



**HAL**  
open science

# Développement d'une application de monitoring et de gestion de parc informatique

Redouane Belkasmi

► **To cite this version:**

Redouane Belkasmi. Développement d'une application de monitoring et de gestion de parc informatique. Systèmes et contrôle [cs.SY]. 2015. dumas-01535649

**HAL Id: dumas-01535649**

**<https://dumas.ccsd.cnrs.fr/dumas-01535649>**

Submitted on 9 Jun 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**CONSERVATOIRE NATIONAL DES ARTS ET METIERS**  
**CENTRE REGIONAL DU MAROC**

---

MEMOIRE

Présenté en vue d'obtenir

**le DIPLOME D'INGENIEUR CNAM**

**SPECIALITE : Informatique**

**OPTION : Systèmes d'Information**

par

**BELKASMI Redouane**

**Développement d'une application de monitoring et de gestion de parc  
informatique**

Soutenu le : 11 Avril 2015

---

**Jury**

Président : **Kamel BARKAOUI**, Professeur des Universités, CNAM Paris

Membre : **Claude GENIER**, Ingénieur, enseignant EICNAM, Rhône-Alpes

**Karim CHEIKH**, Ingénieur en informatique, Casablanca

**ADIL ZEAARAOUI**, Responsable R&D et gérant, Mindware

**Mohamed HAJJI**, Ingénieur Réseau, Mindware



Claude Genier  
Enseignant CNAM



# *Dédicace*

*À mes chers parents,*

*Que nulle dédicace ne peut exprimer mes sincères sentiments,*

*Pour leur patience, leur encouragement, leur aide et pour leur grand sacrifice.*

*À ma chère sœur et à mes chers frères,*

*Pour leur grand amour et leur soutien, qu'ils trouvent ici l'expression de ma haute  
gratitude.*

*À toute l'équipe MindWare pour leur hospitalité et leur aide.*

*À tous mes amis, à toute personne, qui m'a prodiguée des encouragements pendant la  
période de ma formation.*

*Et à tous ceux qui me sont chers*

*Je vous dédie ce modeste travail en signe de reconnaissance.*

# Remerciement

La réalisation de ce mémoire de fin d'étude mené sur plusieurs mois de travail me procure une grande satisfaction mais aussi une grande gratitude pour les personnes qui m'ont beaucoup aidé pour surmonter toutes les difficultés et entraves que j'ai rencontré durant ma période de stage.

Je tiens spécialement à remercier M. Claude Genier mon encadrant pour tous ses conseils et directives qui m'étaient d'un très secours.

A M. Adil ZEAARAOUI mon tuteur au sein de MindWare, pour ses orientations et ses précieux conseils.

Un grand remerciement au CNAM qui ma offert une grande chance de finir mes études supérieures.

A tous mes professeurs et enseignants durant toute la période de ma formation depuis ma première années d'étude en l'année 1980 à l'école Salah-Eddine Alayoubi passant par toute école, lycée, institut et université.

# Table des Matières

<b>Contexte général du projet</b>	<b>9</b>
<b>Chapitre 1</b>	
<b>Etude de marché et fonctionnelle</b>	<b>11</b>
<b>1.1 Etude concurrentielle de marché</b>	<b>11</b>
<b>1.1.1 Solutions opensource :</b>	<b>11</b>
<b>1.1.2 Solutions commerciales</b>	<b>14</b>
<b>1.1.3 Critique des solutions</b>	<b>17</b>
<b>1.1.4 Solution proposée</b>	<b>18</b>
<b>1.2 Description de MindWare</b>	<b>18</b>
<b>1.2.1 Domaines d'intervention</b>	<b>19</b>
a) Edition des logiciels	19
b) ingénierie réseau et système	19
<b>1.3 Méthodologies du travail</b>	<b>19</b>
1.3.1 Analyse générale	19
1.1.1 Analyse détaillée	20
1.1.2 Formation sur les technologies	20
1.1.3 Conception du projet	20
1.1.4 Développement et tests	20
<b>1.4 Conclusion</b>	<b>21</b>
<b>Chapitre 2</b>	
<b>Analyse fonctionnelle détaillée du projet MindWare Remote, Monitoring and control (MRMC)</b>	<b>23</b>
<b>2.1 Première partie : Supervision de parc informatique</b>	<b>24</b>
<b>2.1.1 Gestion de l'inventaire</b>	<b>24</b>
<b>2.1.2 Gestion des alertes et erreurs</b>	<b>26</b>
<b>2.2 Deuxième partie : Gestion d'assistance</b>	<b>27</b>
<b>2.3 Troisième partie : Tableau de bord</b>	<b>30</b>
<b>2.4 Conclusion</b>	<b>31</b>
<b>Chapitre 3</b>	
<b>Modélisation du MRMC</b>	<b>33</b>
<b>3.1 Choix d'UML</b>	<b>33</b>
<b>3.2 Lescasd'utilisation(Usecase)</b>	<b>34</b>
3.2.1 Acteur client	35
a) Envoi de l'état de la machine	35
b) Téléchargement de l'agent client	35
c) Mise à jour de l'agent client	35
3.2.2 Acteur serveur	36
a) identification des machines (identify machine)	36

b)	Sauvegarde dans la base de données (store machine info into DB)	36
c)	Détection des alertes et des erreurs (detect warning and error)	36
3.2.3	Acteur Administrateur :	37
a)	Authentification (log in)	38
b)	Liste des machines (list machines)	38
c)	Etat de la machine cliente (client state)	38
d)	Afficher l'état de la machine cliente (get machine detail info)	39
e)	Extraire les informations de la machine (Extract information)	39
f)	Définir intervalle (Set timer)	39
g)	Télécharger l'agent vers le client (upload new jar)	39
h)	Liste des alertes et erreurs des machines (list machine warning error)	39
<b>3.3</b>	<b>Diagrammes de séquences</b>	<b>40</b>
3.3.1	Client	40
Scénario1		40
Scénario2		40
3.3.2	Serveur	41
3.3.3	Administrateur	42
<b>3.4</b>	<b>Diagramme des classes</b>	<b>43</b>
<b>3.4.1</b>	<b>package « machine »</b>	<b>43</b>
<b>3.4.2</b>	<b>Package surveillance</b>	<b>45</b>
<b>3.4.3</b>	<b>Package assistance « helpDesk »</b>	<b>48</b>
<b>3.5</b>	<b>Conclusion</b>	<b>49</b>
<b>Chapitre 4</b>		
<b>Etude technique du projet</b>		<b>51</b>
<b>4.1</b>	<b>Outils, langages et technologies utilisés</b>	<b>51</b>
4.1.1	Plateforme de développement : JEE	52
4.1.2	Editeur de développement : Eclipse	52
4.1.3	Le serveur d'application Tomcat	53
4.1.4	Le langage JAVA	54
4.1.5	La technologie Java Server Pages (JSP)	54
4.1.6	Les Servlet	54
4.1.7	Le Framework Struts2	55
4.1.8	JPMapper	55
4.1.9	La bibliothèque SIGAR	56
4.1.10	Le langage XML	58
4.1.11	Dom4j	61
4.1.12	Le langage XPATH	61
Exemple :		61
4.1.13	Le langage JavaScript	62
La technologie AJAX		62
La bibliothèque JQuery		63
La librairie jqPlot		63
4.1.14	Bibliothèque bureau à distance (jrDesktop)	63
4.1.15	Méthodologie du codage : Technique MVC	64
4.1.16	SVN	65
<b>4.2</b>	<b>Utilisation des technologies dans MRMC</b>	<b>65</b>
<b>4.2.1</b>	<b>Le dépôt SVN</b>	<b>65</b>

