insérer ou à mettre à jour. Et c'est le serveur web SATURNE qui va interroger, via un composant métier dédié, le composant de service DCOM de SNT.

Le problème posé par ce changement serait d'éviter que SATURNE rapatrie des données déjà présentes en base. Pour cela, nous ajouterions simplement un champ « dt\_maj\_snt » dans la table *bloc\_stat* de SNT, afin de toujours garder en mémoire la date et l'heure de la dernière demande de mise à jour.

Mais une autre question reste à savoir comment planifier les requêtes que SATURNE doit lancer à SNT. La solution mise en œuvre est la suivante :

Cette interrogation se fait à chaque fois que :

- Un utilisateur demande des blocs ou des chronogrammes ;

ET que :

- Personne n'a demandé depuis  $n$  secondes une synchro de blocs provenant d'une machine du même établissement.

Le problème du « depuis  $n$  secondes »<sup>80</sup> peut être résolu par une mémorisation en variables d'application, sur chaque serveur web, de la dernière fois que le serveur web SATURNE a interrogé SNT.

#### 3.8.5 Problème annexe

Un problème resterait à résoudre : si chaque instance de serveur web<sup>81</sup> gère ses propres variables d'application, cela générera des conflits.

---

<sup>79</sup> Sur le client, ce composant de service distant devrait être enregistré avec un 'clireg32 ak\_w3.VBR -s <adresse ip du serveur> -d'.

<sup>80</sup> à quel endroit retenir cette info ?

Les deux solutions possibles sont les suivantes :

A- Simuler ces variables de session via des champs de la table *params* de la base de données

*Avantage* : On garantit la cohérence et la stabilité du système

*Inconvénient* : L'accès à ces variables sera plus long.

B- Ne faire en sorte qu'un seul serveur web sur les 3 communique avec SNT

*Avantage* : On préserve une rapidité d'accès à ces variables

*Inconvénient* : Le plantage de ce serveur entraîne l'indisponibilité générale des données.

### 3.8.6 Comment distinguer les données non encore intégrées des autres données ?

Une autre problématique est d'éviter d'importer plusieurs fois les mêmes données. Il s'agit donc de rendre les applications clientes capables de distinguer les nouvelles données de celles déjà existantes dans leurs bases respectives. La méthode est la suivante :

- Pour les blocs stats : ajout d'un champ « *dt\_maj\_snt* » dans la table *bloc\_stat* de SNT (comme indiqué plus haut). Ce champ comporte la date à laquelle a été inséré ou mis à jour tel bloc stats dans SNT. C'est donc une « date pivot » dont se sert SNT pour délivrer les blocs stats mis à jour ou insérés seulement après une date D.

NB. La date de début de session de tri est inexploitable, car SNT ne reçoit pas toujours les blocs stats dans l'ordre.

- L'import de la donnée méthode liée à un bloc stats se fait au moment de l'import de ce bloc stat, et bien sûr si elle n'existe pas encore.
- L'import des données réceptacles se fait si la donnée méthode vient d'être insérée.
- L'import des durées calculées, des compteurs bruts et calculés ainsi que des valeurs réceptacles a lieu seulement dans le cas d'une création de bloc.
- L'import des événements n'est plus lié à celui d'un bloc stats : il est fait séparément (après chaque synchronisation groupée de blocs stats).

Or, les événements sont très nombreux, et n'ont pas de clé unique. On a donc choisi, pour différencier les événements déjà insérés des autres, de raisonner par « fichiers d'événements » :

---

<sup>81</sup> Pour SATURNE, ils sont au nombre de trois, l'ensemble étant régit par un système de répartition de charge Altéon.

- Deux événements issus du même fichier d'événements ont toujours le champ *dt\_enr* de même valeur <sup>82</sup> ;
- De plus, deux événements de la même machine ayant des *dt\_enr* différentes sont forcément issus de deux fichiers différents (une machine ne peut remonter à SNT 2 fichiers de même nature<sup>83</sup> à la même seconde).
- Deux champs ont été ajoutés dans la table *machine* de SATURNE : « *dt\_last\_maj\_eve* » et « *dt\_last\_nouv\_eve* ». Ainsi, on peut à tout moment connaître les événements qui n'ont pas été insérés depuis une date D pour telle machine de tri, et savoir à quelle date/heure a eu lieu la dernière mise à jour.

Ainsi, on considère qu'un événement existe déjà dans la base de données SATURNE si le couple « machine / dt\_enr » existe déjà.

NB. Les événements « hors-session de tri » sont reliés au bloc stats de type « incident » (dont le *no\_blo* vaut 0).

### 3.8.7 Test du prototype

À l'aide de l'environnement de développement Microsoft Visual Basic, nous avons développé une application cliente basique avec plusieurs champs de saisie et un bouton, permettant simplement d'appeler l'objet distant via COM+ avec les paramètres de notre choix. Le test a fonctionné très facilement.

### 3.8.8 Bilan sur le prototype

Notre prototype a réussi sa mission, qui a été de valider techniquement les nombreux avantages qu'on attendait du scénario mis en place : facilité de maintenance, déspecialisation des technologies employées, économie financière, et surtout résistance à la montée en charge.

Néanmoins, comme cela est souvent le cas dans les grosses entreprises, n'adopte pas n'importe quelle technologie qui veut... Notre solution n'a malheureusement pas été retenue pour des raisons plus ou moins politiques : l'usage du DCOM n'a jamais été testé par la DT (la DT a entre autres pour rôle d'éprouver les diverses technologies afin de les attester ou non en vue d'un usage éventuel dans les applications du SI LA POSTE).

---

<sup>82</sup> Bien qu'un « fichier d'événements » comporte plusieurs « sections d'événements », SNT ne retient que la première « date d'enregistrement » trouvée pour créer le champ *dt\_enr*.

<sup>83</sup> Des fichiers peuvent être de type « FST », « FSE » ou « FLX ».

### 3.9 Optimisations sortant du domaine de la montée en charge

Les optimisations visant à une meilleure scalabilité du système ont été d'une importance vitale pour la survie des données. Mais on ne peut pas clore ce chapitre sans préciser que ce type d'optimisation suffit rarement à lui seul : dans le cas de SATURNE, d'autres types d'optimisation se sont montrés non moins indispensables. Je pense notamment à la nécessité de « blinder » l'application, comme on dit dans le jargon informatique populaire.

Aussi, j'ai participé activement au « blindage » de l'application SATURNE, et à deux niveaux : d'une part le traitement des données, d'autre part l'IHM.

#### 3.9.1 Blindage de la partie SGBD

La nécessité de blinder le traitement des données dans SATURNE vient du fait qu'un certain nombre de statistiques provenant quotidiennement des machines de tri se révèlent aberrantes, comme par exemple une date de début de tri au 01/01/1900, ou encore un indicateur comme « Nombre de plis dépilés » avoisinant un nombre anormalement élevé (tel que 2 milliards de plis dépilés en 1 heure).

Le tableau ci-dessous explique comment nous avons pu contourner les principaux cas d'aberration rencontrés.

Cas de donnée aberrante	Anomalie générée	Solution de contournement
Date de début ou de fin de session de tri à 1900	Lors du calcul de la durée de la session de tri, il y a dépassement de capacité du type de la donnée renvoyée par la fonction DATEDIFF().	Utiliser la fonction NULLIF. Exemple : NULLIF(dt_fin_tri, '01/01/1900').
	À chaque fois qu'un bloc daté à 1900 arrive dans la base de donnée (ce qui est quasi quotidien) la purge n'avance pas à cause de sa règle de fonctionnement, qui est de purger les blocs stats compris dans une période de 24 heures à compter du plus ancien... soit le premier jour de l'année 1900 !	Prendre comme base à purger non pas les 24 heures à compter du plus ancien bloc, mais une union entre le 01/01/1900 et les 24 heures à compter du plus ancien bloc « hors 1900 ».
Valeur avoisinant la limite de l'entier	Lors du calcul d'un total sur plusieurs sessions de tri, il peut y avoir dépassement de la capacité du type de la donnée renvoyée par la fonction SUM().	Passer le type de données de INT à NUMERIC(16).
Durée de session à zéro	Lors du calcul du Taux d'utilisation (= Durée de traitement / Durée de session) il y a erreur de division par zéro.	Remplacer systématiquement '<denom>' par 'NULLIF(<denom>, 0)' dans chaque dénominateur de division.

Tableau 9 – Contournement des aberrations rencontrées avec Sybase

### 3.9.2 Blindage de la partie IHM

Cas de donnée aberrante	Anomalie générée	Solution de contournement
Présence de code HTML dans certains libellés (ex. un « <b> » dans un nom d'opérateur)	La valeur est interprétée en tant que code HTML, ce qui est généralement intempestif car l'exemple du « <b> » dans le nom d'opérateur aurait pour fonction indésirable de mettre en gras <sup>84</sup> tout le reste de la page	Formatage systématique de chaque caractère affiché (exemple : transformation de « <b> » en « &lt;b&gt; »)
Présence de saut de ligne dans certains libellés	Si ce libellé est inséré dans le contenu d'une variable Javascript, alors cela reviendrait à une erreur de type « côte fermante attendue » : var lib = 'blabla bla' ;	Transformer systématiquement les sauts de ligne en code HTML (' ')
Longueur anormale de certains libellés	Dans les tableaux HTML affichant les données, certaines colonnes dont la taille est pourtant fixe peuvent être surdimensionnées par rapport aux autres. De surcroît, le rendu à l'impression peut s'en retrouver tronqué.	On affiche les <i>n</i> premiers caractères de chaque information : s'il y a troncature, alors on ajoute des points de suspensions ainsi qu'une infobulle permettant de dévoiler le texte complet au passage de la souris sur l'information tronquée.
Numéro de centre de tri commençant par un zéro	Quand on mémorise en base la conversion en nombre entier d'un numéro de centre qui commence par un zéro, et qu'on le reconvertit en chaîne de caractères, alors le zéro n'y figure plus : les tentative de comparaison de chaînes échouent alors.	On prend les 6 derniers caractères de la chaîne suivante : '00000' + numCentre
Chevauchement de deux sessions de tri sur la même machine (impossible dans la pratique)	Dans les chronogrammes, deux intervalles viennent se superposer, l'un cachant l'autre.	Aucune solution n'est vraiment envisageable, car aucune méthode ne peut aider à identifier objectivement la session « parasite »... J'ai néanmoins mis en œuvre une possibilité de cacher les intervalles représentant des sessions de tri dites « invalides » (zéro plis dépilé, etc).

Tableau 10 – Blindage de la partie IHM

### 3.9.3 Consolidation de la gestion des erreurs

Le SGBD Sybase prévoit nativement de gérer les différents messages d'erreur possibles dans la table système *sys\_user\_messages*. Il suffit ainsi d'appeler la fonction T-SQL « RAISERROR », paramétrée avec l'identifiant de l'erreur ainsi qu'un ou plusieurs paramètres facultatifs.

<sup>84</sup> En HTML, tout texte mis entre une balise ouvrante <b> et balise fermante </b> est mis en gras ('b' pour 'bold').

En ce qui concerne le Javascript, la manière de gérer les erreurs est d'utiliser le couple d'instructions « try / catch », dont l'intérêt est de ne pas générer de manière visible pour l'utilisateur les éventuelles erreurs Javascript issues de telle ou telle instruction « délicate », mais au contraire traiter de manière particulière l'erreur (ex. envoyer via Ajax -cf. page 129- au serveur web une info sur l'erreur rencontrée, afin de la consigner et de la communiquer aux développeurs).

Enfin, nos composants métier écrits en VB font aussi l'objet d'une gestion dédiée des erreurs, au moyen d'un fichier de ressource contenant, comme avec Sybase, la référence de chaque erreur possible. Ce fichier est lu par un composant spécialisé dans la gestion d'erreurs, et ce composant contient les méthodes permettant d'intercepter les erreurs et de les tracer en base.

À un niveau fonctionnel, il serait dommage de ne pas pouvoir consulter ces erreurs générées dans l'une ou l'autre couche de l'application : aussi avons-nous développé avec l'équipe SNT une interface de recherche et d'affichage des erreurs au sein de chaque application SATURNE et SNT, accessibles seulement en mode administrateur :

**Recherche des erreurs** Résultats

Date début : 07 / 11 / 2008 11 : 43 : 55  
 Date fin : 18 / 11 / 2008 14 : 19 : 55  
 Entité : \*Tous\*  
 Type d'erreur : \*Tous\*  
 Type machine : \*Tous\*, Composant Saturne, Intégration WTX, Exécution des batchs, Exécution de scripts ASP

Rechercher État initial

**Recherche des erreurs** Résultats

TRI

Type d'erreur	Etablissement	Type machine	Machine	Date	Code	Messages	Fichier/Session
Mercator		INCIDENT	0	07/11/2008 11:48:53	20001	Err 1 proc import_etab: Impossible d'insérer l'établissement 620221 de libelle ANNEZIN AMI	aceeaaaa.c9
Mercator	VILLENEUVE GR LES BASTIDES	INCIDENT	0	07/11/2008 11:47:32	20002	Err 2 proc import_etab: Impossible de mettre à jour l'établissement 470470 de libelle AGEN GR PLAINE DE GARONNE	aceeaaaa.c9
Mercator	MARMANDE GR PLAINE DE GARONNE	INCIDENT	0	07/11/2008 11:47:32	20002	Err 2 proc import_etab: Impossible de mettre à jour l'établissement 470460 de libelle AGEN GR LES BASTIDES	aceeaaaa.c9
Mercator	SAINT PAUL LES ROMANS	INCIDENT	0	07/11/2008 11:45:33	20002	Err 2 proc import_etab: Impossible de mettre à jour l'établissement 263230 de libelle SAINT PAUL LES ROMANS BP	aceeaaaa.c9
Mercator	ROMANS JEAN MOULIN	INCIDENT	0	07/11/2008 11:45:29	20002	Err 2 proc import_etab: Impossible de mettre à jour l'établissement 260550 de libelle ROMANS JEAN MOULIN BP	aceeaaaa.c9
Mercator	ST ETIENNE DE FONTBELLON GA	INCIDENT	0	07/11/2008 11:43:55	20002	Err 2 proc import_etab: Impossible de mettre à jour l'établissement 073710 de libelle ST ETIENNE DE FONTBELLON BP	aceeaaaa.c9

6 réponses Page 1 sur 1

Figure 35 – Recherche et affichage des erreurs dans SATURNE (exemple)

### 3.9.4 Mise en correspondance avec la revue de code

Voici, à la lumière de la revue de code détaillée, une synthèse de l'état des lieux que j'ai pu dresser sur l'application SATURNE telle que je l'ai développée dans sa version 2.0 :

État des lieux	Conséquences	Recommandations
De tables temporaires sont utilisées pour effectuer des sélections.	Les tables temporaires utilisent beaucoup de ressources.	Développer des requêtes plus complexes.
Les événements DHTML sont utilisés sur de nombreux objets.	Les événements DHTML utilisent des ressources côté client.	Remonter les événements sur les objets les moins nombreux et utiliser l'instruction <code>event.srcElement</code> pour définir le code à exécuter.
Des requêtes dynamiques sont utilisées.	Les requêtes dynamiques ne sont pas compilées.	Développer plusieurs procédures stockées et/ou mettre des instructions de type IF dans celles-ci. Exécutera du SQL dynamique directement depuis ADO.
Le code Javascript représente une forte volumétrie dans SATURNE.	Le code Javascript a la réputation d'être difficilement maintenable et il charge le réseau.	Remplacer les instructions « <code>document[...]</code> » par une fonction <code>d()</code> par exemple.

Tableau 11 – Bilan de la revue de code de SATURNE 2.0 et principes d'améliorations

Suite à cette revue de code, j'ai accompli un travail d'optimisation générale de l'application SATURNE, dans laquelle j'ai répondu avec succès à l'ensemble des recommandations ci-dessus.

De nouveaux tests de montée en charge ont succédé à cette campagne d'optimisations, et ont confirmé l'acquisition de meilleures performances, tant en allocation mémoire qu'en temps de réponse.

### 3.9.5 Des leçons à tirer pour le développement des projets ?

Jusque là, je n'ai précisé ni comment, ni à quel stade de l'avancement du projet nous avons appris que les données arrivant dans SATURNE pouvaient être invalides.

Dans la réalité, nous n'avons pas su anticiper ces problèmes, la plupart d'entre eux ayant été découverts sur le terrain, c'est-à-dire en production.

Il faut néanmoins considérer que ces problèmes étaient peu prévisibles. En effet, n'est-il pas logique de réagir par une défense du type « *On ne pensait pas que les machines de tri pouvaient envoyer n'importe quoi* » ? Cela dit, je pense que l'on ne peut pas considérer la simple imprévisibilité des aberrations de données comme une excuse : aujourd'hui, après avoir essuyé en production plusieurs mauvaises surprises de ce type au fil des années, il m'apparaît très clairement que toute application doit être testée dans des conditions dites « iso production », c'est-à-dire celles du terrain... pas seulement au regard de la volumétrie

des données, mais aussi de leurs valeurs. Cela doit se faire bien entendu avant que l'application soit livrée<sup>85</sup>, et cela au moyen de ce que l'on appelle un « plan de tests ».

La constitution du « plan de test » est justement ce qui n'avait pas été prévu durant la phase de conception de la toute première version de SATURNE. Mais aujourd'hui, le document de plans de tests fait partie intégrante des livrables prévus par notre contrat de prestation (en l'occurrence, un contrat de TMA).

Cela dit, la véritable problématique liée au blindage des applications, à mon sens, n'est pas de faire ou de ne pas faire de plan de tests : une chose est de rédiger un plan de tests, et une autre est d'y mettre en œuvre tous les cas possibles d'aberrations de données dans lesquels nous considérons que la stabilité de l'application est mise en péril.

C'est justement à ce niveau que l'on échoue si facilement dans les projets informatiques : d'une part, les impératifs de délai font qu'on n'a pas toujours le temps de s'investir dans la constitution de jeux d'essai couvrant un panel complet de données aberrantes possibles. D'autre part, même si on réalise cette tâche avec soin, on ne peut pas prévoir toutes les aberrations auxquelles il est possible de se confronter en production.

Cette problématique ne rejoint-elle pas d'ailleurs le mythe de l'application parfaite ?

### **3.9.6 Des plateformes de tests comme solution complémentaire**

Comme on vient de le voir, un plan de tests rédigé et déroulé par l'équipe de développement ne suffit pas toujours.

Aussi, LA POSTE a mis en œuvre récemment deux plateformes complémentaires visant à stabiliser au mieux ses applications :

- Une plateforme de recette<sup>86</sup> (située à la Direction Technique du Courrier, à Nantes) ;
- Une plateforme de pré-production (située sur le « site pilote technique » de Chalons en Champagne). Y sont réalisés les tests de montée en charge, avec des serveurs iso production et des logiciels d'injection de données simulant les machines de tri.

Précisons qu'avant d'arriver en plateforme de production (à Limoges), les plateformes de recette et de pré-production deviennent respectivement l'avant-dernière et la dernière étape dans le cheminement des livraisons.

---

<sup>85</sup> Ne trouvez pas cela stupéfiant : j'ai déjà vu pour le projet SATURNE des tests de montée en charge effectués seulement après la mise en production par les soins de l'équipe de TMC.

<sup>86</sup> La phase de recette d'une application consiste en ce que la MOA procède à la validation fonctionnelle de celle-ci. Il arrive que la MOA, pour ce travail, se fasse aider directement par de véritables utilisateurs.

### *Quelques limites...*

- Les étapes de recette et de pré-production devraient être simultanées pour permettre un gain de temps ;
- Aucune donnée aberrante (ex. durée négative) ne sort des injecteurs artificiels de données sur la plateforme de pré-production. Dommage, car ce sont très souvent des données aberrantes qui provoquent les bugs dans SATURNE<sup>87</sup>.

#### **3.9.7 Corriger les aberrations à la source**

Au lieu d'optimiser la fiabilité au niveau du traitement des données, ne serait-ce pas plus simple de corriger les données aberrantes (cf. partie 3.9.1, page 92) ou encore de les filtrer, voire les transformer avant de les insérer en base ?

Malheureusement, cela est difficilement envisageable pour différentes raisons :

- La faute incombe aux machines de tri. Certes, une pression est exercée depuis des années auprès des constructeurs pour exiger que celles-ci soient enfin déboguées et envoient des données valides. Mais ces efforts se sont révélés jusqu'ici infructueux, en partie parce que jusqu'à maintenant le système SNT/SATURNE n'avait pas été lui-même suffisamment fiabilisé, et qu'il était difficile de « s'innocenter » auprès des constructeurs. Or en 2008, le système SNT/SATURNE a atteint une maturité remarquable en termes de stabilité, après des efforts longs mais concluants en termes de correction de bugs et de d'optimisation... Malgré cela, nous ne savons pas si un jour, les machines seront vraiment corrigées : ainsi, il me semble qu'une position raisonnable soit de prendre le parti de la défaillance potentielle et irrémédiable des données dès le début de leur cheminement ;
- De même, certaines données sont saisies à la main par les techniciens en centres de tri (comme par exemple le 'nom opérateur', désignant le technicien utilisateur de la machine de tri). On pourrait penser que les écrans de pilotage des machines de tri leur interdisent de saisir autre chose que les caractères autorisés, hélas ça ne semble pas être toujours le cas. Aussi, avec autant de modèles et de constructeurs de machine de tri, il est trop risqué de compter sur un hypothétique blindage des IHM des machines de tri pour assurer celui du traitement des données ;
- Une autre raison pour laquelle on ne peut pas traiter toutes les incohérences, est qu'un besoin fonctionnel à part entière consiste justement à pouvoir visualiser les données que l'on peut objectivement qualifier d'invalides (quand la durée de session est nulle, ou encore quand la fin de session précède le début de session :

---

<sup>87</sup> Pour cause, elles correspondent à des cas qui n'ont pas été anticipés par nos soins (et ce, malgré notre intelligence foudroyante).

toutes ces données s'affichent en rouge dans SATURNE). Moralité : ne surtout pas toucher aux données, même si elles sont fausses !

### 3.10 Résultat des tests d'optimisations

La campagne d'optimisation menée sur SATURNE a permis de dégager des conclusions positives suite aux nouveaux tests de montée en charge (TMC) qui s'en sont suivis :

- Les tests de montée en charge effectués sur SATURNE ont permis de lever un point d'attention sur l'utilisation et le niveau de consommation du CPU sur le serveur de base de données SATURNE.
  - La fidélité des résultats obtenus par les différents tests pourrait être accrue, car les contentions au niveau CPU pressenties par les TMC ne sont pas confirmées par les relevés d'Omnivision<sup>88</sup> ;
  - Cependant, il s'agirait d'améliorer les procédés de tests afin que ceux-ci soient plus représentatifs par rapport aux machines cibles (il s'agirait de maîtriser le calibrage des plateformes) ;
  - De plus, la représentativité des scénarios fournis en entrant des TMC est celle identifiée par la MOE, et non l'usage exact du terrain relayé par la MOA ;
  - En complément du document de TMC SATURNE, les données brutes ont permis de constater qu'en pointe, la répartition des traitements était relativement équilibrée au niveau des « engines » Sybase (62,5% pour l'engine 1 et 58,4% pour l'engine 2 sur les tests en SIE). Aucune information ne permet à ce jour d'identifier quelle était la charge effective des CPU au niveau système.
- Nous avons obtenu des gains potentiellement importants sur une majorité de cas d'usages et en situation d'accès concurrentiels faisant appel aux procédures stockées. Cependant :
  - Certaines procédures, lorsqu'elles sont peu utilisées simultanément (évalué en mono-utilisateur), montrent toutefois une diminution relative des performances ;
  - Il faut s'assurer de la bonne représentativité des tests entrepris (sélection des quelques procédures stockées fortement utilisées et les moins scalables – sur la base des tests TMC).
- En synthèse des tests d'évaluation des optimisations :

---

<sup>88</sup> OmniVision est un outil édité par *Systar* d'observation du comportement des systèmes. Il est basé sur la collecte des données, la normalisation, la consolidation, l'historisation, et la génération et diffusion automatisée des rapports.

- Les performances de quelques procédures stockées comme « liste\_blocs\_stats() » et « journal\_de\_bord() » en mode multi-utilisateur (10 utilisateurs concurrentiels) montrent des facteurs de gains de 1,7 à 50 par rapport aux performances actuelles ;
- La source de gain principale est la réduction du nombre de tables temporaires (par exemple dans les tests, le nombre de tables temporaires sur « liste\_blocs\_stats() » a été réduit de quatre à une, en créant autant de requêtes « statiques » que de cas d'usages qui transparaissaient dans les requêtes dynamiques).

En conclusion, notre campagne d'optimisation de SATURNE, visant à garantir une scalabilité adaptée à l'augmentation prochaine du nombre de centres et d'utilisateurs, est un succès si on s'en tient aux résultats annoncés par les dernières TMC. Et ce succès a été permis tout en minimisant les coûts et les contraintes.

Des scénarios comme celui prévoyant la pré-génération de la dernière journée postale, ou encore la création d'une application à part pour les CDIS, sont en effet superflus... du moins pour l'instant : il est intéressant de les garder dans un fond de tiroir si un jour la MOA prévoit de nouveau une augmentation significative du nombre d'utilisateurs.

## Chapitre 4. CONJUGUER PERFORMANCE ET QUALITÉ DE L'INTERFACE

Le jour où j'ai ressenti une gêne de mon propre métier remonte à la présentation que l'on m'a faite, il y a quelques années, de quelques écrans du fameux progiciel SAP<sup>89</sup>. Le traumatisme était visuel : l'écran permettant de se loguer n'était pas plus avenant qu'un vieux mur de parpaings défraîchi.

