



2009

Rapport de stage 2009



**CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE**

Développement d'une application
collaborative de gestion des données
scientifiques

Langage J2EE

Tuteur : Volker Baecker

Chevron Nicolas

Montpellier RIO Imaging

29/09/2009

Sommaire

1	Remerciements	4
2	Introduction	5
3	Présentation de l'Entreprise	9
3.1	Montpellier RIO Imaging	9
3.2	Le CNRS	10
3.3	Implantation Géographique du MRI	12
3.4	Organisation du MRI	13
3.5	Réalisations informatiques	14
3.6	Positionnement du MRI	16
3.7	Moyens technique mis à disposition	19
4	Analyse de la mission	19
4.1	Analyse du projet	19
4.2	Objectifs de la mission	22
5	Méthodologie utilisée	25
5.1	Méthodologie de travail	25
5.2	Méthode de gestion de projet	26
5.3	Répartitions des tâches	30
6	Réalisations	34
6.1	Outils Technologiques utilisés	34
6.1.1	Java Entreprise Edition	34
6.1.2	Enterprise JavaBeans (EJB)	34
6.1.3	Java Persistence API (JPA)	36
6.1.4	Web-services	36
6.1.5	JSF	37
6.1.6	Richface	37
6.2	Outils utilisés	38
6.2.1	Le Serveur Glassfish Serveur v2.1	38

6.2.2	MySQL.....	38
6.2.3	Eclipse ganymède avec l'édition JAVA EE et les plugins	38
7	Synthèse de la solution apportée	40
7.1	Le système de fichier virtuel	40
7.2	La gestion des dossiers	41
7.3	Effacement des dossiers	43
7.4	Renommer un fichier ou un dossier.....	44
7.5	Gestion de la copie et du déplacement d'un fichier ou d'un dossier.....	45
7.6	Gestion des fichiers de téléchargement du client	45
7.7	Visualisation des aperçus.....	46
8	Conclusion	48
9	Bibliographie	50
10	Glossaire	50
11	Annexes.....	51
11.1	Annexe 1 : Découpage en tâche d'un scénario.....	51
11.2	Annexe 2 : Diagramme de classe du système de fichier virtuel	53

1 Remerciements

Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance aux personnes suivantes, pour l'expérience enrichissante et pleine d'intérêt qu'elles m'ont fait vivre durant ces cinq mois au sein de la Plate-forme Régionale d'imagerie dite au sein du CNRS :

- Monsieur Pierre Travo, responsable de la plate-forme pour son accueil au sein de son équipe et la confiance qu'il m'a accordé.
- Madame Corine Tran-Aupiais, Responsable MRI - MCD (Mesure Contrôle et Développement), pour son accueil et la confiance qu'elle m'a accordée dès mon arrivée au sein de l'équipe MRI.
- Monsieur Volker Bäcker, Responsable du stage et responsable MRI - R&D -Software, pour m'avoir intégré rapidement au sein de MRI et pour m'avoir accordé toute sa confiance. Ainsi que pour le temps qu'il m'a consacré tout au long de cette période, sachant répondre à toutes mes interrogations.
- Monsieur Olivier MIQUEL, Administrateur Réseau&Système MRI, pour son conseil sympathique et son coopération professionnelle.
- Monsieur Yin Wei, stagiaire à MRI - R&D - Software pour sa coopération au long de ces trois mois.

Je tiens aussi à remercier les membres du jury participant à l'évaluation du mon stage.

2 Introduction

Dans le cadre de ma formation en école d'ingénieur en informatique, il m'a été demandé en deuxième année, d'effectuer un stage en entreprise d'une durée de 3 mois (du 1 juin au 31 août 2009). Lors de ma recherche de stage, j'ai privilégié certains critères qui me semblaient prépondérants pour le choix de l'entreprise. En effet, je voulais que l'entreprise dans laquelle j'allais effectuer mon stage soit une grosse structure afin que le travail proposé me permette d'appréhender une technologie inconnue et qui me paraissait primordiale dans la recherche de stage de troisième année. Un de mes professeurs de deuxième année a proposé un stage en J2EE au sein du CNRS. J'ai postulé et après un entretien au CNRS une réponse positive m'était retournée. Ce stage me paraissait correspondre à l'ensemble de mes attentes : il se déroulait au sein du CNRS qui est une importante structure de recherches et de plus la technologie utilisée était celle dans laquelle je voulais acquérir des compétences. C'est donc le groupe Montpellier RIO Imaging qui m'a accueilli.

Le Centre national de la recherche scientifique est un organisme public de recherche (Établissement public à caractère scientifique et technologique placé sous la tutelle du Ministère de l'enseignement supérieur et de la Recherche). Il est le principal organisme de recherches à caractère pluridisciplinaire en France, il mène des recherches dans l'ensemble des domaines scientifiques comme la chimie, la physique, les sciences et technologies de l'information ou de la communication.



Montpellier RIO Imaging est une plate-forme de soutien des programmes de recherches des Sciences de la Vie et de la Santé qui fait appel à l'imagerie microscopique dans tous ses aspects. Sa mission est de proposer, définir et mettre en œuvre les moyens nécessaires, ainsi que les actions qui s'y rattachent. Elle permet aux chercheurs et scientifiques d'effectuer des acquisitions sur microscope et ensuite de traiter leurs images sur ordinateur.



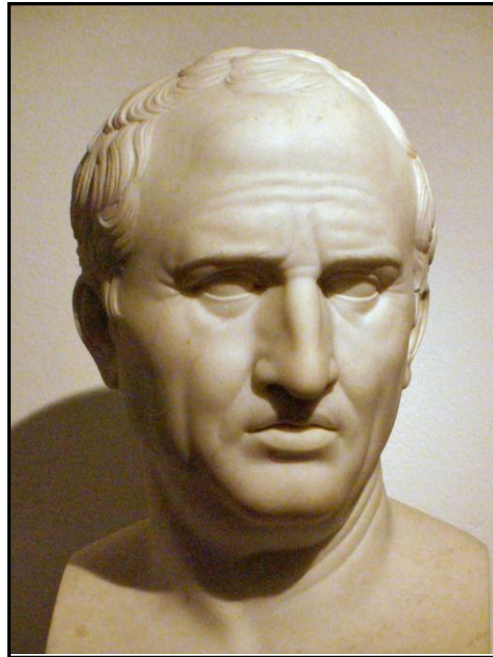
L'acquisition de ces images est un processus maîtrisé par toute l'équipe du MRI, le problème majeur de ce centre étant le partage et la gestion de toutes ces données. Par exemple lorsqu'un utilisateur effectue une acquisition sur un microscope, les images qui en découlent sont stockées sur ce dernier. Pour pouvoir les transporter et les traiter, il est obligé de les transférer sur un disque amovible ou sur le réseau. De plus pour effectuer les traitements sur ces images, il doit se déplacer dans les salles d'analyse du MRI. Plusieurs salles dans différents plateaux techniques sont disponibles, ces plateaux sont situés dans des bâtiments différents séparés pour certains par plusieurs kilomètres. Cependant chaque salle propose ses types de traitements et tous les traitements ne sont pas disponibles sur tous les plateaux.

Pour répondre à ce problème, l'équipe du MRI a élaboré un projet portant le nom de Cicéro. Le stage consistait à participer à ce projet qui était déjà commencé lorsque j'ai accepté la mission. Il consiste donc à permettre aux utilisateurs de la plate-forme de partager, de gérer et de traiter simplement leurs données à partir de n'importe quel poste informatique de leur choix.

Pour cela, ils doivent néanmoins répondre à plusieurs critères :

- L'ordinateur doit avoir une connexion internet.
- L'utilisateur doit avoir accès via un login et un mot de passe au système globale du MRI (site web).

Logo du projet Cicéro



Dans un premier temps, lorsqu'un utilisateur se sert d'un équipement, il faut que les données d'acquisition soient transférées automatiquement vers un serveur central. Ensuite, il faut permettre à l'utilisateur d'accéder simplement à ses données. Celui-ci doit pouvoir accéder depuis n'importe quel poste aux différentes données qui lui appartiennent ou sur lesquels il possède des droits. Il doit donc lui aussi pouvoir autoriser l'accès à ses données à différents utilisateurs de son choix sur la plate forme. Il doit pouvoir gérer ses données :

- Créer des dossiers pour organiser et ranger ses fichiers.
- Copier des fichiers ou les déplacer d'un dossier à un autre.
- Supprimer les dossiers et les fichiers de son choix.
- Partager des fichiers ou des dossiers avec les utilisateurs choisis.

Ensuite dans une troisième étape le système doit permettre aux utilisateurs d'effectuer les traitements nécessaires sur ses fichiers. Il doit pouvoir utiliser les différentes applications de traitements d'images (déjà existantes sur les postes

de travail de la plate-forme du MRI) à partir de n'importe quel plateau technique de son choix.

On peut donc diviser le tout en trois sous parties:

- Une partie acquisition des données.
- Une partie gestion des données.
- Une partie traitement des données.

Ma mission au cours du stage est de m'intégrer à l'équipe de développement et de participer à l'élaboration de l'application. Concrètement : la première partie est déjà commencée, il faut la finir et la tester, la deuxième partie elle n'est pas commencée et il faut créer toute la conception, le développement et les tests de cette partie. En ce qui concerne la troisième partie, elle ne rentre pas dans le cadre de ma mission car ce n'est pas une priorité (il faut d'abord que les deux parties précédentes fonctionnent) puis mon stage ne durant que trois mois, il est impossible de tout réaliser. Le développement au cours du stage est fait en langage J2EE qui est une extension du langage java. Il correspond aux besoins des applications d'entreprise et il facilite la création d'applications réparties. De plus la méthodologie de gestion de projet utilisée durant mon stage était extrême programming.

Mon stage s'est donc déroulé au sein du MRI dans les locaux du CNRS. Le MRI est sous la tutelle du CNRS, c'est pourquoi j'étais stagiaire du CNRS et non du MRI. Chaque personne travaillant au sein du MRI est en réalité employée par une des tutelles de la plate-forme : mon tuteur par exemple est employé par l'INSERM. A mon arrivée au sein du MRI, je n'avais aucune compétence en J2EE et je n'avais jamais eu l'occasion de travailler au sein d'une équipe utilisant extrême programming. Bien sûr j'avais déjà eu des cours de java durant ma formation à l'EPSI mais je n'avais jamais créé d'application en J2EE. Concernant Extrême programming, je ne connaissais que partiellement cette méthodologie sans réellement connaître les concepts qui doivent être appliqués. C'est pourquoi, une partie de mon stage m'a servi à appréhender ces nouveaux concepts avec l'aide de mon tuteur et du stagiaire qui travaillait avec moi.

Ce document a pour but de présenter le travail effectué durant ce stage. Cet exposé des travaux se fera dans un ordre logique : dans un premier temps, je présenterai les structures de mes organismes d'accueils et plus précisément ce qu'est le MRI et son fonctionnement. Ensuite j'exposerai avec plus de détails la problématique de mon stage et j'expliquerai comment le projet Cicéro répondra aux différents besoins. De plus, je présenterai la méthodologie de gestion du projet et au final, je dévoilerai la solution apportée au projet.

3 Présentation de l'Entreprise

3.1 Montpellier RIO Imaging

MRI est la Plate-Forme Régionale d'imagerie du Languedoc-Roussillon. C'est une Plate-Forme Technologique , labellisée ibisa(Infrastructures pour la Biologie, la Santé et l'agronomie. Elle a pour rôle de permettre la coordination des plates-formes de recherches en Sciences du Vivant.).

MRI signifie Montpellier RIO Imaging : Montpellier puisque c'est la plate-forme de Montpellier et de sa région, RIO qui signifie Réunion Inter Opération a été remplacé par le nom d'ibisa et Imaging car MRI est une plate-forme d'imagerie.

La plate-forme du MRI permet à des équipes de recherches d'utiliser du matériel mis à leur disposition sur différents plateaux techniques. Plusieurs types de matériels sont présents sur chaque site. En fait lorsqu'une équipe de recherche a besoin d'un équipement (microscope ou d'un autre type), elle fait une demande auprès d'un organisme comme le CNRS ou le CIRAD pour obtenir des fonds permettant cette achat. Ce dernier peut accepter la demande tout en exigeant que le matériel soit placé sous la responsabilité du MRI. L'avantage de placer un appareil sous la responsabilité du MRI, c'est qu'il peut être utilisé par plusieurs équipes, donc là ou autrefois il fallait acheter le même équipement pour chaque équipe, il suffit d'en acheter un et de le partager entre les équipes, grâce au MRI.