4.2.2	Arborescence du projet Agent/JEE/ MVC	66
4.2.3	Agent MRMC	68
4.2.4	Console d'administration web	70
4.3	Développement de la solution MRMC	73
4.3.1	Développement de l'agent MRMC	73
4.4	Conclusion	83
<b>Chapitre 5</b>		
<b>Présentation de MRMC</b>		<b>85</b>
5.1	Présentation de l'agent	85
5.1.1	Installation	85
5.1.2	Demande d'assistance	87
5.2	Présentation de la Console d'administration	89
5.2.1	Authentification	89
5.2.2	Supervision du parc	90
5.2.3	Paramétrage des alertes et erreurs	96
5.2.4	Gestion de l'assistance aux utilisateurs	98
5.2.5	Tableau de bord	99
5.3	Critique et perspective de la solution	100
5.3.1	Fonctionnalités de l'agent	100
5.3.2	Fonctionnalités de la console d'administration	102
5.4	Conclusion	103
<b>Chapitre 6</b>		
<b>Conclusion générale</b>		<b>105</b>
<b>Références</b>		<b>107</b>
<b>Annexes</b>		<b>109</b>
<b>Struts2</b>		<b>109</b>
	Installation et Configuration	109
	Le fichier xml Struts.xml	111
	Les actions	111
<b>JPMapper</b>		<b>114</b>
	Installation et Configuration	114
	Remarque:	116
<b>Dom4j</b>		<b>116</b>
	Installation et Configuration	116
<b>MySQL</b>		<b>117</b>
<b>Enterprise Architect</b>		<b>117</b>



# Contexte général du projet

## 1. Introduction

Depuis le développement de l'informatisation des entreprises, la question de la sécurité et de la robustesse du système d'information est au cœur des préoccupations des administrateurs réseau et système. Ces craintes sont à l'origine de la création et du succès des outils de surveillance. Les logiciels de supervision permettant de faciliter la gestion des parcs informatiques deviennent de plus en plus complexes. Leur but principal est la collecte de données et la présentation de celles-ci pour que l'administrateur puisse consulter aisément les écrans de contrôle pour mesurer l'état du réseau. Le travail de ce stage se focalise sur le cadre du sujet de monitoring (surveillance) réseau et le contrôle à distance (remote control) il vise :

- Le contrôle et l'utilisation à distance des ressources des différentes machines du parc.
- L'optimisation de l'utilisation des ressources dans différentes composantes (Mémoires, les médias, CPU) tout en gardant une bonne exploitation des machines du parc.
- La diminution du temps réaction lors des maintenances correctives.
- La favorisation de la maintenance préventive.
- La publication des rapports d'état en temps réel.

Mon stage est réalisé au sein de la société MindWare, mon travail consistait à réaliser la partie console d'administration relatives aux machines d'un parc informatique.

Ce rapport sera organisé comme suit :

- Chapitre 1 : Etude de marché et fonctionnelle.
- Chapitre 2 : Analyse fonctionnelle détaillée du projet MindWare Remote, Monitoring and control (MRMC).
- Chapitre 3 : Modélisation du MRMC.
- Chapitre 4 : Etude technique du projet.
- Chapitre 5 : Présentation de MRMC.
- Chapitre 6 : Conclusion générale.

## **2. Problématique**

Pour toute entreprise, la stabilité de son système informatique est considérée comme une brique importante et obligatoire pour son développement. Cette stabilité impose à l'entreprise d'importants investissements matériels et humains ce qui n'est pas toujours disponible et possible pour toute société.

Dès lors, l'utilisation d'une solution logicielle permettant d'optimiser la réactivité des équipes techniques de maintenance devient très demandée.

Pour répondre à cette problématique et malgré l'existence de logiciel libre et propriétaire qui permettent de réaliser la surveillance (monitoring) réseau et/ou le control distant, MindWare a opté pour le développement d'une solution réalisant les deux fonctionnalités en même temps. Cette double fonctionnalité optimisera le temps de réactivité et la durée des interventions de maintenance des équipes techniques.

Aussi, la réalisation de tel logiciel entre dans une perspective générale visée par MindWare pour le développement d'un ERP intégrant un ensemble de sous application de gestion des entreprises. En finalité, Cette application constituera le module responsable de la gestion du parc informatique dans une entreprise.

## **3. Objectifs**

Ce travail compte répondre à trois objectifs principaux qui sont:

- La conception d'une solution qui offre un service de surveillance d'un parc informatique.
- La génération des rapports d'état et des graphes qui décrivent les états des composantes de chaque machines du parc.
- Réalisation d'une application réalisant la surveillance et le contrôle distant intégré dans une seule console d'administration.

# Chapitre 1

## Etude de marché et fonctionnelle

Dans ce chapitre, j'essaierai de réaliser une étude de marché des solutions les plus utilisées et dont leurs fonctionnalités se rapprochent des objectifs de MRMC. En suite j'entamerai une étude fonctionnelle qui listera les différentes tâches que MRMC intégrera.

### 1.1 Etude concurrentielle de marché

Les solutions de surveillance et de contrôle distant peuvent être classées en deux groupes, opensource et propriétaire.

#### 1.1.1 Solutions opensource :

« Open source » désigne un logiciel dans lequel le code source est à la disposition du grand public, et c'est généralement un effort de collaboration où les programmeurs améliorent ensemble le code source et partagent les changements au sein de la communauté ainsi que d'autres membres peuvent contribuer.<sup>1</sup>

Il existe une multitude de produit opensource spécialisé dans la gestion de parc informatique, on peut citer parmi eux :

---

<sup>1</sup> [http://fr.wikipedia.org/wiki/Open\\_source](http://fr.wikipedia.org/wiki/Open_source)

#### a) GLPI

GLPI (gestionnaire libre de parc informatique) est une application web permettant la gestion de parc informatique et de gestion des services d'assistance distribué sous licence GPL. Le projet totalement communautaire a été lancé en 2003 par l'association « Indepnet ».<sup>2</sup>

GLPI offre plusieurs services de base aux administrateurs système et réseau :

- Gestion et suivi des ressources informatiques
- Gestion et suivi des licences
- Gestion et suivi des consommables
- Base de connaissances
- Gestion des réservations
- ServiceDesk (helpdesk, SLA..)
- Inventaire automatisé\*
- Télé déploiement
- ...

