

Rapport de stage DESS Informatique Fondamentale



MOTOROLA

**Sujet : Gestion des délégations et
exceptions dans WEBCongés**

Etudiant : Antoine Jacquet

Maître de stage : Mr Daniel Lazerge

Encadrant universitaire : Mr Marc Boyer

Année Universitaire 2002-2003

Remerciements

Je tiens d'abord à remercier Mr Daniel Lazerge, mon maître de stage, qui m'a encadré efficacement pendant la durée de mon travail et qui a su être disponible pour répondre à toutes mes questions.

Je remercie Mr Jean-Claude Dufau qui m'a permis de m'intégrer efficacement dans son équipe.

Je tiens également à remercier Mr Bruce Gonzalvez qui m'a guidé et conseillé dans la spécification et la conception de mon application.

Un grand merci à Mme Angélique Venica Portenseigne pour ses conseils autour de l'interrogation et de la programmation sur la base de données, et pour le temps qu'elle a accordé aux tests de validation de l'application.

Je remercie aussi Mr Joseph Garnier pour sa validation des différents standards Motorola.

Je salue également toutes ces personnes et les autres membres du département pour leur gentillesse et leur bonne humeur qui contribuent à un climat de travail agréable.

Merci de plus à Mr Marc Boyer, mon encadrant universitaire, pour sa disponibilité, son suivi et ses conseils avisés sur mes présentations et mon rapport.

Enfin je tiens à remercier Mr Arcangeli et Mr Cherbonneau, responsables du DESS Informatique Fondamentale, ainsi que l'ensemble des enseignants de cette formation, pour la qualité de leurs enseignements.

Sommaire

1. INTRODUCTION.....	5
2. PRESENTATION DE L'ENTREPRISE	6
2.1. HISTOIRE DE MOTOROLA	6
2.2. MOTOROLA EN FRANCE	8
2.3. MOTOROLA TOULOUSE	8
2.4. L'ORGANISATION HRIS AU SEIN DE MOTOROLA	9
2.5. PRESENTATION D'ACS.....	10
2.6. LA « QUALITE » LOGICIELLE A MOTOROLA.....	10
3. PRESENTATION DU SUJET DE STAGE.....	13
3.1. WEBCONGES.....	13
3.2. AMELIORATIONS SOUHAITEES	14
3.3. SOLUTION TEMPORAIRE.....	14
3.4. OBJECTIFS	14
4. SPECIFICATION DES BESOINS ET CONTRAINTES	15
4.1. DEFINITIONS.....	15
4.1.1. <i>Délégation</i>	15
4.1.2. <i>Exception</i>	15
4.2. STANDARD MOTOROLA.....	16
4.3. NAVIGATION	17
4.4. FONCTIONNALITES	18
4.4.1. <i>Login</i>	18
4.4.2. <i>Délégation</i>	18
4.4.3. <i>Exception « in-site »</i>	18
4.4.4. <i>Exception « off-site »</i>	19
4.4.5. <i>Super utilisateur</i>	19
4.5. CONTRAINTES DIVERSES	20
4.5.1. <i>Arborescence de délégation</i>	20
4.5.2. <i>Blocage de l'auto validation</i>	20
4.5.3. <i>Délégation multiple</i>	20
4.5.4. <i>Exemple</i>	21
4.6. AUTOMATISATIONS	22
4.6.1. <i>Rappels</i>	22
4.6.2. <i>Echéances de délégation</i>	22
4.6.3. <i>Changements hiérarchiques</i>	22
4.6.4. <i>Construction de la nouvelle hiérarchie</i>	22
4.6.5. <i>Mails managers « off-site »</i>	22
5. PLANIFICATION	23
5.1. PLANNING	23
5.2. HISTORIQUE DES ACTIVITES	24
5.2.1. <i>Intégration</i>	24
5.2.2. <i>Cahier des charges</i>	24
5.2.3. <i>Spécification</i>	24
5.2.4. <i>Phase 1</i>	24

5.2.5.	<i>Phase 2</i>	25
5.2.6.	<i>Documentations</i>	25
5.2.7.	<i>Rapport de stage</i>	25
6.	ENVIRONNEMENT, OUTILS ET LANGAGES	26
6.1.	ENVIRONNEMENT	26
6.2.	SERVEUR DE TEST.....	26
6.3.	SERVEUR DE PRODUCTION.....	26
6.4.	OUTILS	27
6.4.1.	<i>Oracle SQL*Plus 8.0</i>	27
6.4.2.	<i>Oracle Web Application Server</i>	27
6.4.3.	<i>Terminal Service Client</i>	27
6.4.4.	<i>WEBCongés</i>	28
6.4.5.	<i>Temptation</i>	28
6.4.6.	<i>Divers</i>	28
6.5.	LANGAGES	29
6.5.1.	<i>SQL</i>	29
6.5.2.	<i>PL/SQL</i>	29
6.5.3.	<i>Perl</i>	29
7.	CONCEPTION GENERALE	30
7.1.	PRESENTATION	30
7.2.	MODELE CONCEPTUEL DE DONNEES	30
7.3.	MODELE LOGIQUE DE DONNEES.....	31
7.3.1.	<i>employee_sap_data</i>	31
7.3.2.	<i>Exception</i>	32
7.3.3.	<i>Délégation</i>	32
7.3.4.	<i>Résultat_employé</i>	32
7.4.	MODELE PHYSIQUE DE DONNEES	33
7.5.	DIAGRAMME DES FLUX.....	34
7.6.	MAILINGS.....	35
8.	REALISATION	36
9.	TESTS	43
9.1.	TRACABILITY MATRIX	43
9.2.	PLAN DE TEST.....	43
9.3.	JEU DE TEST.....	43
10.	CONCLUSION	44

ANNEXES :

Annexe 1 : Exemple de code PL/SQL (batch.sql)

Annexe 2 : Exemple de code Perl (batch.pl)

Annexe 3 : Extrait du plan de test

1. Introduction

WEBCongés est une application pour la gestion des demandes de congés avec cycle d'approbation. Elle permet à un employé de faire simplement une demande de congés sur un site Web interne, puis à son manager de traiter cette demande.

Dans le cadre de l'utilisation de ce produit à Motorola, un certain nombre de besoins spécifiques à cette entreprise se sont fait ressentir, et l'application WEBCongés n'était pas en mesure d'y répondre.

Mon stage de DESS Informatique Fondamentale a été mis en place afin de résoudre ses problèmes en proposant des solutions adaptées.

Il s'est déroulé du 28 octobre 2002 au 27 juin 2003 au sein du département HRIS à Toulouse.

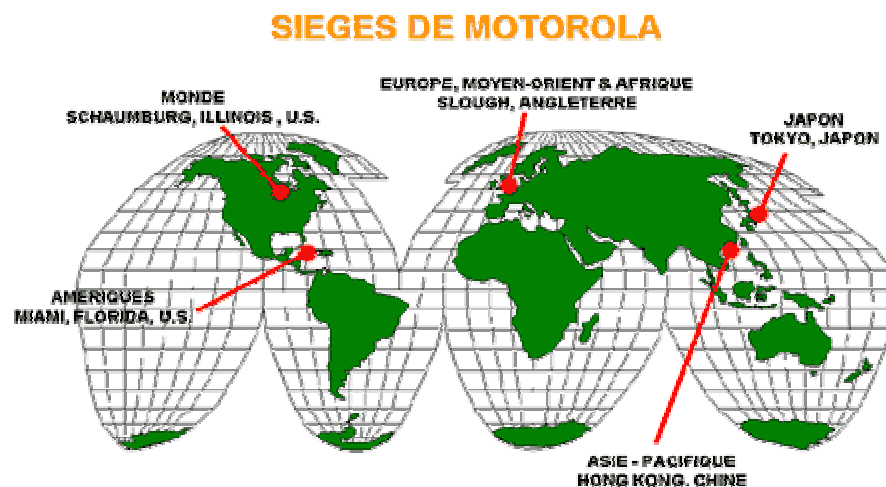
Ce département a pour fonction de gérer le service informatique des ressources humaines de Motorola.

L'encadrement de mon stage est assuré par Mr Daniel Lazerge dans l'entreprise et par Mr Marc Boyer à l'université.

2. Présentation de l'entreprise

Motorola est un grand groupe international spécialisé dans l'électronique professionnelle de haute technologie et plus spécialement dans les radiocommunications et les semi-conducteurs.

La direction Générale de Motorola est installée aux Etats-Unis, à Schaumburg dans l'Illinois. Motorola emploie 110 000 personnes dans le monde dont 20 000 en Europe et plus de 3 000 en France. La société est implantée sur plus de 700 sites dans 69 pays.



2.1. Histoire de Motorola

1928 :

Paul V. Galvin et son frère Joseph E. Galvin rachètent l'usine d'éliminateurs de piles d'une société en faillite. Le 25 septembre 1928, ils fondent la Galvin Manufacturing Corporation qui emploie cinq personnes.

1930 :

La Galvin Manufacturing Corporation met au point le premier autoradio véritablement fonctionnel et de prix abordable.

1940 :

Motorola met au point le premier radiotéléphone portable pour les services de transmissions de l'armée américaine.

1955 :

Motorola adopte un nouveau logo, le fameux M en forme d'ailes de chauve-souris.

1936



1940



1958 :

Le premier produit commercial de Motorola comportant des transistors est un autoradio. Un nouveau produit de radiocommunications fait son apparition. Il s'agit d'un petit récepteur de poche appelé "pager" permettant à des utilisateurs équipés de cet appareil de recevoir des messages radios personnels. La division des produits semiconducteurs est créée.

1958



1969 :

Les premiers mots prononcés par l'astronaute Neil Armstrong depuis la Lune sont retransmis sur Terre au moyen d'un transpondeur conçu et fabriqué par la division "Electronique Gouvernementale" de Motorola.

1969



1974 :

Motorola lance son premier microprocesseur, le 6800.

1989 :

Le téléphone cellulaire personnel Micro-TAC est le plus petit et le plus léger du marché.

1996 :

Motorola lance le StarTAC, premier téléphone cellulaire de poche, le plus petit et le plus léger des portables au monde.

1974



2001 :

Timeport 270 : telephone compatible BlueTooth.

2002 :

Motorola lance le mobile V70. C'est l'équipe PCS Toulouse qui a eu la responsabilité mondiale du produit.

2002



2001



1996



1989



2.2. *Motorola en France*

Motorola en France, c'est aujourd'hui plus de 3 000 personnes et un chiffre d'affaires de 1,1 milliards d'euros, dont près de 50% à l'exportation.

L'usine de **Toulouse**, inaugurée en 1967, possède un centre de développement et de design qui travaille sur les composants de classe mondiale. Avec cette unité, employant près de 2 500 personnes, Motorola est le plus important fabricant de semi-conducteurs en France.



L'usine d'**Angers**, implantée en 1973, fabrique des composants et des modules d'électronique automobile et emploie environ 400 personnes.



La France accueille également un centre de Recherche et Développement "Corporate". Ouvert en août 1996, ce Centre à vocation mondiale, implanté sur le district du plateau de **Saclay**, est dédié aux télécommunications sans fil. Ce centre a également pour mission de coordonner les recherches menées par Motorola au sein de la zone Europe-Moyen-Orient-Afrique.



Enfin le site de Crolles 2, situé près de Grenoble, a été inauguré cette année. C'est une usine pilote de Recherche et Développement en nano-électronique associant trois des principaux fabricants mondiaux de semi-conducteurs dont Motorola.

2.3. *Motorola Toulouse*

- 1990** Motorola Toulouse en collaboration avec le LAAS/CNRS développe la technologie de transistor de puissance « planar » haute tension, utilisée dans l'allumage des lampes fluorescentes.
- 1993** Motorola Semi-Conducteurs a décidé d'implanter à Toulouse un centre de recherche et de développement de circuits intégrés de puissance intelligente.
- 1995** Un centre de recherche commun entre Motorola Semi-Conducteurs et le LAAS/CNRS a également été créé afin de développer de nouvelles technologies de semi-conducteurs avancées.

2.4. L'organisation HRIS au sein de Motorola

L'organisation HRIS (Human Resources Information System) est une organisation répartie dans le monde entier et dont le but est de mettre en place et de maintenir des systèmes d'information pour les Ressources Humaines (RH) de Motorola. Il s'agit en fait du service informatique des ressources humaines.

Le rôle de HRIS est de travailler en étroite collaboration avec les ressources humaines afin de comprendre leurs besoins, de concevoir des solutions informatiques globales (mondiales), de les réaliser et enfin, de les implémenter dans toutes les organisations RH du monde entier.

Cette organisation a notamment mis en place des applications telles que SAP, Global Data Warehouse et ENET que j'ai utilisées pour mon stage puisqu'elles me fournissaient toutes les informations concernant un salarié (nom, prénom, identifiant, manager, ...):



SAP HR est un progiciel permettant aux ressources humaines d'accomplir l'ensemble des tâches liées à la gestion du personnel. Il est composé de deux principaux modules:

- Administration du personnel : état civil, adresse, historique de scolarité, RIB, ...
- Gestion de l'organisation Motorola (Business, Départements, Organigrammes)

SAP a été mis en place dans le monde entier afin que toute l'organisation RH utilise un système unique pour la gestion du personnel.



Global Data Warehouse / Ad hoc reporting

Le "Global Data Warehouse" est la base de données centrale pour les données RH dans le monde (basé sur les données contenues dans SAP). Les ressources humaines ont ainsi la possibilité de construire des rapports à partir de cette base de données en utilisant des outils d'interrogation de base de données comme Brio*Query.



ENET : Application WEB en libre service (Intranet)

Permet aux employés d'accéder et de modifier leurs données personnelles (adresse, personne à contacter en cas d'urgence, ...).

Fournit aux managers la possibilité d'accéder aux informations de son personnel (rapports prédéfinis) et d'initier des changements (demande de promotion, d'augmentation de salaire, changement de département, ...). Ce logiciel est connecté directement avec SAP permettant ainsi des mises à jour en temps réel.

2.5. Présentation d'ACS



ACS (Affiliated Computer Services) est un fournisseur de services pour les entreprises.

Cette société américaine créée en 1988 propose à ses clients des solutions intégrées, notamment informatiques, mais aussi dans de nombreux domaines incluant les ressources humaines.

Implanté dans plus de 48 pays, ACS est un des leaders mondiaux dans son secteur d'activité et emploie plus de 40 000 personnes à travers le monde.

Dans le cadre de l'externalisation de certains de ses départements, Motorola a décidé de faire appel à ACS pour reprendre les activités du département HRIS.

Cette opération de rachat s'est concrétisée pendant mon stage et la transition s'est opérée progressivement.