L'utilisation des équipements est ouverte au secteur privé et au secteur public sans aucune condition thématique, institutionnelle ou géographique. De la même manière, l'accès aux ressources de la plate-forme MRI ne rend pas obligatoire

l'établissement d'une collaboration, sous quelque forme que ce soit et encore moins par la CO-signature obligatoire des articles ou brevets. Enfin, il n'y aucune condition de connaissance, savoir et savoir faire préalables en imagerie.

Néanmoins pour pouvoir avoir accès au matériel certains critères sont à respecter:

- Il faut tout d'abord avoir un compte sur la plate-forme MRI.
- Pour pouvoir utiliser certains matériels un accompagnement ou une formation peut être nécessaire.
- Il faut avoir payé l'accès aux ressources demandées.
- Il faut bien-entendu avoir réservé le matériel.

Le MRI est une plate-forme multi-organisme. Différents organismes permettent au MRI de pouvoir subsister, elle se situe notamment dans les locaux de l'un d'eux : le CNRS.

3.2 Le CNRS

Le Centre national de la recherche scientifique est un organisme public de recherche (Établissement public à caractère scientifique et technologique, placé sous la tutelle du Ministère de l'enseignement supérieur et de la Recherche). Il produit du savoir et met ce savoir au service de la société. Avec plus de 32 000 personnes (dont 26 000 statutaires - 11600 chercheurs et 14400 ingénieurs, techniciens et administratifs), un budget 2009 de 3,367 milliards d'euros dont 607 millions d'euros de ressources propres, une implantation sur l'ensemble du territoire national, le CNRS exerce son activité dans tous les champs de la connaissance, en s'appuyant sur plus de 1200 unités de recherche et de service.

Il est le principal organisme de recherche à caractère pluridisciplinaire en France. Il mène des recherches dans l'ensemble des domaines scientifiques, technologiques et sociétaux. Il couvre la totalité du domaine scientifique, qu'il s'agisse des mathématiques, de la physique, des sciences et technologies de l'information et de la communication, de la physique nucléaire et des hautes

énergies, des sciences de la planète et de l'Univers, de la chimie, des sciences du vivant, des sciences humaines et sociales, des sciences de l'environnement ou des sciences de l'ingénierie. Le CNRS est présent dans toutes les disciplines majeures regroupées au sein de neuf instituts :

- Institut de chimie (INC)
- Institut de l'écologie et de l'environnement (INEE)
- Institut de physique (INP)
- Institut national de physique nucléaire et de physique des particules (IN2P3)
- Institut des sciences biologiques (INSB)
- Institut des sciences humaines et sociales (INSHS)
- Institut des sciences mathématique et de leurs interactions (INSMI)
- Institut des sciences et technologies de l'information et de l'ingénierie (INST2I)
- Institut national des sciences de l'univers (INSU)

Le CNRS développe de façon privilégiée des collaborations entre les spécialistes de différentes disciplines, et tout particulièrement avec les universités, ouvrant ainsi de nouveaux champs d'investigations qui permettent de répondre aux besoins de l'économie et de la société.

Implantation national du CNRS :

Le CNRS est présent sur tout le territoire Français. Il est regroupés en dix-neuf délégations répartie dans différentes régions de France assurant une gestion directe et locale des laboratoires et entretenant les liens avec les partenaires locaux et les collectivités territoriales.



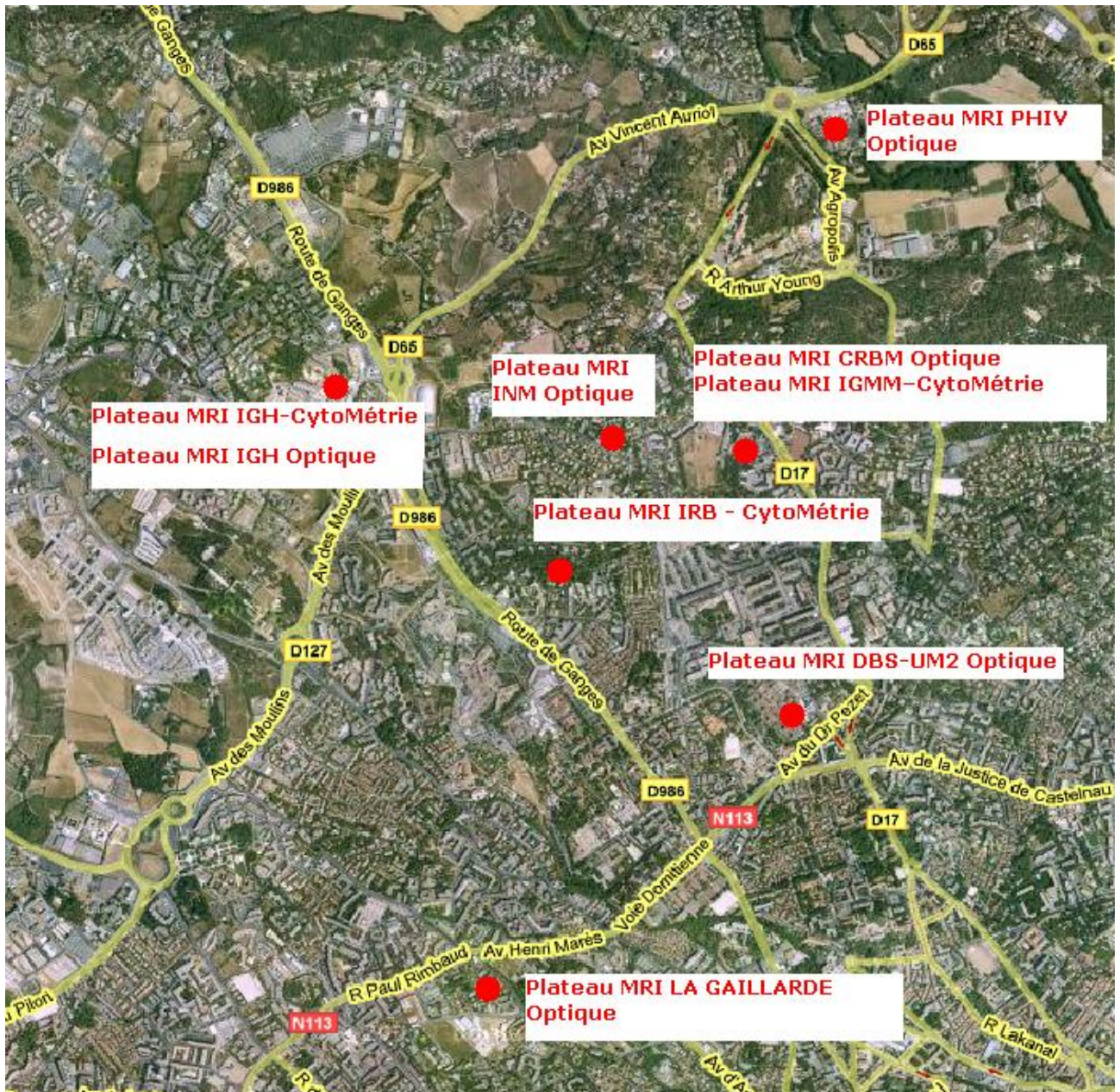
CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

3.3 Implantation Géographique du MRI



Le Montpellier RIO Imaging est implanté comme son nom l'indique dans la ville de Montpellier. Le centre principale du MRI se trouve dans le campus du CNRS, situé en face du zoo de Lunaret au 1919 route de Mende 34293 Montpellier. Néanmoins plusieurs plateaux techniques appartiennent au MRI:

- Plateau MRI CRBM Optique situé au coeur du campus du CNRS.
- Plateau MRI DBS-UM2 Optique situé dans les locaux de l'université de Montpellier II.
- Plateau MRI IGH – CytoMétrie, Plateau MRI IGH Optique situé tous les deux à l' Institut de Génétique Humaine.
- Plateau MRI IGMM – CytoMétrie situé à l'Institut de Génétique Moléculaire de Montpellier.
- Plateau MRI INM Optique situé à l'Institut des Neurosciences de Montpellier.
- Plateau MRI IRB – CytoMétrie situé à l'Institut de Recherche en Biothérapie.
- Plateau MRI LA GAILLARDE Optique situé au campus SUPAGRO.
- Plateau MRI PHIV Optique situé sur le campus du CIRAD.

Implantation géographique des plateaux techniques du MRI à Montpellier :

3.4 Organisation du MRI

Le MRI est divisé en plusieurs départements qui ont chacun leur rayon d'action. Son périmètre de compétence initiale la biologie moléculaire et cellulaire vient récemment de s'élargir. A l'heure actuelle la structure compte deux départements définis par leur périmètre de compétence : la microscopie (MRI-Optique) et la Cytométrie (MRI-Cytométrie). Ces deux départements constituent

le socle de base de l'activité du MRI. A cela, il faut ajouter deux départements dits transversaux par leur mission:

- Le département en charge de la formation (MRI-Formation) qui s'occupe de l'organisation et de la coordination des opérations de formations et notamment celles organisées en partenariat avec les différents Services de formation des organismes publiques ou privés.
- Le département ressources et développement (MRI-R&D) dans lequel mon stage s'est déroulé qui a en charge les opérations de développement et de recherche. Il vient en soutien aux différents plateaux techniques pour le développement ou l'évolution de leurs moyens techniques.

3.5 Réalisations informatiques

Les différentes réalisations informatiques:

HRM:

Huygens Remote Manager (HRM) est un outil en ligne de déconvolution à destination principale des plate-formes multi-utilisateurs d'imagerie et de microscopie de la recherche académique ou industrielle. La déconvolution est une opération de restauration d'une image par un traitement numérique. HRM a initialement été créé à Montpellier RIO Imaging et est maintenant un projet commun du MRI, du Friedrich Miescher Institute (Bâle), de la BioImaging and optics platform, de l'école Polytechnique et fédérale de Lausanne et de la société Scientific Volume Imaging. HRM est composé de deux éléments écrits en PHP (Langage de programmation dont le but est de permettre la création de pages web dynamiques) : une interface Huygens en ligne et un gestionnaire de queue. Brièvement, l'interface en ligne permet à l'utilisateur de créer des jobs (planification, demande de traitement) de déconvolution depuis son poste personnel, en utilisant un navigateur internet (internet explorer, firefox etc...). Les jobs ainsi créés sont ensuite gérés par le gestionnaire de queue qui les distribue aux serveurs sur lesquels s'exécute l'application Huygens. Lorsque les jobs sont terminés, un email informe l'utilisateur que les données déconvoluées peuvent être récupérées.

Système de comptage:

Le système de comptage de MRI (*MRI Accounting System*) permet de rendre compte de l'utilisation des microscopes et des autres stations de travail (cytomètres, stations d'analyse, déconvolution) de la plate-forme. Pour ce faire, la première condition est que les utilisateurs puissent utiliser un login d'accès identique à toutes les stations de travail sur tous les plateaux techniques (et donc sur différents réseaux) ainsi qu'à toutes les applications en lignes offertes par MRI (réservation, déconvolution). Un système hiérarchisé de comptes de groupes, sous-groupes et utilisateurs est utilisé et chacune de ces entités peut acheter un forfait d'heures de référence. Les fractions d'heures de référence sont automatiquement décomptées dès qu'une station de travail de la plate-forme est utilisée. La fraction d'heure de référence débitée par unité de temps donnée dépend du type de matériel et sur certains postes du jour et de l'heure d'utilisation (tarifs WE et nuit). Un outil d'administration développé par MRI R&D permet la gestion des utilisateurs, groupes et crédits et la création de rapport d'utilisation de différents types (par groupe, par utilisateur, par plate-forme, ...).

Le comptage de l'utilisation d'une machine donnée est effectué par un système client-serveur écrit en java. Le serveur utilise une base de donnée mysql. Les communications entre le client et le serveur sont encryptées via un certificat ssl (certificat de cryptage utilisé dans les transactions bancaires notamment). Pour pallier aux problèmes (fréquents) de réseau entre les différents plateaux techniques de la plate-forme, des serveurs d'authentications secondaires locaux peuvent être utilisés sur chacun des sites. En cas de rupture de réseau au cours d'une utilisation d'une des stations de travail, les heures d'utilisation sont comptées localement sur le client puis envoyées automatiquement au serveur lors du retour du réseau. Ce système fonctionne depuis février 2005 et permet la connexion de 42 machines client sur 7 bâtiments au serveur central et aux deux serveurs secondaires.