En plus de ces fonctionnalités, GLPI dispose d'un ensemble de plugins permettant de couvrir un large éventail de tâches des équipes de maintenance.<sup>3</sup>

#### b) OCS Inventory

OCS Inventory NG soit Open Computer and Software Inventory est une application permettant de réaliser un inventaire sur la configuration matérielle des machines du réseau et sur les logiciels qui y sont installés et de visualiser ces informations grâce à une interface web. Il comporte également la possibilité de télé-déployer des applications sur un ensemble de machines selon des critères de recherche.<sup>4</sup>

En plus, Ocs Inventory peut être couplé avec GLPI pour étendre ses fonctionnalités.

#### c) ZABBIX

ZABBIX est un logiciel libre permettant de surveiller l'état de divers services réseau, serveurs et autres matériels réseau et produisant des graphiques dynamiques de consommation des ressources. C'est un logiciel open source créé par Alexei Vladishev.<sup>5</sup>

Zabbix offre un très grand nombre de fonctionnalités regroupées en domaine d'activité (Industrie, banque et finance, médical, aéronautique...), il est caractérisé par :

---

<sup>2</sup> [http://fr.wikipedia.org/wiki/Gestion\\_libre\\_de\\_parc\\_informatique](http://fr.wikipedia.org/wiki/Gestion_libre_de_parc_informatique)

<sup>3</sup> <http://plugins.glpi-project.org/spip.php?mot29&lang=fr>

<sup>4</sup> [http://fr.wikipedia.org/wiki/OCS\\_Inventory](http://fr.wikipedia.org/wiki/OCS_Inventory)

<sup>5</sup> <http://fr.wikipedia.org/wiki/Zabbix>

- Une supervision répartie avec une administration web centralisée, afin que la récolte des données ne soit pas interrompue en cas de problème réseau,
- Une supervision de site web avec recherche de motif et scénarios de navigation,
- Un fonctionnement en actif et/ou passif, le serveur Zabbix va récupérer les données via des agents sur les éléments supervisés, et/ou, ce sont les éléments supervisés qui vont envoyer au serveur Zabbix les données,
- Une interface web pour une visualisation efficace de l'état des éléments réseaux et des données récoltées,
- La notification par e-mail, messagerie instantanée, sms, et pratiquement n'importe quel autre moyen, pour être informé rapidement lors qu'un problème apparaît,
- Le calcul de SLA, pour avoir une vue "commerciale" des services,
- Une supervision sans agent, par snmp, par ssh ou encore par ipmi, pour les serveurs ou éléments réseaux ne permettant pas l'installation de l'agent zabbix,
- Un agent haute performance en natif (Logiciel client pour Linux, Solaris, HP-UX, AIX, Free BSD, Open BSD, OS X, Tru64/OSF1, Windows NT4.0, Windows 2000, Windows 2003, Windows XP, Windows Vista)
- Une supervision proactive, pour une interaction forte avec tous les équipements (relance de services, extinction, reboot ...)
- La découverte automatique des serveurs et périphériques réseaux, pour surveiller les nouveaux serveurs dès que ceux-ci sont présents sur le réseau

#### d) Nagios

Nagios est le standard mondial en ce qui concerne la surveillance système et réseau. Par son architecture modulaire il permet de lancer des alertes lorsque les systèmes ont des dysfonctionnements et quand ils repassent en fonctionnement normal. C'est un logiciel libre sous licence GPL.<sup>6</sup>

Nagios couvre l'ensemble des fonctionnalités demandées par les administrateurs réseau, parmi lesquelles<sup>7</sup>:

- Surveillance des services réseaux (SMTP, POP3, HTTP, NNTP, PING, etc.)
- Surveillance des ressources des hôtes (charge processeur, utilisation des disques, etc.)
- Système simple de plugins permettant aux utilisateurs de développer facilement leurs propres vérifications de services.
- Parallélisations de la vérification des services.
- Possibilité de définir la hiérarchie du réseau en utilisant des hôtes "parents", ce qui permet la détection et la distinction entre les hôtes qui sont à l'arrêt et ceux qui sont injoignables.
- Notifications des contacts quand un hôte ou un service a un problème et quand celui-ci est résolu (via email, pager, sms, ou par tout autre méthode définie par l'utilisateur)

<sup>6</sup> <http://fr.wikipedia.org/wiki/Nagios>

<sup>7</sup> [http://nagios.proy.org/traduction/wiki\\_preview/about.html](http://nagios.proy.org/traduction/wiki_preview/about.html)

- Possibilité de définir des gestionnaires d'évènements qui s'exécutent pour des évènements sur des hôtes ou des services, pour une résolution pro-active des problèmes
- Rotation automatique des fichiers log
- Support pour l'implémentation de la surveillance redondante des hôtes
- Interface web optionnelle, pour voir l'état actuel du réseau, notification et historique des problèmes, fichiers log, etc.

#### e) openNMS

OpenNMS est une application open source développée en JAVA permettant la supervision de réseau informatique. Elle utilise le moteur d'application JETTY et PostgreSQL comme système de gestion de base de données.

Les principales fonctionnalités d'OpenNMS sont les suivantes :

- Découvrir les équipements réseaux à superviser (ping).
- Découvrir les services présents sur un équipement et en mesurer la disponibilité.
- Identifier et lister les interruptions de services réseaux (outages).
- Collecter les informations et recevoir les alarmes provenant des équipements supervisés via le protocole SNMP.
- D'enrichir les informations d'un évènement par des données stockées dans la base de données.
- D'effectuer une corrélation entre les alarmes afin de présenter un affichage clair des problèmes en cours.
- De corréler, notifier et escalader ces évènements sous forme d'alarmes.
- De disposer d'une interface Web permettant d'administrer et de superviser.
- De réaliser des graphiques à partir de polling SNMP.
- De représenter graphiquement les équipements supervisés.

### 1.1.2 Solutions commerciales

Les solutions commerciales sont des produits vendus par de grandes sociétés dont la plus part sont des multinationales. L'ensemble de ces logiciels sont payants et leurs prix sont calculés en fonction du nombre d'utilisateur et/ou du nombre d'équipement réseau à surveiller.

En plus de leurs prix, l'utilisation de ces produits demande dans la plus part du temps une formation spéciale et une bonne expertise dans le domaine de supervision ce qui constitue un investissement important pour les petites et moyennes entreprises.