2.6. La « qualité » logicielle à Motorola

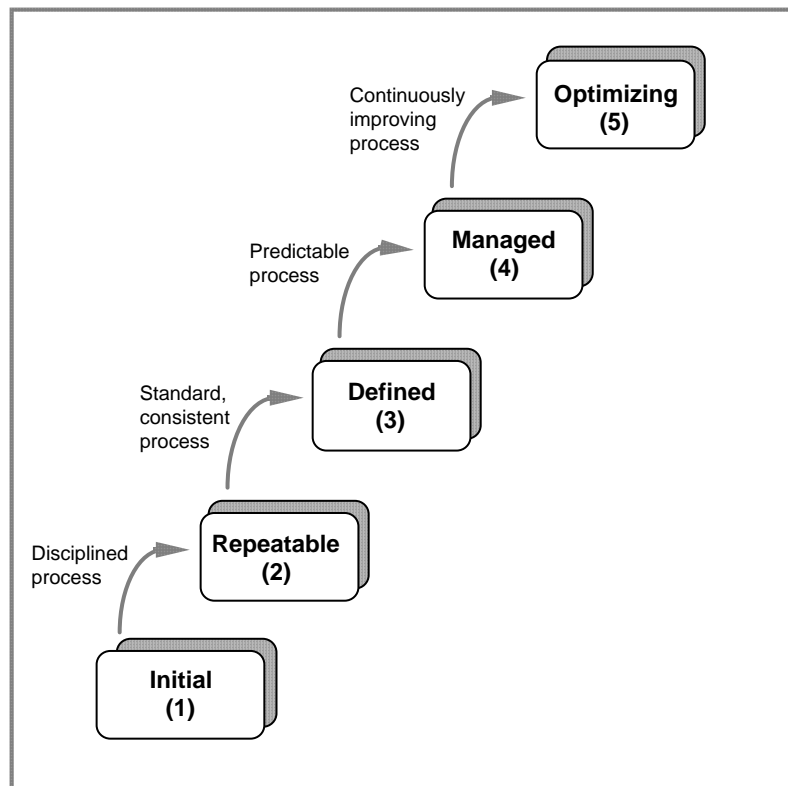
La qualité est une des priorités de Motorola. Au niveau logiciel, les organisations IT (Information Technology) ont été certifiées SEI Level3.

Qu'est ce que le SEI Level3 ?

En novembre 1986, le SEI (Software Engineering Institute) a commencé à développer un modèle permettant de mesurer la maturité des processus afin d'aider les organisations à améliorer leur processus "Logiciel". Le SEI a fait évoluer ce modèle et c'est ainsi que le CMM (Capability Maturity Model) pour le logiciel a vu le jour.

Le CMM est basé sur la connaissance et l'expérience acquise lors de l'évaluation des processus et méthodes logicielles (audits) et grâce aux informations directement fournies par les industries et le gouvernement américain. Ce modèle fournit aux organisations informatiques des recommandations pour établir des programmes d'amélioration de leurs méthodes de gestion de projet et leurs procédures qualité.

Les 5 niveaux du "Software Process Maturity"



Les caractéristiques suivantes des cinq niveaux de maturité permettent de comprendre quels sont les changements faits à chaque niveau :

- 1) *"Initial"* Les méthodes logicielles sont des méthodes "ad hoc", et occasionnellement même chaotique. Très peu de méthodes sont définies et le succès dépend de l'effort individuel.
- 2) *Repeatable* Des méthodes de gestion de projet basiques sont établies pour suivre le coût, le planning du projet, et les fonctionnalités du logiciel. Des méthodes et procédures sont en place afin de réutiliser dans des contextes similaires les expériences acquises sur des projets précédents.

- 3) *Defined* La méthode de gestion de projet et les activités de développement logiciel sont documentées, standardisées et intégrées dans un référentiel standard (procédures qualité). Tous les projets informatiques suivent cette méthode et utilisent les documents standards.
- 4) *Managed* Des mesures détaillées sont effectuées sur le suivi des procédures qualité ainsi que sur la qualité des produits. Les procédures qualité sont comprises et les produits logiciels contrôlés.
- 5) *Optimizing* L'amélioration continue des procédures qualité est un souci permanent. Cela est effectué à partir de commentaires et critiques effectués par les personnes utilisant ces procédures et en prenant en compte de nouvelles idées et méthodes.

Tout au long de mon stage j'ai donc été amené à suivre les procédures qualité et à utiliser les documents standards pour documenter les différentes phases de mon projet.

3. Présentation du sujet de stage

Pour mieux cadrer le sujet de mon stage, nous allons dans un premier temps voir son contexte : WEBCongés.

C'est en effet suite à l'utilisation de cette application que des améliorations possibles sont apparues et que des besoins se sont faits ressentir.

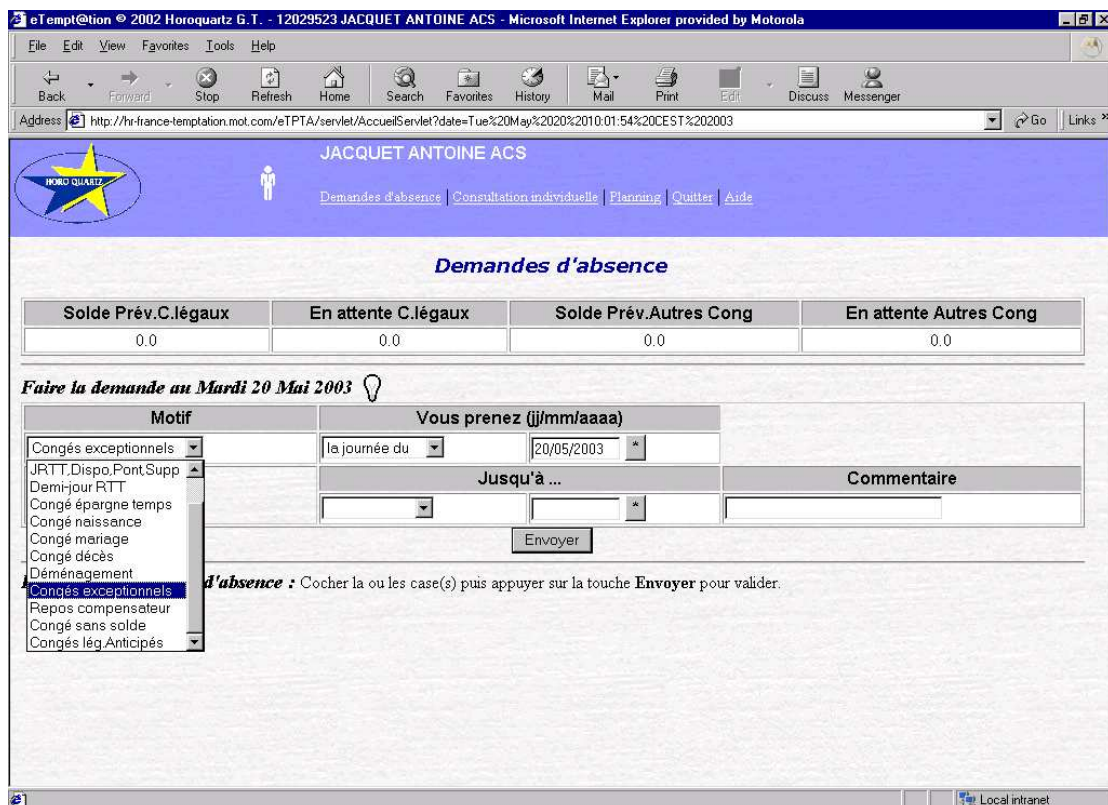
Les objectifs de mon travail seront plus évidents à cerner après cette présentation, puisque le contenu de mon stage et son déroulement sont étroitement liés à l'application WEBCongés.

3.1. WEBCongés

WEBCongés est une application Intranet permettant la gestion des demandes de congés avec cycle d'approbation.

L'employé peut faire une demande de congés sur un site Web interne, puis cette demande est validée par son manager.

C'est un logiciel développé par l'entreprise Horo Quartz qui fournit des solutions logicielles pour la gestion du temps de travail.



Une demande d'absence sur WEBCongés

L'application permet également d'autres opérations comme par exemple la consultation du planning.

WEBCongés a été déployé à Motorola en France par l'équipe HRIS.

3.2. Améliorations souhaitées

Le système WEBCongés gère la liste des employés sous forme de groupe : chaque groupe d'employés a un responsable qui valide les congés.

Par contre la hiérarchie Motorola est une structure arborescente.

Un premier système d'import a été mis en place par l'équipe HRIS pour transformer la structure hiérarchique arborescente en une structure de groupes équivalente pour l'adapter à l'application WEBCongés.

Le cycle d'approbation des congés est maintenant basé sur la hiérarchie Motorola officielle, ce qui n'est pas adapté aux deux cas suivants qui sont la base de mon travail :

- Si le manager est absent (par exemple dans le cas de congés), il n'est pas disponible pour valider les demandes d'absences de ses subordonnés.
- Si le manager est extérieur au site (certains employés de Toulouse ont par exemple des managers à l'étranger), c'est une personne locale (sur le même site que l'employé) qui doit valider les congés pour des contraintes légales.

3.3. Solution temporaire

Le département HRIS a mis en place une solution temporaire couvrant le dernier de ces problèmes en créant une table d'exceptions.

La gestion de cette table est consommatrice en ressource, et d'une manipulation délicate source d'erreur.

De plus cette solution temporaire ne permet pas de gérer tous les cas. Par exemple les congés des managers ne sont pas traités et donc les subordonnées ne peuvent pas faire valider leurs congés pendant l'absence de leur responsable.

3.4. Objectifs

La mise en place d'une solution fiable, facilement maintenable, et couvrant l'ensemble des problèmes ci-dessus, a motivé la mise en place de mon stage.

L'objectif de mon travail est donc de remédier à ces problèmes en proposant une solution automatisée pour que chaque employé puisse gérer les particularités le concernant et ainsi soulager le département HRIS du traitement de ces cas particuliers.

4. Spécification des besoins et contraintes

Dans la documentation que j'ai rédigée pour l'entreprise, l'expression des besoins et des contraintes s'expriment au travers d'un document qui correspond au cahier des charges : le RSD (Requirement Specification Document).

L'ensemble des besoins et des contraintes sont numérotés (ce sont les Requirements), ce qui permet par la suite de facilement construire une matrice de tests (Tracability Matrix).

4.1. Définitions

Pour remédier aux problèmes engendrés par WEBCongés que l'on vient d'évoquer, des règles sont mises en place pour permettre une hiérarchie d'approbation spécifique aux congés.

Ces règles particulières sont classées en deux catégories selon leur nature :

- les délégations
- les exceptions

4.1.1. Délégation

La délégation permet à un manager qui sera indisponible pendant une période de plus d'une semaine de donner ses droits de validation et de consultation des congés à une autre personne.

Elle est temporaire et affecte tous les subordonnés du manager.

Elle concerne potentiellement tous les managers.

4.1.2. Exception

L'exception permet à un employé de faire valider ses congés par une personne locale, plutôt que de faire intervenir son manager.

Elle concerne essentiellement les employés qui ont un manager « off-site » (c'est à dire un manager sur un autre site, par exemple à l'étranger).

Toutefois elle pourra exceptionnellement s'appliquer à certains employés avec un manager « in-site » (c'est à dire un manager sur le même site).

Elle est permanente et n'affecte qu'un seul subordonné du manager.

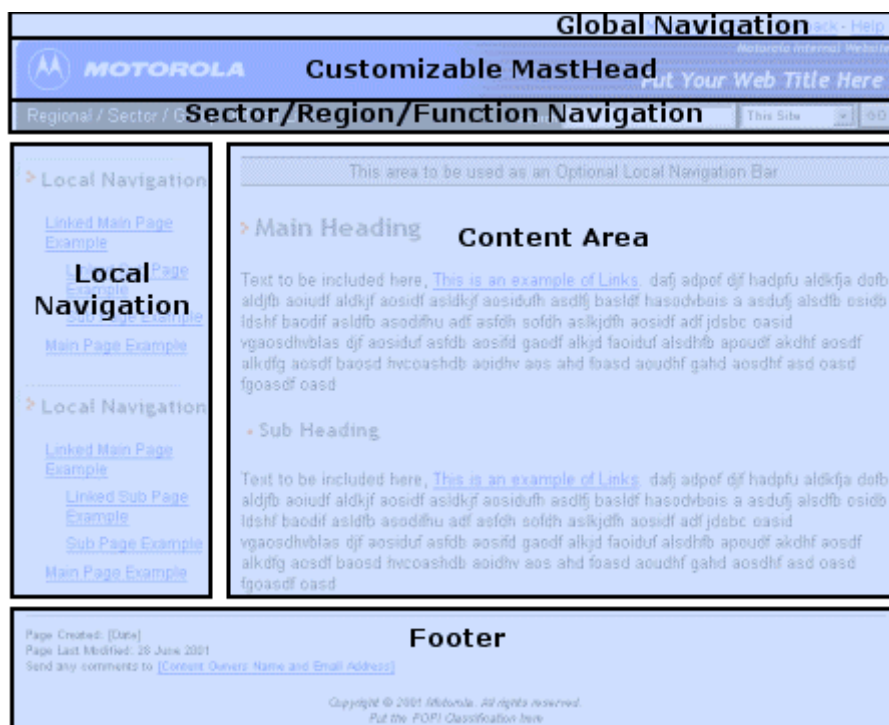
4.2. Standard Motorola

L'entreprise Motorola demande de respecter des standards dans différents domaines (documentation du projet, utilisation du logo Motorola, publications, présentations, etc).

La mise en place d'un site Web interne répond également à certaines contraintes d'aspect et d'utilisation. Il est en effet nécessaire que chaque nouveau site Web s'intègre de façon homogène dans l'Intranet existant de l'entreprise.

Le client souhaite donc que le site Web développé dans le cadre de mon projet soit au standard des sites Intranets de Motorola.

Voici un résumé des principales contraintes visuelles données par ce standard. Elles sont documentées sur un site interne qui liste les points à respecter.