Cell-Image-Analyser :

MRI-CIA est un outil de développement simple et rapide d'applications d'analyse d'images utilisant ImageJ. Il complète les fonctionnalités d'ImageJ en apportant une interface visuelle pour le développement de scripts par *drag and drop* à

partir d'une liste d'opérations existantes. Ainsi, MRI-CIA permet aussi bien la création d'applications automatisées adaptées à un grand volume d'image que la mise en place d'applications interactives.

Pour les besoins des utilisateurs, un certain nombre de scripts d'applications ont déjà été développés et sont installés et utilisables dans MRI-CIA, dont des applications pour l'analyse de peignage moléculaire de l'ADN, de mesure d'intensité de fluorescence dans les cellules, etc....

3.6 Positionnement du MRI

Le MRI n'est pas une société, c'est un organisme qui a pour but de mettre en commun différents matériels pour permettre leur utilisations à toutes sortes d'organismes qu'ils soient public ou privés.

Le MRI doit néanmoins obtenir un soutien financier lui permettant de rémunérer le personnel, de maintenir le matériel et de financer les évolutions techniques de la plate forme.

Le soutien financier global à la plate-forme est défini comme ceci :

- les frais de personnels sont assurés par les organismes et établissements de tutelle.
 - Le CIRAD, Centre de Coopération Internationale en Recherche Agronomique pour le Développement .
 - Le CNRS, Centre National de la Recherche Scientifique .
 - L'INRA, Institut National de la Recherche Agronomique.
 - L'INSERM, Institut National de la santé et de la Recherche Médicale.
 - L'IRD, Institut de Recherche pour le Développement.
 - L'école SUPAGRO, Centre International d'études Supérieures en Sciences Agronomiques.
 - L'université de Montpellier 1 et 2.

- Le GIS IBiSA (Infrastructures pour la Biologie, la Santé et l'agronomie)
Coordination des plates - formes de recherche en Sciences du Vivant.
- Le GIP Cancéropôle GSO (Grand Sud – Ouest).
- Les frais d'infrastructures sont assurés par les laboratoires d'appui qui constituent les principaux utilisateurs de la Plate - Forme, en nombre d'heures d'utilisation. Ils paient leurs heures d'utilisation au même tarif (public) que les autres utilisateurs publics.
 - B&PMP : Biochimie et Physiologie Moléculaire des Plantes
 - CRBM : Centre de Recherche en Biochimie Macromoléculaire
 - DAP : Développement et Amélioration des Plantes.
 - DBS - UM2 : Département Biologie Santé de l'Université de Montpellier 2.
 - DCC : Différenciation Cellulaire et Croissance.
 - DIMNP : Dynamique des Interactions Membranaires Normales et Pathologiques.
 - IGF : Institut de Génomique Fonctionnelle
 - IGH : Institut de Génétique Humaine.
 - IGMM : Institut de Génétique Moléculaire de Montpellier.
 - INM : Institut des Neurosciences de Montpellier.
 - IRB : Institut de Recherche en Biothérapies.
 - ISEM : Institut des Sciences de l'évolution de Montpellier.
- les frais de maintenance, de petite évolution technologique et de fonctionnement sont assurés par les recettes propres de la plate-forme.

- enfin les grosses évolutions technologiques de systèmes existants et l'acquisition de nouveaux équipements ont été réalisés en 2007-2008 avec les soutiens financiers de :
 - L'ANR, Agence Nationale pour la Recherche est une agence ministérielle.
 - La Direction de l'enseignement Supérieur et de la Recherche du Conseil Régional du Languedoc – Roussillon est une collectivité territoriale.
 - L'ANRS, L'agence Nationale pour les Recherches sur le SIDA et les hépatites virales est un Groupement d'intérêt public et scientifique (GIP et GIS) .
 - Le FRM, la Fondation pour la Recherche Médicale est une association reconnue d'utilité publique.
 - L'ARC, l'association de Recherche contre le Cancer est une Associations reconnues d'utilité publique.
 - Le SIDACTION est une association reconnue d'utilité publique qui répartit les fonds collectés entre tous les programmes de recherches contres le SIDA
 - L'AFM, l'association Française contre les Myopathies est une Association reconnue d'utilité publique.

3.7 Moyens technique mis à disposition

En ce qui me concerne l'équipe a mis à ma disposition un ordinateur complet avec pour le développement le logiciel Eclipse ainsi que plusieurs utilitaires pour l'aide au développement, comme un outil qui s'appelle Lin4J qui s'intègre à Eclipse et qui permet de vérifier les erreurs présentent dans le code comme lorsque l'on oublie d'écrire la documentation du code ou lorsqu'une partie de code ou une variable n'est jamais utilisée donc ne sert a rien etc... Ou comme un outil qui permet de créer des tests Unitaire JUnit. Ou encore un autre outil Mylyn qui est un sous-système d'Eclipse s'occupant de la gestion des tâches, il joue le rôle d'assistant du programmeur.

De plus, pour effectuer les tests avant la mise en production d'une version de logiciel, un accès à un serveur dédié nous permettant d'effectuer les vérifications sur un système de tests en condition réel (c'est une copie du serveur de production, il a la même architecture).

4 Analyse de la mission

4.1 Analyse du projet

A l'heure actuelle, lorsqu'un utilisateur de la plate-forme utilise un des équipements mis à sa disposition (microscope ou tout autre équipement appartenant au MRI), il est obligé de stocker ses données sur le poste de travail de l'équipement qu'il a utilisé. Les données qui sont stockées sur le poste de travail de l'équipement restent sur ce poste jusqu'à ce que l'utilisateur les supprime ou jusqu'à leur effacement automatique. Les images ou données vont être ensuite traitées par des applications d'analyse permettant à l'utilisateur d'effectuer ses recherches : dans le cas des applications de traitement des images ou données, la sauvegarde des fichiers bruts est à la charge de l'utilisateur, alors que dans le cas d'un traitement de déconvolution les images brutes sont stockées sur un serveur MRI dont l'espace disque est très limité.

Si un utilisateur souhaite partager ses données/images ou résultats de recherches avec la communauté scientifique ou simplement avec un ou plusieurs

membres de son équipe, il sera obligé de le faire par ses propres moyens : il devra donc copier ses données sur disque, effectuer un transfert par le réseau ou créer un espace disque partagé.

L'utilisation des ressources de la plate-forme s'adresse à des équipes de recherches. Bien entendu plusieurs personnes font partie de ces équipes et elles ont toutes un but commun : obtenir des résultats. Chaque membre d'une équipe a ses propres responsabilités, et chacune de ces responsabilités apportent un élément dans le but final. Chaque équipier peut être amené à utiliser les ressources de la plate-forme MRI. De plus, une équipe de recherches peut avoir besoin ou peut vouloir partager ses découvertes ou données avec d'autres membres de son équipe ou même avec des personnes extérieures.

Comme nous l'avons vu précédemment, à l'heure actuelle, il est très difficile pour un membre d'une équipe de partager simplement ses données. Il est obligé de trouver des solutions de partage tout seul, sachant qu'elles sont stockées sur la ressource de la plate-forme qu'il a utilisé. Par ailleurs, lorsqu'il souhaite effectuer un certain type de traitement d'image, il est obligé de se déplacer d'un plateau technique à l'autre car chaque site possède ses propres logiciels (chacun étant spécialisé dans un type de traitement), ce qui oblige les chercheurs à se déplacer en fonction de leurs besoins.

Le but final est de permettre à tous les utilisateurs et équipes de recherches de pouvoir utiliser simplement leurs données et surtout de leur permettre de partager rapidement et efficacement ces données entre eux et avec les personnes de leurs choix. Tout en protégeant l'intégrité des données. De plus, il faut rendre utilisables les différents logiciels de traitement d'images entre les différents plateaux techniques.

La finalité est d'offrir la possibilité aux utilisateurs de la plate-forme MRI d'utiliser au mieux tous les équipements présents, mais surtout de pouvoir gérer leurs données à partir de n'importe quel ordinateur présent ou non sur la plate forme, à deux conditions :

- L'ordinateur doit avoir une connexion internet.
- L'utilisateur doit avoir accès via un login et un mot de passe au système global du MRI (site web).

Les questions qu'il faut se poser sont : quels sont les besoins d'une équipe de recherche sur la plate-forme MRI? Comment leur permettre de stocker, de gérer et de partager facilement leurs données? Comment améliorer la qualité des services proposés par la plate-forme?

Pour palier à cette problématique : un projet a été mis sur pied par l'équipe du MRI. Ce projet porte le nom de Cicéro.

Le projet Cicéro :

Dans un premier temps, lorsqu'un utilisateur se sert d'un équipement, il faut créer une procédure automatisée de stockage des données acquises sur le poste de travail de l'équipement. Il faut donc les stocker sur un serveur appartenant à la plate-forme.

L'utilisateur doit pouvoir accéder depuis n'importe quel poste aux différentes données qui lui appartiennent ou sur lesquelles il possède des droits. Les données peuvent lui appartenir ou non (elles peuvent appartenir à un autre utilisateur qui lui a attribué des droits). Il doit donc lui aussi permettre l'accès à ses données pour différents utilisateurs de son choix sur la plate forme. Il doit pouvoir gérer ses données :

- Créer des dossiers pour organiser et ranger ses fichiers.
- Copier des fichiers ou les déplacer d'un dossier à un autre.
- Supprimer les dossiers et les fichiers de son choix.
- Partager des fichiers ou des dossiers avec les utilisateurs de son choix.

Ensuite dans une troisième étape, le système doit permettre aux utilisateurs d'effectuer les traitements nécessaires sur leurs fichiers. Ils doivent pouvoir utiliser les différentes applications de traitements d'images (déjà existantes sur les postes de travail de la plate-forme du MRI) à partir de n'importe quels plateaux techniques de leur choix. Les données brutes de toutes les applications sont donc stockées sur la plate-forme et accessibles via le système.

On peut donc diviser le tout en trois sous parties:

- Une partie acquisition des données.
- Une partie gestion des données.
- Une partie traitement des données.

4.2 Objectifs de la mission

Le projet Cicéro doit répondre à différents besoins énoncés au préalable. Pour ce faire plusieurs modules doivent être développés :

En ce qui concerne la première partie, l'acquisition des données se fait via le matériel des différents plateaux du MRI. Sur ces plateaux des équipements sont à la disposition des équipes de recherches. Chacun s'occupe de faire l'acquisition des données transmises par un matériel (microscope ou autre). En d'autres termes, lorsqu'une image est prise au microscope, elle est transmise à un ordinateur (ordinateur relié au matériel) et stockée sur ce dernier. Le rôle de cicéro est de regrouper toutes les données d'une même personne (données qui peuvent être acquises et stockées sur des équipements différents). Pour réaliser cela, il faut placer sur chaque ordinateur qui accompagne un des matériels, deux logiciels :

- le premier est un logiciel qui permet de faire le téléchargement des fichiers présents sur la machine vers un serveur de fichier pour Cicéro. En réalité ce logiciel ne fait qu'une demande de téléchargement, c'est le deuxième logiciel qui va s'occuper du téléchargement. Il doit aussi permettre à un utilisateur de visualiser l'historique des téléchargements, ceux en cours et doit permettre d'arrêter un ou plusieurs en cours.
- Le deuxième lui s'occupe du téléchargement réel vers le serveur de fichier. Les informations doivent être stockées dans des dossiers qui correspondent à l'utilisateur connecté: si deux utilisateurs utilisent la même machine à deux moments différents, chaque fichier téléchargé sera stocké à un endroit différent. De plus, ce logiciel doit être invisible à l'utilisateur car il est lancé en permanence. Cela permet de faire le téléchargement de gros fichiers sans forcer l'utilisateur à attendre sa fin. Ce logiciel doit être lancé en tant que service Windows, ce qui lui permet d'être lancé automatiquement à chaque

démarrage mais surtout de continuer de fonctionner lorsqu'un utilisateur se déconnecte de sa session Windows (bien sûr le logiciel ne fonctionne plus lorsqu'un utilisateur éteint la machine).