#### a) HP OpenView

HP OpenView n'est pas un seul produit mais un ensemble de logiciels spécialisés dans la supervision de réseau et système ou même la surveillance de logiciel spécifique. OpenView a

été développé de façon très modulaire ce qui lui confère une très grande souplesse d'utilisation. Les sous logiciel OpenView peuvent être classés par domaine d'utilisation<sup>8</sup> :

### 1) Supervision de réseau et système

Composants utilisés pour la supervision globale d'un réseau. Cette tâche est assurée par **HP OpenView Network Node Manager (NNM)** qui fournit tous les services de base pour la gestion de la plate-forme réseau. Il permet de connaître la topologie du réseau et la présenter sous forme graphique. Pour la supervision des systèmes d'exploitation, **HP OpenView Operations Manager** est utilisé.

### 2) Analyse des performances

Ces outils sont utilisés pour avoir une vue d'ensemble des performances des équipements réseau du parc informatique. Par le biais de graphique et de rapport très riche d'indicateur, **HP OpenView Performance Manager, Agent, Monitor** sont combinés pour avoir une solution puissante et flexible de supervision et d'analyse de performance.

### 3) Supervision verticale

Ces outils sont utilisés pour la surveillance de produits tiers spécifiques. Par l'intermédiaire de plugins, HP OpenView peut être utilisé pour la supervision d'une multitude d'application verticale comme :

- HP OpenView SPI pour BEA Tuxedo
- HP OpenView SPI pour BEA WebLogic
- HP OpenView SPI pour BEA WebLogic Integration
- HP OpenView SPI pour base de données (Oracle, Microsoft SQL Server, Sybase, and Informix)
- HP OpenView SPI pour IBM WebSphere Application Server

### b) IBM Tivoli

Avant de devenir une branche de la société IBM Tivoli system est créée à Austin en 1989. En mars 1996 elle est rachetée par IBM pour un montant de 750 million de dollars et devient du même coup la filiale IBM dédiée au développement de son logiciel de supervision et administration réseau, de système et d'application.<sup>9</sup>

---

<sup>8</sup> [http://en.wikipedia.org/wiki/HP\\_OpenView](http://en.wikipedia.org/wiki/HP_OpenView)

<sup>9</sup> <https://prezi.com/mf8isjestdoj/presentation-ibm-tivoli/>

A l'instar de HP OpenView IBM Tivoli est subdivisé en sous applications en fonction de leur domaine d'utilisation.<sup>10</sup>

1) Performance, Disponibilité, Configuration et Opération

Tivoli Monitoring, Tivoli Enterprise Console, Tivoli Netview, Tivoli Workload Scheduler, Tivoli Configuration Manager, Tivoli Remote Control

2) Administration du stockage

Tivoli Storage Manager, Tivoli Storage Manager for Space Management (HSM), Tivoli Storage Resource Manager, Tivoli Storage Area Network Manager

3) Sécurité

Tivoli Identity Manager, Tivoli Access Manager, Tivoli Privacy Manager, Tivoli Risk Manager II.4 -Tivoli Enterprise Console (TEC) The Tivoli Enterprise Console® (TEC) L'intégration de l'offre Service Desk fournit une communication bidirectionnelle entre IBM® Tivoli Enterprise Console (TEC) et le composant Service Desk d'IBM Tivoli Service Request Manager® 7.1. Cette section fournit une brève description du produit IBM Tivoli Enterprise Console, les applications Service Desk qui intègrent avec TEC, ainsi que les avantages de l'intégration TEC et la fonctionnalité Service Desk.

c) VMware vRealize Hyperic

vRealize Hyperic est un composant de VMware vRealize Operations qui à son tour est un composant de VMware vRealize Suite<sup>11</sup>. Il est utilisé pour la supervision des systèmes d'exploitation, les équipements réseau et les applications tiers.

Son principal avantage est sa capacité unique d'automatiquement découvrir et contrôler le logiciel et les ressources de réseau, sans tenir compte du type de machine ou de l'endroit. Ce qui donne une vue unifiée de la performance et l'état de performance de l'intégralité du parc informatique.<sup>12</sup>

vRealize Hyperic permet de superviser une très grande variété de ressources :

- OS.
- serveurs web et proxys.

---

<sup>10</sup> <https://prezi.com/mf8isjestdoj/presentation-ibm-tivoli/>

<sup>11</sup> <http://www.vmware.com/products/>

<sup>12</sup> <http://wiki.monitoring-fr.org/various/hyperic-ubuntu-install>

- serveurs d'applications.
- bases de données.
- technologies Microsoft (AD, Exchange, .NET).
- produits de virtualisation.
- ressources réseau.
- et bien d'autres comme Alfresco par exemple.

### 1.1.3 Critique des solutions

Tout logiciel de supervision doit être capable de surveiller des centaines voire des milliers de serveurs et de services. Il doit être aussi facile à installer, simple à utiliser et pouvoir donner une image concrète et fiable de l'état du parc informatique.

Plusieurs critères de comparaison peuvent être définis, mais toute tentative de comparaison doit prendre en compte la stabilité, l'accessibilité, la fiabilité et la pertinence des indicateurs de performance, et le niveau de support de la solution.

Ci-dessous un tableau comparatif dont les indicateurs sont notés de 1 à 5 :

Logiciel / Critère	GLPI	OCS Inventory	ZABBIX	Nagios	OpenNMS	OpenView	Tivoli	Heperic
Facilité de configuration	4	4	5	3	5	4	4	5
Ergonomie de l'interface web	3	3	4	3	5	4	5	5
Multiplateforme	3	3	4	4	5	5	5	5
Mapping réseau	2	2	4	4	4	5	5	5
Niveau d'automatisation	2	2	4	3	4	4	4	5
Génération de rapport	3	3	4	3	4	5	5	5
Niveau des graphiques	2	2	4	3	4	4	4	5
Ticket Helpdesk	3	3	4	4	4	5	5	5
Contrôle à distance	3	3	4	4	4	5	5	5
Support et expertise	4	4	4	4	5	5	5	5
Supervision d'application tierce	3	2	4	4	4	5	5	5
Notification par email ou sms	4	3	4	3	4	5	5	5
Configuration des alertes	2	2	4	3	4	5	5	5
<b>Total</b>	<b>38</b>	<b>36</b>	<b>53</b>	<b>45</b>	<b>56</b>	<b>61</b>	<b>57</b>	<b>65</b>

### 1.1.4 Solution proposée

Dans le domaine de surveillance de parc informatique les niveaux de performance des solutions ci-dessus citées sont assez rapprochés. Toute fois, les logiciels libres ont acquis une notoriété et une stabilité indiscutable.

Pour cette raison, La solution envisagée par Mindware se caractérise sa licence opensource et sa facilité d'installation et d'utilisation. Pour sa première version, elle sera destinée aux petites et moyennes entreprises et proposera dans sa version de base une fonctionnalité Helpdesk avec possibilité d'utilisation d'outil de contrôle à distance

Avant de commencer la description détaillée de la solution, un petit aperçu sur la société hôte de mon stage s'impose.