Esthétique rime aussi avec ergonomie : je peux citer en mauvais exemple les radiateurs électriques chez mes beaux-parents, qu'il faut allumer en faisant coulisser l'interrupteur sur la position rouge... et verte pour éteindre<sup>90</sup>. En effet, le monde de l'ergonomie ne se limite pas au seul domaine de l'informatique, puisque notre confort de vie est directement dépendant de la bonne ou de la mauvaise conception des objets récurrents de la vie quotidienne. Les applications informatiques font partie de ces objets de la vie de tous les jours. Aussi, je pense qu'il faut constamment veiller à soigner leur « utilisabilité »<sup>91</sup>.

Dans ce dernier chapitre, j'expliquerai ce qui m'a amené au cours de mon projet à LA POSTE à refondre de manière exigeante l'interface graphique de l'existant. J'en expliquerai les raisons, et surtout les moyens. C'est ici l'occasion de réfléchir sur, d'une part, l'importance de la qualité d'une interface graphique, et d'autre part, la problématique à laquelle je me suis de fait heurté dans la conception de l'application SATURNE, ce qui pourrait être introduit par la question suivante : quelle doit être la teneur du compromis entre optimisation de la performance et optimisation de l'interface ?



<sup>89</sup> SAP désigne, par abus de langage, le progiciel de gestion intégré conçu par l'éditeur allemand du même nom.

<sup>90</sup> Le constructeur a vraisemblablement associé la couleur rouge à la notion de chauffage, alors qu'il aurait été préférable à mon sens de respecter la norme "vert = allumé / rouge = éteint"...

<sup>91</sup> Vient de l'anglais « usability » — cf. <http://en.wikipedia.org/wiki/Usability>

#### 4.1 L'existant : un peu d'histoire...

L'ancêtre de SATURNE, GIPSI STATS, était lui-même un intranet. Mais je qualifie ce type d'intranet « de première génération », étant donné que GIPSI STATS a été développé par des concepteurs issus du développement « gros système » ayant saisi l'opportunité de faire leurs premiers pas en programmation web. Il en est donc ressorti une application relativement basique d'un point de vue ergonomique, mais ayant le mérite de faire partie des pionniers des intranets dynamiques à LA POSTE, dans un contexte où les postes de travail en centre de tri étaient encore en environnement Windows 3.11.

Ci-dessous figurent à titre d'exemples un ensemble de copies d'écrans de l'application GIPSI STATS telle qu'elle était avant qu'elle ne soit progressivement remplacée par SATURNE :

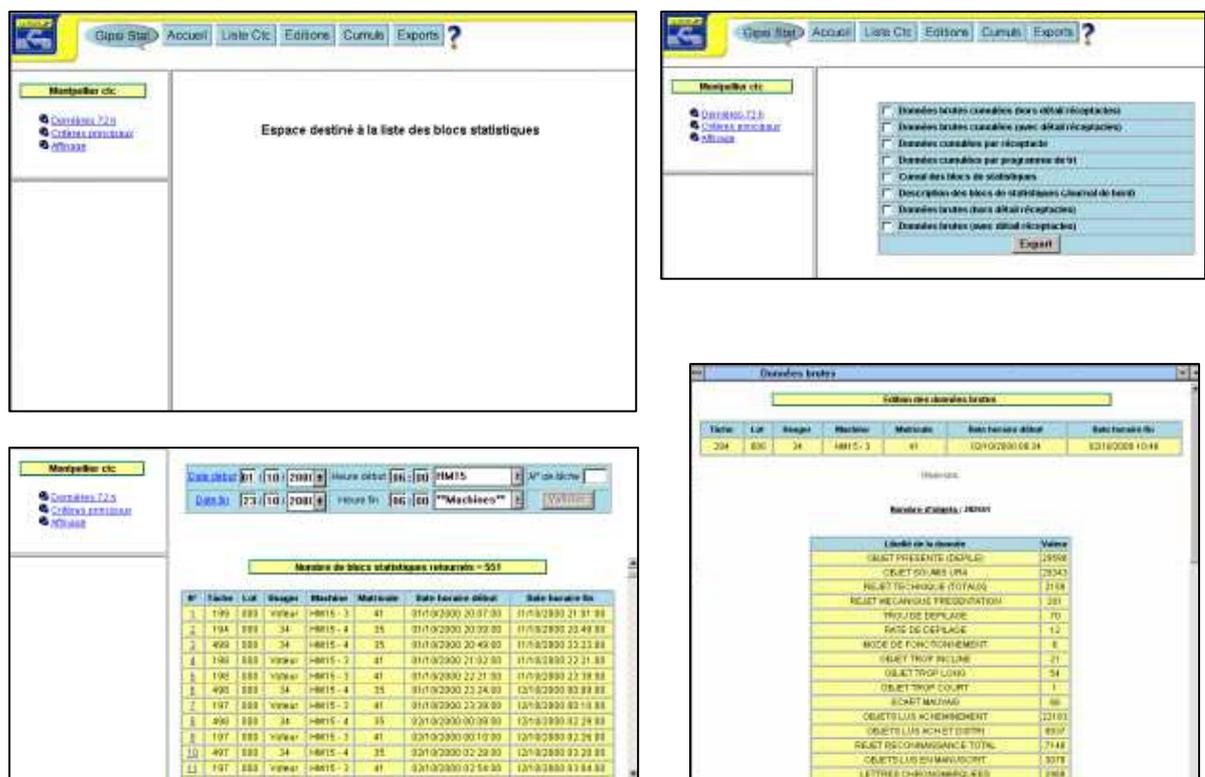


Figure 36 – Copies d'écran de l'application GIPSI STATS

## 4.2 L'IHM de SATURNE, ou la proposition d'une meilleure interface

### 4.2.1 Constat de l'existant

Les principaux défauts que j'ai trouvés à GISPI STATS sont les suivants :

- Le look général évoque celui des premiers sites web amateurs. On peut évoquer comme exemple la limite grossière entre les « frames » (cadres) ou entre les cellules des tableaux HTML, ainsi que les couleurs de fond vieillottes qui sont utilisées (cyan et jaune pâle), et enfin une police peu adaptée au web (Times New Roman), etc. ;
- Les graphiques (histogrammes, chronogrammes...) ne sont pas des objets client réalisés en DHTML, mais des images générées par le serveur web<sup>92</sup>, ce qui entraîne des conséquences en termes de performance d'accès réseau (à cause du poids de l'image), ainsi qu'une insuffisante réactivité (pas de possibilité d'interaction précise avec la souris) ;
- Enfin, j'ai constaté une mauvaise adaptation à la résolution 800x600, encore abondante dans les centres de tri (surtout dans les années où GISPI STATS était encore utilisé).

### 4.2.2 Amélioration progressive de l'ergonomie générale

Les applications telles que GISPI STATS (ou son successeur, SATURNE) étant utilisées par plus de 1500 utilisateurs au quotidien, j'ai considéré ces défauts d'ergonomie comme majeurs.

La portée d'un défaut n'est en effet plus la même si le défaut en question est rencontré 1500 fois par jour, au lieu d'une seule fois ! Aussi, dans ma mission de conception de l'application SATURNE, j'ai entrepris d'accorder à la partie ergonomie une importance toute particulière dans mon budget d'investissement personnel.

J'ai donc mené à bien une première campagne d'optimisation de l'ergonomie par rapport à GISPI STATS, en suivant deux principaux axes d'améliorations : d'abord la modernisation de l'aspect visuel, ensuite la dynamisation de l'interface<sup>93</sup>, qui dans GISPI STATS est assez statique.

---

<sup>92</sup> Cela, via le composant *GraphicServer*

<sup>93</sup> Nous entendons par dynamisation de l'interface la possibilité d'interagir d'avantage avec elle, comme par exemple changer le tri des colonnes en temps réel, ou encore faire apparaître un menu contextuel au-dessus des intervalles de chronogrammes.



Figure 37 – Première version de SATURNE : page d'accueil

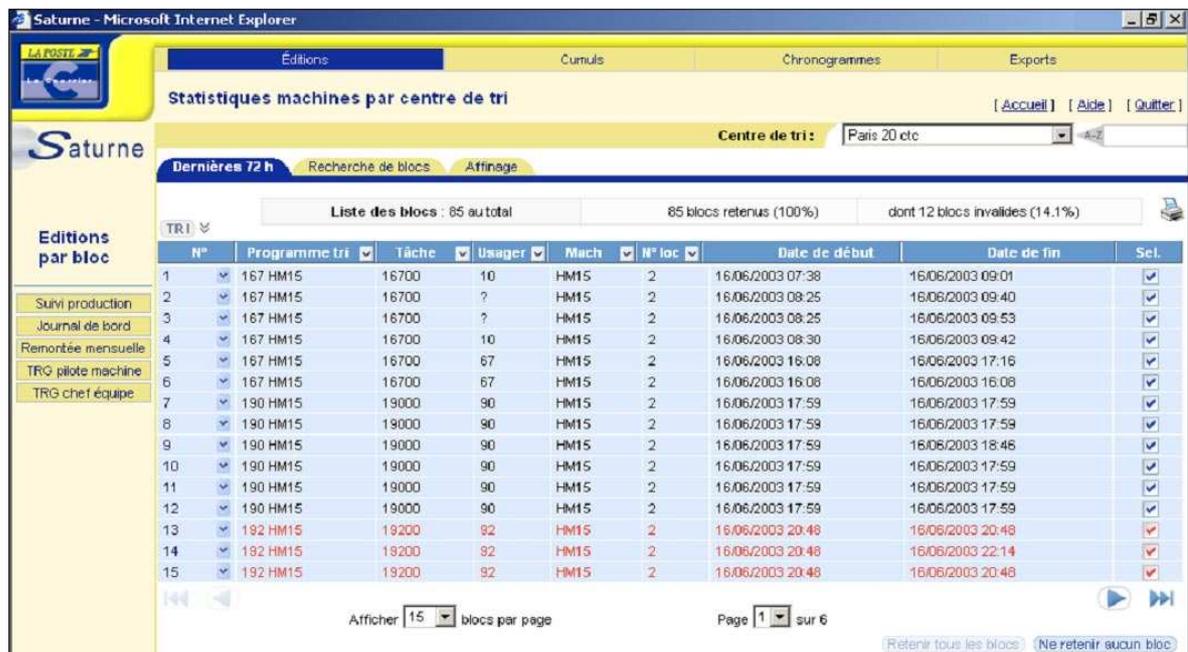


Figure 38 – Première version de SATURNE : liste des blocs statistiques

Puis en 2005, j'ai dû m'atteler à la version 2.0 de SATURNE, afin d'intégrer des fonctionnalités supplémentaires. Fort des enseignements reçus dans le module « IHM » au C.N.A.M.<sup>94</sup>, j'en ai profité pour mener à bien une deuxième campagne d'optimisation de l'ergonomie, et cette fois-ci par rapport à la version 1 de SATURNE :

<sup>94</sup> Je me réfère ici au cours « Principes d'ergonomie des interfaces web », de Jean-Marc PUJOS (module B2, « Interaction Homme-Machine »).

- Adaptation à la nouvelle charte graphique des applications intranet de LA POSTE (nouveau logo LA POSTE, couleur de bannière plus vive...);
- Présentation d'un accès direct et permanent vers les fonctionnalités les plus demandées (exemple : les blocs stats de la dernière journée postale) afin de mieux répondre à l'effet « PIP<sup>95</sup> » de la conduite des utilisateurs;
- Convivialité accentuée (boutons de menus plus esthétiques, planète SATURNE<sup>96</sup> en très léger filigrane sur la page d'accueil, etc.);
- Positionnement plus adéquat de certains éléments de la page : notamment, la liste des établissements déplacée à gauche, afin de respecter le sens naturel de lecture, qui n'aurait convenu parfaitement que dans le monde arabe (les utilisateurs de SATURNE commencent généralement par sélectionner un établissement quand ils se connectent à l'application).

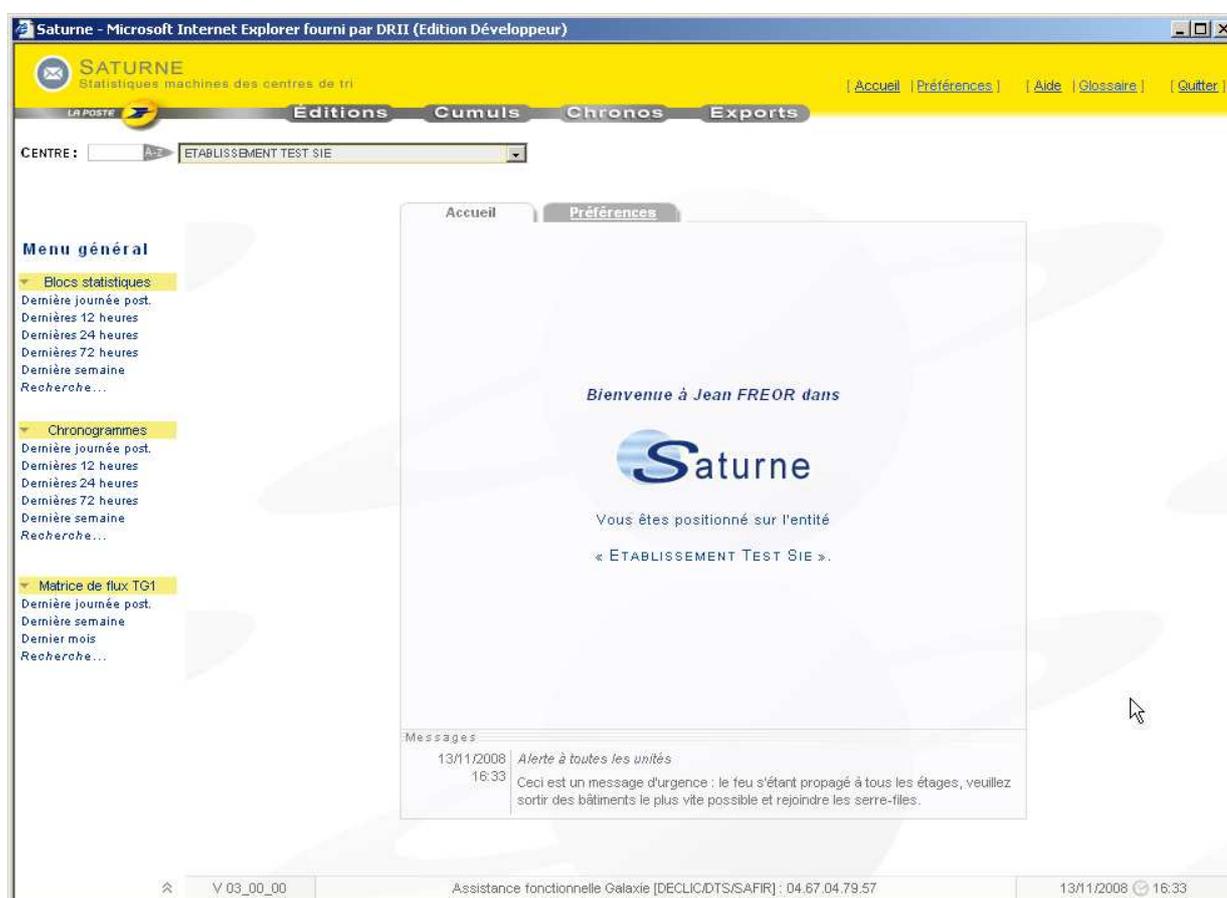


Figure 39 – Dernière version de SATURNE (ici, la page d'accueil)

Une remarque que l'on pourrait me faire à propos des menus est la suivante : je n'ai pas respecté le standard consistant à présenter sur le bandeau vertical (à gauche) les menus de

<sup>95</sup> PIP signifie « Pressés, Ignorants et Paresseux ».

<sup>96</sup> Toutes les images du site, ainsi que son logo, ont été créés de toute pièce par mes propres soins sous Photoshop.

niveau 1 et sur la barre de menus horizontale (en haut) ceux de niveau 2. C'est en effet le contraire qui a été mis en place. Il y a plusieurs raisons concrètes à cette entorse aux règles de base de l'ergonomie web : d'une part, il ne fallait pas trop « chanceler » les utilisateurs de GIPSI STATS, qui accédaient déjà aux données de production du tri postal de cette manière. D'autre part, il est préférable de garder visibles tous les items du menu de gauche à la fois (pour faire économiser des clics aux utilisateurs et leur faire voir d'un seul coup tout ce qu'ils peuvent faire). Or, pour une raison de place et de lisibilité, on ne peut pas se permettre de présenter tous les items en ligne plutôt qu'en colonne. Pour ces deux raisons, j'ai donc conservé ces items dans le menu de gauche plutôt que dans le menu du haut.

### **4.3 Le DHTML, pour servir ou desservir ?**

Il est parfois difficile de conjuguer rapidité et qualité dans une interface : j'en exposerai les raisons. Mais je montrerai aussi comment cela peut être malgré tout possible.

Il est bien connu que dans le monde de l'informatique, les délais de développement obligent souvent à ne se consacrer qu'aux priorités, ce qui revient souvent à négliger la qualité de l'IHM. C'est le cas aussi bien dans le développement des applications intranet que dans celui des logiciels grands public, dans lesquels les priorités se restreignent dans la plupart des cas à la garantie de la fiabilité du système et la réponse primaire aux besoins fonctionnels.

Or, lorsque l'IHM est bâclée, les applications intranet ne comblent pas pour autant les utilisateurs.

Il existe d'autres prétextes communément admis pour la médiocre facture d'une IHM : les améliorations graphiques sont souvent gourmandes en performance, et nécessitent de ce fait une concession entre puissance et rapidité. En l'occurrence, la contrainte de performance constitue souvent une excuse pour une charte graphique peu soignée. Mais dans la plupart des cas, c'est une fausse excuse.

Mon souci a néanmoins été d'éviter une sophistication trop poussée de mes objets d'IHM « maison », le risque étant de ralentir les performances ou encore de rendre le code moins maintenable.

#### **4.3.1 DHTML : un savant mélange de HTML, CSS et Javascript**

##### *4.3.1.1 Le HTML, un incontournable allié du web*

Il n'est sûrement pas nécessaire ici d'expliquer ce qu'est HTML, mais dans le doute rappelons qu'HTML signifie HyperText Mark-up Language : comme XML, il s'agit en effet d'un langage dit « à balises », c'est la base pour afficher du contenu web sur un site ou application internet ou extranet.

#### 4.3.1.2 Le CSS, pour séparer les données de la présentation

Même si le HTML permet de spécifier non seulement le contenu mais aussi la mise en page, il est dorénavant d'usage de recourir exclusivement au CSS (*Cascading Style Sheets*) pour la mise en page. Le CSS, maintenant dans sa version 3, permet en effet de spécifier que telles balises HTML<sup>97</sup>, devront toutes être affichées de telle façon<sup>98</sup>. L'intérêt est alors de rassembler tout le code CSS d'une application dans un seul fichier inclus (à l'extension *.css*) qui est appelée systématiquement dans toutes les pages : c'est la solution technique par excellence pour l'application d'une charte graphique dans un site ou une application web.

Une spécificité du CSS vaut le coup d'être notée ici, puisque je l'ai largement exploitée avec SATURNE : il est possible de définir une mise en page dédiée à la version imprimée, il suffit pour cela d'insérer dans une section à nommer « @media print » les instructions CSS qui seront appliquées seulement dans le cas d'une impression. Ainsi, il n'est pas nécessaire de recourir au Javascript pour optimiser le rendu d'une page web à l'impression !

#### 4.3.1.3 Le Javascript, un langage objet ?

À ceux qui prétendent que Javascript n'est pas un langage objet, il est temps de leur donner tort : il est réellement possible de programmer des classes en Javascript, avec des vraies méthodes et des vraies propriétés. Et c'est même conseillé.

La spécificité objet de Javascript est en effet une aubaine pour les développeurs faisant de ce puissant langage une utilisation privilégiée dans leurs applications web, car développer en objet a tous les avantages : c'est une démarche plus agréable, plus pratique et plus méthodique. Cela permet simplement de ne pas être perdu dans son code source lorsque celui-ci devient complexe.

En ce qui concerne SATURNE, j'ai par exemple créé un fichier de classe et de méthodes pour pouvoir instancier des objets de type « Chronogramme ». Ce n'était pas un luxe, étant donné la grande complexité du code Javascript gérant la génération et le comportement de la page de chronogrammes.

En sus, il est à noter que le langage Javascript en général revêt à mon goût une grande richesse applicative alliée à une grande simplicité d'utilisation, tout en respectant la même syntaxe que bien des langages modernes tel Java, PHP ou encore ActionScript. D'ailleurs, on parle là des mêmes avantages que ceux du langage PHP, mais transposés côté client.

---

<sup>97</sup> L'usage des « classes CSS » peut même cibler seulement certaines parties du code HTML.

<sup>98</sup> Les capacités en CSS sont devenues tellement foisonnantes que certains navigateurs (comme notamment Microsoft Internet Explorer, même dans sa version 8) ont du mal à suivre et n'ont pas encore implémenté certains ajouts pourtant brillants du langage CSS3, comme par exemple les ombrages ou les coins arrondis.

#### 4.3.1.1 Le faux problème de la compatibilité

Le problème bien connu de la compatibilité du code HTML, CSS ou Javascript avec la pluralité des navigateurs web est en général un faux problème dans le contexte du développement d'une application intranet, et c'est tant mieux. En effet, la plupart des intranets d'entreprise, comme celui de LA POSTE, ne s'emploie qu'avec le navigateur officiellement recommandé par la dite entreprise.

On échappe alors à une contrainte de taille en tant que développeur, celle consistant à devoir adapter son code DHTML aux autres navigateurs que celui sur lequel on a l'habitude de tester les développements en cours. Cela nous fait gagner un temps précieux, que l'on peut ainsi réinvestir pour apporter aux sites web dont on a la charge une plus grande qualité applicative.

### 4.3.2 Le DHTML, une aide précieuse pour enrichir l'interface de SATURNE

#### 4.3.2.1 Liste des blocs statistiques

The screenshot shows the 'Liste des blocs' (List of blocks) page in the SATURNE v2 application. The page is displayed in Microsoft Internet Explorer. The interface includes a navigation bar with 'Editions', 'Cumuls', 'Chronos', and 'Exports' tabs. A search bar is set to 'PLOUFRAGAN ST BRIEUC CTC'. The main content area displays a table of statistical blocks with the following data:

N°	Programme de tri	Tâche	Usager	Matériel	Machine	Début de session	Fin de session	Dur. sess	Injectés	Sel.
1	HMI 161 TG3 PERIPH	16160	00000000	HM11	HM11-2	Lun 15/11/2004 06:01:06	15/11/2004 07:01:56	1h 50s	18 455	✓
2	ELIT 394 TG2 PI	39488	40	ELIT	ELIT-1	Lun 15/11/2004 07:25:56	15/11/2004 08:04:29	38mn 33s	8 934	✓
3	ELIT 294 TG2 ECO	29494	30	ELIT	ELIT-1	Lun 15/11/2004 08:08:14	15/11/2004 08:27:50	19mn 36s	8 700	✓
4	ELIT 294 TG2 ECO	29494	30	ELIT	ELIT-1	Lun 15/11/2004 09:03:29	15/11/2004 10:29:45	1h 26mn 16s	39 213	✓
5	HMI TACHE 240 ATNF	24020	00000000	HM11	HM11-2	Lun 15/11/2004 09:11:57	15/11/2004 09:16:11	3mn 14s	0	✓
6	ELIT 294 TG2 ECO	29494	30	ELIT	ELIT-1	Lun 15/11/2004 10:46:20	15/11/2004 11:09:00	22mn 40s	8 576	✓
7	ELIT 394 TG2 PI	39488	40	ELIT	ELIT-1	Lun 15/11/2004 11:16:26	15/11/2004 11:21:28	6mn 3s	2 168	✓
8	ELIT 394 TG2 PI	39488	40	ELIT	ELIT-1	Lun 15/11/2004 11:24:04	15/11/2004 11:28:05	4mn 1s	0	✓
9	ELIT 394 TG2 PI	39488	40	ELIT	ELIT-1	Lun 15/11/2004 11:58:16	15/11/2004 12:00:46	2mn 30s	274	✓
10	ELIT 394 TG2 PI	39488	40	ELIT	ELIT-1	Lun 15/11/2004 12:58:19	15/11/2004 13:02:13	3mn 54s	17	✓
11	ELIT 394 TG2 PI	39488	40	ELIT	ELIT-1	Lun 15/11/2004 13:02:26	15/11/2004 13:14:10	11mn 44s	899	✓
12	HMI 261 TG3 ECO P...	26150	00000000	HM11	HM11-2	Lun 15/11/2004 13:10:12	15/11/2004 14:13:00	1h 2mn 48s	21 193	✓
13	ELIT 394 TG2 PI	39488	40	ELIT	ELIT-1	Lun 15/11/2004 13:15:05	15/11/2004 13:51:48	36mn 43s	11 270	✓
14	ELIT 294 TG2 ECO	29494	30	ELIT	ELIT-1	Lun 15/11/2004 13:52:35	15/11/2004 14:06:18	13mn 43s	5 686	✓
15	ELIT 294 TG2 ECO	29494	30	ELIT	ELIT-1	Lun 15/11/2004 14:07:12	15/11/2004 16:13:32	2h 6mn 20s	57 293	✓

The table summary indicates 46 total blocks, 46 retained (100%), and 7 invalid (15.2%). The total duration is 27h 44mn 16s, and the total number of injected items is 565 281. The interface also includes a 'Choisir les colonnes' (Choose columns) button and a 'Retenir tous les blocs' (Retain all blocks) button.