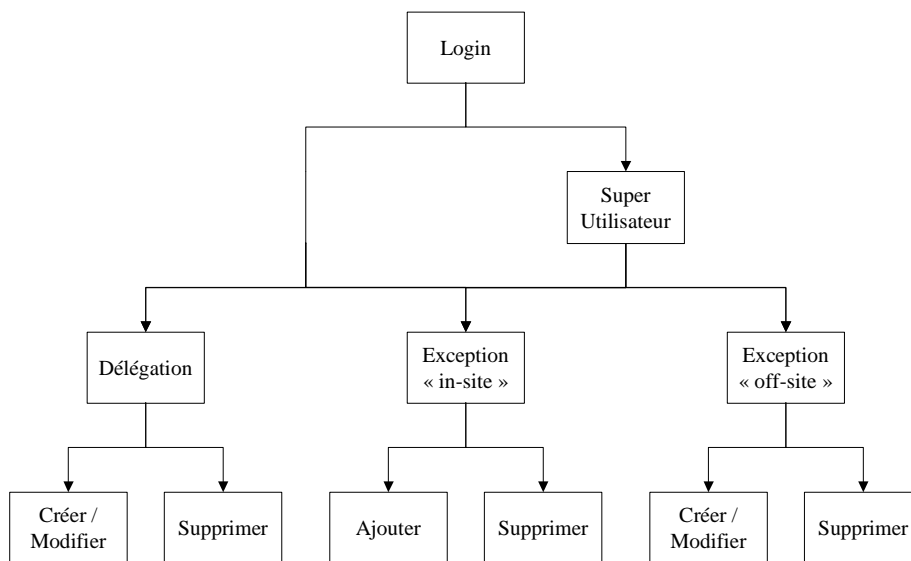
Les parties « Global Navigation », « Customizable MastHead », « Sector Navigation » et « Footer » sont sensiblement les mêmes que pour les sites Intranets HRIS.

L'affichage des menus et la police de caractères utilisée répondent aussi à des normes.

4.3. Navigation

Voici le modèle de navigation retenu pour le site Web dans le cahier des charges. Afin de ne pas surcharger ce diagramme les flèches de retour ne sont pas matérialisées.

On cherche surtout à mettre en évidence les étapes pour réaliser une action souhaitée.



La visibilité du premier niveau n'est pas forcément complète. Ainsi un employé qui n'a pas de responsabilités de validation des congés ne verra pas le menu de délégation.

Le contenu de chaque écran ainsi que les classes d'utilisateurs pouvant y accéder est défini avec les fonctionnalités qui suivent.

4.4. Fonctionnalités

4.4.1. Login

L'utilisateur doit pouvoir se connecter avec la procédure standard d'authentification Motorola, basée sur un serveur LDAP unique.

Il a alors accès à un écran lui permettant de choisir parmi la délégation et les exceptions.

Seules les fonctions disponibles pour cet utilisateur seront accessibles.

4.4.2. Délégation

Ce menu n'est accessible qu'aux personnes qui ont une responsabilité de validation des congés (managers).

C'est ici qu'une personne qui part en congés va pouvoir gérer la délégation de la gestion des congés de ses subordonnés à une autre personne.

Pour toutes les opérations, le délégué choisi est prévenu par mail.

4.4.2.1. Créer / Modifier

Dans cette rubrique, le manager va pouvoir choisir son délégué.

Le choix se fera dans une liste déroulante énumérant les personnes habilitées. Cette liste sera créée à partir de l'arborescence de validation des congés, en tenant compte des exceptions (cf 4.5.1).

Il devra également renseigner la date de début (initialisée au lendemain de l'opération) et la date de fin de délégation.

La période de délégation devra être valide (date de début et de fin cohérentes).

Si le manager a déjà une délégation en cours, le formulaire sera rempli avec les options courantes, et il pourra les modifier.

4.4.2.2. Supprimer

Si le manager est disponible plus tôt que prévu, ou s'il souhaite annuler sa délégation, cette page lui permettra de supprimer sa délégation en cours.

4.4.3. Exception « in-site »

Ce menu n'est accessible qu'aux managers (et pas aux délégués).

C'est ici qu'un manager peut choisir le responsable congés pour un de ses subordonnés.

Pour toutes les opérations, le responsable congés et l'employé concerné sont prévenus par mail.

4.4.3.1. Ajouter

Sur cette page, le manager choisit le responsable congés pour un de ses subordonnés. Il n'y a pas de contraintes sur le choix du responsable congés.

4.4.3.2. Supprimer

Si la désignation d'un responsable particulier ne se justifie plus, il est possible de l'annuler sur cette page.

4.4.4. Exception « off-site »

Contrairement aux exceptions « in-site », les exceptions « off-site » sont plus fréquentes et répondent à des contraintes légales : la personne qui valide les congés d'un employé doit être sur le même site.

Ce menu n'est accessible qu'aux employés qui ont un manager « off-site ».

C'est ici qu'un employé peut choisir un responsable congés local.

Pour toutes les opérations, le responsable congés et le manager « off-site » sont prévenus par mails.

Ces mails seront bilingues (français / anglais) afin d'être compris par les managers « off-site ».

4.4.4.1. Créer / Modifier

Sur cette page, l'employé peut choisir le responsable qui validera ses congés.

Il n'y a pas de contraintes sur le choix du responsable.

L'employé peut également activer une option qui permettra à son manager « off-site » d'être prévenu de chaque période de congés validée par son responsable congés.

4.4.4.2. Supprimer

Si la désignation d'un responsable particulier ne se justifie plus, il est possible de l'annuler sur cette page.

4.4.5. Super utilisateur

Cet écran n'est visible que par un super utilisateur.

Il lui permet de choisir sous quel utilisateur il souhaite se connecter.

Il peut également modifier la liste des supers utilisateurs dans ce menu.

4.5. Contraintes diverses

En plus des besoins et des contraintes que l'on vient de voir, un certain nombre de contraintes diverses ont été énoncées concernant les aspects sécurité, performance et matériel disponible. Elles ne sont pas détaillées ici car elles sont spécifiques à l'architecture de Motorola.

De plus des contraintes spécifiques au client ont été énoncées :

4.5.1. Arborescence de délégation

Les contraintes suivantes ont été énoncées par le client sur l'arborescence de délégation :

- Un manager peut déléguer à son propre manager
- Un manager peut déléguer à un de ses subordonnés directs
- Un manager peut déléguer à un des subordonnés directs de son manager

Dans tous les cas on ne peut pas déléguer à une personne qui a déjà délégué sur une période en chevauchement.

Ces contraintes portent sur la hiérarchie Motorola modifiée par les exceptions.

4.5.2. Blocage de l'auto validation

Lorsqu'un manager délègue à un de ses subordonnés, celui-ci serait en mesure de valider ses propres congés.

Cette situation n'est pas acceptable, et un employé ne peut donc pas s'auto valider ses congés.

Si cette situation vient à se présenter, l'employé devra exceptionnellement faire appel à un super utilisateur.

Le manager doit donc choisir judicieusement son délégué.

4.5.3. Délégation multiple

Dans une première version de la spécification, il avait été envisagé qu'un délégué puisse à son tour re-déléguer.

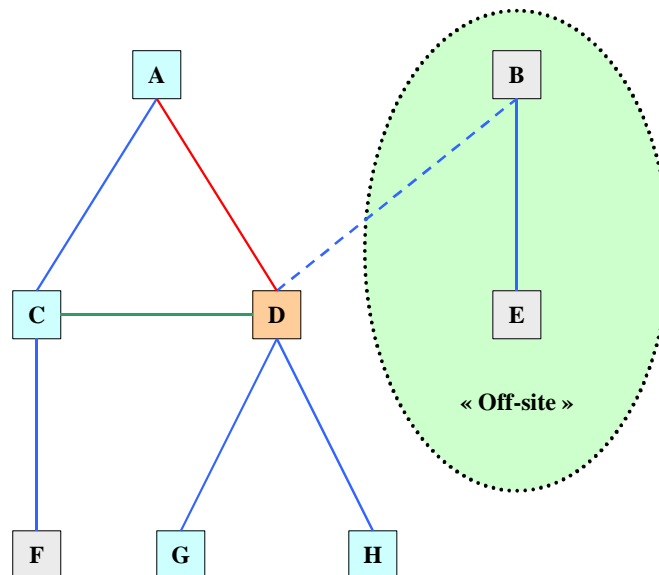
Cette fonctionnalité n'a finalement pas été retenue et la double délégation est donc interdite.

En effet elle mettait en évidence des problèmes de boucles difficiles à résoudre et qui aurait pu rendre le produit difficile à maintenir.

Le client a donc préféré ne pas retenir cette solution complexe qui n'aurait servi que pour des cas particuliers peu fréquents.

4.5.4. Exemple

Pour illustrer les contraintes que l'on vient d'énoncer, voici un exemple :



En bleu : arborescence officielle Motorola

En rouge : exception (permanente)

En vert : délégation (temporaire)

On s'intéresse ici au manager « D » : son manager « B » n'est pas sur le même site. « D » a donc choisi « A » comme responsable congés sur site.

Supposons maintenant que « D » parte lui-même en congés. Il peut désormais déléguer à « A », « C », « G » ou « H ». Il ne peut plus déléguer à « B » ni à « E » car le lien « B-D » est remplacé par le lien « A-D » dans la hiérarchie d'approbations de congés.

Dans cet exemple, « D » a choisi de déléguer à « C ».

4.6. Automatisations

En plus de l'interface Web, certains traitements doivent être automatisés par un programme lancé quotidiennement :

4.6.1. Rappels

Si un manager a déposé des congés d'une durée supérieure à une semaine, mais n'a pas délégué sa responsabilité de validation des congés, le système doit le prévenir par mail une semaine avant son départ.

Ceci a permis de démarrer l'application en donnant l'habitude aux managers de déléguer, et évite également les « oublis » qui peuvent être handicapants pour les subordonnés.

4.6.2. Echéances de délégation

Lorsqu'une délégation arrive à échéance, elle doit être supprimée.

Même si elle n'est plus prise en compte par le système, nous avons préféré la purger une fois qu'elle est terminée afin de ne pas surcharger les tables de vieilles délégations périmées.

4.6.3. Changements hiérarchiques

Lorsqu'un changement de la hiérarchie Motorola a lieu, toutes les exceptions liées doivent être annulées.

Un mail avertit l'employé et le responsable congés que l'exception a été supprimée. Par contre les délégations ne sont pas concernées par les changements hiérarchiques, elles expireront naturellement après la date fin prévue.

4.6.4. Construction de la nouvelle hiérarchie

A partir de la hiérarchie officielle Motorola, les exceptions sont appliquées, suivies des délégations.

Cette nouvelle hiérarchie est stockée en remplacement de celle utilisée par WEBCongés.

4.6.5. Mails managers « off-site »

Pour toutes les demandes de congés qui ont été validées dans la journée par un responsable congés local, le manager « off-site » est prévenu par mail de la période d'absence de l'employé si celui-ci a coché l'option correspondante lors de la définition de l'exception.

5. Planification

Après une phase d'intégration dans l'entreprise, la compréhension du sujet, et un délai nécessaire à estimer la charge de travail nécessaire, j'ai essayé de mettre au point un planning assez tôt dans le stage.

J'ai toutefois attendu que le RSD (Requirement Specification Document) soit rédigé pour ne pas m'engager sur un planning dont je ne mesurais pas la charge.

Le client souhaitait de plus que la délégation soit prête au plus tôt afin de gérer les demandes des managers pour les vacances de Février.

Ceci m'a amené à mettre en place un planning en deux phases pour répondre aux besoins les plus urgents en priorité, avec une contrainte de délai sur la première mise en production (déploiement) de l'application.

Le découpage du projet en deux phases était possible car la partie « délégation » et la partie « exceptions » sont assez indépendantes.

En complément de la planification, l'historique des activités présente plus précisément les activités concrètes que j'ai eu. Il fait référence à certaines tâches que j'ai accomplies parallèlement à mon projet et indique les réunions avec les différents intervenants de mon projet.

5.1. Planning

Mois	Novembre					Décembre					Janvier				Février				Mars				Avril			Mai									
Semaines	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33		
Intégration																																			
Cahier des charges																																			
Spécification																																			
Phase 1																																			
Phase 2																																			
Documentations																																			
Rapport de stage																																			

Les semaines hachurées ont été consacrées à d'autres projets.

Plus de détails sur le contenu de chaque phase sont donnés dans l'historique suivant.

5.2. Historique des activités

Tout au long de mon stage j'ai gardé une trace écrite des activités que j'ai eues (un historique) afin d'avoir une vision globale sur les tâches effectuées.

5.2.1. Intégration

La période d'intégration a consisté en une présentation de l'entreprise et une visite du site, puis une présentation du département.

Ensuite j'ai découvert l'environnement de travail et j'ai commencé à m'habituer aux outils manipulés (notamment un peu d'auto-formation à PL/SQL).

Le client a également présenté plus en détail le sujet.

5.2.2. Cahier des charges

C'est durant cette phase que les besoins ont été approfondis avec le client afin de rédiger le Requirement Specification Document, notamment grâce à une réunion générale.

Plusieurs versions du RSD ont dû être rédigées avant d'obtenir la validation de ce document.

C'est également à cette période que le projet a été découpé en priorités et en phases.

5.2.3. Spécification

Pendant la phase de spécification, j'ai été amené à rédiger un High Level Design (HLD) regroupant notamment le Modèle Conceptuel de Données (MCD) et le Modèle Logique de Données (MLD).

5.2.4. Phase 1

Cette phase regroupe le développement, les tests et l'implémentation de la première version de l'application (partie délégation uniquement).

Des réunions ont eu lieu au fur et à mesure du développement avec mon encadrant pour valider certains aspects de l'application (confirmation que le travail fait répondait aux attentes et discussions sur le sujet pour lever certaines ambiguïtés) et avec mes collègues (validations des standards Motorola, conseils sur la méthodologie de programmation, etc).

Beaucoup de temps a également été accordé à la validation et à l'intégration de cette première version.

5.2.5. Phase 2

Cette deuxième phase m'a permis de compléter le travail pour répondre aux attentes (partie exceptions), d'apporter les modifications nécessaires sur le développement fait lors de la première phase (intégration des exceptions dans la délégation) et d'améliorer le travail déjà fait (évolution du menu super-utilisateur).

La conclusion de cette phase a été la même que la précédente : tests de validations et intégration (mise en production).

5.2.6. Documentations

Durant cette période j'ai tout d'abord rencontré mon encadrant universitaire, Mr Marc Boyer, qui est venu sur le site de Motorola pour voir mon contexte de travail et rencontrer mon maître de stage et mes collègues.

J'ai pu en profiter pour faire un bilan de mon travail avec lui et un point sur mon avancement.

J'ai ensuite été amené à faire une présentation de mon travail à Horo Quartz, l'entreprise fournissant le logiciel WEBCongés, afin de leur présenter les besoins qui s'étaient fait ressentir autour de leur application et les solutions envisagées.