## 1.2 Description de MindWare



Figure 1.1: logo MindWare

Mon stage de fin d'étude a été effectué au sein de la société MindWare qui a été créée en Avril 2010 par M. Adil ZEAARAOUI en association avec l'université Mohammed 1er, MindWare est la première start-up créée par la dite université. MindWare siège à l'intérieur même du complexe universitaire et plus précisément en plein centre de l'Ecole Nationale des Sciences Appliquées d'Oujda (ENSAO).

Ce statut confère à MindWare le soutien et le parrainage de l'université Mohammed 1er et lui offre une place privilégiée tout prêt du corps enseignant, des étudiants et des lauréats de l'ENSAO.

## **1.2.1 Domaines d'intervention**

MindWare offre une gamme de services très variés et axés sur deux pôles :

### **a) Edition des logiciels**

MindWare offre à ses clients l'opportunité de créer des applications spécifiques répondant exactement à leurs besoins et à leurs règles de gestion. Aussi, MindWare propose à ses clients la maintenance et la mise à niveau de leurs applications de gestion déjà en production. Les technologies utilisées par MindWare s'étale sur un large spectre de choix : Technologie : OOA/D, JEE et les Framework (Struts2, JPmapper, hebernate...), PHP5 et Framework (Zend FM), Python, XML, JQuery, Ajax SGBD : Oracle, MySQL, Postgresql

### **b) ingénierie réseau et système**

MindWare propose à ses clients l'installation et la configuration de leurs réseaux mono site et multi site. Pour cela, MindWare couvre un large spectre de technologies : Réseaux et configuration : routeur, passerelle, firewall, proxy, contrôleur de domaine, monitoring, messagerie, VPN, serveur d'impression Solution VoIP: on travaille sur Asterisk Plateforme: linux/Unix (Debian/FreeBSD), Windows.

## **1.3 Méthodologies du travail**

La technique suivie lors de l'élaboration de n'importe quel projet au sein de MindWare est une technique basée le travail en groupe et inspirée de la méthodologie UP (Unified Process) ou itérative incrémentale qui consiste à diviser le projet en parties et pour chaque partie on effectue les étapes suivantes :

- Analyse générale.
- formation sur les technologies.
- Analyse détaillée.
- Conception du projet.
- Réalisation des tests.
- Validation.

### **1.3.1 Analyse générale**

Consiste à comprendre le projet dans l'ensemble en réalisant une étude préliminaire en se focalisant sur les demandes des clients et faire la collecte des informations. Dans cette étape, une analyse du marché avec une comparaisant des différentes solutions existantes est

réalisée. A l'issue de cette étape une idée générale prend forme et qui constituera la base de tout le projet.

### **1.1.1 Analyse détaillée**

Cette étape se base sur les exigences du client pour une bonne implémentation du cahier des charges avec génération des cas d'utilisation. Dans cette étape une étude technique des technologies et outils à utiliser est réalisée. Cette étape est très importante pour le bon déroulement des étapes suivantes c'est pour cela que MindWare lui alloue le délai et les ressources nécessaires.

A la fin de cette étape, l'équipe de développement est constituée et les tâches sont réparties entre ses membres.

### **1.1.2 Formation sur les technologies**

Après la constitution de l'équipe de développement, une formation est réalisée par le chef de projet aux nouveaux membres. Cette formation a comme objet les technologies choisies lors de l'étape précédente et qui seront utilisées dans le développement de la solution.

Le délai imparti à cette étape est proportionnel à la difficulté et la complexité des outils à utiliser et en fonction la nouveauté de ces outils pour les membres de l'équipe de projet.

### **1.1.3 Conception du projet**

En utilisant la méthodologie UML une analyse conceptuelle est réalisée avec la participation de toute l'équipe de projet. En utilisant le logiciel « Entreprise Architect », Dans cette étape les différents scénarios sont catalogués et modélisés sous forme de diagrammes de séquences. En suite, le diagramme de classes est élaboré et affiné à plusieurs reprises. En fin et après une multitude de réunions le chef de projet valide les différents diagrammes un planning de développement est mis en place.

### **1.1.4 Développement et tests**

Le développement de l'application est réparti par le chef de projet sur l'équipe de développement et suit le planning dressé dans la phase précédente. A la fin de chaque semaine une réunion est faite et chaque membre de l'équipe présente sa réalisation.

A la fin de la réunion les tâches correctement réalisées sont validées par le chef de projet et celles présentant des problèmes reportées à la réunion suivante.

Les tâches validées sont affecté à l'équipe chargée des tests et sont validées ou retournées à nouveau à l'équipe de développement en fonction de leur degrés de réponse aux exigences fonctionnelles.

## **1.4 Conclusion**

Dans ce chapitre j'ai présenté MindWare et ses domaines d'intervention ainsi que sa méthodologie de travail avec la réalisation d'une étude de marché comparant un ensemble de solution dont MRMC devra s'inspirer. Dans le chapitre suivant je présenterai une analyse fonctionnelle de la solution proposée.



# Chapitre 2

## **Analyse fonctionnelle détaillée du projet MindWare Remote, Monitoring and control (MRMC)**

MRMC est un logiciel client/serveur full web qui est en cours de réalisation au sein de MindWare pour réaliser la surveillance et le control distant d'un réseau.

Cette application sera intégrée en finalité dans un ERP qui englobe un ensemble de module. MindWare espère par le biais de cet ERP de couvrir l'ensemble ou du moins une grande partie des processus de gestion d'une entreprise. Les modules en cours de développement sont :

- Gestion de parc informatique (MRMC).
- Gestion de messagerie électronique.
- Gestion comptable.
- Gestion commerciale.
- Gestion électronique de document.
- Gestion de courrier.
- Gestion de clientèle.

Dans ce chapitre j'essaierai d'expliquer, indépendamment des technologies utilisées, les fonctionnalités que MindWare envisage intégrer dans MRMC. Pour cette version initiale, ces fonctionnalités sont divisées en trois grandes parties.

## 2.1 Première partie : Supervision de parc informatique

Cette partie du projet consiste en la surveillance des machines et équipements installés dans un réseau. Cette surveillance est divisée en deux parties :

### 2.1.1 Gestion de l'inventaire

L'inventaire des machines et de leurs composants : type du processeur, taille de la mémoire, capacité des disques durs, les cartes réseau, les processus en cours d'exécution, etc.