Figure 40 – SATURNE v2 : liste des blocs statistiques

#### 4.3.2.2 Listes à choix multiples

La liste à choix multiple proposée par le langage HTML a un inconvénient majeur : elle n'est pas intuitive à utiliser. En effet, comme les internautes assidus l'ont remarqué à maintes reprises, pour choisir plusieurs items à la fois, il faut maintenir la touche 'Ctrl' du clavier enfoncée.

J'ai donc entrepris de programmer en DHTML ma propre liste à choix multiple, que voici :



Figure 41 – Exemples d'utilisation d'une liste à multi-sélection

Un avantage de taille est qu'étant maître de mes développements, j'ai pu enrichir cette liste à choix multiple au fil des versions de SATURNE en y intégrant :

- Le total des sélections dans la liste, affiché en temps réel ;
- Une case à cocher pour tout sélectionner ou tout désélectionner ;
- Une infobulle spécifique au passage de la souris sur un item ;
- Une synchronisation possible entre deux listes à choix multiple (comme dans la deuxième copie d'écran de la figure présentée ci-dessus, dans laquelle la désélection d'un type machine entraîne de suite la disparition des machines de ce type dans la liste de droite ; et inversement).

#### 4.3.2.3 Affinage des blocs statistiques

Ce que j'ai baptisé « liste d'affinage » est sans doute l'un des procédés d'IHM les plus pratiques de ceux que j'ai mis en place dans l'interface de SATURNE.

Le fait de cliquer sur une entête de colonne fait apparaître en surimpression une de ces « listes d'affinage », permettant soit de saisir une expression, soit de décocher certaines valeurs. Le résultat consiste en un rafraîchissement immédiat<sup>99</sup> du tableau de données, ne contenant désormais que les valeurs qui intéressent l'utilisateur.

<sup>99</sup> Tout est géré en Javascript à base de tableaux associatifs, ce qui fait que le serveur n'est pas du tout sollicité durant les divers traitements d'affinage du tableau de blocs stats.

L'intensité de la couleur de fond de chaque item est proportionnelle à sa fréquence d'occurrence dans la liste des blocs :

Session	Dur. sess	Injectés	Sel.
04 09:15:11	3mn 14s		
04 14:13:00	1h 2mn		
04 14:40:17	26mn 4s		
04 16:14:58	1h 33mn		
04 17:46:00	56mn 4s		
04 18:31:34	45mn 1s		
04 23:49:02	1h 20mn		
04 03:33:02	32mn 22s	5 775	<input type="checkbox"/>

Figure 42 – Exemple d'utilisation d'une liste d'affinage (par la multi-sélection)

Cet exemple ci-dessus (liste d'affinage de type numérique) montre entre autres que d'affinage peut s'effectuer en cochant ou décochant des items : la liste des blocs statistiques se met à jour lorsque le curseur quitte la liste d'affinage.

Session	Dur. sess	Injectés	Sel.
04 07:01:56	1h 50s		
04 08:04:29	38mn 33s		
04 08:27:50	19mn 36s		
04 10:29:45	1h 26mn 16s		
04 11:09:00	22mn 40s		
04 11:21:28	6mn 3s		
04 13:14:10	11mn 44s		
04 14:13:00	1h 2mn 48s		
04 13:51:48	36mn 43s		
04 14:06:18	13mn 43s	5 686	<input checked="" type="checkbox"/>
04 16:13:22	2h 6mn 20s	57 293	<input checked="" type="checkbox"/>

Figure 43 – Exemple d'utilisation d'une liste d'affinage (par la saisie)

Cet exemple-ci met en évidence la possibilité de débiter sa saisie par le signe '>' ou '<', afin de n'afficher que les valeurs respectivement supérieures ou inférieures à un certain seuil.

#### 4.3.2.4 Éditions multilignes

Voici un exemple d'écran présentant un tableau de données dans SATURNE. Nous pouvons voir qu'il est possible de l'affiner (avec mise à jour en temps réel des résultats dans le bas de la page), le trier, l'imprimer ou même l'exporter sous Excel. Enfin, une image-flèche astucieuse, en haut à gauche, permet de fermer cet écran et revenir immédiatement à la liste de sessions de tri dont il est issu (car le chargement de ces écrans multiligne se fait toujours en surimpression d'une liste de blocs stats ou de chronogrammes).

**Trafic cumulé par séparation**

Période du : 15/11/2004 07:00 au 20/11/2004 02:00 Matériel : HM11-2

Recherche :

	Lots en sortie	Flux	Nb. obj
22100 DINAN	TRI DEFINITIF	TG2	450
22100 DINAN IMPORT	TRI DEFINITIF	TG2	557
22101 DINAN CEDEX	TRI DEFINITIF	TG2 CEDEX	26
22102 DINAN CEDEX	TRI DEFINITIF	TG2 CEDEX	18
22103 104 DINAN CEDEX	TRI DEFINITIF	TG2 CEDEX	23
22105 DINAN CEDEX	TRI DEFINITIF	TG2 CEDEX	11
22106 DINAN CDX	TRI DEFINITIF	TG2 CEDEX	19
22107 108 109 DINAN CDX	TRI DEFINITIF	TG2 CEDEX	23
DINAN 0001	TRI DEFINITIF	TG3 DINAN	19
DINAN 0003	TRI DEFINITIF	TG3 DINAN	24
DINAN 0004	TRI DEFINITIF	TG3 DINAN	23
DINAN 0005	TRI DEFINITIF	TG3 DINAN	39
DINAN 0006	TRI DEFINITIF	TG3 DINAN	35
DINAN 0007	TRI DEFINITIF	TG3 DINAN	17
DINAN 0008	TRI DEFINITIF	TG3 DINAN	27
DINAN 0009	TRI DEFINITIF	TG3 DINAN	15
DINAN 0010	TRI DEFINITIF	TG3 DINAN	25
DINAN 0011	TRI DEFINITIF	TG3 DINAN	31
DINAN 0013	TRI DEFINITIF	TG3 DINAN	31
DINAN 0014	TRI DEFINITIF	TG3 DINAN	21
DINAN 0015	TRI DEFINITIF	TG3 DINAN	22
DINAN 0016	TRI DEFINITIF	TG3 DINAN	24
DINAN 0017	TRI DEFINITIF	TG3 DINAN	25
DINAN 0021	TRI DEFINITIF	TG3 DINAN	29
DINAN 0022	TRI DEFINITIF	TG3 DINAN	12
DINAN 0023	TRI DEFINITIF	TG3 DINAN	19
DINAN 0024	TRI DEFINITIF	TG3 DINAN	17

Nombre de lignes retournées = 42      Total objets = 1 848

Figure 44 – Exemple d'édition multiligne

Notons que dans tout tableau HTML, l'intermittence de la couleur de fond permet une plus grande lisibilité. Mais un contraste trop grand entre les deux couleurs est à éviter : il faut garder une juste mesure afin que l'écran reste agréable à l'œil.

#### 4.3.2.5 Utilisation de bibliothèques extérieures

Le calendrier, qui permet dans SATURNE et SNT de choisir une date avec la seule aide de la souris, est issu d'une bibliothèque extérieure, que j'avais d'ailleurs déjà utilisée lors d'une mission précédente :



Figure 45 – Intégration d'un composant DHTML externe de type calendrier

#### 4.3.2.6 Chronogrammes

L'un des atouts majeurs que j'ai pu offrir à SATURNE est de présenter les données statistiques sous forme de graphiques riches, clairs et souvent interactifs. J'ai notamment migré la fonctionnalité « Chronogrammes » (que l'on trouvait déjà dans GIPSI STATS, mais réduits à leur plus simple expression, sous forme d'une image générée par le composant *GraphicServer*) vers une solution purement DHTML. Les chronogrammes représentent une frise chronologique des sessions de tri qui ont eu lieu sur une ou plusieurs machines, et sur une plage de temps choisie par l'utilisateur. On peut par exemple regarder l'activité d'une machine sur un bloc stats particulier, ou voir une vue d'ensemble des machines d'un centre de tri sur plusieurs jours.

Il est possible, grâce au DHTML, de ne montrer que les intervalles de chronogrammes (comme ceux présentés ci-dessous) correspondant à tel(s) code(s) couleur, et cela de manière immédiate. Il est également possible (et ça d'ailleurs été le cas), d'intégrer un menu contextuel aux différents intervalles de chronogrammes, afin de rendre immédiatement accessible l'accès aux différents écrans statistiques propre à l'intervalle<sup>100</sup> sur lequel a eu lieu le clic gauche ou le clic droit.

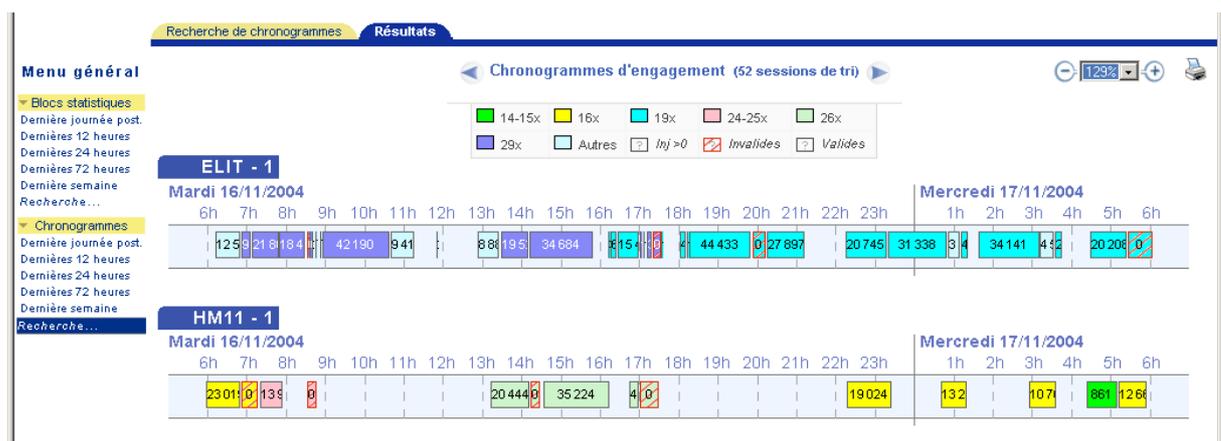


Figure 46 – Page de chronogrammes dans SATURNE

Depuis SATURNE V2, il est possible d'activer une option d'affichage consistant à insérer des « sous-intervalles » à l'intérieur de chaque intervalle, représentant ainsi la succession des temps de fonctionnement de la machine de tri. Si l'on passe la souris au-dessus d'un sous-intervalle, alors une infobulle apparaît (cf. copie d'écran suivante), et affiche la sous-durée de fonctionnement correspondante, en plus des informations générales relatives à la session de tri.

<sup>100</sup> Un intervalle représente le déroulement d'une session de tri.

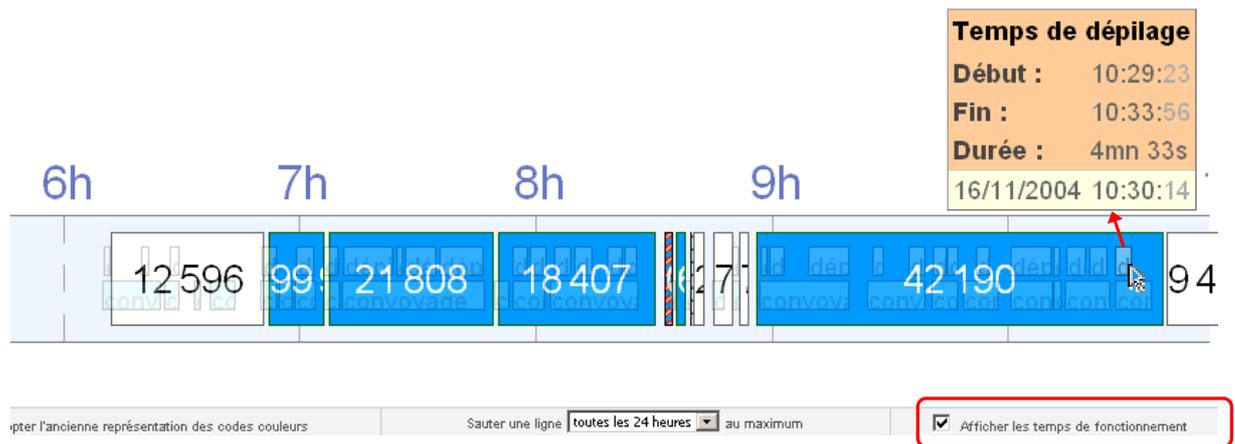


Figure 47 – Visualisation avancée d'un chronogramme dans SATURNE

*Remarque :* Une case à cocher (logée dans la barre de statut de la page des chronogrammes) permet d'activer ou de désactiver cette option en temps réel. Par défaut, cette case est décochée, notamment afin de ne pas compromettre d'office les performances de SATURNE en termes de rapidité d'affichage. Il est à noter que le mode d'affichage à appliquer par défaut est mémorisable dans une page de SATURNE servant aux utilisateurs à mémoriser leurs préférences.

#### 4.3.2.7 Histogrammes

Dans SATURNE, on trouve également des histogrammes, également migrés vers une solution technique DHTML, qui permettent de visualiser des statistiques telles que lots en sortie ou durées d'incidents sur des sessions :

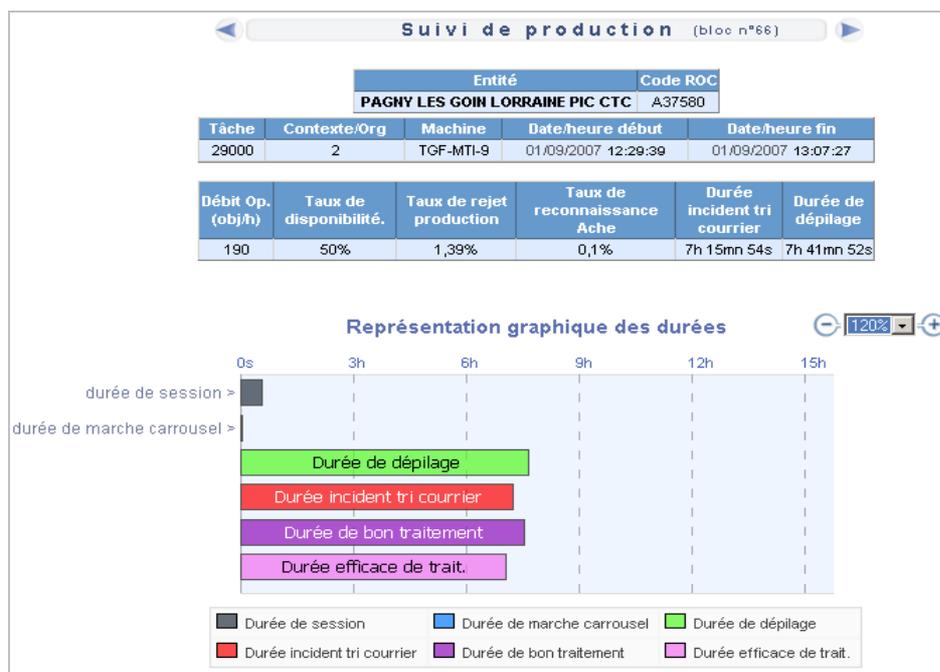


Figure 48 – Exemple d'édition avec histogramme dans SATURNE

#### 4.3.2.8 Autre graphiques

On trouve aussi dans SATURNE des matrices réceptacles, qui permettent de visualiser le nombre de plis en sortie des machines dans chaque réceptacle, ainsi que le nombre de bourrages : cf. page 115 (§4.3.2.11 « Autres procédés d'IHM »).

#### 4.3.2.9 Éditions

Une « édition », dans SATURNE, correspond à un écran affichant une vue statistiques. Les données sont généralement présentées sous forme de tableaux, contenant des valeurs de compteurs, des nombres de plis... En mode « cumulés », ce sont des sommes qui sont affichées, ou des moyennes s'il s'agit de pourcentages.

Statistiques GF (bloc n°10)					
Tâche	Usager	Machine	Date/heure début	Date/heure fin	Entité
29000	12	TOP-5	01/09/2007 06:25:43	01/09/2007 09:14:12	ORVAULT NANTES CTC

Machine	Objets injectés		Objets triés		Rejets
		37 257		35 132	

Antennes auto.	Objets présentés	Objets identifiés	Traités rejets	Rejets vidage	Vidage/Présentés
Antenne 1					
Antenne 2	13 876	10 068	2 467	25	0,18%
Antenne 3	9 565	6 854	1 543	40	0,42%
Antenne 4	13 805	10 049	2 794	40	0,29%
Total	37 246	26 971	6 804	105	0,28%

Carrousel	Durée tâche	Durée tâche h/vidage	Débit exploitation	Débit carrousel
		2h 48mn 29s	2h 48mn 29s	13 268 obj/h

	Durée marche	Durée traitement	Débit marche	Débit traitement	Taux d'utilisation
Antenne 1					
Antenne 2	2h 04mn 08s	2h 15mn 50s	6 707 obj/h	6 129 obj/h	80,62%
Antenne 3	1h 22mn 35s	1h 56mn 15s	6 949 obj/h	4 937 obj/h	69%
Antenne 4	2h 03mn 33s	2h 18mn 23s	6 704 obj/h	5 986 obj/h	82,13%

URA	Soumis URA	Rejetés URA	Taux recon. Ache.	Rejets+Part. Ache.	Rejets+Part. Distri
Antenne 1					
Antenne 2	13 851	2 442	72,69%	3 783	7 689
Antenne 3	9 516	1 503	72,03%	2 662	5 106
Antenne 4	13 764	2 754	73,01%	3 715	7 861
Total	37 131	6 699	72,64%	10 160	20 656

Vidéocodage	Soumis	Identifiés	Rejets URV	Taux codage	Vidéo/Inj.
Antenne 1					
Antenne 2	2 350	2 166	184	92,17%	15,61%
Antenne 3	1 428	1 386	42	97,06%	14,49%
Antenne 4	2 604	2 444	160	93,86%	17,7%
Total	6 382	5 996	386	93,95%	16,1%

Figure 49 – Exemple d'édition volumineuse dans SATURNE : « les stats GF »

4.3.2.10 Exports Excel/Word

Dans SATURNE, de nombreux écrans peuvent être exportés au format Excel ou Word. Cela permet d'éditer des documents qui ont des fonctions précises pour les utilisateurs des centres de tri (qui généralement, utilisent des macros sous Excel pour travailler les données brutes qu'ils exportent).

<b>JOURNAL DE BORD PRODUCTION MACHINE</b>						
<b>Période : Du 03/09/2007 06:00 au 04/09/2007 06:00</b>				<b>Lot client :</b>		
<b>Machine : TPF-APAS</b>				<b>Programme de tri : &lt;Plusieurs&gt;</b>		
Nombre de plis injectés	Nombre de plis non correctement traités	Nombre de plis correctement traités				
		Rejets URA	Plis en sous programme	Plis en tri définitif	TOTAL	
<b>1 810 261</b>	<b>37 243</b>	<b>96 230</b>	<b>478 509</b>	<b>1 195 960</b>	<b>1 770 699</b>	
Durée session		Durée de dépilage		Durée d'incident autre		Rupture d'Exploitation
<b>62h 54mn 06s</b>		<b>47h 08mn 31s</b>		<b>3h 01mn 39s</b>		<b>12h 43mn 56s</b>
DIVERS			DÉBIT		TRG	
Plis indexés	918 031	50,71%	D'ENTREE	38 400	Taux de " Marche "	79,84%
Plis chronomarkés	235 725	13,02%	OPERATIONNEL	33 376	Taux de " Depilage "	93,87%
			PRODUCTION	28 779	TRG Système	
Rjt URA	96 230	5,32%	Lettres injectées		Taux de " Tri Brut "	97,94%
Rjt MANUEL	19 046	1,05%	PRODUCTION	28 150	TRG Système	
Rjt MECA	3 897	0,22%	Plis correctement traités		TRG Production	
À RECYCLER	18 197	1,01%	PRODUCTION	19 013	Taux de " Tri Définitif "	66,07%
Bourrage	837		Plis définitivement triés		TRG Production	

Figure 50 – Exemple d'édition au format RTF dans SATURNE

Le procédé technique utilisé pour ces exports consiste à créer un modèle de fichier CSV (dont le format est ascii, et pas binaire), et écrire un code précis (ex. «\_{0120}\_») dans chaque case du tableau de statistiques qu'il contient. De cette manière, il est possible de générer un écran de statistiques « à la volée » en remplaçant simplement, en ASP, chacun de ces codes par la valeur à afficher.

#### 4.3.2.11 Autres procédés d'IHM

J'ai entre autres mis au point un procédé pour mettre en évidence les valeurs les plus grandes dans un tableau. Pour obtenir l'écran présenté ci-dessous (matrice réceptacles), il suffit simplement<sup>101</sup>, de faire varier l'intensité de la couleur de fond de la cellule en faisant en sorte que ce fond soit une image<sup>102</sup> en dégradé, dont la coordonnée interne  $y^{103}$  varie proportionnellement à la valeur affichée.

Recherche d'une matrice TG1 **Résultats**

**Matrice de flux TG1**

**Flux entrants et sortants de :**  
« RENNES DOTC, VANNES DOTC »

Période du : 15/11/2004      Format : <tous>  
au : 16/11/2004      Catégorie : <tous>

	RENNES DOTC		VANNES DOTC	
	Flux sortant	Flux entrant	Flux sortant	Flux entrant
MONTIGNY LE BRETO...	11 691	22 872	4 132	20 774
MONTIGNY LE BRETO...	11 691	22 872	4 132	20 774
MONTPELLIER CTC	3 972		736	
MONTPELLIER DOTC	7 475		1 393	
MONTREUIL SUR ILL...	1 066			
MORANGIS CC-T1	554			
MORDELLES CC-T1	4 067			
MULHOUSE CTC	2 540		466	
NANCY CTC	2 912		471	
NANCY DOTC	8 312		1 488	
NANTERRE CTC	3 289		6 078	
NANTERRE DOTC	21 728		7 256	
NANTERRE LADEFEN...	1 250		490	
NANTES DOTC	41 715		14 664	
NEVERS CTC	1 040		255	
NICE CTC	5 862		1 216	
NICE DOTC	11 088		2 924	
NIMES CTC	2 222		534	
NIORT CTC	3 596		875	
NOYAL SUR VILAINE...	16 593			
ORLEANS DOTC	15 051		4 186	

Figure 51 – Exemple d'édition « Matrice de flux TG1 » dans SATURNE

<sup>101</sup> Cela, via du code CSS généré en ASP.

<sup>102</sup> Il s'agit de la propriété CSS « background-image »

<sup>103</sup> Il s'agit de la propriété CSS « background-position »

#### 4.4 Beau et performant : une utopie ?

Certaines améliorations d'interfaces ont tendance à être gourmandes en performances, comme par exemple l'usage d'applets Java, d'images de taille importante, ou encore l'usage massif du Javascript.

Pourtant, il existe des solutions pour conjuguer performance du système et qualité graphique. Et c'est ce que je pense avoir réussi dans SATURNE, pour deux raisons principales :

- Les diverses optimisations qui ont pu être pratiquées en faveur d'une meilleure montée en charge (cf. chapitre précédent) n'ont en aucun cas obligé à altérer la richesse de l'interface. L'impact de ces optimisations se situe en effet à des niveaux plus bas que celui de l'IHM (code source, architecture, CPU, etc) ;
- Les problèmes de performances causés par une utilisation trop gourmande en CPU de l'interface de SATURNE ont pu être contournés de diverses manières, comme le montre le tableau suivant :

Élément d'IHM	Performances pénalisées	Solutions et compromis
Couleurs de fond, de police	<i>Aucune</i>	
Images	Risque d'une taille pénalisante pour les temps de réponse	Usage de la fonction d'optimisation d'images sous Photoshop (fonction « Enregistrer pour le web », vraiment efficace)
Tout	Le poids des très nombreux fichiers Javascript, CSS, au même titre que les images, pénalise les temps de réponse.	Externaliser le code Javascript, les images et les fichiers statiques dans des fichiers inclus, puis régler les paramètres du serveur web pour que le répertoire des fichiers inclus soit interrogés le moins souvent possible par le navigateur client <sup>104</sup> .
Tableaux dynamiques générés en Javascript	Délai médiocre (plus de 3 secondes d'attente)	Offrir deux modes de consultation des données : un mode d'affichage perfectionné,

---

<sup>104</sup> La partie 4.4.3 de ce mémoire (page 116) explique la problématique du cache navigateur.

		et un mode d'affichage basique <sup>105</sup> . <i>Inconvénients : Temps de développement, contrainte de maintenance, et plus grande source d'erreur.</i>
--	--	--------------------------------------------------------------------------------------------------------------------------------------------------------------

Tableau 12 – Axes d'amélioration de base des performances d'une IHM

Ainsi, les deux étapes de conception et d'optimisation de SATURNE ont chacune été l'occasion de prouver qu'une application peut être à la fois conviviale et performante. En même temps, cela n'a pu se faire qu'au prix d'un plus grand investissement de temps et d'énergie personnelle.