J'ai également profité de cette période pour finaliser certains documents nécessaires à la démarche de gestion de projet de Motorola :

- **Operation Guide** : c'est un guide de maintenance qui présente la description des tables et des fichiers de l'application, des informations sur le type et la périodicité des traitements, et le contexte d'exécution (serveurs, comptes utilisateurs).
- **User Guide** : c'est la document utilisateur. Dans le cadre de mon application, c'est en fait une aide en ligne qui a été rédigée car elle est plus adaptée à un site Web.

5.2.7. Rapport de stage

Enfin j'ai pu consacrer la dernière période de mon stage pour rédiger mon rapport de stage dans l'entreprise, tout en continuant à avoir des activités sur des petits projets annexes.

6. Environnement, outils et langages

6.1. Environnement

J'ai travaillé sur un PC ayant comme système d'exploitation Windows (NT)2000.

L'équipe HRIS travaille régulièrement sur deux serveurs utilisant également le système d'exploitation Windows (NT)2000 :

- Un serveur de test
- Un serveur de production

Ils partagent le même environnement matériel et logiciel, ce qui permet aux applications d'être déployées de la même façon sur les deux machines.

6.2. Serveur de test

C'est sur ce serveur que s'effectuent le développement et les tests de déploiement des nouveaux programmes.

Ce serveur n'est utilisé que par l'équipe HRIS, et permet de développer et tester les applications sans impacter sur le bon fonctionnement du serveur de production qui est en permanence utilisé par les employés de Motorola.

La majorité des tests (tests unitaires, tests d'acceptation par l'utilisateur, etc) est réalisée sur ce serveur.

6.3. Serveur de production

C'est sur ce serveur que les applications déployées sont visibles par les employés de Motorola.

Son fonctionnement est donc plus critique que le serveur de test.

Le passage de nouvelles applications de l'environnement de test vers l'environnement de production doit donc faire l'objet d'une demande formelle auprès du service informatique de Motorola qui est le seul habilité à effectivement déployer et modifier les programmes sur le serveur de production.

Ces demandes sont historisées, ce qui permet de garder une trace des modifications apportées et de retrouver la cause d'éventuels problèmes rencontrés.

Le passage de l'environnement de test vers l'environnement de production d'un programme est appelé « mise en production ». C'est en fait le déploiement effectif de l'application.

Cette duplication des serveurs est certes plus coûteuse en ressources, mais permet de développer plus efficacement les applications, en évitant tout impact sur les autres programmes.

Certains tests critiques (tests de la charge du serveur, etc) sont également plus faciles à mettre en œuvre dans un environnement qui n'est pas utilisé par les employés.

6.4. Outils

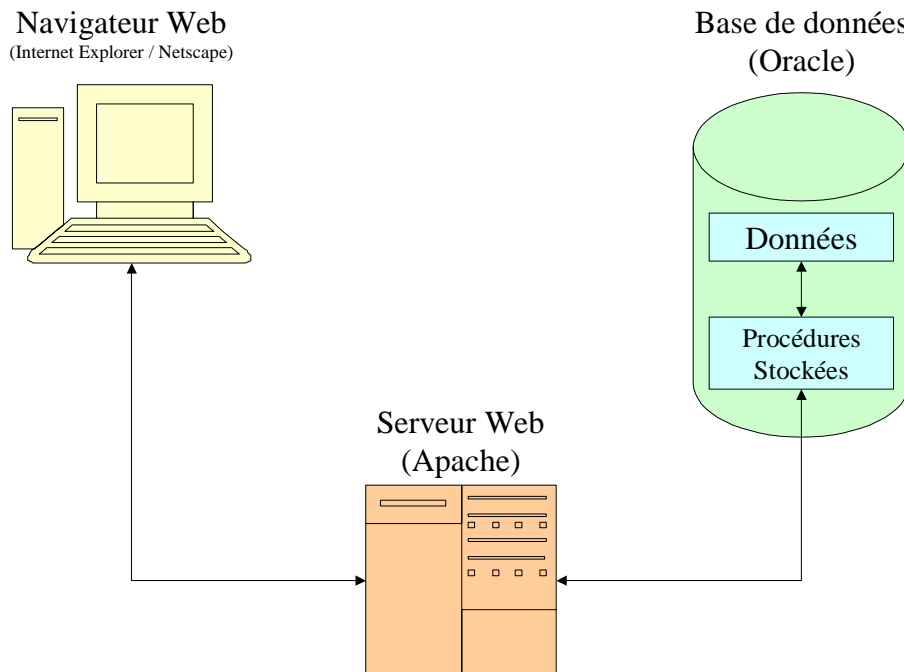
6.4.1. Oracle SQL*Plus 8.0

Cet outil de la suite Oracle permet de se connecter à un serveur de base de données Oracle. Il permet de saisir des commandes SQL ainsi que des commandes spécifiques à Oracle. Je l'ai utilisé pour créer, consulter et mettre à jour les bases de données et les packages PL/SQL de mon projet.

6.4.2. Oracle Web Application Server

Ce programme est également appelé Oracle Internet Application Server (Oracle iAS). C'est le serveur d'Oracle qui permet d'héberger des pages dynamiques programmées en PL/SQL pour les sites Web.

A Motorola, Oracle iAS est couplé au serveur Web Apache.



Architecture générale d'Oracle Application Server

6.4.3. Terminal Service Client

C'est l'outil de Microsoft pour l'administration distante de machines. Il m'a permis de travailler directement dans l'environnement des serveurs qui exécutaient mon application et de mettre en place les tâches planifiées sur ces machines.

6.4.4. WEBCongés

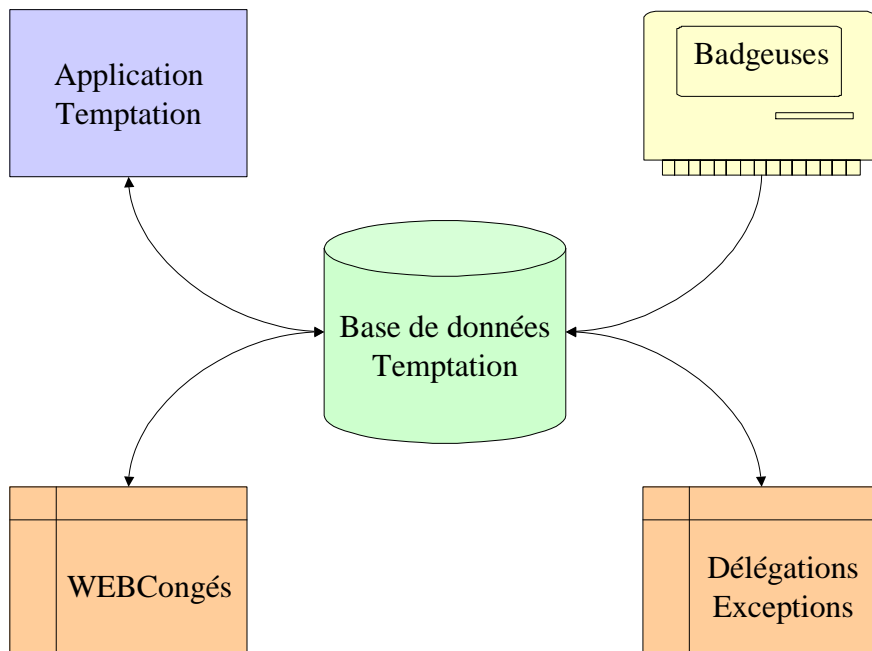
C'est l'application fournie par Horo Quartz autour de laquelle j'ai travaillé (cf présentation du sujet).

J'ai donc été amené à réaliser des opérations sur ce site pour tester ma propre application.

6.4.5. Temptation

Cet outil d'Horo Quartz est un système informatique de gestion des temps et des accès des employés. Il permet de stocker et de consulter les informations relatives au temps de travail. Le site WEBCongés vient en fait se greffer sur ce programme.

J'ai utilisé ce programme pour contrôler l'action de mon site et constater les nouvelles responsabilités de congés suites aux délégations et exceptions.



Applications basées sur la base Temptation

6.4.6. Divers

J'ai été également amené à utiliser les deux navigateurs standards à Motorola (Internet Explorer et Netscape) pour tester la compatibilité de mon site.

Certaines images du site ont été créées ou retouchées à l'aide de Adobe Photoshop.

Enfin la saisie du code source a été facilitée par l'utilisation d'un éditeur de texte avancé : UltraEdit-32.

6.5. Langages

L'environnement logiciel de Motorola, les standards utilisés et certaines contraintes nous ont amenés après réflexion à faire les choix techniques suivants :

6.5.1. SQL

Le langage SQL (Structured Query Language) est le standard pour les systèmes de gestion de base de données relationnelles.

J'ai utilisé ce langage dans l'outil SQL*Plus pour créer, interroger et alimenter les tables de mon application, aussi bien manuellement que dans les programmes.

6.5.2. PL/SQL

C'est le langage procédural proposée par Oracle. Il a une syntaxe proche du langage ADA avec les mêmes structures de programmation (boucles, tests, etc) et la notion de package.

Le code des procédures, des fonctions et des packages programmés est directement compilé et stocké par le serveur de base de données.

Le site Web de mon projet ainsi que les traitements associés sont programmés dans ce langage.

Notre choix s'est porté sur PL/SQL car il est largement utilisé par le département HRIS et le développement est très efficace, ce qui répondait aux contraintes de délai et de maintenance sur le projet.

En effet PL/SQL inclut plusieurs packages pour assister le programmeur dans la création d'un site Web dynamique. Les packages HTP et HTF permettent ainsi d'écrire des balises HTML, de créer des formulaires, de gérer la sécurité et les cookies, etc.

Le partage des données et l'exclusion mutuelle lors des modifications sont également assurés.

6.5.3. Perl

Perl est un langage de script interprété répandu et très souple.

Il est utilisé par le département HRIS pour combler les manques de PL/SQL dans la programmation système.

Il est ainsi utilisé dans ce projet pour lancer les tâches planifiées sur le serveur (lancement des procédures stockées en base) et pour envoyer les mails (ce que ne permet pas PL/SQL avec la configuration adoptée à Motorola).

7. Conception générale

Parmi les documents que j'ai produits pour Motorola dans le cadre de mon projet, le High Level Design (HLD) regroupe les informations concernant les modèles conceptuel et logique de données, le contenu souhaité pour les pages du site, le diagramme des flux de données, etc.

Ce chapitre résume les informations que l'on trouve dans ce document.

7.1. Présentation

L'application « Gestion des délégations et exceptions dans WEBCongés » est divisée en deux composants principaux : le site Web et le batch associé (tache planifiée).

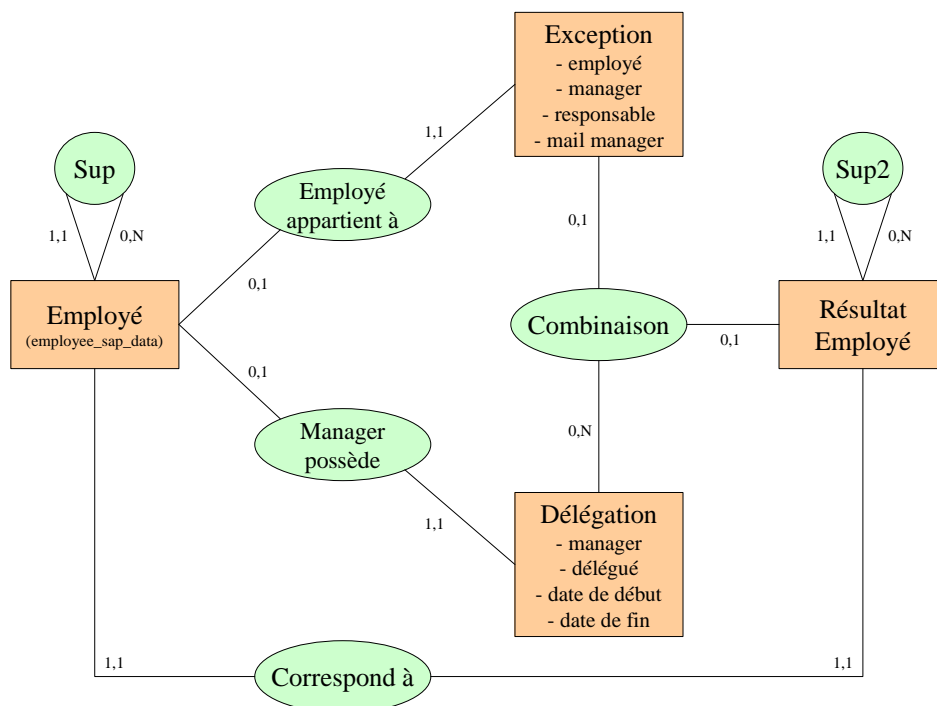
Le site Web permet la consultation et la saisie des informations.

Il offre ainsi toutes les fonctionnalités souhaitées que l'on a vu, en respectant le modèle de navigation retenu.

Il assure également la sécurité (en s'appuyant sur le serveur Web Apache et LDAP) et gère les super utilisateurs (à l'aide de cookies).

Le batch de son côté assure le traitement des données (construction des tables résultats) ainsi que toutes les automatisations et l'envoi des mails.

7.2. Modèle conceptuel de données



Dans ce modèle, on constate que la table originale des employés et la table de résultat se ressemblent et qu'il existe une correspondance entre elles : « sup » dans la première table désigne le manager hiérarchique tandis que « sup2 » dans la table finale désigne le responsable congés.

On remarque qu'un employé peut être concerné par une exception et/ou avoir une délégation en place.

L'association « combinaison » est en fait un traitement par batch qui permet de tenir compte à la fois des exceptions et des délégations pour calculer le résultat.

7.3. *Modèle logique de données*

En complément du Modèle conceptuel de données, le modèle logique de données suivant a été retenu.

7.3.1. *employee_sap_data*

Cette table existe déjà et n'est pas modifiable par l'application. Toutefois comme elle est importante nous rappelons ici sa structure.

Champs	Définition	Type	Clef
cid	Commerce ID	caractères	x
last_name	Last Name	caractères	
first_name	First Name	caractères	
userid	Userid	caractères	
company_cd	Company code (FR02,..)	caractères	
location_cd	Location code (ZFR11, ...)	caractères	
dept	Department code	caractères	
shift	Shift code	caractères	
egrade	Grade	caractères	
lgrade	Legal Grade	caractères	
hvflag	Time And Attendance(T&A) Flag	caractères	
contract_type	Contract type	caractères	
payroll_categ	Payroll category	caractères	
scid	Supervisor Commerce Id	caractères	
sup_last_name	Supervisor Last Name	caractères	
sup_first_name	Supervisor First Name	caractères	
sup_userid	Supervisor Userid	caractères	
sup_dept	Supervisor Department code	caractères	

Cette table contient la liste de tous les employés de Motorola en France et les informations qui leur sont associés (nom, prénom, lieu de travail, manager, etc).