Cet inventaire devra afficher la liste des machines surveillées sous forme de tableau avec pagination. Comme illustré dans la maquette suivante, ce tableau devra afficher les colonnes suivantes :

Environnement

Erreur

Warning

UserName [Disconnect](#)

la zone bleu indique le pourcentage des machines qui presentent des erreurs(warning)

Home

List Machine

Default Setting

Specific Setting

Contact

Search Machine

Search

Parc

Name	Adresse IP	System	ID Machine	status
<a href="#">xxxxx</a>	<a href="#">0.0.0.0</a>	yyyyy	000000	<input checked="" type="radio"/> connect
<a href="#">yyyyy</a>	<a href="#">1.1.1.1</a>	yyyyy	111111	<input type="radio"/> disconnect

copyright @ 2011

dans le cas connecté le voyant sera vert il sera rouge sinn

chaque champ est un lien qui permet d'afficher la liste détaillée de la machine en question

Figure 2.1 : Liste des machines surveillées

En cliquant sur l'adresse IP de la machine, ou en le choisissant dans la zone recherche machine, les informations techniques de l'équipement supervisé seront affichées. Cette fiche technique détaillera les la configuration du CPU, RAM, disques durs, carte réseau ainsi que d'autres composants comme illustré dans la maquette suivante :

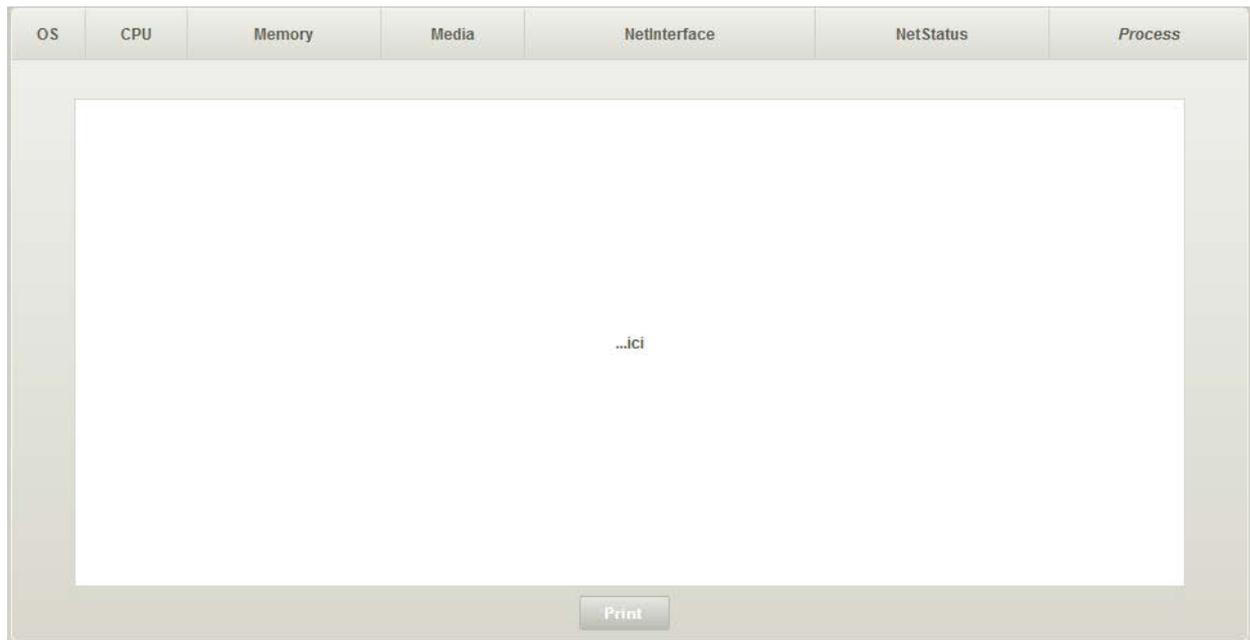


Figure 2.2 : Fiche équipement surveillé

Dans cette fiche, deux types de d’affichage peuvent être utilisés : la disposition tableau pour les données de type chaîne de caractères (système d’exploitation installé, type du processeur, processus monté, ...) et sous forme de graphique pour les informations de type numérique (mémoire, taille du disque dur, ...)

 A screenshot of a monitoring interface showing a table view. The top menu is identical to Figure 2.2. The main content area contains a table with the following data:
 

Description	Microsoft Windows 7
Name	Win32
Vendor	Microsoft

 Below the table, there is a section labeled "Specific Setting" which is currently empty.

Figure 2.3 : Forme d’affichage tabulaire

CPU de la machine : 192.198.52.24

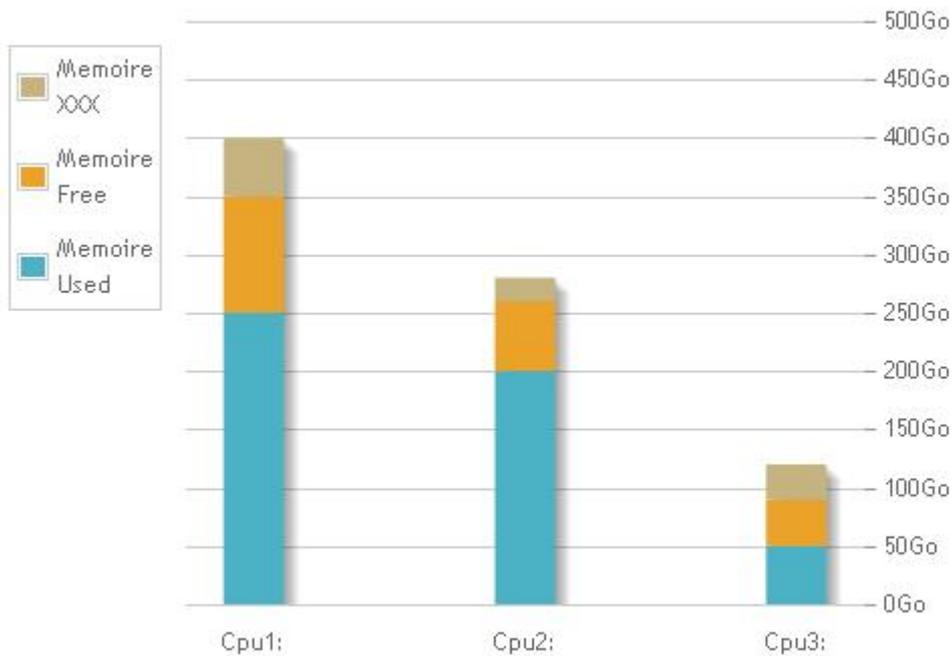


Figure 2.4 : Forme d'affichage graphique

### 2.1.2 Gestion des alertes et erreurs

La deuxième partie consiste en la gestion des alertes et erreurs des machines surveillées. Ces alertes et erreurs doivent être configurées au préalable pour qu'elles soient remontées dans l'interface d'administration. Plusieurs types d'alertes peuvent être suivies, prenons comme exemple les alertes de type disque dur. En effet, MRMC devrait en principe pouvoir détecter les alertes de type seuil d'espace disque atteint comme les alertes de type présence de secteur défectueux.