A mon arrivée cette tâche est en partie réalisée. Le service Windows ou Daemon est déjà créé mais il doit être lancé manuellement et il faut en faire les tests. Le logiciel de téléchargement ou Uploader fonctionne, mais il est encore impossible de visualiser ou stopper les téléchargements en cours ainsi que l'historique. Mon rôle au cours de la mission est de faire les tests du Daemon et de l'intégrer comme service Windows. En ce qui concerne l'Uploader, il faut permettre la visualisation et l'arrêt des téléchargements en cours.

Ensuite pour la deuxième partie qui permet la gestion des données depuis n'importe quel poste le principe est simple, il faut permettre aux équipes de recherches d'accéder à leurs données depuis n'importe quel poste de travail interne ou externe aux locaux du MRI. Pour cela il faut créer un site web. Il permet de satisfaire plusieurs critères importants dont l'accès depuis l'extérieur. A mon arrivée, cette partie n'a pas été commencée et ma mission va se concentrer principalement sur cette tâche. Il faut permettre à un utilisateur d'accéder au site via un login et un mot de passe. Cet accès est déjà réalisé pour d'autres sites appartenant au MRI, l'objectif étant d'intégrer ce nouveau site à ceux déjà existants. De plus, le site doit permettre à un utilisateur de visualiser les différents téléchargements qu'il a effectués depuis n'importe lequel des équipements de la plate-forme. Pour cela l'idée est de créer (à la connexion d'un utilisateur) une arborescence de dossier et de fichier un peu comme un explorateur de dossier sous Windows. A partir de cette arborescence un utilisateur doit pouvoir faire la gestion de ses données :

- Il doit pouvoir créer, copier, couper, coller, renommer et supprimer des dossiers.
- Il doit pouvoir renommer, copier, couper, coller et supprimer ses fichiers.
- Il doit pouvoir avoir accès à certains dossiers et fichiers que d'autres utilisateurs ont partagé pour lui.
- Il doit pouvoir donner l'accès, à d'autres utilisateurs de son choix, à certains de ses dossiers ou fichiers.

Pour permettre à un utilisateur de partager ses données, nous devons instaurer ce que l'on appelle « des droits d'accès appliqués au dossier et au fichier ». Reprenons : lorsqu'il se connecte, une arborescence du système de fichier apparaît, dans celle-ci n'apparaît que les dossiers et fichiers auxquels l'utilisateur connecté a des droits de lecture. Donc en permettant à un utilisateur de donner à un ou plusieurs utilisateurs des droits sur ses fichiers et dossiers, il peut par cette intermédiaire partager ses fichiers. Ainsi la notion de droit d'accès lui permet de protéger ses données.

En effet, d'autres fonctions de gestions des données doivent être ajoutées, comme la création d'une corbeille qui permet lorsque l'on supprime un fichier ou un dossier de revenir en arrière (un peu comme Windows le fait avec sa corbeille).

De plus, un moteur de recherche sera proposé aux utilisateurs. Ceci leur permettra de retrouver leurs données plus facilement en tapant simplement les mots clefs correspondant à leur recherche. Pour cela, il faut ajouter à un fichier une notion de tag. Un tag c'est une note (post-it) attribuée à un fichier que l'on peut ajouter à la main via le site. Ce tag permet lorsqu'un utilisateur tape un mot de rechercher tous les tags correspondant et de lui proposer en réponse les images affiliées.

Pour la dernière partie qui concerne la gestion des traitements d'images via le site web, à mon arrivée au sein de l'équipe MRI aucun élément n'était mis en place. Cette tâche n'est pas abordée au cours du stage car elle n'est pas prévue dans le sujet de stage par manque de temps, mais elle fait partie du projet Cicéro et sera développée au cours d'une deuxième partie. Le but principal est de permettre aux utilisateurs de Cicéro d'effectuer leurs traitements sur leurs données, directement à partir du site web. Cela apporte un avantage certain : le travail pour les utilisateurs est simplifié (ils n'ont plus besoin de se déplacer sur les machines des différents plateaux pour pouvoir effectuer leurs traitements).

En résumé ma mission au cours de ce stage est de modifier et de retoucher la première partie sur l'Uploader, il faut effectuer les tests, intégrer le Daemon en tant que service Windows et modifier le logiciel de téléchargement. Pour le site web il faut créer une interface web permettant la connexion d'un utilisateur, la création de son arborescence et la gestion de ses dossiers et fichiers. Il doit

pouvoir les copier, les déplacer où il en a envie, créer des fichiers et des dossiers. Il doit pouvoir visualiser ses fichiers sous forme d'aperçu (de grande, petite et moyenne taille) et télécharger les fichiers ou les dossiers qu'il veut.

5 Méthodologie utilisée

5.1 Méthodologie de travail

L'équipe de développement est composée de trois personnes:

- Volker Baecker Développeur, responsable de l'équipe et tuteur.
- Yin Wei Développeur, stagiaire.
- Moi-même, Nicolas Chevron Développeur Stagiaire.

Nous sommes donc deux stagiaires sur le même projet sous la responsabilité d'un même tuteur: V. Baecker.

Il existe un responsable de projet qui s'occupe de la création du cahier des charges, de l'organisation des réunions, de la gestion du budget et des équipes. Cette responsable s'appelle Corinne Tran-Aupiais.

Dans un premier temps il a fallu écrire le cahier des charges. Pour cela Corinne Tran-Aupiais a décidé d'organiser plusieurs réunions avec tous les membres de l'équipe MRI. Elles ont permis à chacun de définir les besoins auxquels devait répondre Cicéro. Ces réunions sont organisées sous forme de Brainstorming, ce qui permet à chaque membre présent de donner son opinion sur les différentes missions de Cicéro. Ensuite une fois tous les besoins collectés, le cahier des charges est défini. Il est en réalité un condensé des différentes réunions organisées au préalable. De plus il contient une partie de la gestion des coûts et délais à respecter tout au long du projet.

Afin de pouvoir gérer les délais du projet un logiciel collaboratif est utilisé : il s'appelle DotProject, il permet notamment de créer des tâches et de définir des dates de réalisation (dates de début et date de fin). Il est également possible pour chaque tâches de définir des sous-tâches. Pour chacune d'entre elles le logiciel permet de renseigner précisément une description, de définir son

avancement (en pourcentage), d'attribuer la tâche à une personne de l'équipe, ou encore de définir sa priorité dans le projet. De plus chaque tâche et sa ou ses sous tâches associés, appartiennent à une grande tâche que l'on appelle une release. Elle permet de définir un moment où le projet passe à une étape dans sa réalisation (cela peut être une première version du logiciel).

Le cahier de charges réalisé ; V. Baecker commence la réalisation du projet. Il doit dans un premier temps décider de l'architecture technique du projet, et faire valider les choix par la hiérarchie et par l'administrateur Réseau.

Enfin le développement du projet peut commencer. Pour travailler V. Baecker décide d'utiliser la méthodologie de travail : Extrême Programming.

5.2 Méthode de gestion de projet

En ce qui concerne le développement, le choix s'est porté sur l'utilisation des méthodes agiles, et en particulier XP ou Extrême Programming.

Extrême Programming (XP) est une méthodologie adaptée à des équipes de taille réduite qui développent des logiciels dans un contexte où les besoins sont mal connus ou susceptibles d'évoluer rapidement.

Elle se démarque des approches traditionnelles en mettant l'accent sur l'implication du client, le travail d'équipe, l'adaptation au changement et la robustesse des applications développées.

Cette méthode de gestion de projet s'articule autour de quatre principes :

- La communication : Comme la priorité est donnée à la collaboration avec le client plutôt qu'aux négociations contractuelles, il faut que la communication avec ce dernier soit bonne afin qu'il comprenne bien les enjeux de ce genre de projet et qu'il s'implique au même titre que les développeurs. De plus une bonne communication entre développeurs est primordiale car le travail s'effectue en équipe et si un membre de l'équipe ne suit pas le rythme, c'est l'ensemble de l'équipe qui s'en retrouve pénalisé.
- La simplicité : XP met l'accent sur la simplicité, il faut développer ce dont on a

besoin maintenant et ne pas penser à ce dont on pourrait avoir besoin ultérieurement.

- Le Feedback : Comme XP est une méthode de développement agile, le principe est de faire des itérations courtes. C'est à dire que rapidement une première version du logiciel doit être livrée et installée chez le client pour qu'il puisse l'utiliser. Bien sûr cette première version ne contient que les fonctionnalités nécessaires et obligatoires pour pouvoir commencer à l'utiliser. Mais l'avantage majeur est de permettre d'avoir un retour rapide de la part du client (retour sur certain bug ou sur certaines fonctionnalités qui ne fonctionnent pas comme il l'entendait). Le Feedback est aussi garanti par l'implémentation des tests unitaires qui permettent d'avoir un retour permanent sur l'état du logiciel (ceci est expliqué plus en détails dans la suite du document).
- Le courage : Les développeurs doivent avoir du courage : Le courage de se concentrer sur la fonctionnalité en cours sans penser aux fonctionnalités suivantes (si deux fonctionnalités ont une même base, il ne faut pas penser à la prochaine même si lorsqu'on sera à la seconde il faudra recommencer la première). Le courage de mettre à la poubelle du code trop complexe et de le recommencer pour le simplifier.

Les principes mis en œuvre dans XP sont pour la plupart des bonnes pratiques mises au point et éprouvées dans d'autre type de projet. Aucune de ces pratiques n'est fondamentalement révolutionnaire. XP remet ces pratiques en application dans un contexte plus propice, une méthodologie allégée de développement. La nouveauté dans l'approche XP consiste à organiser des pratiques éprouvées, en un tout cohérent où chaque pratique renforce les autres, et de pousser l'ensemble à l'extrême (d'où le nom de la méthode). Si les tests sont une bonne pratique, XP en fait un point de passage obligé et permanent de la programmation (test-driven development) et de l'intégration (pas d'intégration si tous les tests unitaires ne passent pas).

Voici ci-dessous toutes les pratiques de la méthodologie :

- Le Planning Game : Dans ce type de projet, il faut dans un premier temps définir les scénarios du logiciel (c'est en réalité tous les cas d'utilisation du logiciel ou ses fonctionnalités). Cette partie a été réalisée par Corinne Tran-Aupiais, Volker Baecker et Olivier Miquel. Une fois que les scénarios sont définis, je les ai découpés en tâches avec l'aide de Yin Wei. Dans cette partie a été définie une liste de tâche correspondant à toutes les fonctionnalités du logiciel. Une fois toutes les tâches développées le logiciel est fini. Ensuite il faut lister toutes les tâches par ordre de priorité allant de la plus importantes à la moins importante. Enfin cette liste permet de définir à partir de quel moment la première itération est terminée, et donc de savoir quels sont les tâches à développer pour que le premier logiciel puisse être installé chez le client.
- Le cycle de développement rapide : Le principe est de mettre très rapidement le logiciel en phase de production (au départ avec des fonctionnalités de base, le minimum requis), puis de livrer des versions successives selon un cycle itératif très court (typiquement 2 semaines). Cela permet d'avoir un retour, sur ce qui est développé, de la part du client de façon très rapide. Ce qui va donc permettre de pouvoir corriger les différents problèmes rencontrés par le client avant la prochaine itération. Au cours de mon stage, j'ai travaillé sur la première itération du logiciel. Quelques jours après mon départ cette première version était mise en production.
- Méthodes basées sur les tests: Les développeurs écrivent en permanence des tests automatisés qui doivent valider une fonction. Le but dans Extrême Programming, c'est d'écrire les tests avant de l'écrire. La fonction est finie lorsque les tests sont valides. De plus à chaque modification du code déjà écrit ou à chaque fois qu'une fonction est écrite, il faut relancer tous les tests du projet. Si tous les tests sont validés alors on peut passer à la fonction suivante. Si un ou plusieurs des tests n'est pas validé, dans ce cas il faut revenir sur cette partie et réparer les erreurs. Ce sont les tests qui permettent de garantir la robustesse de l'application.
- Le Refactoring : C'est un processus qui consiste à modifier du code déjà existant sans en altérer les fonctionnalités. Le but dans extreme programming est d'effectuer cette tâche de façon régulière pour permettre d'avoir un code

qui est en permanence optimisé. Au cours du stage le refactoring s'effectue durant une journée par semaine. Il est terminé lorsque tous les tests unitaires sont validés.