#### 4.4.1 La charte graphique, une base de l'ergonomie

La charte graphique d'une application web est un peu le point d'entrée de son ergonomie, puisqu'il implique la mise en forme de chacune des pages. Mais il est important de préciser dans ce chapitre que je ne vois pas en quoi une charte graphique aurait la moindre incidence sur les performances, à moins, bien sûr, que cette charte implique l'affichage d'images lourdes ou de nombreuses *iframes*, ou encore la simulation d'un OS multifenêtrage<sup>106</sup>. Dans SATURNE, ce n'est pas le cas.

#### 4.4.2 Les fichiers inclus, le réflexe du bon développeur

Il n'est peut-être pas nécessaire d'évoquer la nécessité incontestable de rendre générique la moindre fonction utilisée dans le code source de toute application web, aussi bien côté serveur (ASP, PHP, JEE...) que côté client (Javascript, CSS...).

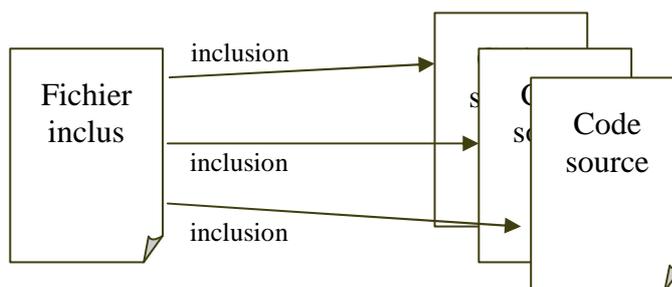


Figure 52 – Symbolisation de l'usage de fonctions générique

<sup>105</sup> Un exemple connu sur le web est celui des sites de bourse en ligne, qui proposent deux modes de consultations des courbes graphiques : l'un dynamique (mais plus lent à charger, et plus lourd), l'autre statique (mais plus rapide à charger, et plus sobre).

<sup>106</sup> Cela implique un très grand nombre d'objets DHTML chargés à l'écran et des dialogues Ajax gourmands, le tout entraînant un temps d'affichage pénalisant. J'ai connu cela lors de ma participation dans *Genese*, un autre projet LA POSTE dont les choix du chef de projet ont peut-être conduit à une utilisation abusive d'Ajax et du DHTML.

La généricité consiste à déclarer à un seul endroit de l'application toute fonction ou bout de code susceptible d'être utilisé dans plusieurs pages (même du code CSS). Cet endroit est en général un « fichier inclus » contenant des fonctions dites génériques, une sorte d'API personnelle, donc. Cette bonne pratique apporte de nombreux avantages, dont :

- Éviter les sources d'erreur (car si on modifie une fonction qui est déclarée en double dans une autre page web de l'application, on peut oublier d'effectuer cette correction aux autres endroits que celui dans lequel le besoin d'évolution s'est fait ressentir) ;
- Alléger le poids du code source téléchargé sur le navigateur client (puisque'une fonction écrite une seule fois prend moins de place qu'en double).

#### 4.4.3 L'importance du paramétrage du cache

De nombreux fichiers inclus sont ainsi appelés par chacune des pages d'une application web : on peut alors penser que cela oblige le navigateur client à télécharger inutilement, à chaque appel de page, l'intégralité des fichiers inclus (fichiers Javascript, fichiers CSS, tout comme des images comme la bannière). Pourtant, c'est nullement le cas, et cela grâce aux bienfaits prodigieux du cache navigateur, qui garde en mémoire chaque fichier Javascript, CSS, image ou autre fichier déjà téléchargé.

Mais j'ai découvert dans ma mission à LA POSTE que l'utilisation du cache navigateur était absolument à optimiser manuellement au moyen d'un réglage adapté du serveur web. Pourquoi ? Car le serveur web ne peut pas savoir par lui-même combien de temps il doit dire au navigateur de garder les fichiers en cache. Il est en effet possible avec le serveur web IIS (comme avec Apache, au moyen du module *mod\_expires*) de régler des temps spécifiques de rétention dans le cache navigateur pour tel ou tel type de fichier. Aussi, pour les images, qui ne changent normalement jamais, j'ai prévu une durée de rétention de 7 jours. Quant aux fichiers Javascript, j'ai choisi une durée de mise en cache de 2 heures. Cela nous impose donc, à chaque livraison en production d'une nouvelle version de SATURNE, d'interrompre la disponibilité de SATURNE durant au moins 2 heures, sans quoi les utilisateurs risqueront de surfer sur une nouvelle version de SATURNE mais avec des fichiers Javascript datant de l'ancienne version (ce qui peut provoquer des bugs). Et pour les pages ASP, qui doivent absolument rester dynamiques, il n'y a bien sûr aucune durée de rétention : le réglage du serveur doit préciser cela.

Le « nec plus ultra » est que, si la durée de rétention est dépassée, il existe un dialogue client-serveur caché et systématique, consistant simplement à demander au client de quelle heure date la dernière mise à jour de tel fichier. Et si cette heure de mise à jour diffère entre le client et le serveur, alors ce fichier n'est pas téléchargé inutilement par le navigateur.

#### 4.4.4 Procédés d'optimisation des objets DHTML

Le temps d'affichage de la page HTML suivante, dont le tableau d'informations comporte 650 lignes, est de 3 secondes. Ce temps est tout-à-fait correct compte tenu du fait qu'il s'agit d'une part d'un cumul (un temps non nul est mis par la procédure stocké pratiquant une instruction *GROUP BY* – groupement de données -) et d'autre part d'une longue liste.

Trafic cumulé par séparation			
Période du :	15/11/2004 07:00	au	20/11/2004 02:00
Matériel :	HM11-2		
Libellé séparation	Lots en sortie	Flux	Nb. obj
22 ST BRIEUC CTC T140	A RETRIER	TG2	8 445
22000 ST BRIEUC	TRI DEFINITIF	TG2	750
22001 ST BRIEUC CDX 1	TRI DEFINITIF	TG2 CEDEX	50
22002 ST BRIEUC CDX 1	TRI DEFINITIF	TG2 CEDEX	16
22003 ST BRIEUC CDX 1	TRI DEFINITIF	TG2 CEDEX	20
22004 ST BRIEUC CDX 1	TRI DEFINITIF	TG2 CEDEX	3
22005 ST BRIEUC CDX 1	TRI DEFINITIF	TG2 CEDEX	25
22015 ST BRIEUC CDX 1	TRI DEFINITIF	TG2 CEDEX	11
22016 ST BRIEUC CDX 1	TRI DEFINITIF	TG2 CEDEX	1
22017 ST BRIEUC CDX 1	TRI DEFINITIF	TG2 CEDEX	5
22021 ST BRIEUC CDX 1	TRI DEFINITIF	TG2 CEDEX	17

Nombre de lignes retournées = 650 Total objets = 820 302

Figure 53 – Affichage complet d'une édition multiligne dans SATURNE

Mais l'intérêt fonctionnel est de pouvoir affiner ou trier cette liste (chose qui n'était pas prévu dans GIPSI STATS, l'ancêtre de SATURNE).

Le défi, que j'ai pu relever avec succès, a donc consisté à ne pas recourir de nouveau à un dialogue serveur à chaque action d'affinage, mais d'afficher la liste dans un ordre différent ou alors la retrouver aux éléments contenant le texte saisi dans la zone de recherche.

Dans l'exemple ci-dessous (affinage sur les flux postaux à destination de Dinan), j'obtiens finalement un temps d'1 seconde, ce qui est parfait en termes de réactivité pour satisfaire l'utilisateur ; sans compter que cela économise de la ressource serveur, ce qui améliore donc la scalabilité générale de l'application.

dinan	Lots en sortie	Flux	Nb. obj
DINAN 0033	TRI DEFINITIF	TG3 DINAN	15
DINAN 0034	TRI DEFINITIF	TG3 DINAN	24
DINAN 0038	TRI DEFINITIF	TG3 DINAN	2
DINAN AUCALEUC 0026	TRI DEFINITIF	TG3 DINAN	29
DINAN BRUSVILY 0031	TRI DEFINITIF	TG3 DINAN	20
DINAN CALORGUEN 0025	TRI DEFINITIF	TG3 DINAN	17
DINAN LA VICOMTE 0037	TRI DEFINITIF	TG3 DINAN	29
DINAN PLEUDIHEN 0035	TRI DEFINITIF	TG3 DINAN	19
DINAN PLEUDIHEN 0036	TRI DEFINITIF	TG3 DINAN	13
DINAN ST SAMSON 0018	TRI DEFINITIF	TG3 DINAN	30
DINAN TREVIRON 0030	TRI DEFINITIF	TG3 DINAN	28

Nombre de lignes retournées = 42 Total objets = 1 848

Figure 54 – Affichage partiel d'une édition multiligne dans SATURNE

Le procédé utilisé consiste à cacher en Javascript chacune des lignes de tableaux ne contenant pas l'expression recherchée.

En revanche, pour ce qui est du tri, il m'a été impossible d'obtenir un temps de réponse aussi performant.

Libellé séparation	Lots en sortie	Flux	Nb. obj
YFFINIAC 0202	TRI DEFINITIF	TG3 YFFINIAC	59 055
A RECYCLER	A RECYCLER	REJETS	52 532
REJETS LECTURE	REJETS MANUELS	REJETS	21 173
ST BRIEUC 0707	TRI DEFINITIF	TG3 ST BRIEUC	18 915
RENOI T140 T 240 TG3 22100	A RETRIER	TG3	17 079
YFFINIAC 0021	TRI DEFINITIF	TG3 YFFINIAC	11 887
22 ST BRIEUC CTC T140	A RETRIER	TG2	8 445
REJET VIDAGE FIN DE MACHINE	A RECYCLER	REJETS	4 662
PLOUFRAGAN TREMUSON 0011 0012	TRI DEFINITIF	TG3 PLOUFRAGAN	3 810
PLOEUC 12 13 22 HENON	TRI DEFINITIF	_TG3 22150 PLOEUC	3 635
LOUDEAC LA MOTTE 0028 29 30	TRI DEFINITIF	TG3 LOUDEAC	3 568

Nombre de lignes retournées = 650 Total objets = 820 302

Figure 55 – Tri d'une édition multiligne dans SATURNE

Pour cause, trier un tableau HTML en Javascript ne peut se faire que de deux façons différentes :

- Solution A : on ajoute au code source de la page HTML un dictionnaire Javascript contenant l'intégralité des colonnes à afficher. Ensuite, si l'utilisateur demande un tri sur une colonne, cela reviendra à :
  1. Effacer le tableau HTML existant ;
  2. Trier le dictionnaire Javascript en mémoire vive sur le champ souhaité ;
  3. Le parcourir, et pour chaque occurrence créer une nouvelle ligne HTML.
- Solution B : le code source de la page ne contient que du code HTML. Si jamais l'utilisateur demande un tri ou un affinage, alors une étape s'insère en première position dans l'enchaînement présenté ci-haut : la lecture du tableau HTML visant à

créer en temps réel le dictionnaire Javascript dont on a besoin pour éviter un appel au serveur.

La solution B s'est finalement avérée la plus avantageuse car il vaut mieux privilégier le temps de chargement initial de la page. En effet, la solution A doublait le poids des fichiers HTML transitant du serveur vers le client, et ceux-ci ayant déjà une taille problématique, les temps de réponse de chaque page appelée s'en trouvent amoindris (même en environnement intranet).

Finalement, on obtient un temps de 4 secondes pour l'application du tri décroissant sur la dernière colonne. (Il est à noter que ce temps est plus rapide pour les tris successifs, puisque le dictionnaire Javascript est déjà constitué en mémoire vive.)

Comme dernier exemple, prenons ma liste à sélection multiple telle qu'elle est instanciée dans le but d'afficher une liste d'établissements : lorsque SATURNE a dû afficher beaucoup plus de centres de tri en raison de l'arrivée des centres de distribution dans la cible des utilisateurs, je me suis posé la question de la performance, puisqu'il s'agissait non pas d'afficher 130 centres, mais désormais 367. Heureusement, le temps d'affichage de cette liste multi-sélection de 367 items ne dépassait pas les 2 secondes.

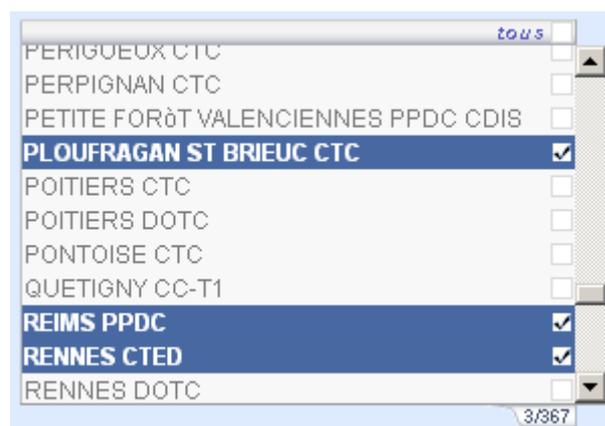


Figure 56 – Affichage d'une liste multi-sélection volumineuse dans SATURNE

#### 4.4.5 Pendant ce temps-là, que fait le serveur ?

DHTML n'est interprété que par la partie cliente des applications web.

Le serveur web, lui, intervient d'abord au début de la session ou du lancement d'une autre fonctionnalité. Cette intervention ressemble à une initialisation : elle a pour but d'envoyer au client l'ensemble des pages HTML/CSS/Javascript et images nécessaires au navigateur pour gérer l'affichage et les interactions utilisateurs via DHTML.

Il intervient également au cours de la session, notamment quand l'utilisateur soumet des informations ou lorsqu'il émet une nouvelle requête. D'ailleurs, en DHTML, il existe une

façon très intéressante de faire communiquer la partie cliente d'une application internet/intranet et son serveur web sans que cela se traduise par le rafraîchissement d'une page : il s'agit d'Ajax<sup>107</sup>.

Enfin, le serveur intervient généralement à la fin de la session, pour libérer de la mémoire vive l'environnement propre à un client qui vient de se déconnecter (ce que l'on appelle les variables de session).

#### **4.4.6 Conclusion sur la problématique ergonomique**

L'ergonomie dans SATURNE a été l'objet d'une mise en pratique intéressante. À cette occasion, je pense avoir poussé la qualité de l'ergonomie beaucoup plus loin que le minimum requis, ce qui m'a valu une reconnaissance de la part de la MOA et des autres équipes de la MOE.

Pour ceux qui douteraient qu'il est possible de faire des interfaces web à la fois belles, puissantes, interactives et performantes, on peut répondre que le DHML ne dessert jamais, dès lors que l'on est capable de s'en servir avec justesse.

### **4.5 Beau et performant, ça ne suffit pas**

#### **4.5.1 Facilité de maintenance et réutilisabilité**

Beau et performant oui, mais : réutilisable ? Facile à maintenir ? J'ai également veillé à cela dans le code source dans SATURNE, et ceci en créant des bibliothèques pour chaque objet.

La belle théorie voudrait qu'on rende partageable les objets graphiques entre plusieurs applications (exemple : SATURNE, SNT et SPOT pourraient partager la même bibliothèque de fichiers Javascript permettant d'afficher des chronogrammes). En effet, cela aurait pour effet de centraliser la correction des bugs ainsi que le développement des améliorations. Mais je doute que dans la pratique cela fonctionne bien, car :

- On peut évoquer la problématique de la localisation des fichiers : faudrait-il opter pour une centralisation ou une démultiplication de celui-ci ?

*Hypothèse de la centralisation :*

- ❖ Elle entraînerait le coût d'un serveur web dédié pour les bibliothèques ;

---

<sup>107</sup> Cf. partie 4.6.3, page 118.

- ❖ Elle fragiliserait la stabilité du système : une panne sur ce serveur influerait sur l'ensemble des serveurs web y faisant appel.

*Hypothèse de la démultiplication :*

- ❖ À chaque fois qu'un code source évolue (ajout d'une méthode à un fichier de classe Javascript, correction d'une anomalie...), il faut penser à le distribuer à toutes les applications qui l'utilisent.
- Qu'en est-il de la nécessité de faire évoluer pour chaque application les classes et méthodes de ces objets partagés, en fonction des besoins de chacune : y a-t-il un responsable de l'objet ? On peut évoquer la facilité de maintenance, qui ne serait guère au rendez-vous puisqu'aucune application ne serait responsable du code source des objets partagés auxquels il ferait appel, ce qui implique une interaction entre les différents programmeurs : cela engendrerait une interdépendance entre eux, ainsi qu'une perte de temps.
- Enfin, une question anecdotique se pose : dans un projet mené à LA POSTE, est-il préférable d'écrire les commentaires et noms de variables en langue anglaise ? A priori, on aurait raison de penser que non, car LA POSTE est typiquement un univers « franco-français », et au vu du temps et de l'énergie nécessaires à la traduction ou la compréhension de l'anglais dans le code source, cela ne vaut probablement pas le coup. En même temps, il serait judicieux de tenir compte d'un argument de poids qui ferait peser nettement la balance en faveur de l'anglais : nous avançons de plus en plus vers une externalisation de nos développements à l'étranger, et dans de telles circonstances nous aurions beaucoup à gagner que notre code source soit compréhensible par autrui afin que sa reprise soit facilitée.

## **4.5.2 Aide en ligne**

### *4.5.2.1 Nécessité de l'aide en ligne*

Que serait une application « belle et performante » si le contenu de celle-ci n'était pas expliqué aux utilisateurs ?

En reprenant GISPI STATS, dont l'aide en ligne était particulièrement pauvre, je me suis fait force de proposition sur cette partie. Notamment, j'ai remarqué que les formules de calcul des diverses données statistiques présentées dans les écrans n'étaient en général pas documentées de manière précise dans l'aide en ligne.



Figure 57 – Exemple de page de l'aide en ligne contextuelle de SATURNE

#### 4.5.2.2 Un glossaire pour SATURNE

Aussi, j'ai entrepris d'intégrer un glossaire dans l'aide en ligne. Ceci, afin de ne maintenir aucun secret dans l'interface de SATURNE et que tout utilisateur puisse savoir à tout moment la signification de tel terme ou le mode de calcul de tel indicateur. Pour assurer la publicité de ce glossaire, un lien permanent dans la bannière de SATURNE permet d'y accéder directement. J'ai divisé le glossaire en trois parties :

- Les termes généraux du tri postal (vidéocodage, TG1...);
- La définition de chaque indicateur statistique, ainsi que sa formule de calcul (durée de marche, taux de rendement global...);
- Les éléments de l'interface de SATURNE (liste d'affinage...).

#### 4.5.2.3 L'aide en ligne : un exercice d'IHM à part entière

En réalisant l'aide en ligne de SATURNE, un constat est apparu : celle-ci constitue à elle seule une véritable petite application web. En effet, d'une grande navigabilité, les fonctions qu'elle met à disposition exploitent les possibilités du DHTML tout autant que celle du langage serveur, ASP, dont je me suis servi pour créer de la généricité. Voyons en quoi consiste celle-ci dans le contexte de l'aide en ligne.

#### 4.5.2.4 L'aide en ligne : l'importance de la généricité

Grâce à mes efforts de conception, l'aide en ligne HTML de SATURNE relève le déficit de documenter plusieurs centaines d'indicateurs statistiques présents dans les différents écrans de l'application : chacun de ces indicateur est documenté sous la forme d'une définition suivie d'une formule de calcul. Or, dans SATURNE, la formule de calcul d'un indicateur fait elle-même très souvent appel à d'autres indicateurs, de même qu'un indicateur est généralement présent dans plusieurs autres écrans statistiques : il est alors crucial de ne pas tomber dans le piège du copier-coller des définitions HTML !

Pour ce faire, j'ai rassemblé dans l'ordre alphabétique toutes les définitions des termes et indicateurs statistiques et les ai positionnées à suivre dans une unique page HTML, qui n'est ni plus ni moins le « glossaire » de SATURNE. Dans cette page de référence, chaque définition de terme ou indicateur est imbriquée dans des balises HTML spécifiques permettant à mon système de voir cette page comme une sorte de base de données hypertexte, sachant qu'on utilise les expressions régulières pour identifier les définitions auxquelles accéder. Cela permet, sans dédoubler la moindre définition, d'inclure n'importe quelle d'entre elles à tout endroit possible de l'aide en ligne. Cela permet également d'afficher une infobulle contenant une définition au survol de la souris sur celle-ci, et ce, au moyen d'une simple fonction Javascript.

#### 4.5.2.5 L'aide en ligne a ses limites

Les audiences des pages consultées ainsi que certains témoignages directs m'ont permis de me rendre compte que la plupart des utilisateurs font l'impasse sur l'aide en ligne. J'ai tenté d'en étudier les raisons, ce qui m'amène au constat suivant. De part l'attention particulière avec laquelle j'ai veillé dans l'ergonomie de SATURNE, les utilisateurs ont le sentiment de maîtriser intuitivement l'IHM, ce qui les amène à faire l'économie de l'aide en ligne. C'est dommage, car du coup ils font l'impasse sur des astuces d'IHM qui pourtant leur ferait gagner beaucoup de temps<sup>108</sup>.

---

<sup>108</sup> Comme par exemple la possibilité de saisir le numéro de tâche « 2\* » (pour filtrer la liste des blocs sur les plis économiques) au sein de la liste d'affinage de la colonne « tâche » de la liste des blocs, et pas via le

## 4.6 Et le « web 2.0 » ?

### 4.6.1 Qu'est-ce que le « web 2.0 » ?

Pour certains, c'est un effet de mode inventé par les médias pour désigner les nouveaux outils et services web que sont blogs, podcasts, vidéo en ligne, communautés virtuelles, etc. Pour d'autres, c'est une réalité tangible dans le monde de l'informatique, car basée sur des apports conceptuels et technologiques pratiques. De quoi s'agit-il donc ?

Socialement parlant, le web 2.0 évoque l'aspect communautaire et collaboratif d'un site web. Olivier Fécherolle, Directeur Général de Viadeo, rappelle un des grands principes du 2.0 : « *la conception de l'entreprise comme un écosystème ouvert* ». Selon lui, pour être crédible et efficace dans sa démarche communautaire, l'entreprise doit mettre en place des outils facilitant la transparence, l'ouverture et le ton appropriés. En termes de communication extérieure, cette « Entreprise 2.0 », doit d'autant plus respecter ces règles qu'elles influent directement sur son image de marque.

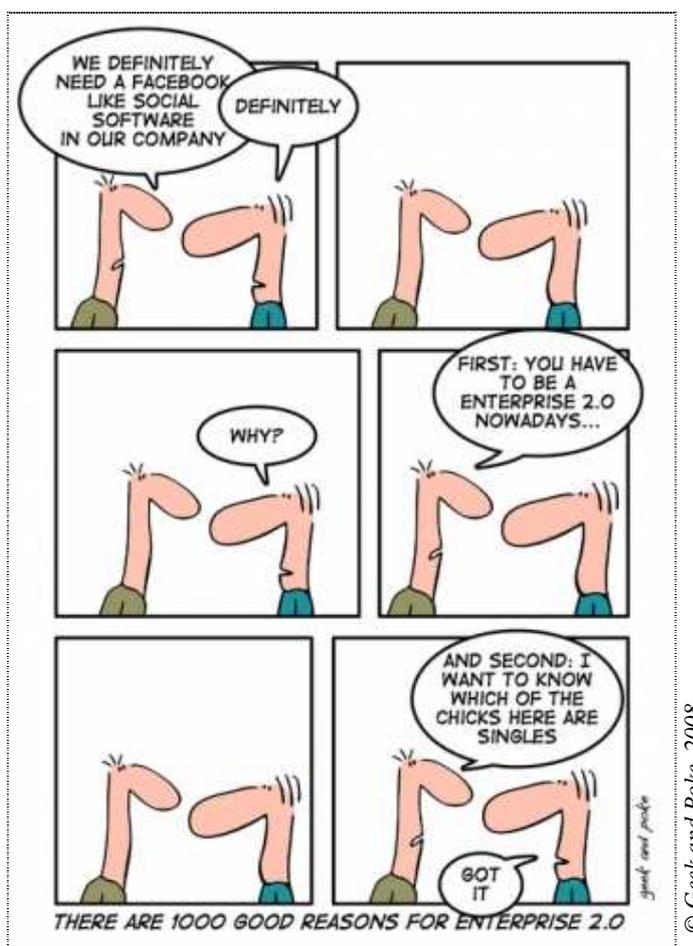


Figure 58 – « Les bonnes raisons de devenir une Entreprise 2.0 »

---

formulaire de recherche, qui entraîne un temps de traitement plus long et une nouvelle sollicitation du serveur.

#### 4.6.2 SATURNE est-il un intranet 2.0 ?

Nous pouvons croire qu'une application intranet de LA POSTE telle SATURNE est située loin de toute notion de réseau social, ni d'aspect collaboratif, et, en cela, loin du « web 2.0 ». Mais détrompons-nous, la définition du concept n'est pas aussi étriquée.

##### 4.6.2.1 L'intranet 2.0 vu de l'extérieur

Pour savoir si SATURNE est un intranet 2.0, posons-nous d'abord la question sur ce qu'est un intranet 2.0 par rapport à un intranet de première génération.