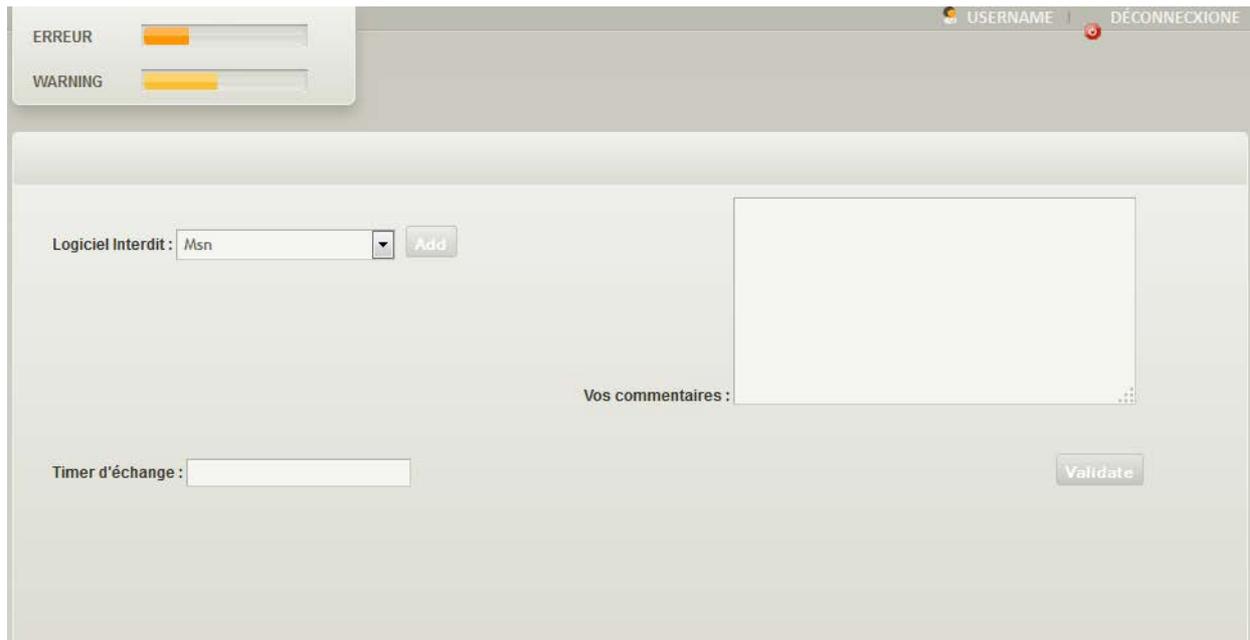


Figure 2.5 : Configuration des alertes et erreurs

## 2.2 Deuxième partie : Gestion d'assistance

La gestion des incidents dans toute entreprise est un composant primordial dans la stratégie des équipes techniques. En effet, plus la taille du parc informatique est grande plus sa gestion est complexe et plus la satisfaction des demandes d'assistance devient difficile.

MRMC être capable de générer des demandes assistance automatique basées sur les alertes et erreurs détectées et remontées dans la phase supervision, ainsi que la création des demandes d'assistance par les utilisateurs des machines et destinées à l'équipe technique.

La satisfaction des demande d'assistance pourra être réalisée soit par conférence audio, soit par chat, soit par prise de contrôle à distance ou dans le cas échéant en se déplacement vers le lieu d'incident

MindWare envisage dans un premier niveau intégrer une gestion de ticket de demande d'assistance destiné aux techniciens et dont le suivi est automatisé par l'application.