- Conception simple : La conception doit être la plus simple possible, une complexité inutile doit être immédiatement supprimée. Simple ne veut pas dire simpliste, le code doit être clair, facilement testable, concis et sans duplication. Cela n'est réalisable qu'après plusieurs refactoring.
- Pair Programming : Le développement s'effectue par paire. Nous sommes donc deux pour un seul ordinateur, un développeur s'occupe de taper le code et l'autre dicte au premier. Les rôles doivent être inversés très régulièrement, dans notre cas le binôme passe deux heures sur la tâche d'un des développeurs, le roulement s'effectue donc toutes les heures. Le but du pair programming est de permettre la réalisation de la conception en binôme.

L'avantage de ce genre de méthode vient du fait que le premier livrable arrive très rapidement. Il ne contient pas encore toutes les fonctionnalités du logiciel, mais il permet déjà d'étancher la soif du client qui lui ne demande qu'une chose : voir le logiciel en fonctionnement. De plus, le client recevant le premier livrable rapidement, il commence à faire les premiers tests et à découvrir les premières erreurs du logiciel. Erreurs qu'il est simple de résoudre puisque le but de ces méthodes est de permettre la flexibilité, donc dans ce cas il est très simple de corriger les erreurs.

Dans ce type de méthode c'est l'équipe de travail qui a le plus gros des pouvoirs de décision. Le travail se fait en binôme et ils changent au fur et à mesure du temps. Des réunions doivent être organisées quotidiennement entre les membres de l'équipe ce qui permet de se mettre au courant des difficultés de chacun et de prendre des décisions en groupe.

Au début du stage le projet Cicéro a été divisé en plusieurs parties, chacune d'entre elles représentant une fonctionnalité du logiciel. Ensuite pour chaque partie il a fallu diviser le travail en tâches.

Le développement se déroulera avec une équipe de trois personnes. Le travail en binôme étant rendu difficile à cause du nombre impairs des membres de l'équipe, il a été décidé de travailler avec un binôme et un développeur seul. Le travail

s'est donc découpé ainsi : lorsque l'on arrive le matin nous faisons une réunion, nous discutons des difficultés rencontrées, nous cherchons rapidement des solutions et nous nous séparons par groupe. Chaque groupe travaille ensemble sur une durée de quatre heures. Il est donc nécessaire de changer de groupe toutes les quatre heures. Chaque membre de l'équipe a la responsabilité de développer une fonctionnalité (composée de tâches). Durant les quatre heures deux heures sont accordées aux tâches de chacun des membres. Durant ces deux heures, un des membres du groupe se trouve en possession du clavier (c'est lui qui s'occupe de taper au clavier) et l'autre s'occupe de dire au premier ce qu'il doit faire. Ensuite au bout d'une heure les rôles sont inversés. Les roulements sont interrompus de temps en temps par des réunions avec les autres membres de l'équipe MRI pour faire un point sur l'avancement du projet et discuter d'éventuels problèmes rencontrés.

5.3 Répartitions des tâches

Comme vue précédemment le travail sur le projet a été divisé en scénarios. Ces scénarios vont donc être définis précisément. Chaque scénario sera découpé en tâche et chaque membre de l'équipe aura la responsabilité de réaliser certaines de ces tâches. En réalité elles sont distribuées au fur et à mesure au cours du développement, et ceci suivant un ordre de priorité établi au préalable : c'est à dire que lorsqu'un membre a terminé sa tâche, il prend la suivante.

Les différents scénarios sont établis au cours des réunions concernant les besoins du projet Cicéro. Les réunions sont organisées par Corinne Tran-Aupias. J'ai participé à quelques unes d'entre elles (les autres ayant eu lieu avant mon arrivée). Seules les tâches auxquelles j'ai participé, sont reprises dans les définitions suivantes:

- 1) La gestion des dossiers.

Cela permet de faire la création et la modification d'un dossier.

- 2) La possibilité d'effacer des données.

Il permet l'effacement des données d'un utilisateur (image ou donnée). Bien sûr un utilisateur doit posséder des droits afin de supprimer des données.

3) La possibilité de renommer les fichiers ou les dossiers.

Chaque dossiers et fichiers doivent pouvoir être renommés. Bien sûr il ne peut y avoir dans un dossier plusieurs fichiers ou sous-dossier portant le même nom.

4) La possibilité de copier des dossiers ou fichiers.

Les fichiers ou les dossiers peuvent être copiés vers un autre dossier. Il faut que dans le dossier de destination, il n'existe aucun fichier ou dossier portant le même nom qu'un des éléments devant être copiés.

5) La possibilité de déplacer des dossiers ou fichiers.

C'est le même principe que pour le copier, à la seule différence que les fichiers ou dossiers sources doivent être supprimés (c'est l'équivalent du 'couper').

6) La possibilité de renommer plusieurs fichiers ou dossiers en même temps.

Le but est de renommer plusieurs fichiers ou dossiers en même temps. Pour cela l'utilisateur détermine un nom, et celui-ci sera utilisé pour tous les autres en y ajoutant un nombre à la fin, correspondant à un compteur. Le premier se nommera « Nom1 », le deuxième « Nom2 » et ainsi de suite.

7) L'établissement d'une gestion des droits d'accès.

Pour permettre à un utilisateur de protéger ses données, une notion de droit d'accès est ajoutée. Différents droits doivent permettre à un utilisateur d'autoriser d'autres utilisateurs à lire ou à modifier ses données.

8) La gestion du format de données du Nanozoomer.

Un des formats de fichier le plus important est le format du nanozoomer (c'est un appareil du MRI). Cette partie doit permettre la gestion de ce format.

9) La création des aperçus web (viewer).

Dans l'interface Web, chaque fichier doit avoir un aperçu visible depuis une page web.

Pour finir, j'ai découpé chaque scénario en tâche définie précisément. Pour cela, il faut présenter toutes les méthodes, les classes et les attributs qui doivent être créés. L'exemple de découpage du scénario « Gestion des dossiers » est présenté en annexe 1.

Il est bien évident que certaines tâches ne peuvent pas être commencées si d'autres ne sont pas finies. C'est pour cela que dans la définition des tâches d'un scénario, une partie d'un autre peut être commencée. Par exemple en ce qui concerne les scénarios « gestion des droits d'accès » et « effacer les données », il est évident que pour effacer des données il faut vérifier les droits de l'utilisateur sur ces données. Par exemple s'il faut développer dans un premier temps la partie permettant d'effacer. Dans ce cas une partie de la gestion des droits d'accès devra être développée durant une des tâches permettant d'effacer des données.

Ensuite à partir des tâches établies nous avons pu définir une première itération (voir la partie concernant la méthode de gestion de projet). Pour cela il faut définir tous les scénarios qu'il contient par ordre de priorité.

La liste des scénarios dans l'ordre (de la plus à la moins importante) est la suivante : scénario1, scénario2, scénario 3, scénario 8, scénario 9, scénario 4, scénario 5.

Planning des tâches prévisionnel

Numéro du scénario	Temps impartie à sa réalisation
Scénario 1 : Gestion des dossiers	70 heures
Scénario 2 : Effacer les dossiers	40 heures
Scénario 3 : Renommer les données	30 heures
Scénario 8 : Le Format Nanozoomer	100 heures
Scénario 9 : Création des aperçus	70 heures
Scénario 4 : Copier des données	30 heures
Scénario 5 : Déplacer des données	30 heures

Phases	Date Début	Date de Fin
Apprentissage	01/06/2009	05/06/2009
Analyse du projet et répartitions des tâches	08/06/2009	30/06/2009
Développement	01/07/2009	26/08/2009

6 Réalisations

Le projet Cicero est un logiciel contenant une application classique (client lourd) et une application web (un client léger), il utilise pour cela la technologie JEE5 (EJB, JPA, web-services, JSF et Richfaces).

6.1 Outils Technologiques utilisés

6.1.1 Java Enterprise Edition

La technologie J2EE est une spécification de la technologie java de SUN, elle est particulièrement adapté aux développements d'entreprise d'où son nom. Son implémentation contient un ensemble d'extension au Framework java standard facilitant la création d'application répartie. Ce type d'implémentation facilite notamment le développement d'une application déployée sur une architecture 3-Tiers. Cette forme d'architecture permet de séparer la couche applicative, contenant toute la logique métier, de la couche présentation, qui est la partie visible par l'utilisateur, et la partie stockage et accès aux données, qui s'occupe de conserver dans le temps des données de façon structurée. Ce type d'architecture permet de simplifier les travaux de maintenance sur l'application. Comme l'application est divisée en trois parties plus ou moins indépendante, la modification sur une des parties n'influence pas les deux autres donc si l'on veut par exemple modifier l'interface graphique, nous ne sommes pas obligés de modifier la partie applicative et stockage.

6.1.2 Enterprise JavaBeans (EJB)

La technologie Enterprise JavaBeans (EJB) est une architecture de composants logiciels côté serveur pour la plateforme de développement J2EE. C'est la partie applicative vue précédemment.

Cette architecture permet de créer des composants distribués (c'est-à-dire déployés sur des serveurs distants) écrit en langage de programmation Java hébergés par un serveur applicatif permettant de représenter des données (EJB dit entité), de proposer des services avec ou sans conservation d'état entre les appels (EJB dit session), ou encore d'accomplir des tâches de manière

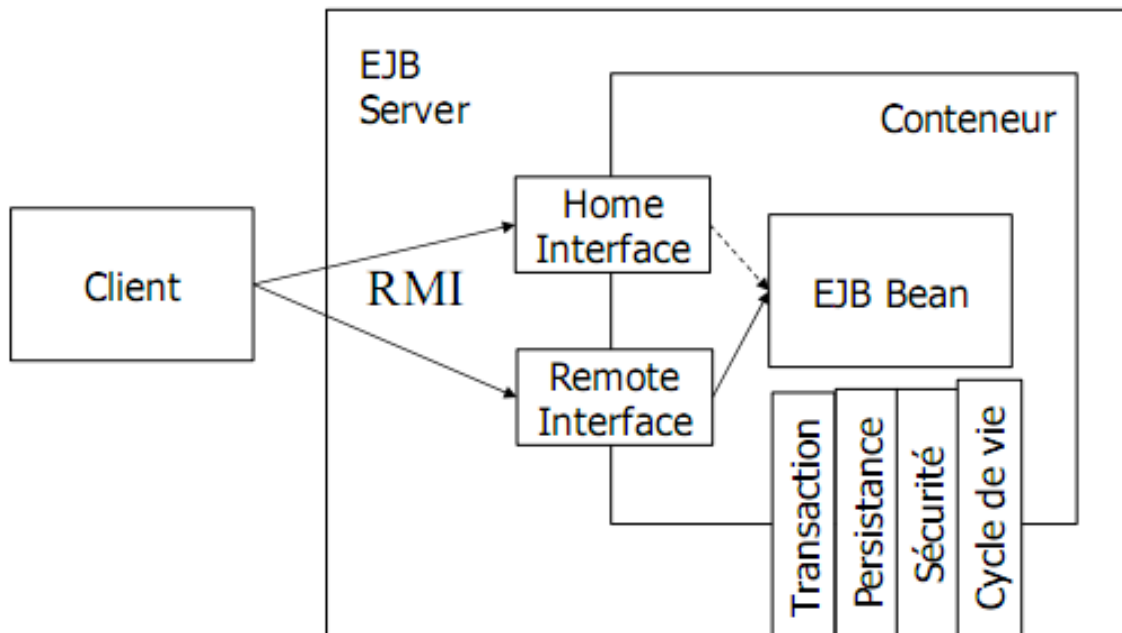
asynchrone (EJB dit message). Tous les EJB peuvent évoluer dans un contexte transactionnel.

EJB définit trois types d'objet que les réalisateurs peuvent construire et adapter aux besoins du client, comme suit :

- EJB sessions sont des objets proposant des services à leur appelant. Ils proposent un certain nombre de méthodes écrites par le développeur.
- EJB message permettant de déclencher un processus côté serveur applicatif lors de la publication d'un message asynchrone.
- EJB entité sont des beans ayant majoritairement pour vocation d'être persistants, c'est-à-dire pouvant être stockés sur un support physique entre deux sessions.

Les EJB ont été utilisés dans le projet Cicéro et ils permettent de fournir à partir du serveur d'applications, différents services.