Le « cid » est un numéro unique attribué à l'employé et sert souvent d'identifiant clef dans les tables.

Cette table est importée quotidiennement depuis le Data Warehouse (base de données centralisée aux Etats-Unis et alimentée par SAP) et sert de référence aux différents systèmes informatiques des ressources humaines.

Mon application se base donc naturellement sur cette table pour interroger la hiérarchie Motorola.

7.3.2. Exception

Champs	Définition	Type	Clef
employe	Identifiant de l'employé	caractères	x
manager	Sauvegarde de l'identifiant de l'ancien manager	caractères	
responsable	Identifiant du responsable congés	caractères	
mail	Contrôle si l'ancien manager est prévenu par mail	booléen	

Cette table contient la liste de toutes les exceptions à la hiérarchie pour la validation des congés.

Le « vrai manager » est sauvegardé le champ « manager » de la table afin de détecter les changements hiérarchiques par comparaison avec la hiérarchie officielle.

Le champ « mail » permet de spécifier si l'on souhaite que le « vrai manager » soit prévenu des congés validés par le responsable.

7.3.3. Délégation

Champs	Définition	Type	Clef
manager	Identifiant du manager	caractères	x
delegue	Identifiant du délégué	caractères	
date_debut	Date de début de délégation	date	
date_fin	Date de fin de délégation	date	

Cette table contient la liste de toutes les délégations en cours et à venir (les délégations passées sont purgées quotidiennement).

La période d'application de la délégation est bornée par les dates « date_debut » et « date_fin ».

7.3.4. Résultat_employé

Champs	Définition	Type	Clef
cid	Identifiant du manager	caractères	x
scid	Identifiant du responsable	caractères	

Cette table contient la hiérarchie finale pour la validation des congés.

Le champ scid ne désigne plus forcément le manager de l'employé mais son responsable pour la validation des congés.

7.4. Modèle physique de données

Le modèle physique de données est évidemment très proche du modèle logique que l'on vient de voir.

Toutefois, dans un souci d'optimisation et de simplification de l'application, la table « Résultat Employé » qui devait contenir la hiérarchie complète d'approbation des congés a été remplacée par une table différentielle « t_ext_deleg_manager ».

En effet malgré les délégations et les exceptions, la hiérarchie d'approbation des congés reste très proche de la hiérarchie officielle Motorola et un différentiel est donc moins complexe à construire.

De plus une interface d'import existait déjà pour la traduction de la hiérarchie Motorola en groupes WEBCongés avec prise en compte de la table « t_ext_deleg_manager » ce qui simplifie le déploiement et la validation de mon application.

Certaines tables annexes et temporaires ont également été rajoutées à l'application.

Ainsi la table « webcong_esd_exception » est une copie de la table « employee_sap_data » où le champs scid pointe sur le responsable congés permanent (soit son vrai manager, soit son responsable par exception).

Les délégations qui sont temporaires n'affectent pas cette table.

Cette table est utilisée partout dans l'application comme remplacement de « employee_sap_data » sauf pour les requêtes permettant de retrouver le vrai manager.

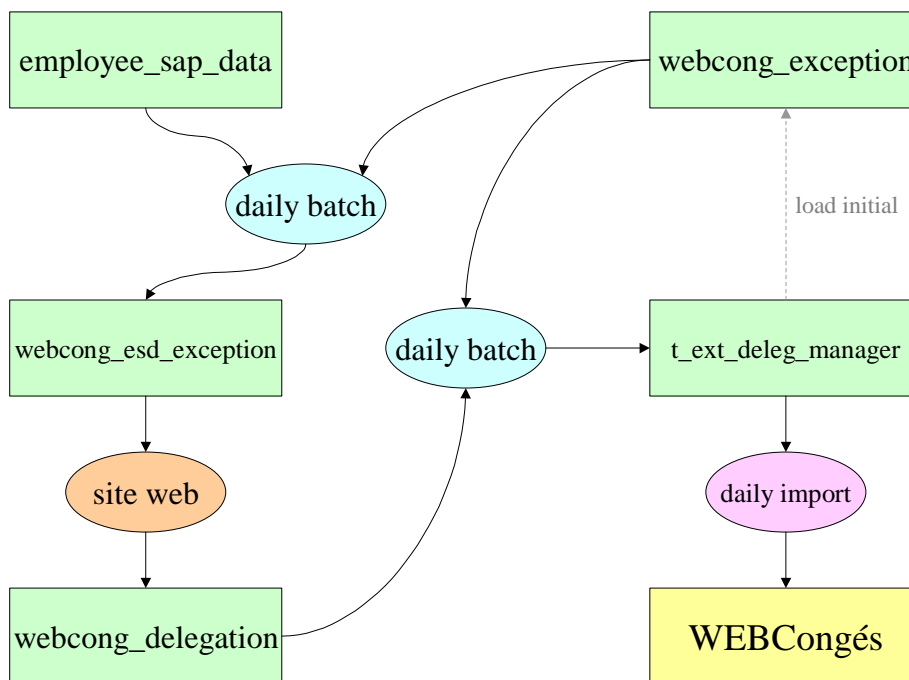
Cette substitution facilite beaucoup la programmation, la performance et la maintenance de l'application, surtout pour la partie délégation où il est nécessaire de tenir compte des exceptions, ce qui peut engendrer des calculs complexes.

Une autre table annexe permet de gérer la liste des super utilisateurs.

7.5. Diagramme des flux

Comme l'application fait appel à deux composants et s'appuie sur de nombreuses tables, il était nécessaire de construire un diagramme des flux afin de matérialiser le chemin que parcourent les données dans le programme.

Le schéma suivant est une vue simplifiée des différentes sources d'alimentation des tables, et est donné ici à titre d'illustration :



Les rectangles verts sont des tables SQL.

Les deux batchs en bleu sont en fait deux traitements différents du même batch.

On constate que la structure des tables au niveau physique diffère du modèle logique pour des raisons de simplification.

Plus globalement et pour se rapprocher du modèle conceptuel produit au début du projet, on constate que la source principale est la table « employee_sap_data » et que la destination est le système « WEBCongés ».

L'application agit donc comme un filtre entre ses deux systèmes en s'appuyant sur des tables internes pour gérer la délégation et les exceptions.

7.6. Mailings

Afin de prévenir les employés des différentes opérations qui les concernent, l'application envoie des mails spécifiques.

La quantité et la variété (21 types de mails différents) nécessitaient également de synthétiser ces informations dans un tableau.

Il illustre les nombreux cas qui se présentent dans l'application :

Mails selon motif	Délégation	"In-site"	"Off-site"
Création	mail délégué (create)	mail employé et responsable (create)	mail manager et responsable (create)
Suppression	mail délégué (delete)	mail employé et responsable (delete)	mail manager et responsable (delete)
Modification (significative)	mail délégué (update) ou mail ancien délégué et nouveau délégué (delete / create)	mail employé (update) et responsables (delete / create)	mail manager (update) et responsables (delete / create)
Expiration	mail délégué (expire) et éventuellement mail manager si congés à valider	n/a	
Départ employé (terminated)	rien (on attend expiration)	rien (simplement suppression)	
Départ délégué / responsable	rien (on attend expiration)	mail manager (remind)	mail employé (remind)
Changement manager	rien (on attend expiration)	mail employé et responsable si nouveau manager est "in-site"	
Approche de congés	mail si manager (remind)	n/a	

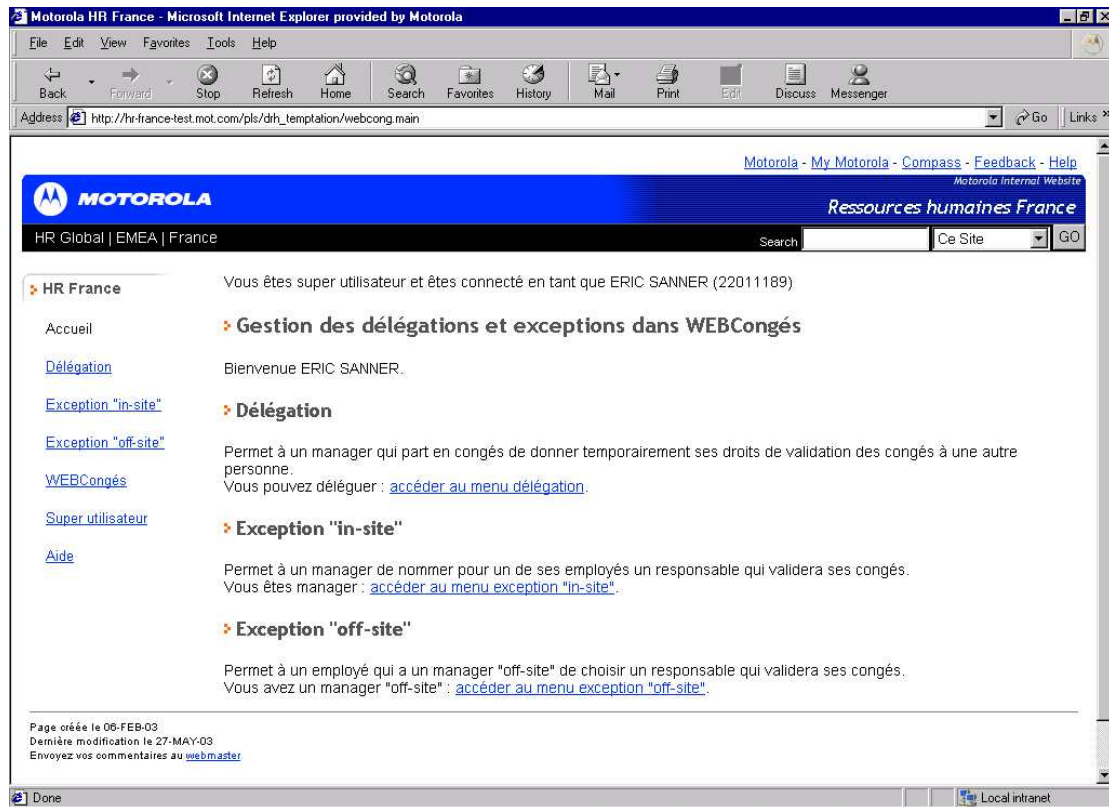
Ce tableau résume les destinataires et l'intitulé des mails selon le motif et le type de l'opération.

Les colonnes donnent la catégorie (délégation, exception « in-site » ou exception « off-site »).

Les lignes indiquent l'opération ou le motif (création, suppression, modification, etc). L'intersection de la ligne et de la colonne indique les personnes qui sont prévenues par mail et une indication sur le type de notification. Ces indications (create, delete, update, etc) reprennent les codes internes utilisés par l'application.

8. Réalisation

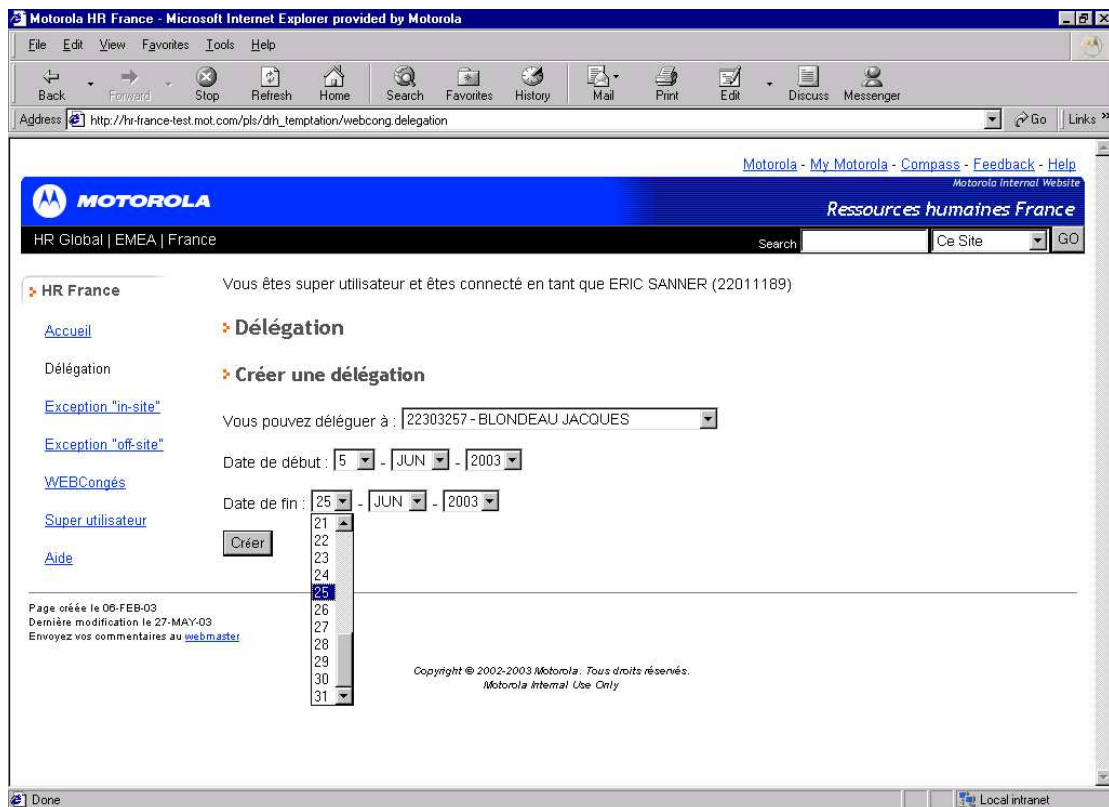
Accueil du site : une fois l'authentification passée, c'est l'écran qui s'affiche à l'utilisateur.



Le menu de gauche est ici complet. Dans le cas général il s'adapte à l'utilisateur qui ne voit donc que les opérations auxquelles il a accès.

En plus des fonctionnalités, ce menu contient un lien vers l'application WEBCongés et un lien vers l'aide en ligne.