Stéphane Rossard, Fondateur de Capeotravel<sup>109</sup>[WEB 09] a défini plusieurs signes permettant de savoir à quelle étape une entreprise en est précisément dans la réalisation de son intranet :

- Un intranet 2.0 est « profilé ». Autrement dit, il part de l'individu et de sa connaissance qu'on en a, que ce soit en termes de métier, de statut et de centres d'intérêt<sup>110</sup> ;
- Un intranet 2.0 est organisé de façon structurante, c'est-à-dire qui répond aux trois conditions suivantes : le recours aux nouvelles technologies, la formalisation des procédures (qui met à jour quoi, et de quelle façon) et le fait que ces procédures soient portées par le management ;
- Un intranet 2.0 implique un pilotage par les indicateurs grâce à une traçabilité précise (cession, identifiant...) qui permet d'obtenir une « photographie de très haute résolution » sur les usages, ce que les collaborateurs utilisent ou pas et leurs attentes. D'où une meilleure optimisation du fonctionnement de l'intranet 2.0 par une pertinence des contenus et une pertinence de l'organisation.

Il se trouve qu'en dépit d'une dimension collaborative, SATURNE répond en tout point aux trois règles définies ci-dessus. On peut notamment évoquer en exemple l'outil de trace que j'ai créé et intégré à SATURNE, permettant de journaliser en permanence les actions utilisateurs, puis de les afficher dans un module de surveillance permettant de grouper ces statistiques par profil utilisateur. Ce module m'a permis de mieux appréhender le besoin des utilisateurs en centre de tri, et ainsi d'affiner les fonctionnalités de SATURNE ou d'en proposer de nouvelles.

##### 4.6.2.2 Le web 2.0 vu par le développeur

Dans SATURNE, j'ai également appliqué la définition technique du web 2.0, qui n'est ni un standard ni une technologie, mais plutôt une série de principes d'utilisation concernant des technologies existantes. En effet, le principe technique général du web 2.0 est qu'une page

---

<sup>109</sup> Capeotravel (<http://www.apeotravel.com/>) est une société fournissant des voyages sur mesure.

<sup>110</sup> Comme le rappelle l'Observatoire de l'Intranet, le « mieux » remplace le « trop » d'information.

web échange des données avec le serveur sans pour autant recharger entièrement la page à chaque interaction ou rafraîchissement (ce qui dans la pratique est implémenté en Ajax<sup>111</sup>).

Le terme a été inventé en août 2004 par un certain Dale Dougherty<sup>112</sup>... Et une fierté personnelle est d'avoir fait du web 2.0 sans le savoir, et surtout avant même que le terme eut été inventé ; En effet, ma mission précédente, en 2001<sup>113</sup>, m'avait amené à trouver des solutions techniques équivalant à ce concept<sup>114</sup>.

Bridé par le parc de navigateurs installés, le web 2.0 s'appuie donc sur des technologies que l'on peut qualifier sans risques de « mûres » :

- HTML (dont la dernière version date de décembre 1999) ;
- Un sous-ensemble de CSS 2.0 supporté par Internet Explorer (la recommandation CSS 2.0 a été publiée en mai 1998) ;
- Javascript (introduit par Netscape dans son navigateur en 1995) ;
- XML (recommandation publiée en 1998) ;
- Syndication Atom ou RSS (RSS –feuilles de style- a été créé par Netscape en 1999) ;
- Protocole HTTP (dernière version été publiée en 1999) ;
- Identifiants universels URI<sup>115</sup> (publiés en 1998) ;
- REST<sup>116</sup> (thèse publiée en 2000) ;
- Services web (les premières API XML-RPC pour Javascript ont été publiées en 2000).

Du point de vue du développeur, le web 2.0 est donc l'appropriation de technologies âgées de dix ans pour apporter une ergonomie différente aux sites internet et intranet. Donc, si c'est une révolution, c'est une révolution dans l'utilisation des technologies et non une révolution des technologies elles-mêmes.

---

<sup>111</sup> L'Ajax est un procédé technique permet à une page web d'en invoquer une autre sans que cela soit visible) – cf. partie 4.6.3, page 118.

<sup>112</sup> Dale Dougherty, de la société O'Reilly\_Media, avait un brainstorming avec Craig Cline de MediaLive pour développer des idées pour une conférence conjointe. Il a suggéré que le web était dans une période de renaissance, avec un changement de règles et une évolution des business model.

<sup>113</sup> Je m'employais alors, pour le répartiteur pharmaceutique OCP, à trouver des solutions de 'profiling' et d'informatisation de tâches administratives diverses sur leur intranet.

<sup>114</sup> Dès 2001, mes développements web répondaient à la définition de l'Ajax.

<sup>115</sup> En gros, les liens hypertextes

<sup>116</sup> REST est une architecture d'échange asynchrone de données reposant sur le protocole HTTP.

### 4.6.3 Vous avez dit Ajax ?

Un an plus tard (en 2005) fût donc inventé le terme « Ajax », signifiant *Asynchronous JavaScript and XML*. Ce mot désigne la manière concrète dont se fait le dialogue sous-jacent entre un serveur web et un navigateur dans une application web nouvelle génération. Une communication conforme au concept « Ajax » se fait en mode asynchrone, via l'objet *XMLHttpRequest* manipulé en Javascript, dans lequel sont injectées des données au format XML, puis qui est envoyé sur le réseau via le protocole habituel (HTTP). Un exemple connu est celui des messageries web modernes (Zimbra, Hotmail, etc) permettant de manipuler les mails exactement comme si on était dans une application traditionnelle comme Outlook.

#### 4.6.3.1 Les avantages d'Ajax

Dans le mode traditionnel d'échange de données du web, une requête est envoyée au serveur web qui agit en fonction de l'action de l'utilisateur et des données reçues, et renvoie une nouvelle page. Dans le jargon du web, ce type de requête est dite « synchrone ». Avec Ajax, ce mode de communication entre client et serveur web est révolu pour deux raisons. D'une part, ce n'est pas la page entière qui est renvoyée au client à chaque interaction, mais seulement les données utiles. Les applications sont alors plus réactives, car la quantité de données échangées entre le navigateur et le serveur web ainsi que le temps de traitement de la requête côté serveur sont tous deux réduits (une partie du traitement est réalisée en Javascript sur l'ordinateur d'où provient la requête). D'autre part, le fait d'avoir des échanges asynchrones entre client et serveur suppose que l'utilisateur ne subisse plus d'interruption, ce qui se rapproche finalement du mode événementiel en vigueur avec les applications GUI (Graphic User Interface).

#### 4.6.3.2 Ma mise en garde sur l'utilisation d'Ajax

Mon projet à LA POSTE m'a appris qu'Ajax n'est pas à utiliser de manière systématique pour implémenter n'importe quel type de dialogue entre le navigateur et le serveur web. En effet, les performances côté client seront plombées notamment dans le cas du chargement et l'affichage d'une liste de données volumineuse : cela vient du fait que le rafraîchissement des données prend beaucoup plus de temps via Ajax (donc via DHTML<sup>117</sup>) que dans un dialogue classique où la page entière est rechargée.

---

<sup>117</sup> Exemple-type d'un code DHTML utilisant Ajax :

```
document.getElementById("data_list").innerHTML = <code_provenant_de_dialogue_ajax>;
```

#### 4.6.4 Le « web 2.0 » est-il techniquement une révolution ?

Ma réponse sera assurément : Oui ! Aussi, il est intéressant ici d'identifier chacune des affirmations présentes en 2003/2004 dans le cours C.N.A.M. de Jean-Marc PUJOS [IHM 03], pour montrer à quel point ces dernières ne correspondent plus du tout au contexte du web 2.0... ce qui, bien évidemment, les rend dépassées pour bon nombre d'entre elles puisque depuis 2004, l'eau a coulé sous les ponts. En effet, ce que l'on pouvait affirmer il y a 10 ans voire un peu moins à propos du web est effectivement à remettre totalement en question puisque le navigateur web est devenu plus que jamais le « client universel » par excellence, les potentialités du DHTML alliées à la puissance des PC actuels ayant considérablement explosé.

Affirmations de J.-M. PUJOS	Remise en question avec le web 2.0
<i>Contrairement aux GUI<sup>118</sup>, le web n'offre presque pas de possibilité de manipulation.</i>	Les versions récentes des langages utilisés en DHTML permettent depuis plusieurs années aux interfaces internet/intranet de véritablement manipuler les objets de la page, et cela sans avoir recours à des applets Java ou des contrôles ActiveX de type <i>Flash</i> . Par exemple, dans les applications de messagerie web comme Gmail ou Zimbra, lorsque l'on déplace des mails dans un dossier, cela se fait par un glisser-déplacer exactement comme dans Outlook <sup>119</sup> .
<i>Les possibilités sont réduites mais suffisantes.</i>	Il est pourtant possible de créer ses propres objets graphiques, sans pour autant qu'ils soient lents et incompatible avec certains navigateurs. Par exemple, j'ai pu créer dans SATURNE une liste à choix multiple, sachant qu'un tel objet n'a pas été prévu par le langage HTML. De même, on ne compte plus le nombre de sites web disposant d'une barre de menus dont le comportement est semblable à celles des applications « GUI » (arborescence, déroulement...).
<i>Il ne se passe rien à l'intérieur d'une page web sans que l'utilisateur ne clique sur un lien ou sur un bouton.</i>	C'est justement là le seul but du DHTML, et plus encore, de l'Ajax <sup>120</sup> : permettre à une page de rafraîchir les informations et les objets graphiques qu'elle contient sans pour autant faire appel au serveur pour un nouveau chargement de page.
<i>Les combinaisons de touches et raccourcis clavier ne sont pas possible avec le web.</i>	Pourtant, l'objet <i>Event</i> (gérant les événements souris ou clavier) est là pour remédier à ce besoin. Par exemple dans SATURNE, on peut faire « F1 » pour obtenir l'aide en ligne, en plus du lien dédié dans la bannière.
<i>Le WISIWIG<sup>121</sup> n'est pas possible avec le web.</i>	Aujourd'hui, beaucoup de sites comme les messageries web telles que Gmail ou Zimbra n'intègrent-ils pas une zone d'édition de texte avec mise en forme ? Pourtant, celles-ci sont faites exclusivement en DHTML.

Tableau 13 – Remise en question des anciens prédicats du web

<sup>118</sup> « Guy User Interface » (application graphique traditionnelle), opposé aux « Network User Interface » (application graphique internet/intranet)

<sup>119</sup> Programmer en DHTML les possibilités habituellement disponibles d'emblée dans les logiciels traditionnels, tels que le glisser-déplacer est très compliqué d'autant que le code doit s'adapter à la diversité des navigateurs. Cependant, il existe des bibliothèques Javascript gratuites telles que *MooTools* ou *JQuery*, épargnant ce travail au développeur web.

<sup>120</sup> Cf. partie 4.6.3, page 118.

<sup>121</sup> « *What I See Is What I Get* » : WISIWIG désigne la manipulation des objets dans la page en temps réel, à l'instar des applications de bureautique). Exemple : la saisie d'un mail avec mise en forme dans Gmail.

Certes, dans son cours de 2003/2004, Jean-Marc PUJOS parlait des limites du web en des termes réducteurs, mais il affirmait lui-même qu'il « existe des possibilités pour réduire cette placidité », en mentionnant d'ailleurs le DHTML comme sauveur historique.

Aujourd'hui, on pourrait bien dire que cette « placidité » est tout simplement réduite à néant. C'est en cela qu'on peut considérer que oui, le web 2.0 constitue bel et bien une révolution...

Néanmoins, il faut reconnaître que pour des applications à très fort besoin en termes d'IHM (rafraîchissement de graphiques lourds en temps réel, etc.) les applications traditionnelles (GUI) ont tendance à garder une avance sérieuse, notamment en termes de rapidité d'affichage : c'est sur ce point que j'ai senti les principales limites du développement en DHTML dans SATURNE.

Enfin, notons un tout dernier bémol, et je dirais plutôt un avertissement, dans la conception des applications web : comme le dit lui-même Jean-Marc PUJOS, « il suffit de peu pour dériver des standards du web ». Autrement dit, comme j'ai pu moi-même le ressentir en développant SATURNE, l'intérêt du DHTML nouvelle génération est que tout devient possible avec le web, et donc nous sortons facilement des objets standards prévus par le langage HTML, préférant tantôt développer les nôtres, tantôt utiliser les bibliothèques Javascript libres comme *mootools* ou *JQuery*<sup>122</sup>. Mais gardons conscience que plus une application est sophistiquée, plus elle est difficile à maintenir... De plus, respecter un standard veut généralement dire que la page sera lue par tous les navigateurs. Ce qui, si on ne le fait pas, représente un risque pour la portabilité de l'application, particulièrement si c'est une application internet ; généralement dans les intranets, le navigateur est clairement identifié et maîtrisé<sup>123</sup>. Au final, il ne faudrait sortir des standards du web que si ça en vaut vraiment le coup, et s'assurer dans tous les cas que l'application que l'on développe se comporte de manière égale et fluide sur l'ensemble des navigateurs.

---

<sup>122</sup> Nous sommes maintenant en 2010, et ces librairies n'existaient pas (ou peu) lors du développement de SATURNE.

<sup>123</sup> En dépit de cette généralité, il existe dans le cas de l'application SATURNE une minorité d'utilisateurs n'utilisant pas la version recommandée du navigateur (ex. IE7 au lieu de IE6, alors que le fonctionnement de l'application suite à un changement de version n'a pas été garanti par nos soins).

## Chapitre 5. UNE MÉTHODE COMME UWE AURAIT-ELLE ÉTÉ UTILE ?

Les spécifications du projet SATURNE n'ont pas été réalisées avec une « véritable » méthode de conception web, comme peut l'être parmi d'autres la méthode UWE<sup>124</sup>. Je me suis initié à cette dernière sous les conseils de mon tuteur CNAM (Jacky AKOKA), qui m'a donné comme objectif de dégager une analyse synthétique du bien fondé d'une méthode de conception web comme UWE, par rapport à la démarche employée dans le projet SATURNE, que les puristes de l'ingénierie pourraient qualifier d'« empirique ».

Mais, avant d'évoquer UWE, faisons d'abord un point sur la méthode que j'utilise à LA POSTE dans la conception de l'application SATURNE.

### 5.1 Méthode de conception actuellement employée dans SATURNE

#### 5.1.1 Modélisation de l'architecture

On modélise l'architecture en la schématisant, mais sans utiliser de méthode précise si ce ne sont les normes de documentation de projet prévues par LA POSTE, néanmoins très précises : ces spécifications font l'objet d'un DAL (*Dossier d'Architecture Logicielle*) et d'un DAT (*Dossier d'Architecture Technique*), le contenu de chacun de ces documents respecte une table des matières type.

#### 5.1.2 Modélisation des données

À LA POSTE, on modélise les données à l'aide d'AMC Designer. Cet outil permet de :

1. Spécifier les données en Merise – production d'un MCD<sup>125</sup> ;
2. Lancer une transformation automatique de ce MCD en un MPD<sup>126</sup> ;
3. Enfin, lancer la génération automatique de ce MPD en un script SQL de création des tables, colonnes et index au format de base de données de son choix<sup>127</sup>.

---

<sup>124</sup> UML-based web engineering

<sup>125</sup> Modèle conceptuel de données

<sup>126</sup> Modèle physique de données

<sup>127</sup> Le choix de l'éditeur/version de la base de données se fait plus exactement au moment de générer le MPD.

La bonne habitude a été prise de maintenir à jour le MCD de SATURNE à chacune des évolutions ou corrections mettant en œuvre une modification de la structure de la base de données. À chacune des évolutions de ce MCD (visible en *Annexes n°1* de ce mémoire), on insère ce dernier dans le *Document de Conception Générale*.

### 5.1.3 Modélisation des besoins fonctionnels

On modélise les besoins fonctionnels, la navigation entre pages et le contenu (présentation) de chaque page web, mais sans utiliser de méthode précise : ces spécifications néanmoins précises sont rédigées dans Word, agrémentées de schémas, et sont intégrées au DCG.

Il est aussi d'usage d'intégrer dans le DCG des copies d'écran de ce dont on souhaite disposer à terme. Ces copies d'écran sont la plupart du temps issues d'un début de développement web côté client ayant abouti à des pages temporairement statiques, dont le choix et la disposition des éléments d'interfaces (tableaux, boutons, menus...) sont néanmoins fidèles à ce que l'on souhaite proposer à la MOA.

## 5.2 Mise en pratique de la méthode UWE avec le projet SATURNE

### 5.2.1 Qu'est-ce qu'UWE ?

UWE signifie « UML-based Web Engineering » et désigne une méthode permettant de modéliser une application web dans tout son cycle de développement.

Il s'agit d'une extension d'UML, voire même d'un « profile UML » puisque UWE utilise les 3 mécanismes d'extension suivants :

- Stéréotypes (classes UML spécifiques UWE, comme par exemple les « menus », donnant la possibilité de modéliser en 1 clic la barre de menus d'une application web)
- Tagged values (attributs spécifiques UWE dans ces stéréotypes)
- Contraintes OCL (langage UML pour décrire les conditions d'enchaînement dans les diagrammes)

Une spécification UWE est donc semblable à une spécification en UML, avec en plus des classes ou diagrammes écrits dans un formalisme faisant l'objet d'une extension d'UML : extension justement appelée UWE dans notre cas ; cela en vue d'une meilleure adaptation aux applications web.

Suivre la méthode UWE revient à produire dans l'ordre les modèles suivants (*je me suis essayé à la plupart d'entre eux : les diagrammes que j'ai produits sont visibles en guise d'illustration dans la partie 5.2.2, commençant à la page 136*).

#### 5.2.1.1 Modélisation des besoins fonctionnels

Cette phase n'utilise rien de spécifique à UWE, il s'agit pour l'instant de simples diagrammes UML :

- *Diagramme(s) des cas d'utilisation* : c'est le diagramme montrant les différentes actions possibles pour tel et tel acteur du système. Cela permet de représenter de manière synthétique les différents besoins fonctionnels de l'application par profil applicatif.
- *Diagramme(s) d'activités* : symbolise les workflows (enchaînements d'interactions entre l'utilisateur et le système), permet de documenter la logique fonctionnelle de l'application.

#### 5.2.1.2 Modélisation du contenu

Là aussi, rien n'est encore spécifique à UWE :

- *Diagramme(s) de classes* : permet de représenter la structure des données. Cela serait comparable à un MCD si on employait la méthode Merise, à ceci près que dans un *diagramme de classes* UML, chaque classe (ex. bloc stats, machine de tri...) en plus d'une liste d'attributs, intègre aussi une liste de méthodes (fonctions applicables spécifiquement à ces objets). Rappelons qu'UML permet avant tout de modéliser des applications à écrire dans un langage objet.
- *Diagramme(s) d'états* : permet de modéliser le cycle de vie d'un objet de l'application. Cela se rapproche de ce qu'on appelle de manière générique un « *diagramme états/transitions* ».

#### 5.2.1.3 Modélisation de la navigation

C'est là que commence intervenir la spécificité UWE.

MagicUWE initialise automatiquement la transformation d'un *diagramme de classes* en un « *diagramme de la structure de navigation* » :

- *Diagramme(s) de la structure de navigation* : définit le « quoi » (les objets visités)
- *Diagramme(s) d'accès* : définit le « comment » (comment ces objets sont visités)

#### 5.2.1.4 Modélisation de la présentation

Cette phase utilise également des diagrammes UML signés UWE. MagicUWE permet en effet d'initialiser la transformation d'un *diagramme de classes* en un « *diagramme de la structure de navigation* » :

- *Diagramme de présentation statique* : Ce diagramme permet simplement de définir explicitement la disposition des éléments au sein des différentes pages.
- *Diagramme de présentation dynamique* : Il s'agit ni plus ni moins d'un « *diagramme de séquence* » UML : celui-ci permet de modéliser le déroulement d'une interaction entre un ou plusieurs acteurs et le système, pour tel cas d'utilisation.

#### 5.2.1.5 Modélisation de l'adaptation

La « modélisation de l'adaptation » consiste à ajouter dans les *diagrammes de présentation* une précision concernant tant un contexte qu'une implication sur la présentation. Cette règle est modélisée via une classe stéréotypée « rule ».

## 5.2.2 Essais de modélisation du projet SATURNE en UWE

### 5.2.2.1 Diagramme des cas d'utilisation (général)

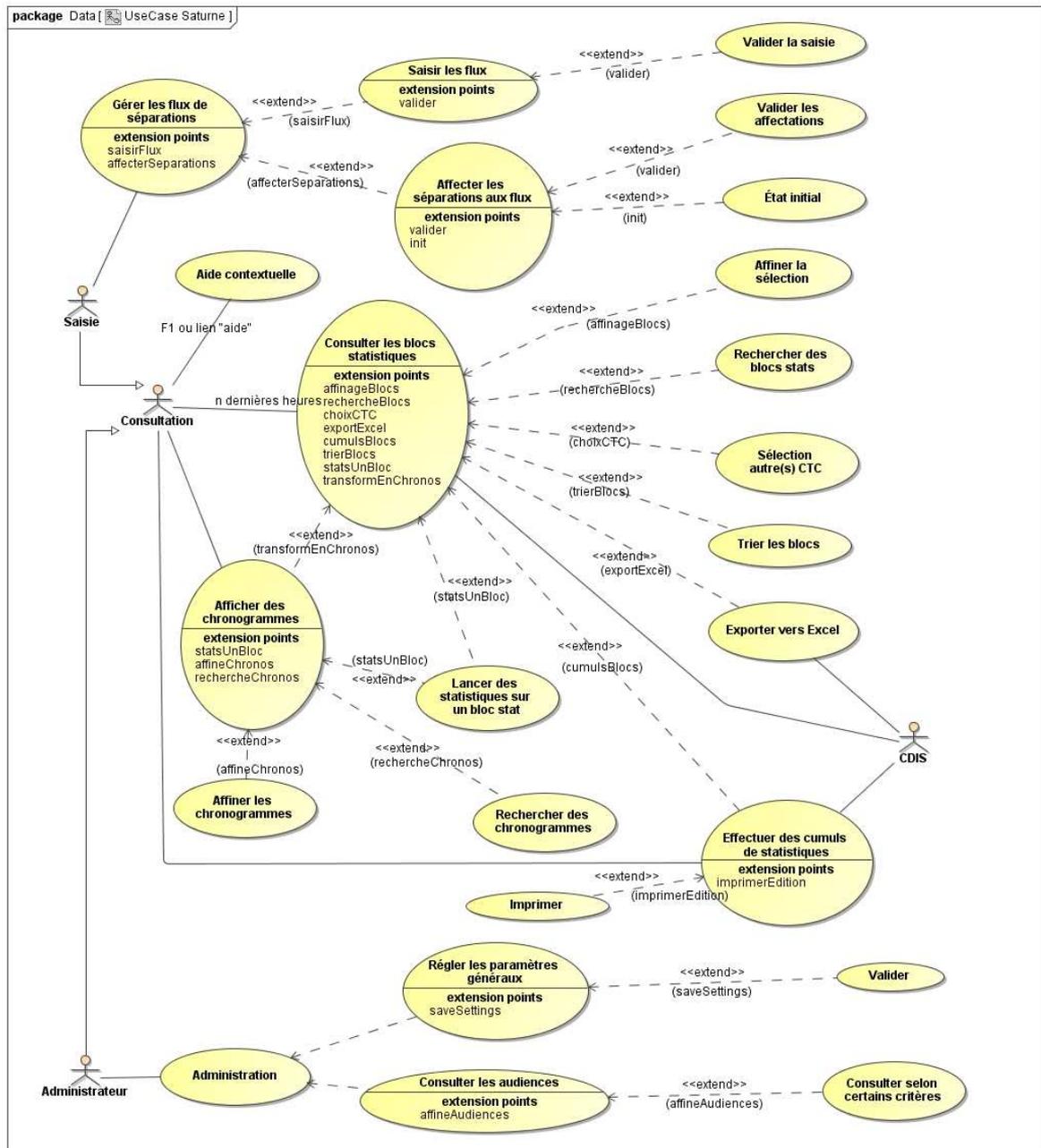


Figure 59 – Diagramme des cas d'utilisation de SATURNE (général)

Remarque : J'ai entrepris d'utiliser la notion d'héritage entre acteurs du système. En effet, il existe plusieurs profils utilisateurs dans l'application SATURNE : ces profils correspondent

en UML à ce qu'on appelle des « acteurs »<sup>128</sup>. Or, certains acteurs ont accès à un sous-ensemble de fonctions par rapport aux autres.

Recentrons-nous à présent sur le cœur fonctionnel de SATURNE, à savoir la consultation des blocs statistiques.