Schéma de fonctionnement des EJBs



6.1.3 Java Persistence API (JPA)

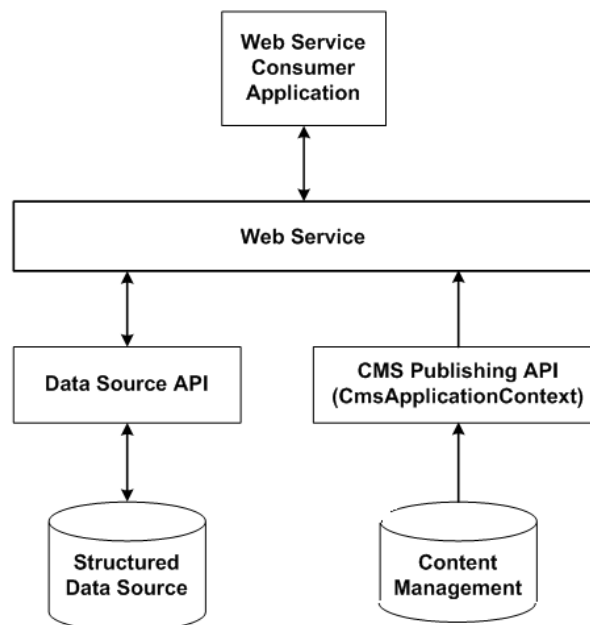
La technologie JPA a pour objectif d'offrir un modèle d'ORM (Object Relational Mapping) indépendant d'un produit particulier (comme Hibernate, TopLink, etc.). Elle correspond à la partie stockage et accès aux données vue dans le chapitre. Cette technologie est basée sur :

- Un jeu d'interfaces permettant de séparer service de persistance et implantation,
- Un jeu d'annotations pour préciser la mise en correspondance entre classes Java et tables relationnelles,
- Un fournisseur de persistance (par exemple Hibernate),
- Un fichier XML « persistence.xml » décrivant les moyens de la persistance (fournisseur, datasource, etc.)

Cette technologie est utilisable dans les applications Web (conteneur Web), ou dans les EJB (serveur d'applications) ou bien dans les applications standards (Java Standard Edition).

6.1.4 Web-services

Schéma de fonctionnement d'un web-service



Un service web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués à l'aide des fichiers XML.

Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et en temps réel.

6.1.5 JSF

Java Server Faces est un framework d'interface utilisateur pour les applications web, basé sur les technologies JSP et Servlets. Le but de JSF est d'accroître la productivité des développeurs dans le développement des interfaces utilisateur tout en facilitant leur maintenance. Il correspond à la partie présentation de l'architecture 3-Tiers. JSF permet :

- Une séparation nette entre la couche de présentation et les autres couches
- Le mapping HTML/Objet
- Un modèle riche de composants graphiques réutilisables
- Une gestion de l'état de l'interface entre les différentes requêtes
- Une liaison simple entre les actions côté client de l'utilisateur et le code Java correspondant côté serveur
- La création de composants customisés grâce à une API
- Le support de différents clients (HTML, WML, XML, ...) grâce à la séparation des problématiques de construction de l'interface et du rendu de cette interface

Il existe plusieurs Framework web Java dédiés au développement d'interfaces utilisateur mais aucun n'est un standard et va aussi loin que JSF.

6.1.6 Richface

Richfaces est une librairie de composants JSF pour le développement d'applications web riches (RIA) Rich Internet Application avec Ajax, elle est le résultat de la fusion de deux projets développés par exadel qui sont:

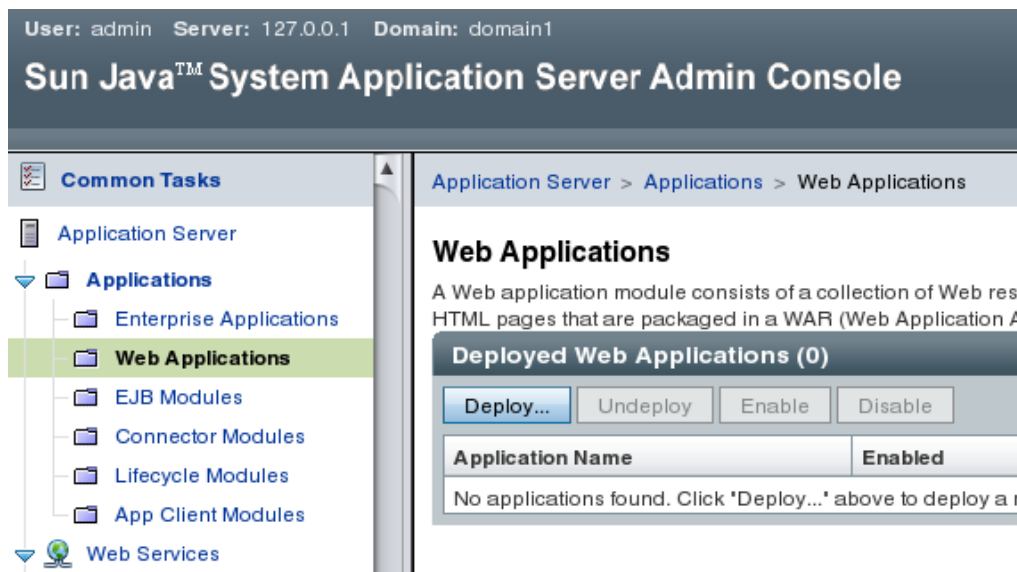
- Ajax4jsf est né dans la fondation sourceforge.net en 2005 sous le nom de Télamon, son concepteur a intégré ensuite la société Exadel qui a commercialisé le projet dans une première période avant de le rendre open source sur Java.net.
- Richfaces : C'est une librairie commerciale de composants JSF fournie par Exadel.

6.2 Outils utilisés

Le projet Cicero est une application distribuée qui est développée en utilisant la technologie J2EE. Les outils suivant sont donc nécessaires pour le développement de ce type d'application:

6.2.1 Le Serveur Glassfish Serveur v2.1

Interface d'accueil du serveur Glassfish



Glassfish est un serveur d'applications compatibles J2EE. Il a été créé et il est maintenu par Sun Microsystems. Nous utilisons pour la persistance le framework Toplink d'Oracle. Glassfish est la réponse de Sun aux développeurs Java désireux d'accéder aux sources et de contribuer au développement des serveurs d'applications de nouvelle génération de Sun.

6.2.2 MySQL

Système de gestion de base de données pour le projet Cicero.

6.2.3 Eclipse ganymède avec l'édition JAVA EE et les plugins

Eclipse est un environnement de développement intégré libre, extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Il dispose d'un éditeur visuel pour Java qui permet de gérer l'ensemble de son code très facilement.

Le plugin subclipse a beaucoup participé à la méthode de travail. Cette outil permettant de manipuler un répertoire SVN, nous permet de stocker les différentes versions du logiciels sur un serveur configuré à cet effet (serveur SVN). Ainsi il est possible de retrouver toutes les versions modifiées depuis le début du développement, mais il est surtout possible de revenir à des versions précédentes du logiciel dans le cas d'un mauvais fonctionnement ou d'une erreur. Ce module est très intéressant car il permet à chaque membre d'une équipe d'apporter ses améliorations au logiciel et d'ensuite le mettre à jour sur le serveur sans perdre les anciennes versions. Bien entendu cette fonctionnalité d'éclipse est d'autant plus intéressante que le projet est complexe et le nombre de personnes concernées important.

Mylyn est le nom d'un sous-système d'Eclipse s'occupant de la gestion des tâches, il joue le rôle d'assistant du programmeur. Une fois ces tâches intégrées, Mylyn scrute le profil d'activité du programmeur pour identifier à tout moment la tâche en cours et afficher les informations pertinentes relatives à cette tâche en fonction du contexte en commandant l'interface graphique d'Eclipse. Son but est de réduire le temps consacré par le programmeur à :

- La recherche d'information.
- L'exploration des textes de la source et de l'aide.
- La navigation dans les fichiers

7 Synthèse de la solution apportée

Interface de connexion au système Cicéro



MRI Central Login

Login

If you already have an account enter your username and password in the fields in the upper right corner and press the login button. You can then access one of the applications listed below.

Get an account

To get an account, please fill in the form on our [inscription page](#). Any research person can ask for an account.

Applications:

- [Reservation System](#)
Go there to reserve one of our imaging systems. Note that you must login to reserve but you can check if a system is free without being logged in.
- [Manage your MRI-account](#)
Update your email address or change your password.
- [Huygens Remote Manager](#)
This is the remote image restoration software from MRI. The page offers a remote interface to accomplish deconvolution of microscopic images.
- [Helpdesk](#)
To contact the support staff, report a technical problem on a given workstation, or report any other anomaly, please use the MRI helpdesk.

Name:

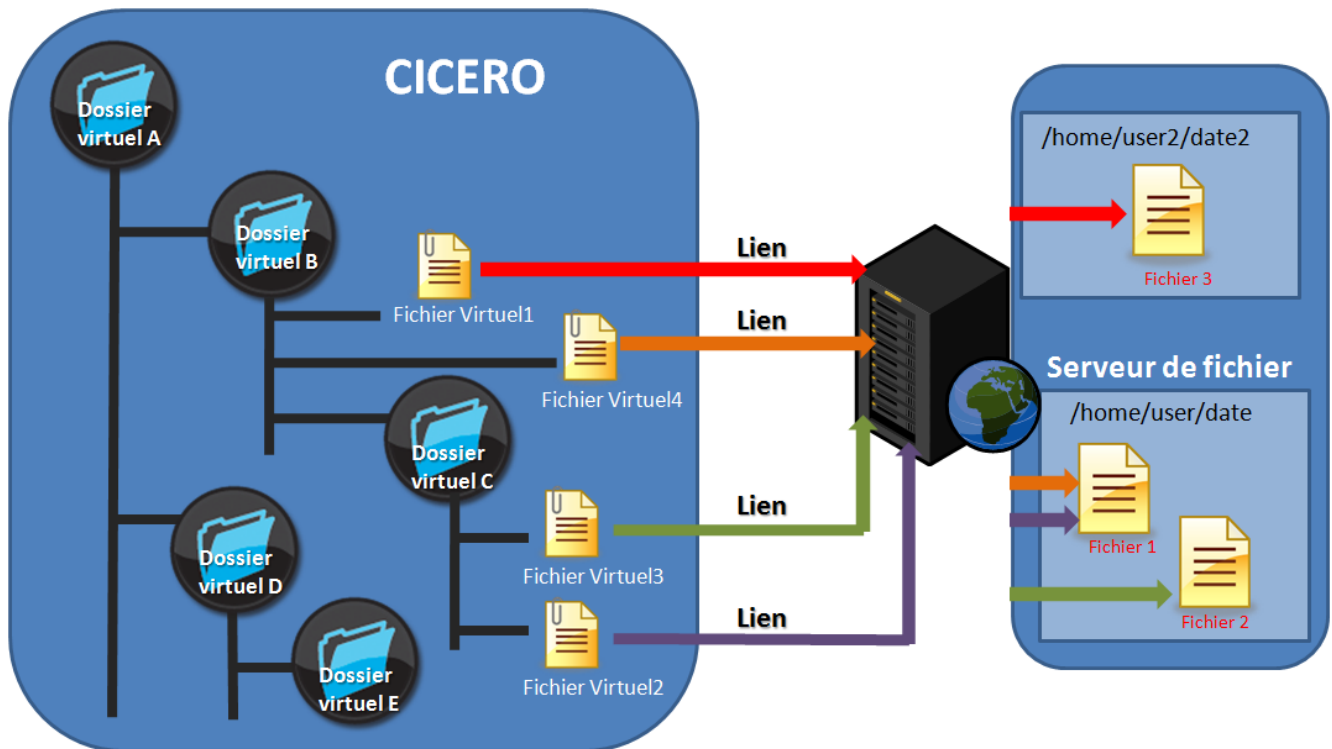
Password:

La première interface web ne peut être visionnée que par un utilisateur déjà connecté sur le système global MRI. Elle est donc accessible par l'intermédiaire de la page de login du MRI déjà existante. La vérification de session permet de contrôler les droits de l'utilisateur avant l'accès à la page web. Si la session est validée, la page web affiche les dossiers de cet utilisateur.