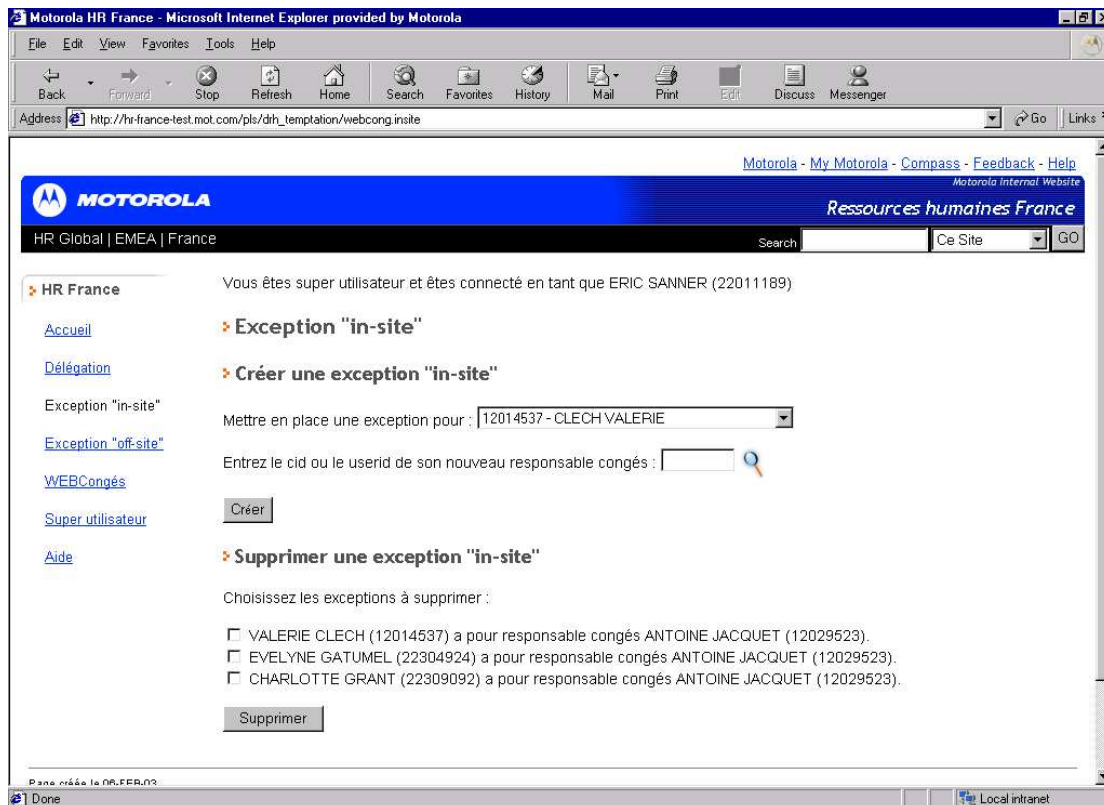
Menu de délégation : ce menu n'est donc accessible qu'aux personnes qui ont une responsabilité de validation des congés (managers).



Une liste déroulante permet de sélectionner l'employé à qui l'on souhaite déléguer parmi les personnes autorisées.

Si une délégation existe déjà, il est possible de la supprimer ou de la modifier (le formulaire est déjà rempli avec les valeurs précédemment saisies).

Menu d'exception « in-site » : ce menu n'est accessible qu'aux managers. Il leur permet de désigner pour un de leur subordonnés la personne qui validera les congés de cet employé.



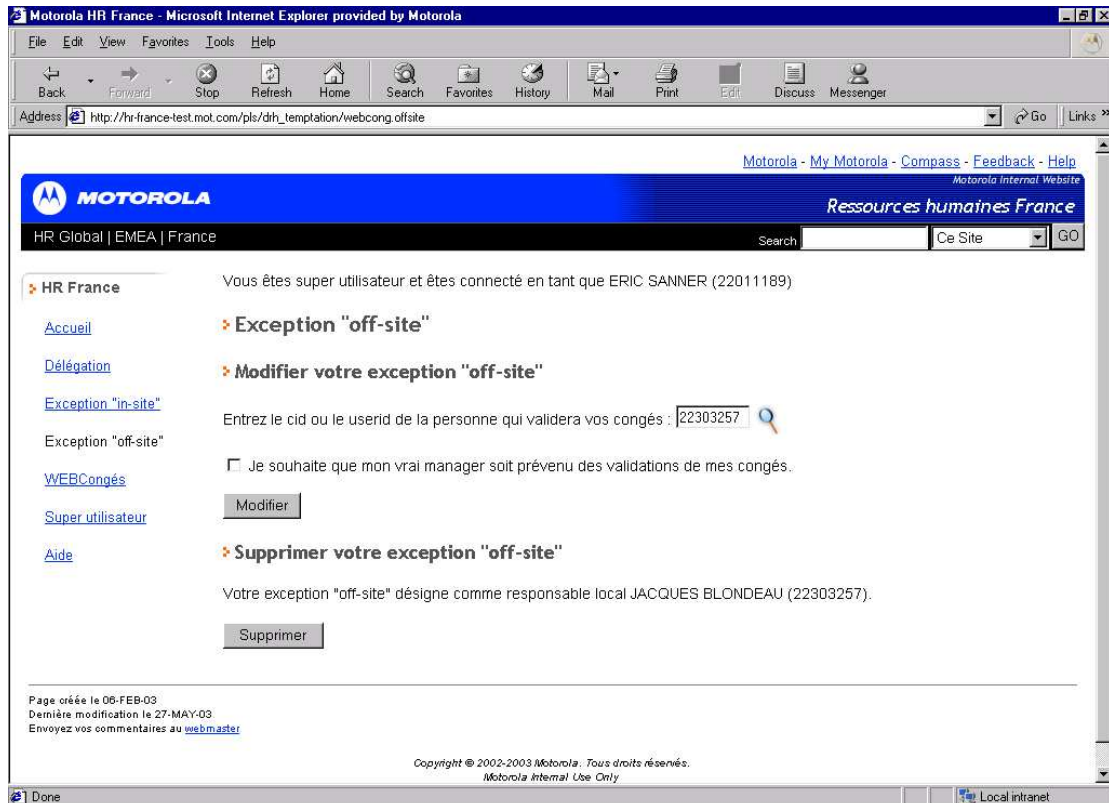
La liste déroulante permet de sélectionner le subordonné pour lequel on souhaite mettre en place une exception.

Comme il n'y a pas de contraintes pour le choix du responsable, on laisse à l'utilisateur le soin de saisir le numéro de la personne qu'il veut désigner.

La petite loupe permet de lancer une recherche sur l'annuaire Motorola pour retrouver facilement le numéro de l'employé.

Il y a déjà 3 exceptions en place pour ce manager. En cochant les cases il peut facilement en supprimer une ou plusieurs.

Menu d'exception « off-site » : ce menu n'est accessible qu'aux employés qui ont un manager « off-site » (à l'étranger ou sur un autre site en France). Il leur permet de choisir eux-mêmes leur responsable congés.

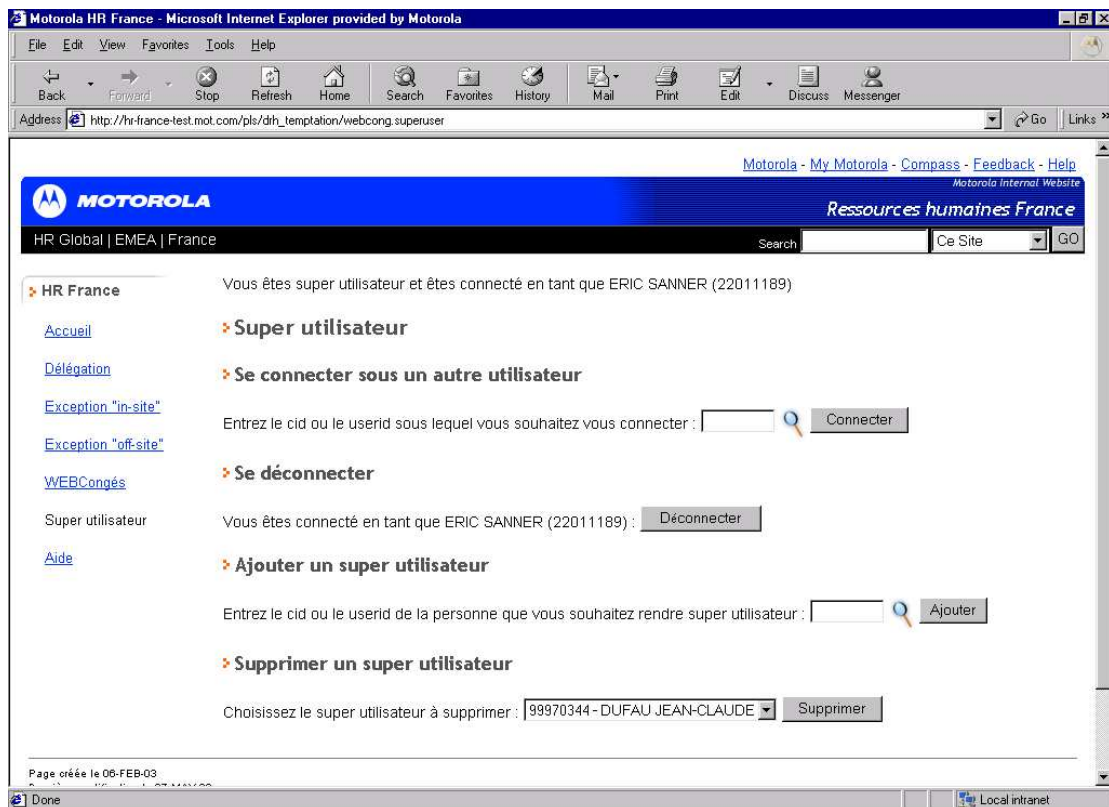


De la même façon que pour une exception « in-site », l'employé entre directement le numéro du responsable qu'il souhaite voir valider ses congés.

Il peut de plus cocher une case afin que son manager soit prévenu par mail des congés validés par le responsable.

Enfin un autre bouton lui permet de supprimer l'exception en cours s'il y en a une.

Menu super utilisateur : ce menu n'est accessible qu'aux super utilisateurs de l'application. Il leur permet de gérer certains cas particuliers.

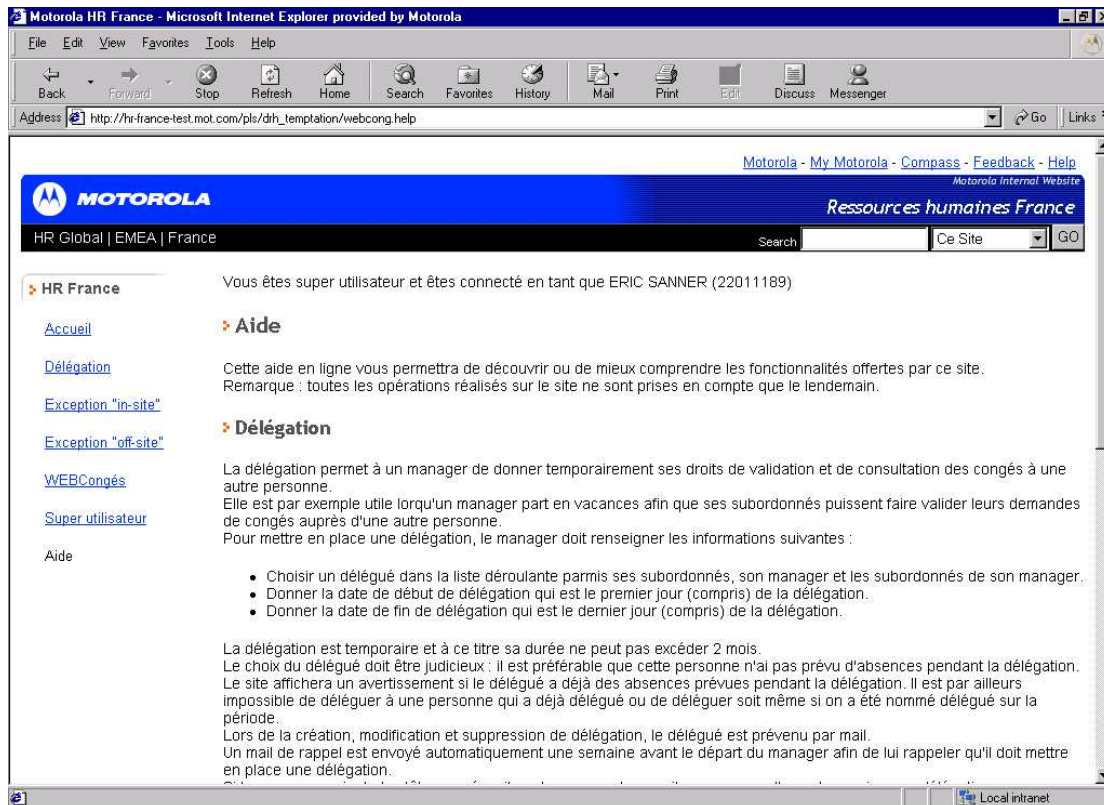


Le super utilisateur peut se connecter sous l'identité de n'importe quel autre employé.

Il peut également se déconnecter pour retrouver sa propre identité.

Enfin, il peut modifier la liste des super utilisateurs (ajouter ou supprimer).

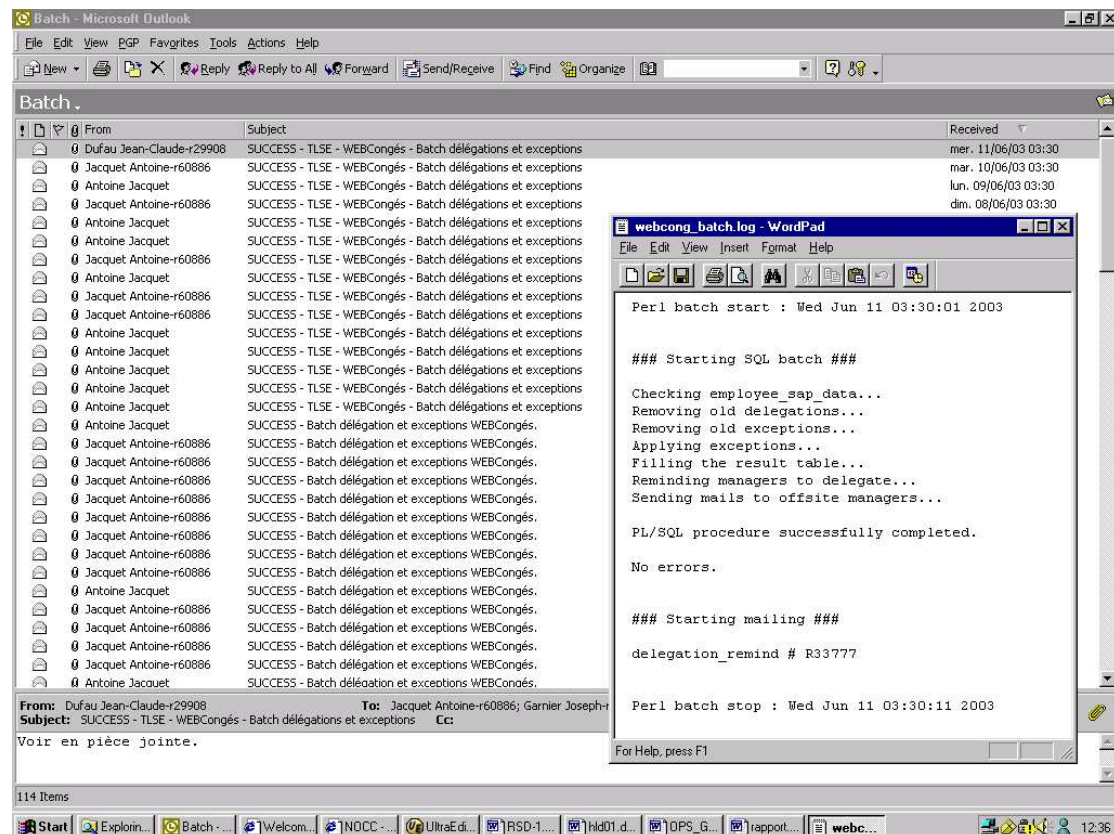
Aide en ligne : cette page est accessible à tous les utilisateurs et leur explique comment utiliser le site.