### 5.2.2.2 Diagramme des cas d'utilisation (centré sur la consultation de blocs statistiques)

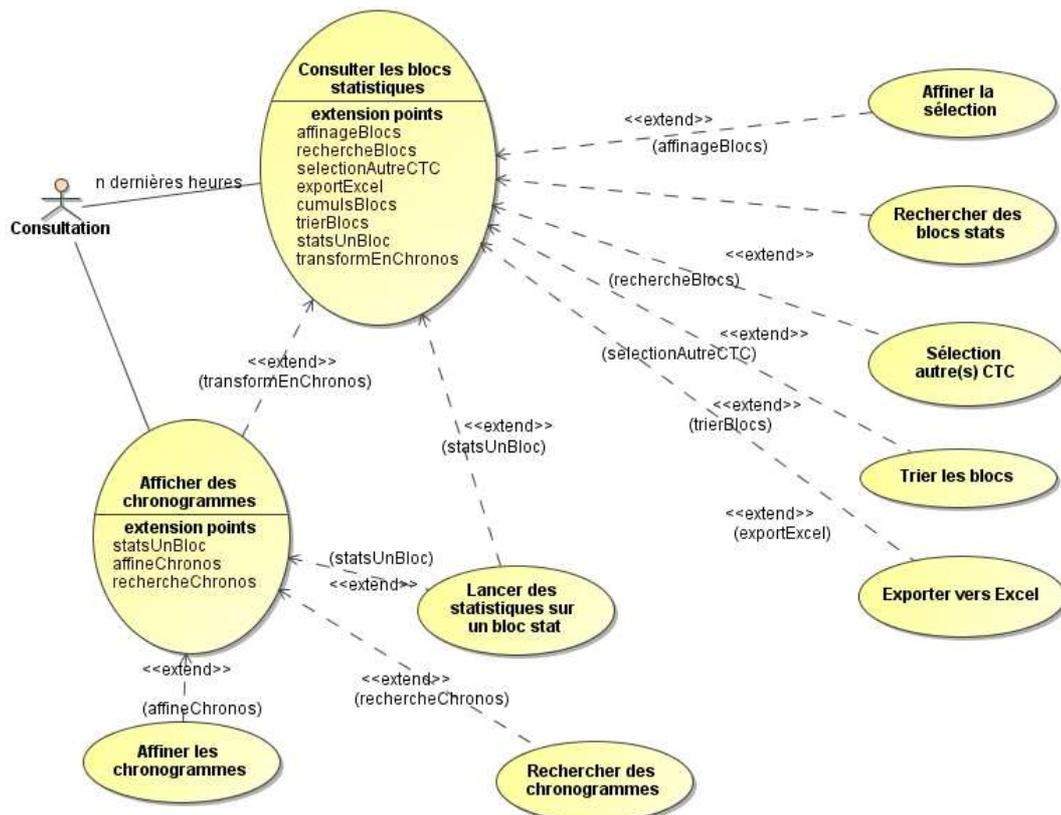


Figure 60 – Diagramme des cas d'utilisation de SATURNE (la consultation de blocs stats)

On notera que dans ces *diagrammes des cas d'utilisation*, j'ai systématiquement fait l'usage des relations de type « extend » : celles-ci permettent de modéliser des « sous-actions » possibles (ex. « affiner la sélection ») au sein d'un cas d'utilisation général (ex. « consulter les blocs statistiques »). Cela n'est donc pas à confondre avec la notion d'héritage, mettant en jeu des « sous-types » d'action (ex. « consulter les blocs stats » serait un héritage de « actions de consultation »).

<sup>128</sup> Il s'agit de ceux qui saisissent, ceux qui consultent, ceux qui ont les droits d'administration et ceux qui n'ont besoin qu'à des fonctions réduites de consultation, dédiées aux CDIS (*Centres de distribution* – cf. partie 1.1.3.1, page 14)

### 5.2.2.3 Diagramme de classes

Voici un exemple simplifié d'un *diagramme de classe* mettant en scène celles qui correspondent aux caractéristiques des données de l'application SATURNE :

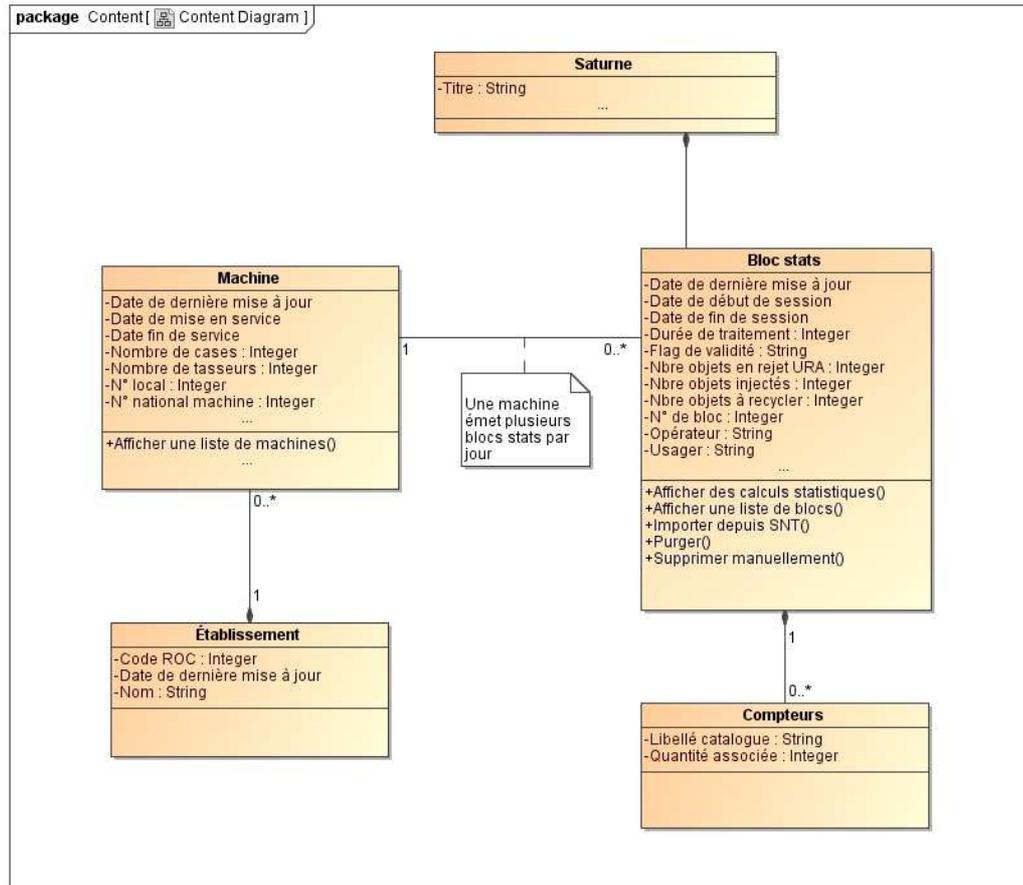


Figure 61 – Diagramme de classes de SATURNE (version simplifiée)

### 5.2.2.4 Diagramme d'états (pour un bloc stats)

Le cycle de vie le plus utile à modéliser dans SATURNE est celui d'un bloc stats :

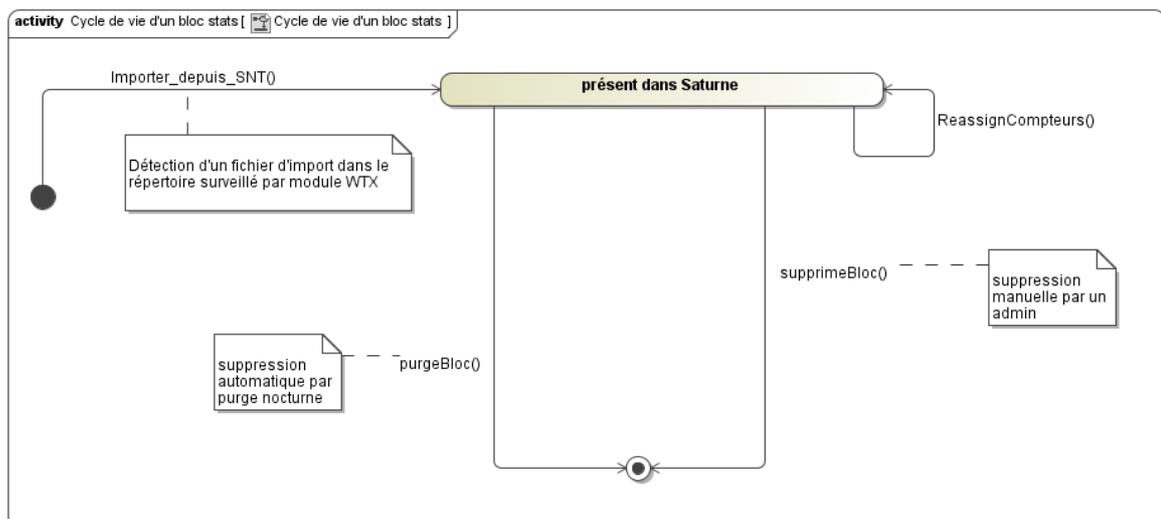


Figure 62 – Diagramme d'états (pour un bloc stats)

### 5.2.2.5 Diagramme de la structure de navigation

Dans MagicUWE, la transformation du diagramme de classes en *diagramme de navigation* s'exécute en un clic :

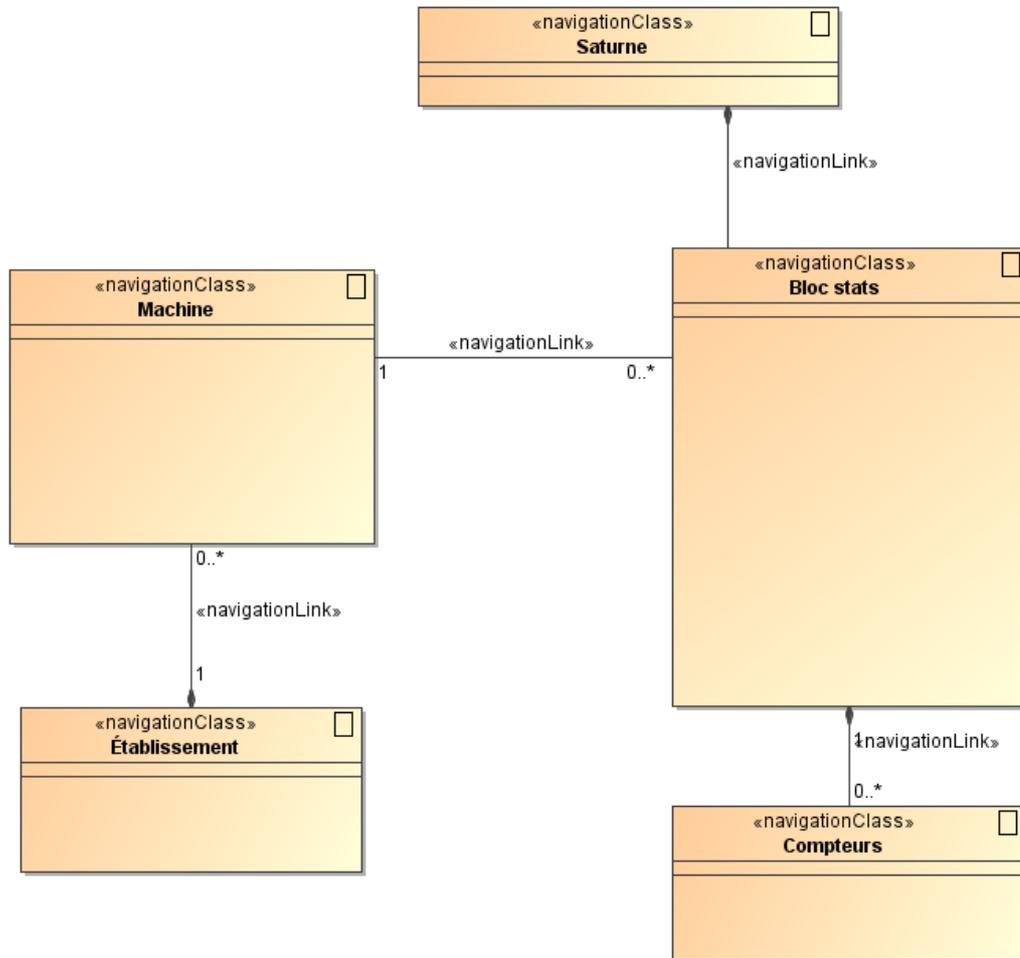
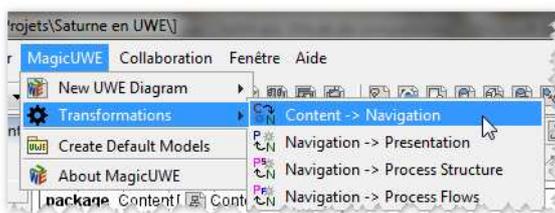


Figure 63 – Diagramme de la structure de navigation de SATURNE (version simplifiée)



Ce diagramme ne me semble pas avoir d'intérêt en tant que tel : il sert juste de point de départ pour préparer le « *diagramme d'accès* », qui constitue la suite directe de la méthode.

### 5.2.2.6 Diagramme d'accès

Nous aboutissons rapidement au *diagramme de navigation* suivant, mais seulement après avoir :

- retiré les classes inutiles pour la présentation à l'écran ;

- retouché ou ajouté des liens entre les différentes classes du diagramme ;
- intégré les « stéréotypes » que sont notamment les objets *menus*, *index* et *Query*.

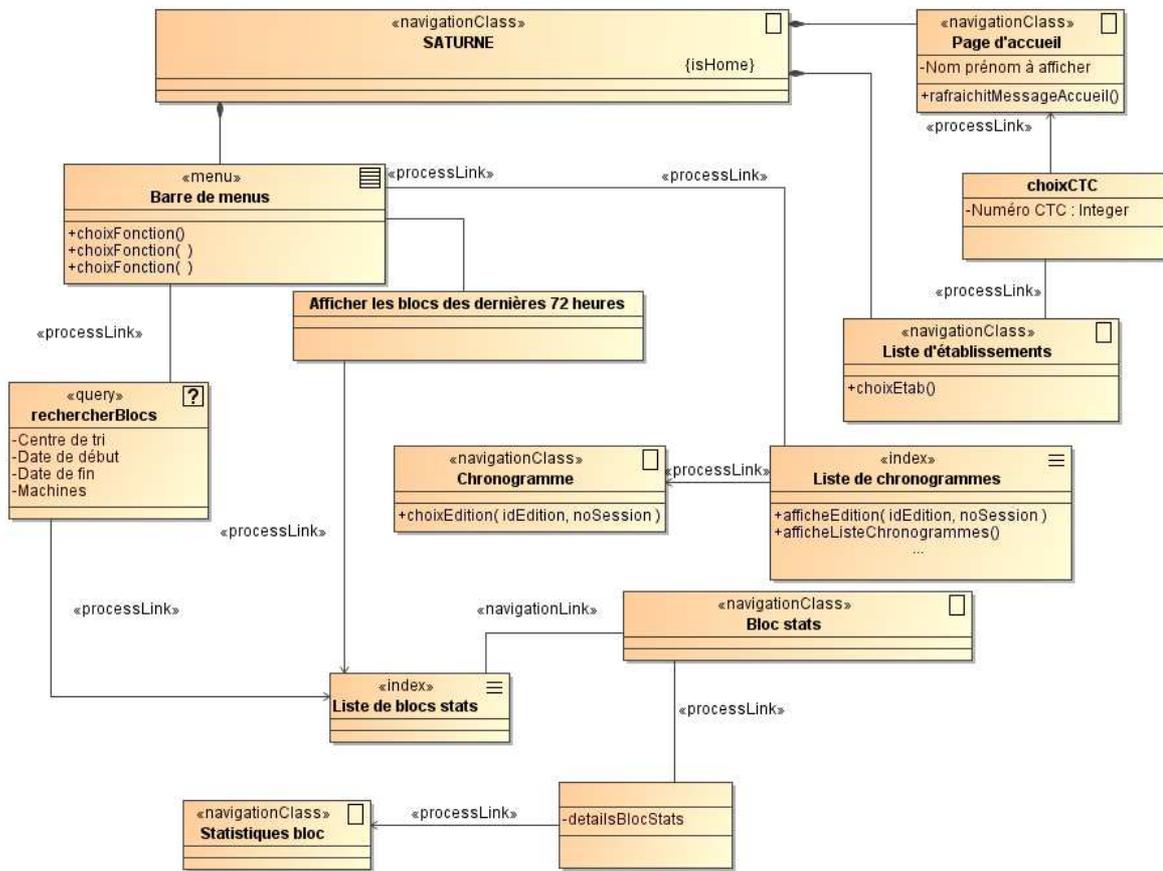
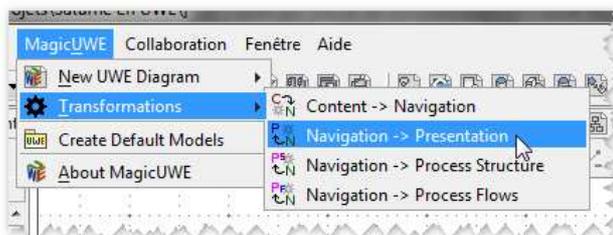


Figure 64 – Diagramme d'accès de SATURNE (version simplifiée)

Remarque : Concernant l'usage du stéréotype « Query » (à gauche du diagramme), qui modélise la recherche de blocs stats, il m'est apparu légitime d'ajouter en tant que propriété chacun des critères de tri pouvant conditionner le résultat de la recherche.

### 5.2.2.7 Diagramme de présentation

Là aussi, un simple clic dans MagicUWE permet de générer une version initiale de diagramme de présentation :



Il faut ensuite enrichir le diagramme ainsi généré, et disposer les classes de la manière dont seront disposés les objets qu'elles définissent au sein des futurs écrans de l'application.

Voici un *diagramme de présentation* donnant une vue générale de l'application SATURNE :

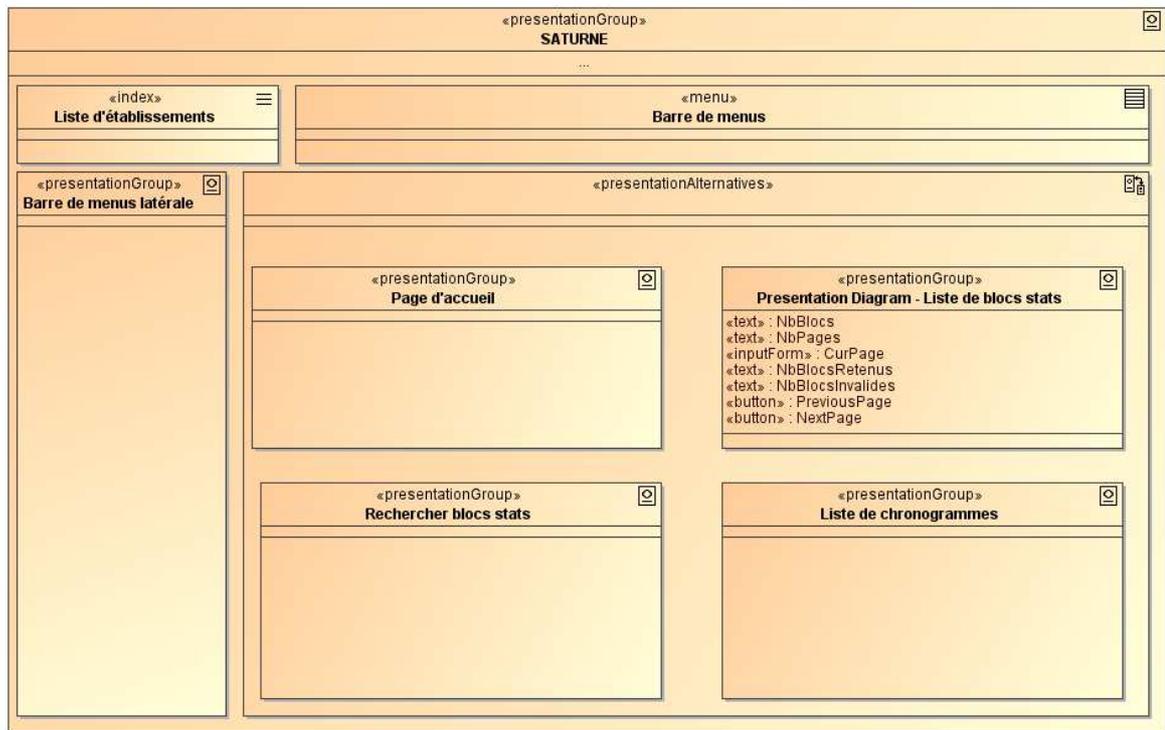
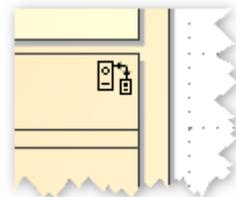


Figure 65 – Diagramme de présentation général de SATURNE (version simplifiée)

*Remarque* : J'ai utilisé le type de classe « *Alternative* », prévu par la méthode UWE (cf. l'icône ci-contre, extraite de la copie d'écran MagicDraw UWE ci-dessus). Cela symbolise le fait que selon la fonctionnalité en cours d'utilisation, la frame centrale de SATURNE contient au choix l'un des 4 écrans que contient cette classe de type « *Alternative* » : Page d'accueil, Recherche, Liste ou Chronogrammes.



Voici à présent une vue du *diagramme de présentation* recentrée sur l'écran « Liste de blocs stats » :

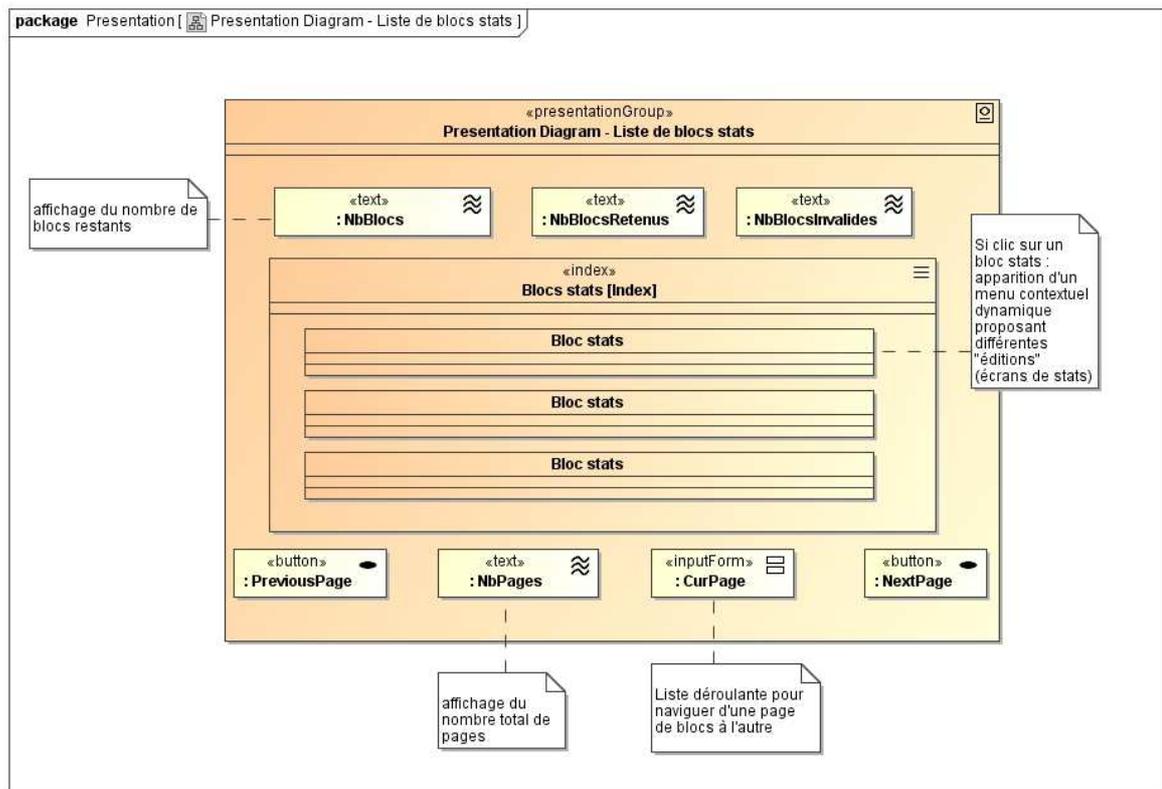


Figure 66 – Diagramme de présentation de la fonction «Liste de blocs stats » (version simplifiée)

C'est dans ce type de diagramme UWE que l'on perçoit de suite la différence avec une représentation purement UML, puisqu'on y met en jeu des « éléments de présentation UWE », ces classes pré-typées permettant de modéliser les éléments d'IHM d'une page HTML (boutons, champs de saisie, etc).

Les infobulles, même si elles n'ont aucune incidence sur l'éventuel code source généré, sont un complément utile pour préciser ce qui ne peut pas l'être de manière rationnelle au travers de la méthode. Elles me semblent indispensables si on souhaite faire de ces diagrammes un support de documentation pour un projet informatique.

### 5.2.2.8 Modèle dynamique de présentation (diagramme de séquence)

Pour finir, je me suis essayé à la version dynamique du « *diagramme de la présentation* », à savoir un diagramme de séquence UML. Le *diagramme de séquences* représenté ci-dessous modélise le scénario « *Lancement d'une édition via les chronogrammes* ».