7.1 Le système de fichier virtuel

Les dossiers et les fichiers de Cicéro font partie du système de fichier virtuel. Celui-ci a pour rôle de simuler un système de fichier permettant aux utilisateurs de gérer leurs données. Il faut savoir que tous les fichiers d'un utilisateur sont stockés sur un serveur de fichier. L'arborescence permettant de stocker ces fichiers n'est pas la même sur le serveur de fichier que sur le système Cicéro (visible par un utilisateur). Quand un utilisateur crée un dossier sur Cicéro ce dernier n'est pas créé sur le serveur de fichier.

En réalité dans Cicéro, il n'existe que des dossiers virtuels et des fichiers virtuels. Ces dossiers et fichiers virtuels sont conceptualisés de cette manière :

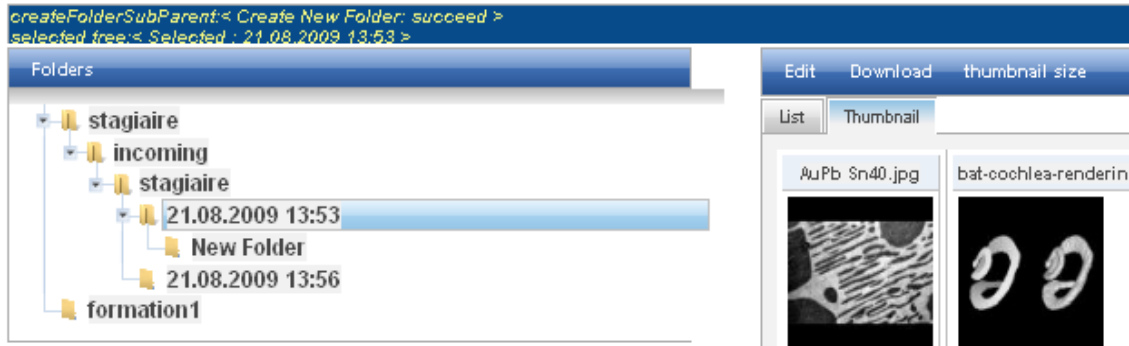
Schéma du système de fichier du projet Cicéro

Un dossier virtuel peut contenir un ou plusieurs dossiers virtuels ou fichiers et ainsi de suite. Un fichier virtuel possède un lien. Ce lien pointe vers un fichier réel présent sur le serveur de fichier. Ces fichiers et dossiers virtuels n'existent pas réellement, ils sont modélisés par des objets représentés par le diagramme de classe en annexe 2. Ces objets sont stockés dans une base de données. Le lien vers le fichier réel est stocké dans l'objet `fileReference`, il est en réalité composé : « /home/'propriétaire'/'date-de-réation'/'identifiant'.'extension' » qui correspond au chemin du fichier sur le serveur de fichier.

7.2 La gestion des dossiers

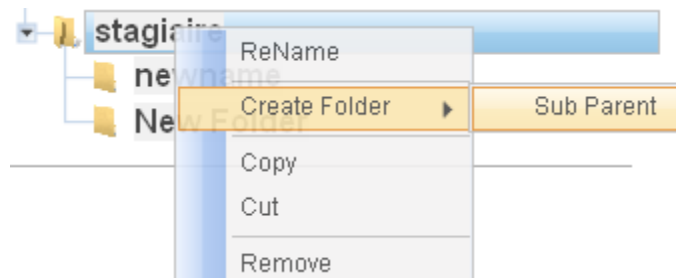
L'interface de gestion des dossiers permet de visualiser les dossiers virtuels dans une arborescence. Les fichiers et les aperçus sont visibles dans le panel de droite avec les noms, le type (dossier ou fichier), la taille, la dernière date de modification et un lien de téléchargement.

Affichage des aperçus d'un dossier



Le menu contextuel doit permettre à l'utilisateur de pouvoir créer un nouveau dossier dans le dossier sélectionné sur l'arborescence. Ensuite la création d'un dossier donne lieu à l'attribution des droits d'accès propriétaire sur ce nouveau dossier.

Interface qui permet de créer un dossier



Cette partie a été divisée en quatre tâches:

1) Récupérer les dossiers virtuels

Dans cette tâche, Il faut créer un EJB qui va permettre en fonction de l'utilisateur connecté de récupérer les dossiers et les fichiers virtuels.

2) Création de la page web

Ensuite il faut créer une page web avec un affichage de l'arborescence des dossiers virtuels. Pour cela nous devons créer une page JSF. Ensuite il faut donc faire appel à la méthode de l'EJB créé au dessus et donc d'afficher tous les dossiers et les fichiers virtuels dans un arbre.

3) Création du dossier

Pour créer le dossier nous avons besoin de différentes informations:

- Nom du dossier
- Le dossier parent

Il faut donc ajouter une méthode à l'EJB permettant la création de ce dossier. Elle reçoit comme paramètre l'identifiant du client ainsi que le nom du dossier à créer et l'identifiant du dossier parent.

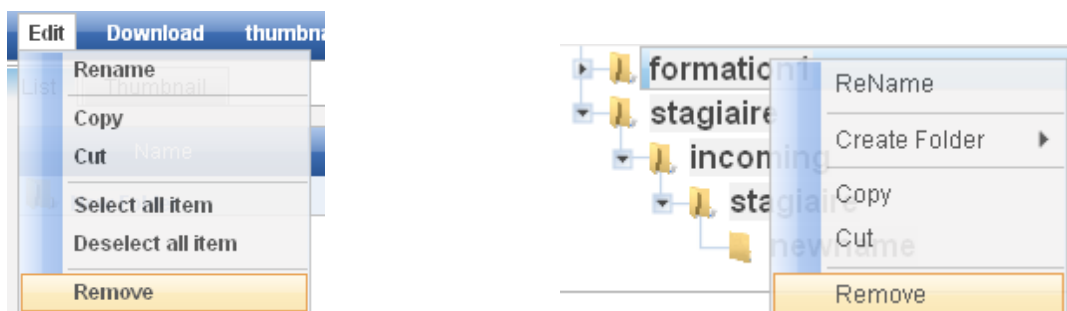
4) Affectation des droits sur le dossier

Pour modéliser cela nous avons décidé de créer une entité AccessRight qui permet de définir un type de droit d'accès. Nous avons vu précédemment que la création d'un dossier donne lieu à l'attribution de droit d'accès propriétaire. Les droits d'accès propriétaire correspondent aux droits en lecture, téléchargement, écriture et en visualisation.

7.3 Effacement des dossiers

Lorsque l'on clique sur le bouton 'remove' du menu déroulant le dossier de sélectionné est supprimé à condition que l'utilisateur ait les droits nécessaire pour cette opérations.

Interface permettant de supprimer un dossier ou un fichier

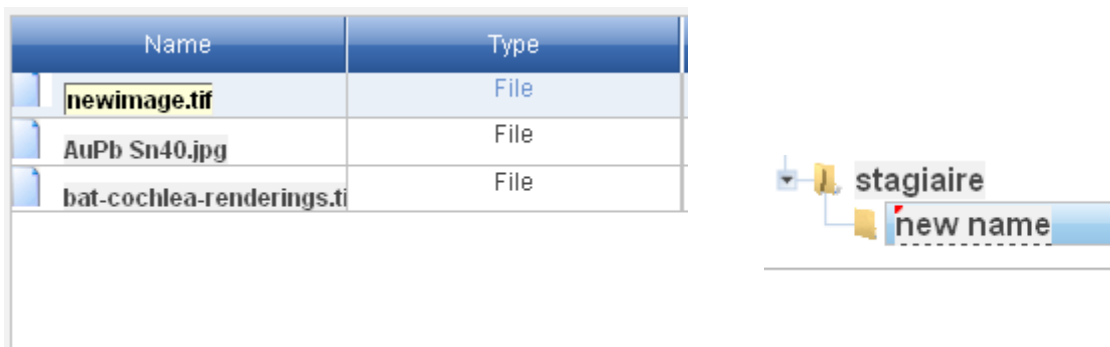


Pour cela, il faut en premier ajouter à l'EJB une méthode pour effectuer le traitement. Cette méthode doit vérifier qu'il a les droits pour supprimer le dossier et tous les sous-dossiers. S'il est propriétaire de tous les dossiers, la méthode se chargera de supprimer tous les fichiers et dossiers. Ensuite, on peut ajouter cette fonctionnalité dans le menu contextuel de l'interface graphique pour permettre au clique droit sur le dossier de le supprimer.

7.4 Renommer un fichier ou un dossier

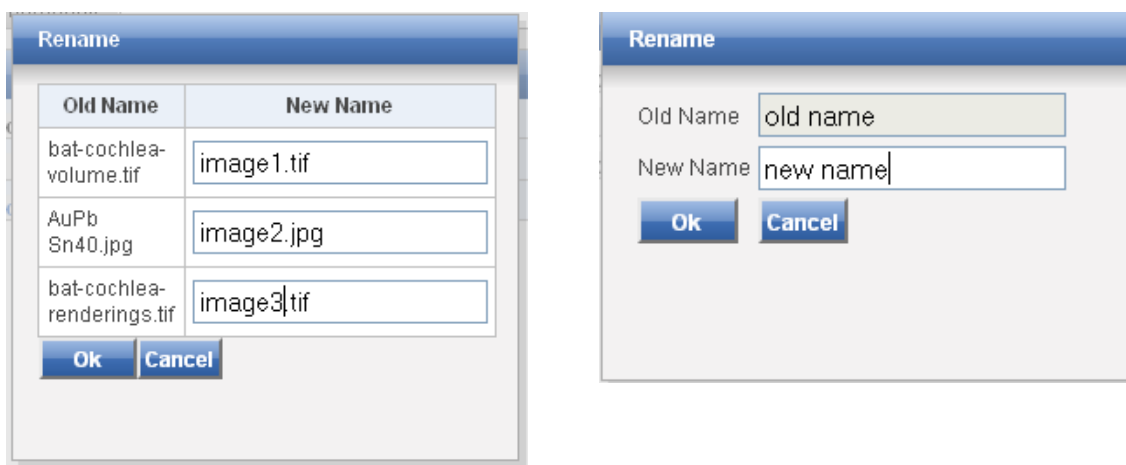
Lorsque l'on double clique sur le nom d'un dossier dans l'arborescence des dossiers ou dans la liste des fichiers et dossier, cela permet à l'utilisateur de modifier son nom. Si le nouveau nom du fichier ou dossier existe déjà dans le dossier parent alors à la fin du nom il faut ajouter un numéro. Par exemple si le nom 'stagiaire' est tape pour renommer un dossier et que ce no, de dossier existe déjà alors il sera remplacé par 'stagiaire2'.

Interface permettant de renommer un fichier ou un dossier



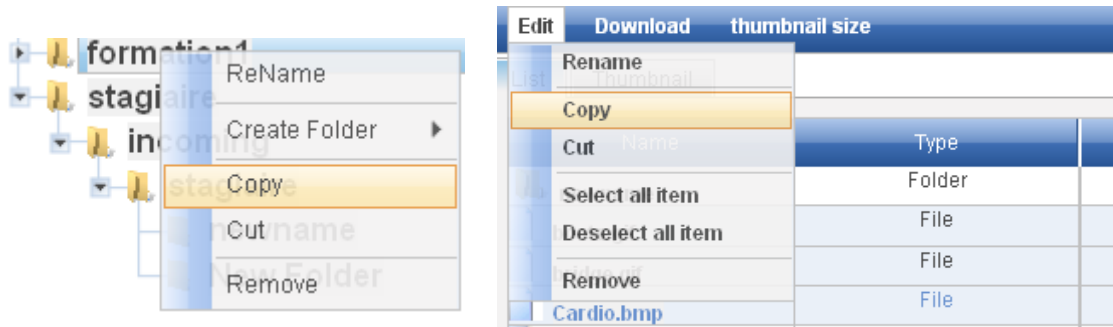
L'utilisateur peut aussi renommer un dossier ou un fichier en sélectionnant le dossier puis en cliquant dans le menu déroulant ou dans le menu contextuel sur 'rename'. A ce moment le nom du dossier est éditable à partir de page pop-up. Si l'utilisateur choisi plusieurs dossier ou fichiers, il peut les renommer en une fois.