Cette aide explique les notions de délégation, exception « in-site », exception « off-site » et indique comment mettre en place de telles règles.

Comme cette aide est suffisamment détaillée, elle fait également office de documentation utilisateur (l'aide en ligne est plus adaptée pour un site Web).

Résultats du batch : afin de se tenir informé des résultats du batch, un mail est envoyé aux responsables de l'application.



Le batch est programmé toutes les nuits. L'heure précise de son exécution (3h30) est imposée par les dépendances avec les autres batchs et permet de limiter la charge du serveur.

Cette capture d'écran de mon logiciel de messagerie montre les résultats des derniers batchs.

Un fichier de log en pièce jointe (deuxième fenêtre sur la capture) donne le détail des actions et permet d'analyser le problème en cas d'échec.

9. Tests

Tout au long de mon stage, les tests ont été une condition nécessaire avant la mise en production (déploiement) de l'application.

J'ai ainsi systématiquement testé le programme au fur et à mesure du développement (tests unitaires).

Pour bien vérifier que le produit répondait aux attentes du client (tests d'acceptation par l'utilisateur), des collaborateurs ont participé afin qu'ils ne soient pas influencés par le développement.

Avant d'installer l'application en production il était également nécessaire de réaliser des tests (tests d'intégration).

J'ai donc eu à rédiger les documents suivants à deux reprises pendant mon stage, pour les deux phases de développement :

9.1. *Tracability Matrix*

C'est une table qui permet de tracer la validation des besoins par les tests afin de s'assurer de la complétude des tests.

Pour chaque « Requirement » énoncé dans le RSD (Requirement Specification Document), on effectue un ou plusieurs tests permettant de le valider.

On rapporte ensuite le résultat dans la table.

Le fait d'avoir numéroté les « Requirements » dans le RSD prend ici tout son sens car cette table devient plus aisée à rédiger.

9.2. *Plan de test*

Le plan de test énonce pour chaque « Requirement » le ou les tests qu'il convient d'effectuer.

Certains « Requirements » peuvent ne pas nécessiter de tests explicites (la simple existence de l'application répond à la contrainte).

Par contre certains « Requirements » nécessitent plusieurs tests afin de les valider. En effet plutôt que de formuler un test compliqué, on préfère l'éclater en plusieurs petits tests afin de faciliter le travail de la personne qui valide.

9.3. *Jeu de test*

Le jeu de test est à mettre en parallèle du plan de test. Il indique pour chaque test le jeu de valeurs à utiliser et la démarche détaillée à suivre pour effectuer le test.

Il permet à plusieurs testeurs de travailler simultanément en écartant les risques d'interférences entre leurs tests respectifs.

10. Conclusion

Ce stage d'une durée totale de huit mois m'a beaucoup apporté sur les plans professionnel et personnel.

J'ai consolidé mes connaissances dans certains langages (SQL, Perl, HTML), appris de nouvelles technologies (site Web dynamique en PL/SQL) et appliqué d'autres méthodes (démarche de gestion de projet, planification, qualité SEI Level 3).

L'utilisation des outils associés à ses technologies (Oracle) et d'outils spécifiques aux ressources humaines (SAP, Temptation) a également été enrichissante, notamment la compréhension des interfaces nécessaires à la collaboration de ces applications hétérogènes.

La migration du département vers ACS m'a également permis de vivre une opération de rachat dans un contexte où les grands groupes font de plus en plus appel au partenariat pour les opérations informatiques.

De plus, la prise en compte de mon travail et le déploiement de mon application ont été des étapes très encourageantes dans mon stage. Le site est effectivement utilisé et compte 170 exceptions actives et une soixantaine de délégations créées à ce jour.

Mon implication, la réussite de la mise en place de mon projet, ainsi que d'autres travaux annexes réalisés pour Motorola pendant mon stage, me permettent aujourd'hui de continuer cette expérience chez ACS cet été, preuve de leur confiance.

Enfin, dans cette expérience humaine enrichissante, je tiens à souligner l'esprit d'équipe et le professionnalisme de l'ensemble du groupe dans lequel j'ai su m'intégrer.

Annexe 1

Exemple de code PL/SQL (batch.sql)

```
-- This program is the daily batch for the application "Gestion des délégations et exceptions dans WEBCongés"
-- http://hr-france.mot.com/pls/drh_temptation/webcong.main
--
-- A. Jacquet 21 Jan 2003   Initial Version
-- A. Jacquet 18 Feb 2003   Mailing correction for delegation reminding
-- A. Jacquet 05 Mar 2003   End date correction for delegation
-- A. Jacquet 10 Mar 2003   Purge exception was added to remove terminated exceptions
-- A. Jacquet 26 May 2003   Offsite notification request updated in order to work at any time
```

```
-- log messages
SET SERVEROUTPUT ON
```

```
-- *****
-- * Declaration of the program *
-- *****
DECLARE
```

```
-- write mails to send in mailing log
PROCEDURE mail(line IN VARCHAR) IS
    mailing_file UTL_FILE.FILE_TYPE;
BEGIN
    BEGIN
        mailing_file:=UTL_FILE.FOPEN('f:\var\opt\applications\temptation','webcong_mailing.lst','a');
        UTL_FILE.PUT_LINE(mailing_file,line);
        UTL_FILE.FCLOSE(mailing_file);
    EXCEPTION
        WHEN OTHERS THEN
            NULL;
    END;
END mail;
```

```
-- checks the table "employee_sap_data" is not empty
FUNCTION check_employee_sap_data RETURN BOOLEAN IS
    counter NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO counter
    FROM employee_sap_data;
    RETURN counter>0;
END check_employee_sap_data;
```

```
-- removes the old delegations
PROCEDURE purge_delegation IS
    CURSOR old_delegation IS
        SELECT d.userid AS delegate_userid,m.userid AS manager_userid,m.first_name AS manager_first_name,m.last_name
        FROM webcong_delegation,employee_sap_data m,employee_sap_data d
        WHERE manager_cid=m.cid
        AND delegated_cid=d.cid
        AND SYSDATE>=date_end+22/24;
    counter NUMBER;
BEGIN
    -- check for old delegations
    FOR old_delegation_rec IN old_delegation LOOP
        DBMS_OUTPUT.PUT_LINE(' removing delegation for manager '||old_delegation_rec.manager_cid||' because it is 1
        mail('delegation_expire_delegate # '||old_delegation_rec.delegate_userid||' # '||old_delegation_rec.manager_
        -- if there are holidays to validate then we also mail the manager
        SELECT COUNT(*)
        INTO counter
        FROM hopabsw,webcong_esd_exception
        WHERE visa='2'
        AND matri=cid
        AND scid=old_delegation_rec.manager_cid;
        IF counter>0
        THEN
            mail('delegation_expire_manager # '||old_delegation_rec.manager_userid||' # '||counter);
        END IF;
    END LOOP;

    -- remove all old delegations
    DELETE FROM webcong_delegation
    WHERE SYSDATE>=date_end+22/24;
    COMMIT;
END purge_delegation;
```

```
-- removes the old exceptions
PROCEDURE purge_exception IS
    CURSOR check_employee IS
        SELECT *
        FROM webcong_exception
        WHERE employee_cid NOT IN (SELECT cid FROM employee_sap_data);
```

```

CURSOR check_responsible IS
SELECT *
FROM webcong_exception,employee_sap_data
WHERE responsible_cid NOT IN (SELECT cid FROM employee_sap_data)
AND cid=employee_cid;
CURSOR check_manager IS
SELECT e.userid AS employee_userid,e.first_name AS employee_first_name,e.last_name AS employee_last_name,e.c
FROM webcong_exception,employee_sap_data e,employee_sap_data r
WHERE employee_cid=e.cid
AND manager_cid!=e.scid
AND responsible_cid=r.cid
AND e.scid IN (SELECT cid FROM employee_sap_data WHERE company_cd=e.company_cd);
counter NUMBER;
BEGIN
-- check for employee terminated
FOR check_employee_rec IN check_employee LOOP
DBMS_OUTPUT.PUT_LINE(' removing exception for employee '||check_employee_rec.employee_cid||' because this e
DELETE FROM webcong_exception
WHERE employee_cid=check_employee_rec.employee_cid;
COMMIT;
END LOOP;

-- check for responsible terminated
FOR check_responsible_rec IN check_responsible LOOP
DBMS_OUTPUT.PUT_LINE(' removing exception for employee '||check_responsible_rec.employee_cid||' because his
-- check if the user has an off-site manager
SELECT COUNT(*)
INTO counter
FROM employee_sap_data e,employee_sap_data m
WHERE e.cid=check_responsible_rec.employee_cid
AND m.cid=check_responsible_rec.manager_cid
AND e.company_cd=m.company_cd;
IF counter=0
THEN
-- offsite manager : remind the employee to set a new exception
mail('offsite_remind # '||check_responsible_rec.userid);
ELSE
-- insite manager : remind the manager to set a new exception
mail('insite_remind # '||check_responsible_rec.sup_userid||' # '||check_responsible_rec.first_name||' '||c
END IF;
-- now remove it
DELETE FROM webcong_exception
WHERE employee_cid=check_responsible_rec.employee_cid;
COMMIT;
END LOOP;

-- check for hierarchy changes
FOR check_manager_rec IN check_manager LOOP
DBMS_OUTPUT.PUT_LINE(' removing exception for employee '||check_manager_rec.employee_cid||' because he has
mail('exception_expire_employee # '||check_manager_rec.employee_userid||' # '||check_manager_rec.responsible
mail('exception_expire_responsible # '||check_manager_rec.responsible_userid||' # '||check_manager_rec.emplc
DELETE FROM webcong_exception
WHERE employee_cid=check_manager_rec.employee_cid;
COMMIT;
END LOOP;
END purge_exception;

-- build the employee hierarchy using employee_sap_data and webcong_exception
-- in the table webcong_esd_exception (this is a temporary solution)
PROCEDURE apply_exception IS
CURSOR responsible IS
SELECT *
FROM webcong_exception,employee_sap_data
WHERE cid=responsible_cid;
BEGIN
-- purge the old employee hierarchy
DELETE FROM webcong_esd_exception;

-- fill with the standart hierarchy
INSERT INTO webcong_esd_exception
SELECT *
FROM employee_sap_data;

-- update informations with the exceptions
FOR responsible_rec IN responsible LOOP
UPDATE webcong_esd_exception
SET scid=responsible_rec.responsible_cid,
sup_last_name=responsible_rec.last_name,
sup_first_name=responsible_rec.first_name,
sup_userid=responsible_rec.userid,
sup_dept=responsible_rec.dept
WHERE cid=responsible_rec.employee_cid;
END LOOP;

COMMIT;

```

```
END apply_exception;
```

```
-- fill the table t_ext_deleg_manager using delegation and exception  
-- currently it only supports simple delegations and exceptions  
-- this procedure is safe and will "rollback" in case of error
```

```
PROCEDURE fill_t_ext_deleg_manager IS
```

```
BEGIN
```

```
  COMMIT;
```

```
  BEGIN
```

```
    -- purge the old table
```

```
    DELETE FROM webcong.t_ext_deleg_manager;
```

```
    -- fill the new table with current delegations
```

```
    INSERT INTO webcong.t_ext_deleg_manager
```

```
      SELECT e.cid,delegated_cid,d.last_name,d.first_name
```

```
      FROM webcong_delegation,webcong_esd_exception e,employee_sap_data d
```

```
      WHERE e.scid=manager_cid
```

```
      AND d.cid=delegated_cid
```

```
      AND e.cid!=delegated_cid
```

```
      AND SYSDATE BETWEEN date_begin-2/24 AND date_end+22/24;
```

```
    -- fill the new table with exceptions BUT DON'T overwrite because the delegation is always good
```

```
    INSERT INTO webcong.t_ext_deleg_manager
```

```
      SELECT employee_cid,responsible_cid,last_name,first_name
```

```
      FROM webcong_exception,employee_sap_data
```

```
      WHERE cid=responsible_cid
```

```
      AND employee_cid NOT IN (SELECT commerce_id FROM webcong.t_ext_deleg_manager);
```

```
  EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
      DBMS_OUTPUT.PUT_LINE(' !!! ERROR filling the result table : previous table was kept !!!');
```

```
      ROLLBACK;
```

```
      RETURN;
```

```
  END;
```

```
  COMMIT;
```

```
END fill_t_ext_deleg_manager;
```

```
-- reminds by email the manager who must delegate but have not
```

```
PROCEDURE remind_delegation IS
```

```
  CURSOR remind IS
```

```
    SELECT matri,userid
```

```
    FROM hophabs,employee_sap_data
```

```
    WHERE matri=cid
```

```
    -- look for the next week
```

```
    AND dat BETWEEN SYSDATE+7 AND SYSDATE+14
```

```
    AND motif IN (SELECT motif FROM webcong_motif_significatif)
```

```
    AND matri IN (SELECT scid FROM webcong_esd_exception)
```

```
    AND matri NOT IN (SELECT manager_cid FROM webcong_delegation)
```

```
    -- did we already remind him this/last week ?
```

```
    AND matri NOT IN (SELECT cid FROM webcong_delegation_reminded WHERE dat BETWEEN SYSDATE-7 AND SYSDATE)
```

```
    GROUP BY matri,userid
```

```
    -- if he has 5 days of more in the next week then he is not present
```

```
    HAVING count(*)>=5;
```

```
  BEGIN
```

```
    FOR remind_rec IN remind LOOP
```

```
      mail('delegation_remind # '||remind_rec.userid);
```

```
      INSERT INTO webcong_delegation_reminded
```

```
        VALUES (remind_rec.matri,SYSDATE);
```

```
    END LOOP;
```

```
  COMMIT;
```

```
END remind_delegation;
```

```
-- this procedure sends mail to offsite manager for employees who have asked for this feature
```

```
PROCEDURE notify_offsite IS
```

```
  CURSOR notify IS
```

```
    SELECT sup_userid,last_name,first_name,cid,a.datdeb,a.datfin
```

```
    FROM hopabsw a,hopvisa v,employee_sap_data,webcong_exception
```

```
    WHERE a.matri=v.matri
```

```
    AND a.datdeb=v.datdeb
```

```
    AND a.ordre=v.ordre
```

```
    AND a.motif IN (SELECT * FROM webcong_motif_significatif)
```

```
    AND a.visa=3
```

```
    AND a.annul=0
```

```
    AND a.matri=employee_cid
```

```
    AND a.matri=cid
```

```
    AND v.datevisa=TRUNC(SYSDATE)-1
```

```
    AND mail_flag='Y';
```

```
  BEGIN
```

```
    FOR notify_rec IN notify LOOP
```

```
      mail('offsite_notify # '||notify_rec.sup_userid||' # '||notify_rec.first_name||' '||notify_rec.last_name||'
```

```
    END LOOP;
```

```
END notify_offsite;
```