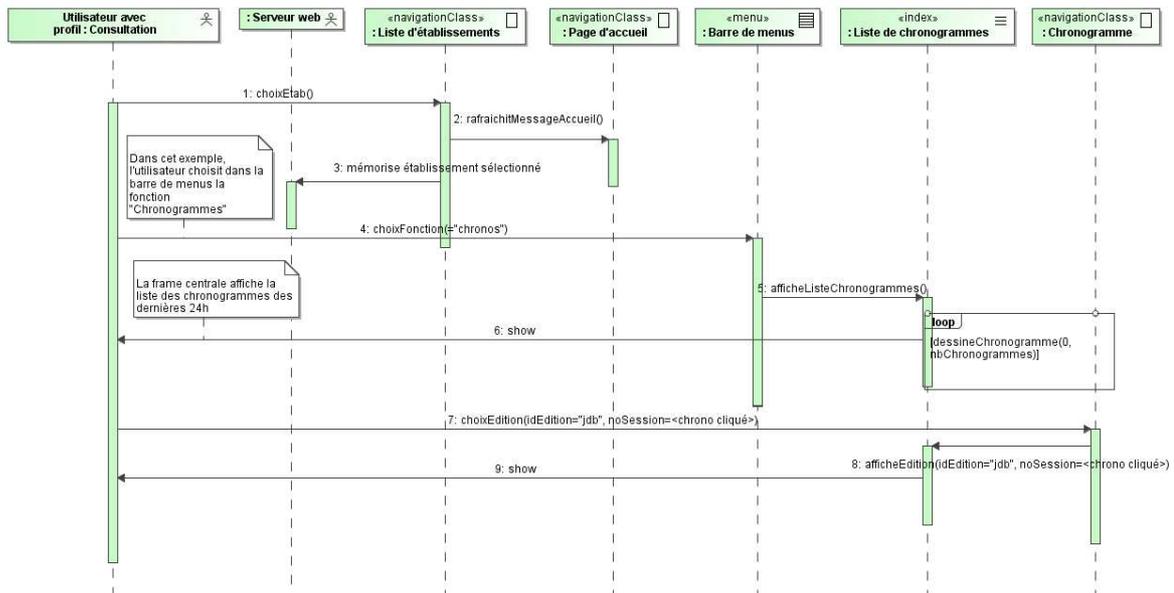
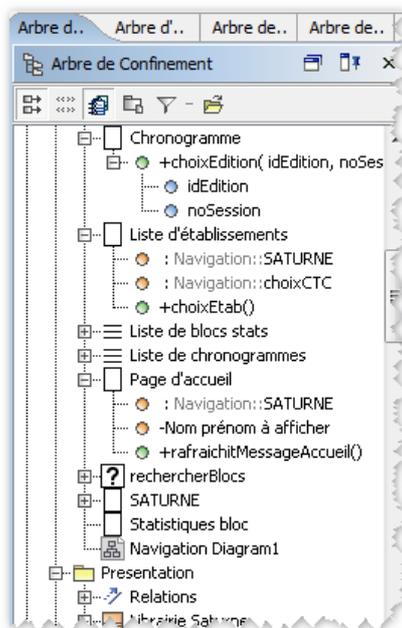


Figure 67 – Diagramme de séquence pour le scénario « Lancement d'une édition via les chronogrammes »



*Remarques :* Dans MagicDraw, plus on avance dans la création successive de diagrammes, plus la librairie sur la gauche de l'écran MagicDraw se remplit avec les différentes classes déjà créées dans les précédents diagrammes. Une même classe va en effet être représentée à la fois dans un *diagramme d'accès* et dans un *diagramme de séquence*. Pour concevoir par exemple un *diagramme de séquence*, il est alors plus simple d'effectuer avec la souris un « glisser-déplacer » des classes référencées sur la gauche que l'on souhaite retrouver sur notre *diagramme de séquence*.

De même, je trouve préférable de définir à l'avance les différentes méthodes de chaque classe présente dans le *diagramme d'accès*. Ainsi, il suffit de commencer à saisir leur nom lorsqu'on nomme un message reliant deux classes dans le *diagramme de séquences* : apparaît alors la liste des méthodes disponibles pour cette classe.



langage ou un outil « populaire » est gage à la fois de réussite et de pérennité, notamment parce que l'aide sur internet est dans ce cas fournie et vivante (forums de développeurs, aide en ligne avec possibilité de commenter ou de poser des questions) et aussi parce que le succès généralisé d'un outil de développement informatique entraîne des moyens mis en œuvre pour les mises à jour ou corrections régulières de celui-ci. Dans l'immédiat, ce que je tire de mon peu d'expérience avec MagicUWE est que je me suis senti seul et désorienté dans ma tentative d'apprentissage du produit, car il offre des possibilités complexes et tout n'est pas intuitif. J'ai certes pu produire quelques diagrammes, mais j'aurais fortement apprécié l'aide d'une personne ayant déjà utilisé jusqu'au bout, pour un vrai projet informatique, la méthode UWE à l'aide de MagicDraw UML.

- En effet, UWE s'interface avec l'un des deux logiciels de modélisation UML suivants : ArgoUML (on dit alors « ArgoUWE ») et MagicDraw UML (on dit alors « MagicUWE »). Mais ce choix vaut pour le passé, car l'équipe du projet UWE a cessé de maintenir un plugin compatible avec les dernières versions d'ArgoUML, qui est pourtant un logiciel open source... C'est regrettable, car :
  - ArgoUML est une application Open Source, alors que MagicDraw UML est un logiciel commercial qui dans sa version d'essai impose une limitation du nombre de classe qui rend le logiciel inexploitable pour un vrai projet informatique, à moins de sortir la monnaie : cela a d'ailleurs été ma première difficulté ;
  - Contrairement à ArgoUML, MagicDraw UML ne prend pas en charge la génération de code en PHP, qui est un langage objet (depuis la version 5) puissant simple d'utilisation et massivement utilisé dans le monde du développement web, et dont je constate un réel retour en force à l'approche de 2010<sup>130</sup> ;
  - ArgoUML me semble plus sobre et plus simple d'utilisation que son concurrent payant, MagicDraw UML.
- Cette méthode est moins intuitive qu'une approche empirique. Il faut bien le dire, le « bon sens » est souvent source de réussite dans un projet informatique (même si tout excès n'est bon en rien : un bon ingénieur ou un bon développeur prend bien

---

<sup>130</sup> Certains ingénieurs se lassent en effet de la complexité de mise en place et de développement de pages web sous JEE, et n'hésitent pas à revenir à la simplicité de mise en œuvre de PHP, tout en continuant à développer des composants en Java/JEE pour les parties les plus complexes ou les plus gourmandes de leur application, comme par exemple des besoins de multi-threading.

soin de ne pas se précipiter sur le code source avant d'avoir pris le temps de la réflexion à l'aide d'un feuille et d'un crayon de papier<sup>131</sup>).

### 5.3.3 Limites de la méthode UWE dans son application à SATURNE

J'ai tenté une application d'UWE à SATURNE, ce qui équivaut à suivre les étapes successives de cette méthode pour spécifier les objets et fonctionnalités principales de mon application web. Voyons quelles ont été les principales difficultés dans cette mise en pratique.

#### 5.3.3.1 Une méthode certainement privilégiée pour les conceptions ex-nihilo

Cette méthode est sûrement très adaptée à la création d'un projet *ex nihilo* (à partir de rien), alors que dans mon projet à LA POSTE, le plus clair de mon temps a été de décrire, puis de spécifier des sous-fonctionnalités : il serait intéressant de voir de quelle manière UWE modifie du code déjà auto-généré, surtout si ce code a été retouché manuellement entre temps.

#### 5.3.3.2 Une génération de code non compatible avec SATURNE

À LA POSTE, une condition imposée était d'écrire le code source de SATURNE côté serveur en ASP.

Donc, si je devais répondre de manière figée à la question posée dans cette partie, qui est « *Une méthode comme UWE aurait-elle été utile ?* », je n'aurai même pas eu besoin d'installer UWE sur mon ordinateur et m'en serais tenu à la conclusion que non, au seul motif de la non-compatibilité avec le langage cible utilisé dans SATURNE, je n'aurais pas pu utiliser UWE<sup>132</sup>.

Mais jouons quand même le jeu « expérimentation / analyse » un peu plus loin, en émettant simplement l'hypothèse que SATURNE ait été écrite en PHP, seul langage que je trouve vraiment similaire à ASP dans ses caractéristiques en dehors de l'aspect objet de PHP.

---

<sup>131</sup> Dans une telle « réflexion concrète », il est en général question de décoder les besoins transmis par le client/MOA, ou encore tenter de découvrir toutes les contraintes et implications diverses qu'ils supposent.

<sup>132</sup> Sans doute cette absence de compatibilité est-elle due au fait qu'ASP n'est pas un langage objet, donc pas adapté à l'utilisation d'une méthode de conception web utilisant UML (les classes décrites dans des diagrammes UML ont des propriétés et des méthodes).

### 5.3.3.3 Une compatibilité ArgoUML qui appartient malheureusement au passé

La première difficulté à laquelle j'ai été confronté en allant donc plus loin dans ma tentative de produire des spécifications de SATURNE avec UWE, est le fait que ce dernier ne gère plus sa compatibilité avec ArgoUML.

C'est en effet ArgoUML qui m'intéressait le plus pour la génération de code source : contrairement à UWE qui permet de générer du Java/C#/C++, seul ArgoUML aurait permis de générer du code PHP.

Mais ne nous arrêtons pas là, et considérons simplement que SATURNE ait pu être écrit dans un langage géré par MagicDraw UML : j'ai donc installé ce logiciel, me suis battu au passage avec les limitations dues à la version d'essai (nombre d'objets limité et pas de possibilité de générer du code source), et ai produit les modélisations visibles page 136 (partie 5.2.2 « Essais de modélisation du projet SATURNE en UWE »).

### 5.3.3.4 Les fonctions et l'IHM de SATURNE exploitent peu les capacités d'UWE

Même en ayant tenté de créer des diagrammes pour SATURNE en suivant la méthode UWE, je reste extrêmement sceptique sur le bien fondé d'un outil comme UWE pour une application comme SATURNE. Pour cause, je ne trouve pas l'outil suffisamment adapté à la fois au type de fonctionnalité et aux éléments d'IHM que j'y ai intégrés. Expliquons chacun de ces deux points :

1. D'un ordre général, et concernant davantage UML que UWE en tant que tel, j'ai constaté que les logiciels comme MagicDraw offre des possibilités à la fois simples et puissantes pour dessiner des *diagrammes d'activité* (pour représenter les différents enchaînements dans l'interactivité avec l'utilisateur) des *diagrammes d'états* (pour représenter le cycle de vie de chaque objet) ou encore des *diagrammes de séquence*. En revanche, vis-à-vis des caractéristiques fonctionnelles générales de SATURNE, mon constat est que, contrairement au back-office, la partie web de SATURNE n'aurait pas beaucoup profité de ce type de diagrammes, auxquels je me suis pourtant essayé : mon application n'offre quasiment que des fonctionnalités de consultation de données, avec des enchaînements d'interactions limités ; ce qui réduit fortement l'intérêt de recourir à des *diagrammes d'activité*. En réalité, la complexité de la partie web de SATURNE se situe à un autre niveau : cette application gère notamment un grand nombre d'éditions<sup>133</sup> possibles, avec pour chacune, un certain nombre de règles de calcul bien établie et documentées de manière générique et rationnelle dans l'aide en ligne.

---

<sup>133</sup> Dans SATURNE, on appelle « édition » tout écran de statistique, comme par exemple le « Journal de bord » ou encore le « Tableau technique ».

Le même constat est à tirer pour le bien fondé des *diagrammes d'états* dans SATURNE. En ce qui concerne les cycles de vie des objets, celle-ci n'est pas trépidante : il n'arrive pas des milliers de choses aux blocs stats ou aux machines de tri entre leur création et leur destruction dans le système.

Néanmoins, les *diagrammes de séquence* (ou, en langage UWE, « *modèles dynamiques de présentation* ») représentent peut-être une exception à ce constat moyen, car ceux-ci auraient pu être utilisés sans peine dans la documentation de SATURNE ;

2. En ce qui concerne l'intérêt de la surcouche UWE, rappelons que celle-ci est en théorie sensée rendre les diagrammes UML pleinement adaptés à la spécification d'applications web, notamment en offrant des objets « clé en main » pour chaque élément d'interface propre aux pages HTML (boutons, images, etc), qu'il est possible de disposer tels quels dans la modélisation de la présentation, dans laquelle UWE fait intervenir un nouveau diagramme appelé « *diagramme de présentation* ». Le problème est que ces objets sont basiques et me semblent adaptés au design de pages web tout aussi basiques, ne faisant appel qu'à des objets HTML simples. Or, si j'ai veillé à offrir à SATURNE des éléments d'IHM simples d'utilisation, ce n'est pas pour autant que le code source DHML qui les génère est simple. En effet, tout ce qui n'existe pas en HTML, comme par exemple les onglets, doit être codé en DHTML – ou à la limite en Flash, mais le Flash est déconseillé car les applications Flash sont lourdes et leur maintenabilité est moyenne, sans compter que Flash sort des standards. Je pense notamment au menu contextuel de la liste des blocs stats, programmée spécifiquement pour SATURNE, ou encore ma liste à choix multiple à base de cases à cocher. On mentionnera aussi que, tout simplement, les différents boutons de l'application ne sont pas des « boutons » au sens objet HTML<sup>134</sup>, mais, pour les besoins esthétiques, des images cliquables avec effet *roll-over* (gérant les événements souris *onclick*, *onmouseover* et *onmouseout*). Or, il est permis de douter que ces spécificités d'interface ne peuvent que difficilement être spécifiées et générées en UWE...

#### 5.3.4 Bilan sur la question de l'adaptation d'UWE à SATURNE

Étudier une véritable méthode de conception de projet informatique est toujours instructif, surtout si celle-ci est adaptée aux projets d'applications web.

En revanche, mon constat est que UWE n'aurait pas été adapté à la conception de SATURNE, et ce, pour les multiples raisons présentées dans les deux paragraphes précédents.

---

<sup>134</sup> <input type="button" ...>

Et combien même UWE aurait été adaptable à SATURNE, un autre problème plus terre à terre se serait posé : les standards de LA POSTE en terme de documentation de projets sont suffisamment chronophages en l'état actuel, aussi, il serait difficilement envisageable de trouver le temps nécessaire à la fois pour maintenir les spécifications en UWE et pour produire la documentation classique du projet (fournies au travers des DAL, DAT, DCG, DCD<sup>135</sup>).

Rappelons, pour pondérer ce bilan à tendance réfractaire, que la méthode UWE n'est pas sans avantage. Elle a d'abord été pour moi l'occasion de m'initier à UML, ce qui est un des plus grands standards modernes en termes de modélisation de systèmes informatisés. Ensuite, même si la possibilité de générer du code source à partir de diagrammes UWE est pour moi restée à l'état théorique, le simple fait de pouvoir dessiner facilement des *diagrammes des cas d'utilisation* (fonctions générales par profil utilisateur) ou des *diagrammes d'état* (cycle de vie d'un objet), ou encore des *diagrammes d'activité* (représentation du workflow) est un vrai plus pour la qualité de documentation d'un projet informatique, même si celui-ci n'aboutit pas sur l'utilisation d'un langage objet.

### 5.3.5 Alors, quid de la méthode actuellement employée ?

Résumons d'abord en quelques lignes les différentes « notations » que j'emploierais personnellement pour noter la qualité de gestion technique d'un projet informatique :

1. *L'extrême minimum attendu* : correctement documenter ce qui a été développé.
2. *Un peu plus que le minimum* : écrire les spécifications du projet avant de passer à la réalisation (même si ces spécifications sont écrites sous Word)
3. *La méthode de projet « quali »* : écrire ces spécifications à l'aide d'une méthode de conception utilisant un standard universel comme UML.
4. *Le must* : que cet outil de conception puisse aussi générer du code source pour faciliter la tâche de réalisation.

Dans le cas de SATURNE, il est difficile de se situer à un niveau précis de cette échelle de notation qui est la mienne. En effet, même si une chose certaine est que les deux premiers points sont pleinement acquis, les deux derniers ne sont que partiellement vérifiés : n'oublions pas que l'outil de modélisation de données employé à LA POSTE (*AMC Power Designer*) nous permet à la fois de spécifier un modèle et de générer le code de la base de données.

---

<sup>135</sup> Cf. partie 1.4.2.1 « Phase d'analyse », page 31

## CONCLUSION

### *Un bilan de projet positif*

Mon projet à LA POSTE m'aura offert des satisfactions variées : diversité des outils, richesse du contenu, diversité des phases de développement, qualité des rapports humains, autonomie appréciable et évolution vers une fonction d'encadrement<sup>136</sup>. De plus, l'analyse, le développement et l'optimisation de SATURNE ont été l'occasion de mettre en œuvre des capacités qui sont celles que l'on peut attendre d'un profil d'ingénieur. En effet, j'ai aidé LA POSTE à trouver une réponse aux impératifs de performance de SATURNE, à différents niveaux de l'architecture de l'application, tout en apportant ma propre valeur ajoutée qui consiste à ne pas faire de l'IHM une laissée pour compte. Ainsi, je suis heureux d'avoir posé ma brique dans le gigantesque chantier du Cap Qualité Courrier, et ce avec succès puisque SATURNE est même devenue une référence en terme d'IHM pour tous les autres intranets postaux !

Quant à ce mémoire, je tiens à préciser que sa rédaction, ainsi que le travail de synthèse qu'il a occasionné, m'auront aidé à mieux savoir analyser et conceptualiser l'architecture d'une application client-serveur, de même que le travail sur UWE m'a permis d'approfondir mes connaissances sur le standard universel qu'est UML.

### *Les bémols*

C'est bien normal, quelques aspects du projet viennent légèrement ternir ma satisfaction globale : ils tiennent d'une part à la façon dont l'application SATURNE a été conçue, et d'autre part à la forme qu'a pu revêtir ma mission au cours des années.

En ce qui concerne les choix techniques imposés pour SATURNE, on notera en premier lieu qu'une migration du système actuel (ASP 2.0 / Sybase / WTX) vers une solution plus moderne de type « services web REST » (.NET ou J2EE / Oracle), que j'espérais depuis longtemps, ne correspondrait pas qu'à un effet de mode... cela, en dépit de tous nos efforts d'optimisation du MCD et du code T-SQL. Le remplacement de certaines technologies employées à LA POSTE offrirait en effet une plate-forme plus stable, et améliorerait encore la capacité d'SNT/SATURNE à résister à la montée en charge. Le maintien de cette capacité devra toujours rester une priorité, parce que la quantité d'utilisateurs ciblés par SATURNE croît constamment : LA POSTE se rapproche de plus en plus de l'ouverture du marché à la concurrence, et doit ainsi faire ses preuves plus que jamais en matière de performance. De

---

<sup>136</sup> Cf. partie sur la TMA dans la conclusion, page 151.

plus, s'il s'avérait que les web services peuvent remplacer WTX pour la partie EAI, ce que je crois, alors l'économie financière réalisée serait de taille<sup>137</sup> !

La deuxième chose à regretter est la rigidité de la structure postale, rendant très difficile, voire parfois impossible, de franchir les couches du métier. Certes, en temps normal, le fait de maintenir à sa place chaque membre d'une équipe est sensé accroître la qualité globale du projet, et éviter les pertes de temps pouvant être dues à de mauvaises initiatives. Mais à LA POSTE, la réalité est étonnante : même la MOA n'a pas toujours la possibilité de s'adresser directement aux utilisateurs pour recueillir les besoins.

Même si un effort de ressenti positif de la contrainte structurale est toujours possible (un tel apprentissage peut être vu comme « l'école de la vie ») il n'empêche que cet état de fait rentre en conflit avec l'idéal que je me fais du métier d'ingénieur-développeur : être à la fois proche de la technique (savoir comment implémenter le besoin utilisateur) et proche des utilisateurs (savoir appréhender leur besoin via des réunions, des questionnaires ou encore des tests sur le terrain, de type « Think aloud<sup>138</sup> »). Cette tâche bien spécifique du recueil des besoins revient à la MOA, mais cela ne doit pas empêcher à la MOE d'exercer un regard non concurrent, mais complémentaire au leur. Durant cette mission, j'ai justement prouvé, au travers des rares opportunités de proposer de nouvelles fonctionnalités pour SATURNE, que l'œil de la MOA avait besoin à grand renfort de celui du développeur-concepteur. La raison est double. D'une part, les concepteurs sont très à même d'imaginer ce qui serait techniquement possible. D'autre part, et peut-être est-ce un paradoxe, ils connaissent souvent mieux les applications que la MOA elle-même, car ce sont eux qui les développent. C'est pour cela que notre situation de développeur-concepteur ou d'ingénieur est, je crois, extrêmement privilégiée, car nous pouvons imaginer aussi bien, sinon mieux que la maîtrise d'ouvrage, ce que pourraient être les améliorations à apporter aux applications que l'on conçoit.

### *Réinventons le Think Aloud*

Il faut toutefois souligner que l'autonomie dont j'ai pu bénéficier les premières années de ma mission m'a permis de contourner la contrainte structurale et mieux appréhender malgré tout les besoins des utilisateurs. Dans SATURNE, j'ai mis en place de ma propre initiative un module de traces des actions utilisateurs, avec trois modes de consultations : le taux d'audience de chaque page, le détail à la seconde près des actions de chaque utilisateur d'un échantillon pris au hasard, et enfin la visualisation dans une frise du nombre de

---

<sup>137</sup> Une licence serveur WTX coûte actuellement à LA POSTE environ 250.000 euros. (À savoir qu'il s'agit d'un prix négocié.)

<sup>138</sup> « Think aloud » est une méthode d'évaluation très simple, mais puissante : elle consiste à se placer derrière l'utilisateur, sans l'aider, mais en encourageant celui-ci à penser à voix haute pendant qu'il utilise le système.

connexions à SATURNE dans chaque tranche horaire. Finalement, pourquoi ne pas appeler cette méthode un « Think aloud virtuel » ?

Le développement de cet outil de trace a été fait trois ans avant mon départ, or depuis, la rigidité de séparation des couches hiérarchiques à LA POSTE s'est endurcie de telle façon qu'à l'heure actuelle, il serait inenvisageable de prendre des initiatives comme celle-ci. Précisons néanmoins que le type de contrat qui me lie à LA POSTE y est pour quelque chose. Depuis mai 2007, mes collègues d'APSIDE travaillant à LA POSTE<sup>139</sup> et moi-même sont liés à cette société cliente par un contrat de TMA<sup>140</sup>, rendant le contenu de la prestation beaucoup plus cadré : respect d'un « plan action qualité » (engagement sur les résultats), forfaitisation du travail avec demande officielle de LA POSTE pour chaque action (corrections, évolutions ou études), envoi hebdomadaire de compte-rendus (journal de bord, planning...), etc.

*TMA : fini de rigoler !*

Un premier avantage d'un contrat de TMA est que notre mission a pu perdurer. En effet, les conventions collectives stipulent qu'un prestataire de service est dans le droit de se faire embaucher au bout de 3 ans de mission, ce qui conduit alors LA POSTE à ne prendre aucun « risque »<sup>141</sup>. Or, en TMA nous sommes devenus pratiquement interchangeables.

Un autre avantage du passage de la mission en mode TMA est que j'encadre désormais un autre prestataire<sup>142</sup>. L'objectif est à la fois de renforcer notre équipe de développement, et tout simplement procéder à un passage de connaissance en vue de mon départ à moyen terme vers une autre mission. Cela peut surtout être considéré comme une opportunité personnelle, puisque ce changement implique un début d'expérience en management. C'est aussi pour cette raison que le statut officiel qui m'a été attribué à l'occasion du passage en

---

<sup>139</sup> J'ai ajouté au jargon local de ma mission le terme de « postataire » : un prestataire qui reste plus de 6 ans à LA POSTE n'est-il pas à mi-chemin entre postier et prestataire ?

<sup>140</sup> Tierce maintenance applicative

<sup>141</sup> Cette notion de « risque » est relative, et je la trouve personnellement exagérée puisque l'embauche d'un prestataire par LA POSTE entraîne bien souvent une baisse conséquente de son pouvoir d'achat : ce qu'en général il n'est pas prêt à sacrifier, même pour les jours de congés additionnels, ni même pour la « sécurité de l'emploi » pourtant encore bien réelle (LA POSTE, comme France Télécom, peut embaucher des personnels selon les règles du droit privé, mais n'en est pas moins apparenté à la fonction publique – cf. partie 1.1 page 10, expliquant le nouveau statut de LA POSTE –).

<sup>142</sup> Le principe de la TMA est que nous sommes tous devenus « interchangeables ». De plus, le client n'a pas son mot à dire sur le choix des personnes physiques qui développent le projet, puisque tout est régi par un contrat de service. Est-ce là l'avenir du développement informatique ? J'ai pour ma part mon opinion sur la question. En témoigne le choix que j'ai fait en 2009 de m'orienter vers une structure de type « start-up »... pour rien au monde je ne regretterai mon choix.

TMA est « Ingénieur Chargé de Mission » (soit celui qui précède directement le statut de « Chef de projet junior »<sup>143</sup>).

En revanche, l'inconvénient d'un contrat de TMA avec une société cliente est que les rapports entre les deux sociétés sont très formalisés : la moindre demande d'évolution ou de correction d'anomalie doit s'acheminer en suivant un processus beaucoup plus lourd qu'en temps normal. La conséquence à mes yeux est que LA POSTE ne peut pas être aussi réactive qu'une petite structure, ce qui la pénalise forcément dans sa préparation à la concurrence... est-ce là un sacrifice nécessaire pour satisfaire les objectifs du Cap Qualité Courrier ? Un autre inconvénient est qu'en temps normal, un travail régi par un contrat de TMA doit s'effectuer dans les locaux de la société de service, et non dans les locaux de la société cliente (exceptionnellement pour moi, cette règle n'est pas appliquée<sup>144</sup>, mais la société SOPRA, qui a pris le relais<sup>145</sup>, exerce son contrat de TMA à l'extérieur de LA POSTE). Cela pose des limites évidentes en matière de travail collectif : la proximité avec le client est à mon sens une condition de réussite d'un contrat de service.

Dans tous les cas, on peut dire que l'évolution de mon contrat vers une TMA confirme une tendance de plus en plus persistante dans les entreprises françaises à vouloir sous-traiter leurs développements informatiques. En cela, je n'ai aucun doute sur le fait que le monde du service informatique a encore devant lui de beaux jours à vivre ! D'autant que, pour les années à venir, les besoins des entreprises clientes en solutions informatisées s'avèrent énormes.