Interface qui permet de renommer plusieurs dossiers et fichiers



7.5 Gestion de la copie et du déplacement d'un fichier ou d'un dossier

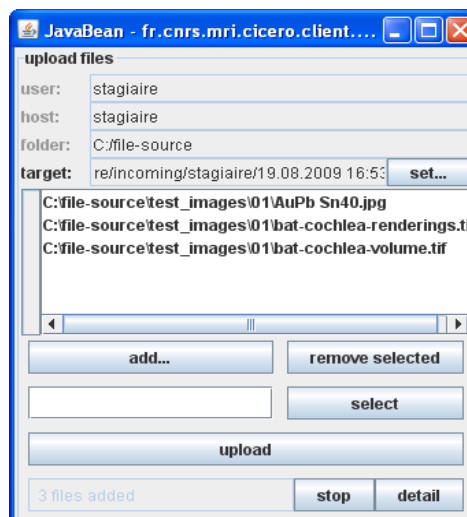
Interface de copie d'un fichier ou d'un dossier



Sur l'interface de gestion des dossiers, l'utilisateur peut copier ou déplacer un dossier à partir du menu déroulant ou du menu contextuel en sélectionnant le dossier qu'il veut copier puis en cliquant sur 'copie'. A ce moment l'utilisateur n'a plus qu'à sélectionner le dossier de destination puis il peut coller son dossier en cliquant sur 'coller'. S'il s'agit du déplacement le dossier source est supprimé. Si l'utilisateur essaie de copier un dossier ou un fichier dans un dossier où ce nom est déjà existant le système change le nom automatiquement avec un suffixe. Si un utilisateur n'a pas les droits nécessaires pour effectuer son opération un message échec doit apparaître.

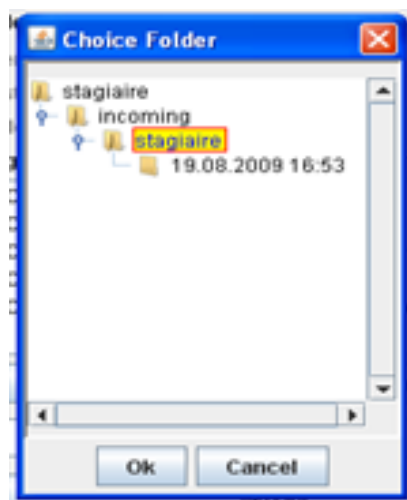
7.6 Gestion des fichiers de téléchargement du client

Interface du logiciel de téléchargement



La gestion du téléchargement des fichiers du client permet à l'utilisateur de choisir les fichiers qu'il veut télécharger sur le serveur. A ce moment la, l'application permet à l'utilisateur de choisir un chemin de destination en cliquant sur le bouton « set ». Une arborescence des dossiers virtuels de l'utilisateur connecté apparait. A partir de là, l'utilisateur sélectionne un des dossiers et valide son choix, Le chemin validé sera mis en mémoire et le téléchargement des fichiers se fera vers ce chemin.

Interface de modification du chemin de destination



7.7 Visualisation des aperçus

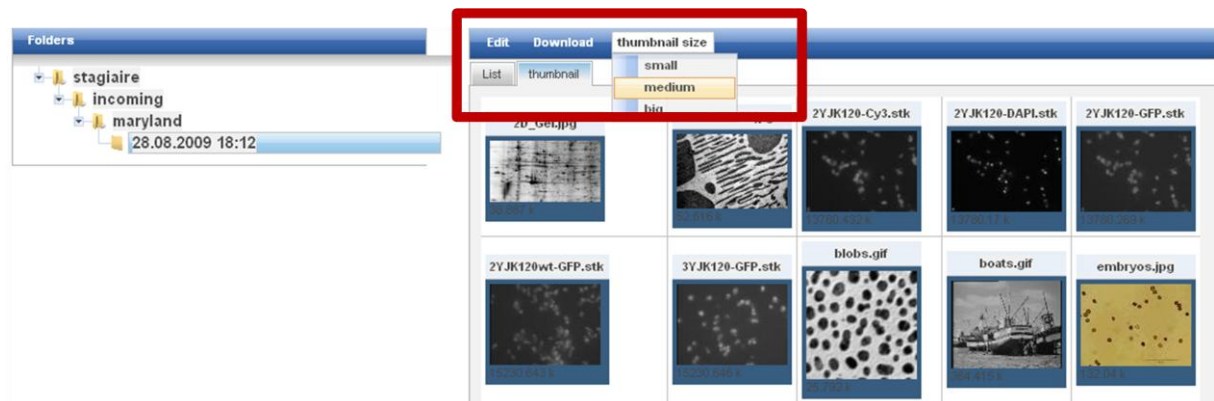
La visualisation de l'aperçu se fait par l'intermédiaire de l'interface web principale. Il suffit de sélectionner l'onglet 'Thumbnail' comme ci-dessous encadré en rouge.

Exemple des aperçus d'un dossier :



Ensuite il est possible pour l'utilisateur de modifier la taille des aperçus. Trois tailles lui sont proposées : petite, moyenne et grande. Pour modifier la taille des aperçus il lui suffit de la sélectionner à partir du menu déroulant comme ci-dessous encadré en rouge.

Changement de taille des aperçus :



8 Conclusion

Au final, la mission a été réalisée. Les fonctionnalités demandées ont été développées. L'élaboration de ce projet, même si elle a parfois été très difficile, a été bénéfique en tout point et je suis assez content du résultat obtenu malgré tous les problèmes survenus.

Les plus grandes difficultés rencontrées durant la mission sont dues à la découverte d'un langage et d'une conception que je ne connaissais pas auparavant. L'apprentissage du J2EE est très dur au début car il y'a beaucoup de concepts spécifiques à retenir. De plus il a été difficile de se familiariser rapidement au vocabulaire J2EE : par exemple il est très dur de savoir à quoi correspond un EJB juste en lisant ce mot 'Entreprise java Bean', il faut approfondir en lisant des articles et des cours pour comprendre le concept. Ensuite je n'avais jamais travaillé sur un serveur d'application, il m'a donc fallu apprendre comment il fonctionnait et comment l'utiliser. En arrivant au MRI il m'a été demandé de travailler sous l'environnement de développement Eclipse que je ne connaissais pas du tout car je travaillais avec le logiciel Netbeans en cours de java. Toutes ces difficultés cumulées les unes aux autres ne m'ont pas simplifié la tâche, c'est pourquoi les premières semaines ont été difficiles à appréhender mais elles ont surtout été véritablement très enrichissantes pour moi.

En plus j'ai eu la chance d'avoir intégré une équipe de développement fonctionnant avec la méthode Extrême Programming. Cette méthode m'a paru être très enrichissante car elle m'a permis de travailler avec une personne d'expérience et d'apprendre à ses côtés (avantage du Pair Programming). Elle a aussi été enrichissante et intéressante pour une équipe de travail car les développeurs expérimentés sont mélangés avec les moins expérimentés ce qui leur permet de transmettre leur savoir et pourquoi pas d'apprendre de ces derniers. De plus elle a mis en évidence le travail de l'équipe de développement car elle les a placée au centre des décisions.

Les objectifs d'un stage professionnel sont multiples : le stagiaire doit tirer une double expérience (immersion dans le monde du travail et acquisition de nouvelles connaissances sur les activités en entreprise) et pouvoir apporter à l'entreprise un bénéfice aussi bien sous forme de nouvelles compétences liées à sa formation qu'à sa personnalité.

Ce stage a été enrichissant, aussi bien au niveau humain que professionnel et sera un atout pour mon entrée dans la vie active. Effectivement, j'ai eu par l'intermédiaire de ce stage une expérience en J2EE et j'ai intégré un projet fonctionnant avec Extrême Programming, ce qui est pour moi une mine d'or à rajouter sur mon CV.

Ce qui est aussi enrichissant c'est de savoir que le travail effectué va servir dans l'application et être utilisé par de multiples utilisateurs. Il a permis au CNRS d'avancer sur le projet Cicéro et d'intégrer une première version dans l'environnement de production.

9 Bibliographie

- www.CNRS.fr

Site officiel du CNRS.

- www.developpez.com

Site internet contenant des cours pour l'apprentissage des langages de programmation comme le java et le J2EE.

- www.mri.fr

Site officiel du Montpellier RIO Imaging.

- Livre : J2EE aux éditions Eyrolles par j.Molière

Livre présentant des solutions de développement en J2EE.

- Présentation Extrême Programming

Document réalisé par un cabinet de consultant Optima Consulting présentant tous les concepts de cette méthodologie de gestion de projet.

10 Glossaire

Cytométrie : La cytométrie en flux est une technique permettant de faire défiler des particules, molécules ou cellules à grande vitesse dans le faisceau d'un laser.

Framework : Un Cadre d'applications (en anglais, Application Framework) est un ensemble de bibliothèques permettant le développement rapide d'applications. Ils fournissent suffisamment de briques logicielles pour pouvoir produire une application aboutie. Ces composants sont organisés pour être utilisés en interaction les uns avec les autres. Ils sont en principe spécialisés pour un type d'application.

J2EE : Java 2 Platform, Enterprise Edition est un framework pour le langage de programmation Java de Sun plus particulièrement destiné aux applications d'entreprise.

Application distribué : Ensemble compose d'éléments reliés par un système de communication ; les éléments ont des fonctions de traitements, de stockage, de relation avec le monde extérieur. Les différents éléments du système ne fonctionnent pas séparément.

Nanozoomer : Le nanozoomer est un appareil qui permet de faire de la microscopie virtuel. A partir d'une coupe il crée une image, il est possible d'utiliser cette image avec un logiciel informatique pour effectuer des zooms sur une partie. En réalité à partir de l'image il est possible d'utiliser un ordinateur comme un microscope, c'est ce qu'on appelle la microscopie virtuel.

11 Annexes

11.1 Annexe 1 : Découpage en tâche d'un scénario

Scénario : La gestion des dossiers

L'utilisateur de Cicéro arrive sur une page web, il est déjà connecté au système. Sur la page une arborescence de ses dossiers est présente. Lorsqu'il clique sur un de ses dossiers, un aperçu des fichiers apparaît dans une liste présente sur la droite de l'arborescence.

Tâche1 : Récupérer les dossiers virtuels

Pour permettre à un utilisateur de voir son arborescence, il faut avant tout créer un EJB qui va permettre en fonction de l'utilisateur de récupérer les dossiers et les fichiers virtuels.

L'EJB s'appellera donc VirtualFileSystemServices et contiendra la méthode:

- getVirtualFileSystemComponent () Cette méthode permet d'afficher tous les dossiers et tous les fichiers virtuels(même s'ils ne correspondent pas à cette utilisateur dans un premier temps).

Tâche 2 : Création de la page web

Ensuite il faut créer une page web avec un affichage de l'arborescence des dossiers virtuels. Pour cela nous devons créer une page JSP avec un jTreeView permettant l'affichage des dossiers

Dans cette partie, il faut faire appel à la méthode de l'EJB créé au dessus et donc d'afficher tous les dossiers et les fichiers virtuels. Sur cette page, on veut pouvoir afficher les dossiers virtuels dans une arborescence et on veut créer un bouton permettant de créer un dossier (ou de pouvoir faire un clique droit sur un dossier de l'arborescence puis de cliquer sur nouveau dossier dans le menu déroulant).

Tâche 3 : Création du dossier

Sur le clique du bouton, il faut pouvoir créer un dossier. Pour créer le dossier nous avons besoin de différentes informations:

- Nom du dossier.
- Le dossier parent.

Pour cela, il faut afficher à côté du bouton 'créer un dossier', une zone de saisie lui permettant d'indiquer le nom du dossier est créé. Il faut aussi récupérer le dossier parent, il sera sélectionné par l'utilisateur dans l'arborescence et si aucun dossier parent n'est sélectionné dans ce cas le dossier créer est un dossier racine.

Si l'utilisateur a utilisé la solution du clique droit sur un dossier alors un nouveau dossier est crée à l'endroit où le clique droit a été effectué. Ce dossier est éditable ce qui permet à l'utilisateur de donner le nom de son choix au dossier.

Ensuite l'utilisateur valide sa demande en cliquant sur le bouton et un appel est fait auprès de l'EJB VirtualFileSystemServices. Il faut donc ajouter une méthode à l'EJB permettant la création de ce dossier.

- AddVirtualFolder: Cette méthode permet donc la création d'un Virtual Folder. Elle reçoit comme paramètre l'identifiant du client ainsi que le nom du dossier à créer et l'identifiant du dossier parent.

11.2 Annexe 2 : Diagramme de classe du système de fichier virtuel