-- Main program

BEGIN

DBMS_OUTPUT.ENABLE;

DBMS_OUTPUT.PUT_LINE('Checking employee_sap_data...');

IF NOT check_employee_sap_data()

THEN

DBMS_OUTPUT.PUT_LINE(' !!! ERROR employee_sap_data is empty : batch aborted !!!');

RETURN;

END IF;

DBMS_OUTPUT.PUT_LINE('Removing old delegations...');

purge_delegation();

DBMS_OUTPUT.PUT_LINE('Removing old exceptions...');

purge_exception();

DBMS_OUTPUT.PUT_LINE('Applying exceptions...');

apply_exception();

DBMS_OUTPUT.PUT_LINE('Filling the result table...');

fill_t_ext_deleg_manager();

DBMS_OUTPUT.PUT_LINE('Reminding managers to delegate...');

remind_delegation();

DBMS_OUTPUT.PUT_LINE('Sending mails to offsite managers...');

notify_offsite();

END;

/

SHOW ERROR

Annexe 2

Exemple de code Perl (batch.pl)

```

#/perl/bin/perl -w

# This script launches the SQL batch and sends emails
# for the application "Gestion des délégations et exceptions dans WEBCongés"
# http://hr-france.mot.com/pls/drh_temptation/webcong.main
#
# A. Jacquet    13 Jan 2003           Initial Version
# A. Jacquet    25 Feb 2003           Corrections in mail text of
delegation remind (wrong URL and formulation)
# A. Jacquet    11 Mar 2003           Changed SMTP server from
'msgtls2.sps.mot.com' to 'smtp.mot.com'
# A. Jacquet    12 Mar 2003           Update to handle mails needed for
exceptions (insite/offsite)
#
Mail title for the log is now
different if there are errors
# A. Jacquet    21 May 2003           Changed mail title for the log to be
standard

# packages
use File::Basename;
use Mail::Sendmail 0.75;
use MIME::QuotedPrint;
use MIME::Base64;

# global variables
$path_log='f:\var\opt\applications\temptation' ;
$path_sql='f:\opt\applications\WebRh\OraWeb\Webcong_Deleg_Excep\SQL' ;

# *****
# * Procedure which send an email with the following parameters : *
# * from, to, subject, text, optional file1, optional file2, ... *
# *****
sub mail
{
    my ($from,$to,$subject,$text,@files)=@_;

    %mail=(
        SMTP => 'smtp.mot.com',
        from => $from,
        to => $to,
        # support "french" charset in subjects
        subject =>
'=?iso-8859-1?Q?' . encode_qp($subject) . '?=',
    );
    $boundary="====".time()."====";
    $mail{'Content-Type'}="multipart/mixed; boundary=\"\$boundary\"";

    # text message
    $mail{body}="--$boundary\n";
    $mail{body}.="Content-Type: text/plain; charset=\"iso-8859-1\"\n";
    $mail{body}.="Content-Transfer-Encoding: quoted-printable\n";
    $mail{body}.=" \n".encode_qp($text)." \n";

    # files attachments
    foreach $file (@files)
    {
        $mail{body}.="--$boundary\n";
        $mail{body}.="Content-Type: application/octet-stream;
name=\"\".basename($file)."\" \n";
        $mail{body}.="Content-Transfer-Encoding: base64\n";
        # basename do avoid full path in attached filename
        $mail{body}.="Content-Disposition: attachment;
filename=\"\".basename($file)."\" \n";
        open(F,$file) or die "Cannot read $file: $!";
        binmode F; undef $/;
    }
}

```

```

        $mail{body}.="\\n".encode_base64(<F>)."\\n";
        close F;
    }

    $mail{body}="--$boundary--";
    sendmail(%mail) || print "Error: $Mail::Sendmail::error\\n";
}

# *****
# * Procedure which start a sql program on tlsptpt *
# * the program name is the parameter *
# * the log_file is optional and/or can be included in the command *
# *****
sub sql_process
{
    my ($prog_file,$log_file)=@_;
    # initialization of the variable command with the sqlplus command
    $command= "| plus80 -s webtation/*****\\@tlsptpt";
    if ($log_file) { $command.=">>$log_file"; }
    # run the command
    open SQLPLUS,$command;
    # execute the sql program
    print SQLPLUS "\\@$prog_file\\n";
    # close the command
    close SQLPLUS;
}

# *****
# * Remove all the log files *
# *****
sub delete_logs
{
    unlink($path_log.'\\webcong_mailing.lst');
    unlink($path_log.'\\webcong_batch.log');
}

# *****
# * Read lines in the mailing file and sends the mails *
# *****
sub mailing
{
    if (!open(MAILING_FILE,$path_log.'\\webcong_mailing.lst'))
    {
        print LOG_FILE "No mails to send.\\n";
        return;
    }
    while ($line = <MAILING_FILE>)
    {
        print LOG_FILE $line;
        chomp($line);
        @fields=split(/ # /,$line);

        $from='R56797@motorola.com';
        #$to='R60886@motorola.com';
        $to="$fields[1]\\@motorola.com";
        $subject='';

        # no "switch" statement available in Perl 5.0 !
        if ($fields[0] eq 'delegation_create')
        {
            $subject='WEBCongés : Création délégation';
            $message="$fields[2] vous a nommé délégué pour la
validation des congés de ses subordonnés du $fields[3] au
$fields[4].\\nWEBCongés : http://hr-france.mot.com/webconges";

```

```

    }
    elsif ($fields[0] eq 'delegation_update')
    {
        $subject='WEBCongés : Mise à jour délégation';
        $message="$fields[2] à mis à jour la période de
délégation du $fields[3] au $fields[4].\nWEBCongés :
http://hr-france.mot.com/webconges" ;
    }
    elsif ($fields[0] eq 'delegation_delete')
    {
        $subject='WEBCongés : Suppression délégation';
        $message="$fields[2] à supprimé sa délégation.\nVous
n'êtes donc plus son délégué." ;
    }

    # ... résumé pour le rapport de stage

    elsif ($fields[0] eq 'offsite_notify')
    {
        $subject='WEBCongés : Notification de congés';
        $message="$fields[2] sera absent du $fields[3] au
$fields[4].\n\n";
        $message.=" $fields[2] will be on holidays from
$fields[3] to $fields[4]." ;
    }
    elsif ($fields[0] eq 'exception_expire_employee')
    {
        $subject='WEBCongés : Expiration exception';
        $message="Suite à un changement hiérarchique,
$fields[2] n'est plus responsable de vos congés." ;
    }
    elsif ($fields[0] eq 'exception_expire_responsible')
    {
        $subject='WEBCongés : Expiration exception';
        $message="Suite à un changement hiérarchique, vous
n'êtes plus responsable des congés de $fields[2]." ;
    }

    if (!$subject)
    {
        print LOG_FILE "          !!! ERROR : unknown mail
type !!!\n";
    }
    elsif (!$fields[1])
    {
        print LOG_FILE "          !!! WARNING : no userid for
recipient !!!\n";
    }
    else
    {
        mail($from,$to,$subject,$message);
    }
}
close(MAILING_FILE);
}

#
*****
*****
# * Parse a log file and returns 'SUCCESS', 'WARNING' or 'FAILURE' for the
mail title *
#
*****
*****
sub find_errors
{

```

```

$result='SUCCESS';
my ($parse_file)=@_;
open(PARSE_FILE,$parse_file) or die "Cannot open file : $!";
while ($line = <PARSE_FILE>)
{
    # "error(s)" and "failure(s)" are failures, but "No errors."
is a good message from SQL
    if ($line =~ /error/i && $line !~ /^No errors./ || $line =~
/failure/i)
    {
        $result='FAILURE';
        # this is the worst case so we can quit the loop
        last;
    }
    # "warning(s)"
    if ($line =~ /warning/i)
    {
        $result='WARNING';
    }
}
close(PARSE_FILE);
return $result;
}

# *****
# * Main program *
# *****

open(LOG_FILE,">$path_log.\webcong_batch.log') or die "Cannot open file :
$!";
print LOG_FILE "Perl batch start : ".localtime()."\n\n\n";

print LOG_FILE "### Starting SQL batch ###\n\n";
close LOG_FILE;
sql_process($path_sql.\batch.sql',$path_log.\webcong_batch.log');
open(LOG_FILE,">>$path_log.\webcong_batch.log') or die "Cannot open file :
$!";
print LOG_FILE "\n\n";

print LOG_FILE "### Starting mailing ###\n\n";
mailing();
print LOG_FILE "\n\n";

print LOG_FILE "Perl batch stop : ".localtime()."\n";
close(LOG_FILE);

$subject=find_errors($path_log.\webcong_batch.log');
$subject.=' - TLSE - WEBCongés - Batch délégations et exceptions';

mail('R29908@motorola.com','R60886@motorola.com,R29665@motorola.com,R35131@m
otorola.com,R29908@motorola.com,R16092@motorola.com',$subject,'Voir en pièce
jointe.', $path_log.\webcong_batch.log');
delete_logs();

```

Annexe 3

Extrait du plan de test

REQ Description et tests

Résultat

Commentaires

1 Compatibilité du site avec les navigateurs utilisés.

Controler l'affichage et la navigation du site sous Internet Explorer.

Controler l'affichage et la navigation du site sous Netscape (Windows / Unix).

2 Temps d' exécution du batch d' automatisation inférieur à une demi-heure.

Mettre en place un nombre significatif (50) de délégations et controler la durée du batch.

3 Identité des utilisateurs garantie par l' authentification.

N/A (Naturellement vérifié par la nature de l' authentification).

4 Les supers utilisateurs peuvent réaliser toutes les opérations.

Vérifier que l' on peut mettre en place une délégation ou une exception pour quelqu' un d' autre.

5 On ne peut pas déléguer à une personne qui a déjà délégué sur une période en chevauchement.

Créer une délégation pour l' utilisateur Y nommant le délégué Z ; puis créer une délégation pour l' utilisateur X nommant le délégué Y sur une période en chevauchement (tester les différents cas). Le système doit refuser.

6 On ne peut pas valider ses propres congés suite à une délégation.

Mettre en place une délégation pour le manager X nommant un de ses subordonnés Y. Vérifier que Y n' est pas responsable de ses propres congés.

7 Le système empêche les délégations multiples.

Créer une délégation pour l' utilisateur X nommant le délégué Y ; puis créer une délégation pour l' utilisateur Y nommant Z sur une période en chevauchement (tester les différents cas). Le système doit refuser.

8 Le site Web correspond au standard Motorola.

Vérifier visuellement que le standart est respecté avec le reste de l' intranet, par exemple en cliquant sur un lien du bandeau pour comparer la présentation avec une page existante.

9 Intégration du site dans l' espace Intranet des ressources humaines.

Vérifier que la navigation WEBCongés <---> Site Délégation / Exception est facile.

10 Lors du choix de l' opération, l' utilisateur ne voit que les opérations auxquelles il a accès.

Se loguer sous différents utilisateurs et vérifier que l' on ne voit que les menus nous concernant parmiés délégation, exception "in-site" et exception "off-site".

11 Un super utilisateur peut modifier la liste des supers utilisateurs.

Ajouter un super utilisateur à partir de son CID ou de son UserID. Vérifier avec son compte qu' il est bien devenu super utilisateur.

Supprimer un super utilisateur dans la liste. Vérifier avec son compte qu' il n' est plus super utilisateur.

12 Lors de l' ajout, modification, ou suppression de délégation, le délégué est prévenu par mail.

Sur une délégation, effectuer les 3 opérations (ajout, modification puis suppression). Vérifier que suite au lancement automatique du batch qui suit les mails sont bien envoyés.

13 Un manager peut créer ou modifier une délégation.

Créer une délégation et vérifier sa prise en compte dans WEBCongés après le délai nécessaire au passage du batch.

Modifier la délégation et vérifier la prise en compte de la modification dans WEBCongés.

14 Un manager peut supprimer une délégation.

Supprimer une délégation et vérifier que WEBCongés est également mis à jour.

15 Lors de la création ou suppression d' une exception "in-site", le responsable congés et l' employé sont prévenus par mail.

Sur une exception, effectuer les 3 opérations (ajout, modification puis suppression).
Vérifier que suite au lancement automatique du batch qui suit les mails sont bien envoyés.

16 Un manager peut créer une exception "in-site" pour un de ses subordonnés.

Créer une exception "in-site" et vérifier sa prise en compte dans WEBCongés après le délai nécessaire au passage du batch.

Vérifier que l' exception se répercute aussi sur la liste des employés visibles pour la délégation : le manager ne doit plus voir l' employé concerné par l' exception et le nouveau responsable doit le voir.

17 Un manager peut supprimer une exception "in-site" pour un des ses subordonnés.

Supprimer une exception "in-site" et vérifier que WEBCongés est également mis à jour.

Vérifier que la liste des employés visibles pour la délégation redevient l' originale pour le manager et le responsable.

18 Lors de la création ou suppression d' une exception "off-site", le responsable congés et le manager "off-site" sont prévenus par mail.

Sur une exception, effectuer les 3 opérations (ajout, modification puis suppression).
Vérifier que suite au lancement automatique du batch qui suit les mails sont bien envoyés.