#### *Méthodologie : ce que je n'ai pas pu faire*

Au-delà du « Think aloud virtuel » dont il est fait mention précédemment, j'aurais aimé appliquer d'autres méthodes permettant d'optimiser le respect des besoins, mais à LA POSTE je n'en ai pas eu l'opportunité. Parmi celles-ci, on pensera d'abord au prototypage, l'approche en spirale ou encore le contact avec l'utilisateur. J'ai également compris, au travers de mon travail sur UWE<sup>146</sup> que le recours à de vrais logiciels de modélisation utilisant un standard reconnu comme UML constituait un vrai plus, même si ceux-ci ne sont utilisés qu'à 10% de leurs capacités et ne servent qu'à enrichir les documentations existantes.

---

<sup>143</sup> Depuis que je suis chez Apside, on m'a attribué les statuts successifs suivants : « Ingénieur Débutant », puis « Ingénieur d'étude », puis « Ingénieur chargé de mission ».

<sup>144</sup> La raison principale est que la licence de certains logiciels comme IBM WTX coûterait trop cher à Apside.

<sup>145</sup> Malgré la grande satisfaction de LA POSTE envers Apside, nous ne faisons pas partie de leur vivier de SSII : ce vivier ne comporte que de grosses entreprises. Nous n'avons donc pas pu éterniser notre prestation.

<sup>146</sup> Cf. dernière partie de ce mémoire, page 123.

### *Un plan de carrière ?*

À la fin de cette longue mission à LA POSTE, mon commercial chez APSIDE m'a interpellé sur la nécessité de définir un plan de carrière. Traduction : j'ai été invité à réfléchir à la place que je me vois tenir d'ici les 15 prochaines années dans le monde de l'informatique<sup>147</sup>. Sinon, le risque est d'échapper à la perspective d'un profil professionnel sécurisé et d'un emploi plus rémunérateur.

Le résultat de ma réflexion est le suivant : je pense qu'il est primordial de m'adapter à l'évolution que semble prendre les métiers de l'informatique, sans pour autant renier mes aspirations. Sachant qu'en France, l'informatique a de plus en plus tendance à être soustraitée, notamment vers les pays du Maghreb, je me vois évoluer vers une fonction de chef de projet fonctionnel (au travers de formations, mais surtout d'une expérience progressive), établissant les allers-retours entre MOA locale et MOE distante, et tout en essayant de transmettre aux équipes de développeurs mon savoir-faire et ma sensibilité. J'envisage même de me consacrer à cette fonction sur le terrain, en passant quelques années à l'étranger afin d'établir un dialogue sur place, et accessoirement de m'ouvrir plus largement aux autres cultures.

Dans l'attente, tant que je reste ingénieur développeur, il me faut « passer à l'action », c'est-à-dire adapter mon savoir faire à l'évolution très récente des usages : tout le monde s'accorde depuis l'été 2010 à annoncer « la mort du web »<sup>148</sup>. Tout en gardant un pied dans le monde du web, il faut donc foncer plus que jamais dans le développement des applications mobiles<sup>149</sup>, qui sont en train de devancer les applications web en termes d'usage.

---

<sup>147</sup> On dit qu'il est bon pour toute personne soucieuse de sa carrière de prendre des décisions de trajectoire professionnelle avant ses 35 ans.

<sup>148</sup> Les statistiques récentes relatent que les gens passent désormais plus de temps sur internet avec un téléphone portable plutôt qu'avec un ordinateur.

<sup>149</sup> Pour programmer les applications utilisables sur iPhone ou iPad, le kit de développement mis à disposition des développeurs par Apple, bien que moderne et intéressant à utiliser, n'a rien à voir avec le développement web. (Le langage en question, *ObjectiveC*, est un dérivé du C.)

## GLOSSAIRE

<b>AMOA</b>	Assistance à maîtrise d'ouvrage. L'AMOA a pour mission d'aider la MOA (maîtrise d'ouvrage) à définir, piloter et exploiter le projet réalisé par la MOE.
<b>ASP</b>	ASP (Active Server Pages) est un standard Microsoft permettant de développer des applications web interactives, c'est-à-dire dont le contenu est dynamique. Une page web ASP (repérable par l'extension .asp) aura un contenu pouvant être différent selon certains paramètres (des informations stockées dans une base de données, les préférences de l'utilisateur,...) tandis que page web "classique" (dont l'extension est .htm ou .html) affichera continuellement la même information.
<b>Bloc stats</b>	Diminutif pour « bloc statistiques ». Un bloc stats désigne dans SATURNE l'ensemble des informations statistiques relatives à une session de tri.
<b>C.N.A.M.</b>	Ne signifie évidemment pas dans ce mémoire « Caisse nationale d'Assurance Maladie », mais « Conservatoire National des Arts & Métiers ».
<b>Consolidation</b>	Regroupement en un seul système, et en un même endroit, de données contenues dans des serveurs géographiquement dispersés afin d'en faciliter l'accès, la restitution, la sauvegarde et l'analyse. Cela apporte à la fois gain de temps et productivité.
<b>CQC</b>	Cap Qualité Courrier : vaste projet de modernisation de la branche Courrier, visant un taux de 90% d'arrivée du courrier à « J+1 » pour l'horizon 2010.
<b>CSS</b>	“Cascading Style Sheet” : Il s'agit des feuilles de style adjointes aux fichiers HTML, permettant de séparer les données de leur présentation.
<b>CTC</b>	Centre de tri du courrier
<b>DCD</b>	Document de conception détaillée : Il s'agit du document, dans tout projet à LA POSTE, dans lequel sont rédigées les spécifications techniques de l'application, c'est-à-dire par quels moyens techniques l'application va implémenter les besoins. Ce document n'est pas livré à la MOA.
<b>DCG</b>	Document de conception générale : Il s'agit du document, dans tout projet à LA POSTE, dans lequel sont rédigées les spécifications générales de l'application. Ce document doit être compréhensible par la MOA.
<b>DCOM</b>	DCOM est une couche Microsoft permettant à des processus de se transférer des informations d'un serveur à l'autre, en toute simplicité. Cela se fait au moyen d'une invocation de méthodes d'un objet à distance. (On parle de COM ou COM+ pour les invocations de méthodes d'un objet non distant.)
<b>DFB</b>	Définition fonctionnelle des besoins : Il s'agit du cahier des charges pour tout projet à LA POSTE. Ce document est rédigé par la MOA, et retouché à chaque avenant à l'expression des besoins.
<b>EAI</b>	Enterprise Application Integration : Un outil d'EAI garantie l'interopérabilité et

l'organisation de la circulation de l'information entre des applications hétérogènes, c'est-à-dire la communication des différentes applications constituant le système d'information de l'entreprise, voire même celles des clients, des partenaires ou des fournisseurs.

---

<b>ETL</b>	Extract Tranform Load : Outil chargé d'extraire, nettoyer et transformer les données émanant de sources diverses pour ensuite les insérer dans une base de données. Les ETL sont quelquefois considérés comme l'ancêtre des EAI.
------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<b>http</b>	Hyper Text Transfer Protocol : Il s'agit du protocole de communication utilisé par le web.
-------------	--------------------------------------------------------------------------------------------

---

<b>IUP MIAGE</b>	Institut universitaire professionnalisant en Maîtrise de méthodes informatiques appliquées à la gestion des entreprises
------------------	-------------------------------------------------------------------------------------------------------------------------

---

<b>JIT</b>	Just In Time : mode de compilation de code source qui se fait automatiquement, et juste avant son exécution (cas des servlets Java, ou encore des pages '.ASPX' utilisées en .NET).
------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<b>Mainframe</b>	Ordinateur d'une grande puissance de traitement, fonctionnant sur le modèle centralisé (en opposition aux applications multi-tiers, comme SATURNE). Ces systèmes vieillissants sont néanmoins reconnus pour leur fiabilité et leur performance.
------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<b>MOA</b>	Maîtrise d'ouvrage : C'est le donneur d'ordre. En informatique, il s'agit de l'équipe qui envoie à la MOE (Maîtrise d'œuvre) un cahier des charges de l'application à réaliser.
------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<b>MOE</b>	Maîtrise d'œuvre : Il s'agit de l'équipe d'informaticiens qui, à partir d'un cahier des charges transmis par la MOA, est chargée des spécifications et de la réalisation du projet informatique.
------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<b>Multi-threading</b>	Capacité d'un programme à lancer simultanément plusieurs processus de traitement.
------------------------	-----------------------------------------------------------------------------------

---

<b>ODBC</b>	Open DataBase Connectivity : Il s'agit d'une couche Microsoft rendant universelle la méthode d'accès aux différentes bases de données présentes sur le marché.
-------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<b>Pool</b>	Il s'agit d'un ensemble de ressources réutilisables, géré de façon commune pour un ensemble d'utilisateurs.
-------------	-------------------------------------------------------------------------------------------------------------

---

<b>Revamping</b>	Technologie qui permet d'accéder aux anciennes transactions pour terminal passif des mainframes à l'aide d'une interface graphique sur micro (Windows, généralement).  Le revamping est une vraie interface active (elle charge, voire contrôle les zones de saisie). Elle ne doit donc pas être confondue avec l'émulation, qui ne fait qu'afficher l'écran de terminal passif à l'identique (d'ailleurs, si c'est une page web qui émule, alors on parle de « web-to-host »).
------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<b>Scalabilité</b>	Vient de l'anglais 'Scalability' : Capacité d'un produit ou d'un système à changer de taille, selon l'évolution des besoins, tout en conservant ses propriétés fonctionnelles (extensibilité).
--------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<b>Scaleout</b>	Évolutivité verticale : il s'agit d'une méthode de dimensionnement d'un système informatique consistant à disposer de plusieurs instances de serveurs au lieu d'un seul.
-----------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<b>Scaleup</b>	Évolutivité horizontale : il s'agit d'une méthode de dimensionnement d'un système informatique consistant à augmenter les capacités du hardware (mémoire vive, nombre de
----------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

processeurs, etc.).

---

**Service web REST** Un service web est un programme informatique qui permet et facilite l'échange de données entre des programmes, applications et systèmes hétérogènes entre plates-formes distinctes. Il repose sur un réseau du web (internet ou intranet), à travers des applications et machines en temps réel.  
Un service web de type REST expose entièrement ses fonctionnalités comme un ensemble de ressources (URI) identifiables et accessibles par la syntaxe et la sémantique du protocole HTTP. Les services web de type REST sont donc basés sur l'architecture du web et ses standards de base que sont HTTP et URI.

---

**Session de tri** C'est l'application, sur telle machine de tri du courrier, de tel programme de tri sur un ensemble donné de plis postaux, commençant et terminant à telle heure.

---

**SGBD** Système de Gestion de bases de données : Outil assurant la gestion des relations et des requêtes sur les tables d'une base de données, comme par exemple Sybase. Dans le langage courant, on utilise souvent ce terme tout simplement comme synonyme de « base de données ».

---

**SI** Système d'informations : dans le contexte de ce mémoire, il s'agit des systèmes informatiques consacrés au traitement de l'information relative à toute l'activité courrier de LA POSTE.

---

**SQL** Structured query language : Langage commun à toutes les bases de données relationnelles, servant à effectuer des requêtes sur une base de données pour en extraire des informations.

---

**SSII** Société de Service en Ingénierie Informatique : La mission d'une SSII est d'accompagner une société cliente dans la réalisation d'un projet. Une SSII peut englober plusieurs métiers autour de l'informatique : conseil, conception et réalisation d'outils, maintenance ou encore formation.

---

**Sybase** Fait partie des SGBD les plus importants du marché avec Oracle, IBM DB/2, SQL Server, MySQL et PostgreSQL.

---

**TMA** Tierce Maintenance Applicative : consiste à externaliser la maintenance des applications, c'est-à-dire confier tout ou partie de la maintenance des applications à un prestataire informatique tiers contre rémunération en définissant des niveaux de disponibilité, de délais et de qualité.

---

**URA** Unité de Reconnaissance d'Adresse (concerne les machines de tri à LA POSTE)

---

**UML** Unified Modeling Language : Langage mondialement connu permettant de visualiser, spécifier, construire et documenter les artefacts d'un système. Il ne s'agit pas d'une méthode mais d'un langage permettant d'utiliser toute méthode orientée objet.

---

**UWE** UML-based web engineering

---

**Vidéocodage** Sur une machine de tri, système permettant de photographier l'adresse d'un courrier qui n'a pas pu être lu automatiquement par la machine : celui-ci est identifié manuellement par la suite, pendant que la lettre continue à voyager. La lettre est ensuite identifiée grâce au code barre qui y avait été apposé à cette occasion.

---

**WTX** (Websphere Transformation Extender)  
Outil d'EAI vendu par IBM

## INDEX

- Active X, 44, 49, 53  
Alain MICHEZ, 6  
Arts & Métiers de Paris, 7  
ASP, 44, 49, 53, 54, 78, 81, 114, 115, 117, 118, 125, 146, 150, 155  
Banque Postale, 10  
C.N.A.M., 132  
CDIS, 14, 15, 60, 61, 62, 63, 64, 65, 69, 73, 76, 77, 78, 79, 99, 137  
COM+, 53, 88, 91, 155  
Consolidation, 93, 155  
Courrier, 7, 9, 10, 11, 12, 17, 18, 19, 20, 21, 26, 36, 48, 67, 96, 150, 153, 155  
CQC, 8, 9, 17, 18, 19, 25, 26, 60, 71, 155  
CSS, 55, 105, 106, 107, 115, 116, 117, 118, 121, 128, 155  
CTC, 14, 15, 16, 21, 26, 28, 60, 62, 63, 64, 65, 78, 80, 155  
DCG, 22, 30, 31, 34, 35, 36, 40, 133, 149, 155  
DCI, 11  
DCOM, 49, 87, 89, 91, 155  
DECLIC, 11, 17, 20, 22, 23, 24, 28, 30, 31, 32, 33  
Deutsch Post, 13  
DFB, 30, 31, 34, 35, 36, 38, 78, 155  
DHTML, 55, 71, 95, 102, 105, 107, 108, 110, 111, 112, 117, 119, 121, 125, 129, 130, 131, 148  
DLL, 53  
DTC, 11, 36  
DUT, 7  
EAI, 25, 40, 59, 151, 155, 156, 157  
ETL, 156  
Europe, 11, 12, 13, 18  
GSI, 23  
HTML, 7, 52, 55, 58, 93, 102, 105, 106, 107, 108, 110, 119, 120, 121, 125, 128, 130, 131, 142, 148, 155  
HTTP, 48, 128, 129, 157  
IGP, 21, 26, 60, 62  
IIS, 25, 118  
Internet, 48, 55, 106, 128  
Intranet, 127  
IUP MIAGE, 7, 156  
J2EE, 18, 25, 150  
Jacky AKOKA, 6  
Javascript, 7, 44, 55, 56, 57, 71, 72, 93, 94, 95, 105, 106, 107, 108, 116, 117, 118, 120, 121, 122, 123, 125, 128, 129, 130, 131  
JIT, 53, 156  
Machine de tri, 57  
Mainframe, 42, 156  
MCD, 35, 132, 133, 134, 150  
MOA, 20, 23, 30, 34, 35, 36, 39, 62, 79, 81, 84, 96, 98, 99, 122, 133, 146, 151, 154, 155, 156  
MOE, 20, 23, 30, 35, 62, 79, 81, 84, 86, 98, 122, 151, 154, 155, 156  
MSMQ, 25, 49, 59  
ODBC, 72, 89, 156  
Oracle, 18, 25, 150, 157  
Recette fonctionnelle, 22  
Recette technique, 22  
Revamping, 156  
Scalabilité, 156  
Scale out, 156  
Scale up, 156  
Service web, 157  
Session de tri, 157  
SGBD, 7, 25, 42, 45, 49, 50, 54, 57, 58, 60, 69, 70, 73, 79, 80, 84, 85, 92, 93, 157  
SI, 6, 9, 20, 21, 23, 25, 42, 60, 91, 157  
Site pilote, 22  
SNTP, 11  
SSII, 7, 18, 153, 157  
Sybase, 18, 25, 40, 49, 69, 70, 72, 81, 82, 83, 84, 85, 86, 92, 93, 94, 98, 150, 157  
TMA, 8, 17, 29, 37, 96, 152, 153, 157  
TMC, 60, 67, 69, 70, 79, 96, 98, 99, 159  
UML, 8, 23, 61, 132, 133, 134, 135, 137, 142, 143, 144, 145, 146, 147, 148, 149, 150, 153, 157  
URA, 16, 21, 26, 157  
UWE, 8, 132, 133, 134, 135, 136, 141, 142, 144, 145, 146, 147, 148, 149, 150, 153, 157, 159  
VB, 25, 91  
Vidéocodage, 157  
Web, 48, 133, 159  
WTX, 25, 40, 49, 59, 84, 87, 88, 150, 151, 153, 157  
XML, 48, 105, 128, 129

## RÉFÉRENCES

- [ACTU WEB] Actualité du courrier (lien intranet)  
URL : <http://www.comco.courrier.intra.laposte.fr/>
- [COUR WEB] Le courrier en chiffres (lien intranet)  
URL : <http://www.chiffres.courrier.intra.laposte.fr>
- [D-POST 06] Rapports « *Deutsche Post et LA POSTE* » Nicolas LECAUSSIN, 2006
- [CS 99] « *Client/serveur, guide de survie* » (3<sup>ème</sup> édition), nov. 1999  
Auteurs : *Robert Orfali, Dan Harkey et Jeri Edwards*  
Éditions : *Vuibert Informatique*
- [CS WEB] Client Serveur, présentation  
URL : <http://phortail.org/webntic/Client-Serveur-presentation.html>
- [COL WEB] Cadres Online (base de connaissance sur l'emploi des cadres)  
URL: <http://www.cadresonline.com/>
- [COMP 05] Formation LA POSTE « *Traitements et composants* », 2005  
Support PowerPoint : *Formation composants.ppt*
- [IHM 03] Cours C.N.A.M. « *Principes d'ergonomie des interfaces web* », oct. 2003  
Module B2, « *Interaction Homme-Machine* »  
Auteur : *Jean-Marc PUJOS*
- [ING 09] Cours C.N.A.M. « *Ingénierie des Systèmes d'Information Web* », 2009,  
J. Akoka & I. Wattiau
- [UWE] Site officiel d'UWE  
URL : <http://uwe.pst.ifi.lmu.de>
- [WEB 09] Article de Stéphane Rossard synthétisant un débat diffusé sur techtoc.tv  
intitulé « *Qu'est-ce qui justifie le passage de l'Intranet 1.0 au 2.0 ?* », 2009  
URL : <http://techtoc.tv/event/19/qu-est-ce-qui-justifie-le-passage-de-l-intranet-1-0-au-2-0>
- [TMC 05] Rapports de TMC SATURNE, 2005  
(*document interne LA POSTE*)

## TABLE DES FIGURES

Figure 1 – Organisation du groupe LA POSTE.....	10
Figure 2 – Évolution du chiffre d'affaires du Courrier dans le groupe (en millions d'euros).....	11
Figure 3 – Répartition du chiffre d'affaires du groupe LA POSTE .....	12
Figure 4 – Cycle d'acheminement et de distribution du courrier .....	14
Figure 5 – Devenir d'un courrier à l'issue de son passage dans une machine de tri.....	15
Figure 6 – Vues intérieure et extérieure d'une machine de tri .....	16
Figure 7 – Le process industriel du Cap Qualité Courrier.....	17
Figure 8 – Machine de tri industriel Grand Format (MTI GF).....	18
Figure 9 – Cycle de vie d'un projet à LA POSTE : les acteurs .....	22
Figure 10 – Planification des phases de développement à LA POSTE .....	22
Figure 11 – Typologie du réseau informatique national à la DECLIC.....	24
Figure 12 – Imbrications des différentes applications du projet GALAXIE.....	27
Figure 13 – Planning du projet SATURNE.....	28
Figure 14 – Cycle de vie type du projet SATURNE .....	29
Figure 15 – Schématisation du cycle en V d'un projet informatique .....	30
Figure 16 – Interactions entre les acteurs LA POSTE et APSIDE pour les projets SNT et SATURNE..	39
Figure 17 – Les différents modèles client-serveur .....	43
Figure 18 – Découpage d'une application client-serveur 3 tiers avec un serveur web Microsoft....	45
Figure 19 – Schéma d'architecture technique de l'application SATURNE .....	47
Figure 20 – Schéma de la perspective logicielle de SATURNE .....	50
Figure 21 – Architecture de SATURNE : la décomposition en briques logiques .....	51
Figure 22 – Environnement de développement VB6 d'un composant métier de SATURNE .....	54
Figure 23 – Outil de compactage des fichiers Javascript de la solution SATURNE .....	56
Figure 24 – Imbrications fonctionnelles des différents profils dans SATURNE .....	61
Figure 25 – Répartition de la montée en charge prévisionnelle dans SATURNE.....	65
Figure 26 – Schéma des différentes facettes architecturales d'un système .....	68
Figure 27 – Architecture fonctionnelle du scénario « Prise en compte des préférences utilisateurs » .....	75
Figure 28 – Architecture fonctionnelle du scénario « Création d'une application CDIS ».....	76

Figure 29 – Architecture fonctionnelle du scénario «Profil CDIS » .....	77
Figure 30 – Architecture fonctionnelle du scénario «Pré-génération de la dernière journée postale » .....	80
Figure 31 – Architecture fonctionnelle du scénario «Base d'historisation / Base à J+1 » .....	83
Figure 32 – Scénario prévisionnel d'une solution complète de type « scale-out » dans SATURNE...	85
Figure 33 – Fonctionnement actuel de l'échange de données de SNT vers SATURNE.....	87
Figure 34 – Scénario d'évolution de l'échange de données vers un composant de service .....	88
Figure 35 – Recherche et affichage des erreurs dans SATURNE (exemple).....	94
Figure 36 – Copies d'écran de l'application GIPSI STATS.....	101
Figure 37 – Première version de SATURNE : page d'accueil .....	103
Figure 38 – Première version de SATURNE : liste des blocs statistiques .....	103
Figure 39 – Dernière version de SATURNE (ici, la page d'accueil) .....	104
Figure 40 – SATURNE V2 : liste des blocs statistiques.....	107
Figure 41 – Exemples d'utilisation d'une liste à multi-sélection .....	108
Figure 42 – Exemple d'utilisation d'une liste d'affinage (par la multi-sélection).....	109
Figure 43 – Exemple d'utilisation d'une liste d'affinage (par la saisie).....	109
Figure 44 – Exemple d'édition multiligne.....	110
Figure 45 – Intégration d'un composant DHTML externe de type calendrier .....	110
Figure 46 – Page de chronogrammes dans SATURNE.....	111
Figure 47 – Visualisation avancée d'un chronogramme dans SATURNE.....	112
Figure 48 – Exemple d'édition avec histogramme dans SATURNE .....	112
Figure 49 – Exemple d'édition volumineuse dans SATURNE : « les stats GF » .....	113
Figure 50 – Exemple d'édition au format RTF dans SATURNE.....	114
Figure 51 – Exemple d'édition « Matrice de flux TG1 » dans SATURNE .....	115
Figure 52 – Symbolisation de l'usage de fonctions générique .....	117
Figure 53 – Affichage complet d'une édition multiligne dans SATURNE .....	119
Figure 54 – Affichage partiel d'une édition multiligne dans SATURNE .....	120
Figure 55 – Tri d'une édition multiligne dans SATURNE .....	120
Figure 56 – Affichage d'une liste multi-sélection volumineuse dans SATURNE.....	121
Figure 57 – Exemple de page de l'aide en ligne contextuelle de SATURNE.....	124
Figure 58 – « Les bonnes raisons de devenir une Entreprise 2.0 » .....	126
Figure 59 – Diagramme des cas d'utilisation de SATURNE (général).....	136

Figure 60 – Diagramme des cas d'utilisation de SATURNE (la consultation de blocs stats).....	137
Figure 61 – Diagramme de classes de SATURNE (version simplifiée) .....	138
Figure 62 – Diagramme d'états (pour un bloc stats) .....	138
Figure 63 – Diagramme de la structure de navigation de SATURNE (version simplifiée).....	139
Figure 64 – Diagramme d'accès de SATURNE (version simplifiée) .....	140
Figure 65 – Diagramme de présentation général de SATURNE (version simplifiée) .....	141
Figure 66 – Diagramme de présentation de la fonction «Liste de blocs stats » (version simplifiée) .....	142
Figure 67 – Diagramme de séquence pour le scénario «Lancement d'une édition via les chronogrammes ».....	143

## LISTE DES TABLEAUX

Tableau 1 – Deutsche Post versus LA POSTE .....	13
Tableau 2 – Ensemble des documents à livrer dans le cadre de notre prestation de service .....	33
Tableau 3 – Répartition de la montée en charge prévisionnelle dans SATURNE .....	64
Tableau 4 – Pistes d'optimisation du module SGBD .....	70
Tableau 5 – Évolution de l'architecture fonctionnelle .....	79
Tableau 6 – Évolution de l'architecture logicielle.....	81
Tableau 7 – Évolution de l'architecture technique logique .....	84
Tableau 8 – Évolution de l'architecture technique physique, ou « scaleup/scaleout » .....	86
Tableau 9 – Contournement des aberrations rencontrées avec Sybase .....	92
Tableau 10 – Blindage de la partie IHM .....	93
Tableau 11 – Bilan de la revue de code de SATURNE 2.0 et principes d'améliorations .....	95
Tableau 12 – Axes d'amélioration de base des performances d'une IHM.....	117
Tableau 13 – Remise en question des anciens prédicats du web.....	130