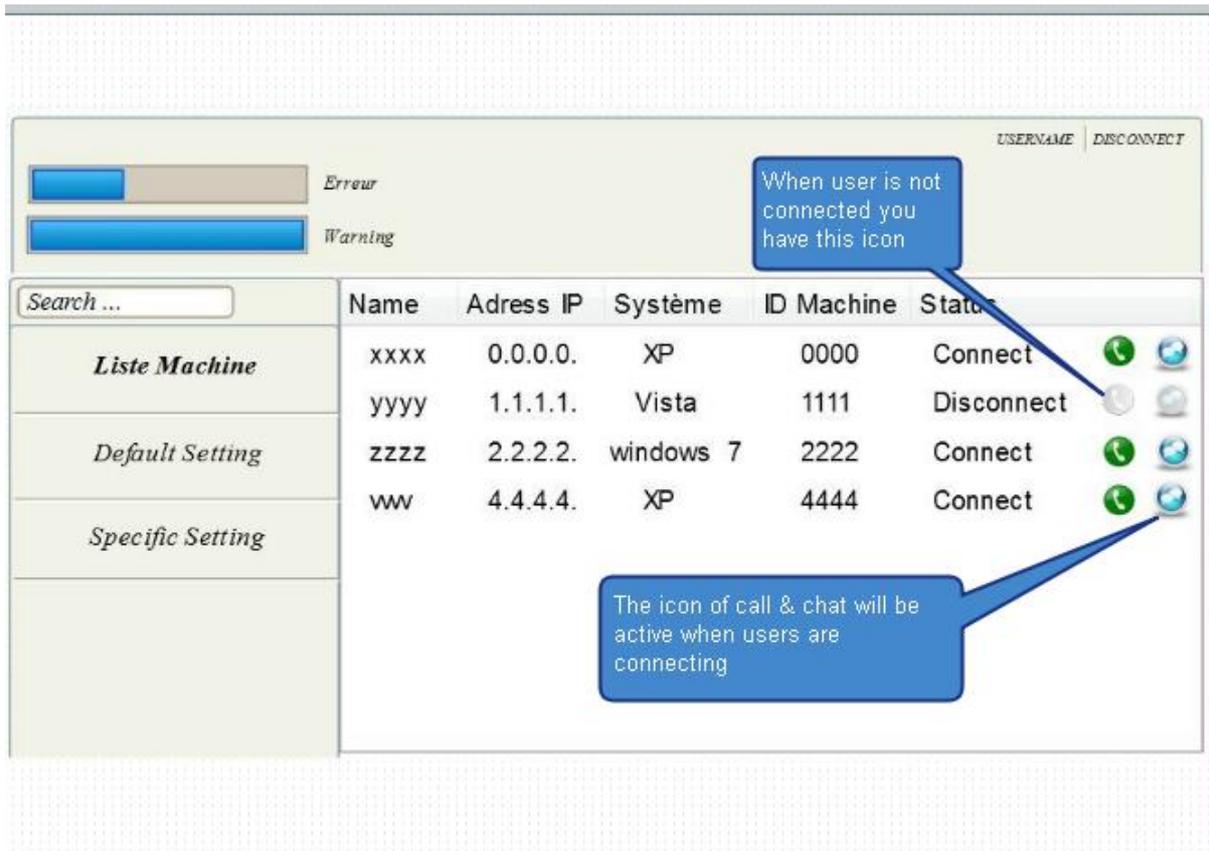


Figure 2.6 : Gestion d'assistance à distance par conférence audio ou chat

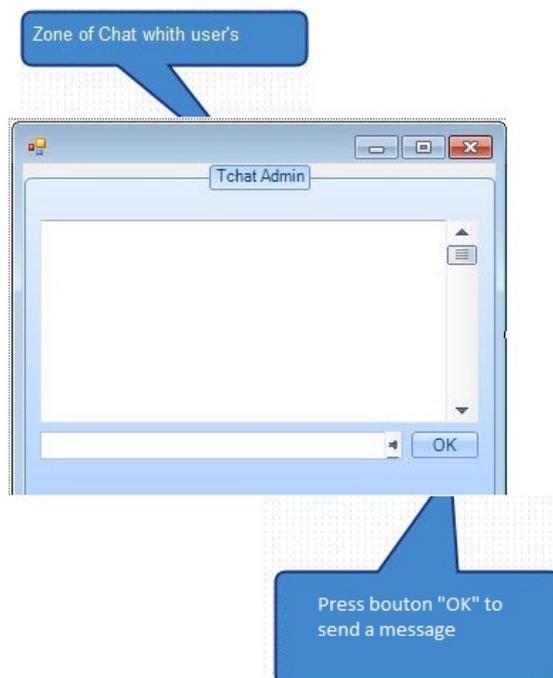


Figure 2.7 : Interface de chat



Figure 2.8 : Conférence audio

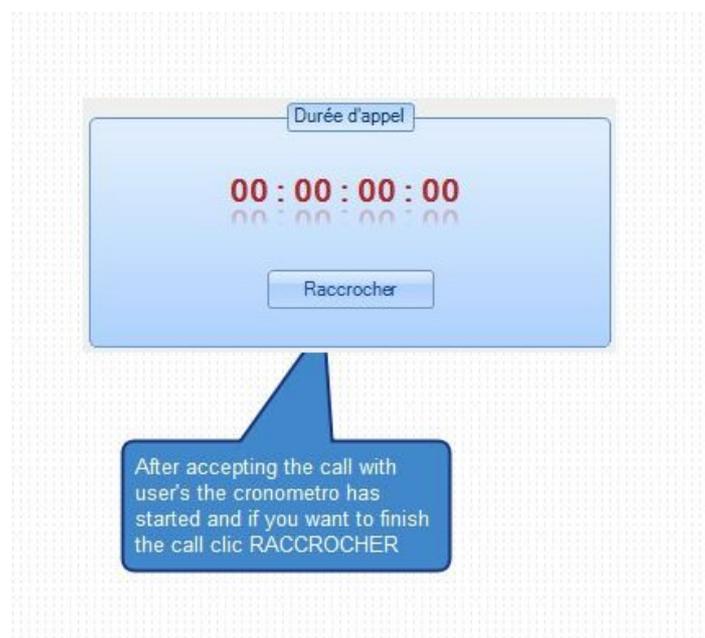


Figure 2.9 : Compteur de conférence audio



Figure 2.10 : Prise de contrôle à distance

## 2.3 Troisième partie : Tableau de bord

Dans une application de gestion de parc informatique dont le nombre d'équipement est assez grand, le suivi des machines, une par une, est très complexe et nécessite des efforts de synthèse pour dresser une vue générale de l'état du réseau, d'où la nécessité d'un tableau de bord.

MRMC devra être capable de dresser un tableau de bord synthétisant l'état global du parc informatique et devra aussi avoir la capacité de filtrer les statistique en fonction d'une machine spécifique du réseau voir une durée ou une date précise.

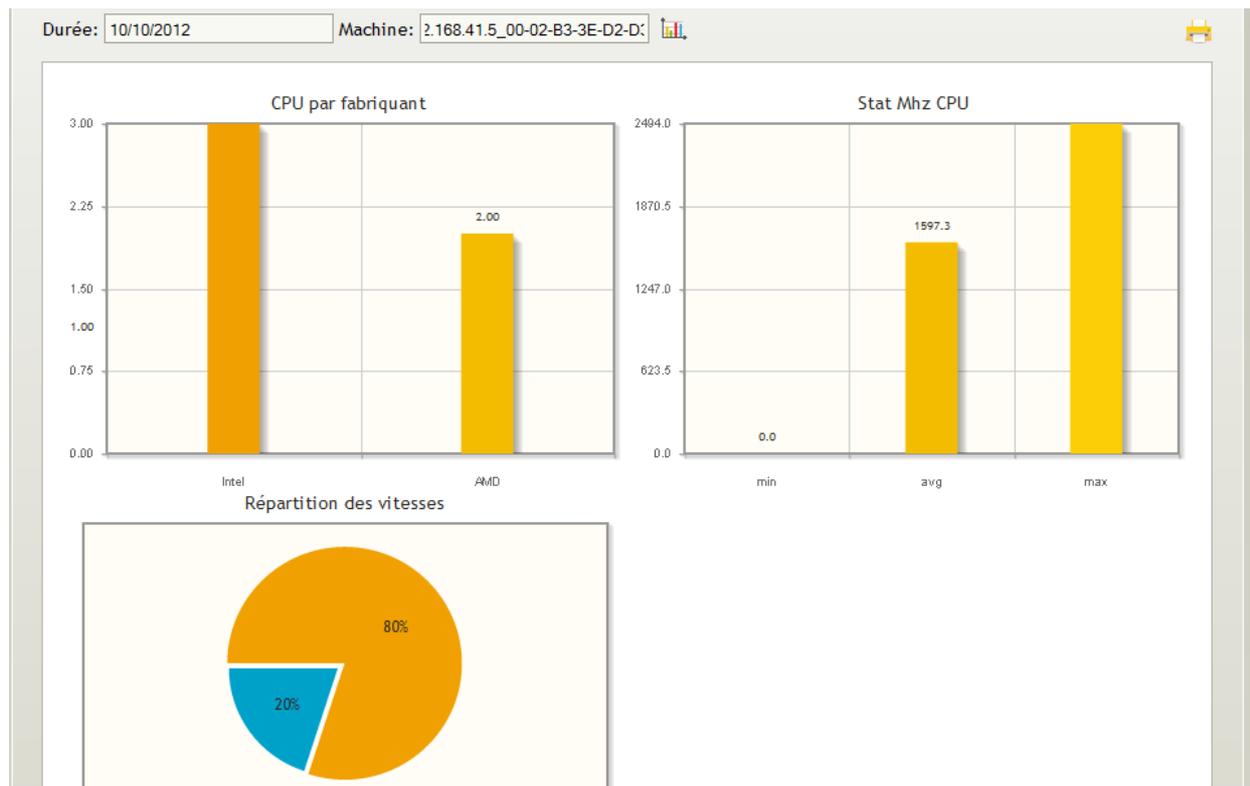


Figure 2.11 : Tableau de bord

## 2.4 Conclusion

Au cours de ce chapitre, une étude fonctionnelle a été réalisée, les fonctionnalités les plus importantes ont été décrites et détaillées. Néanmoins, d'autres fonctionnalités plus simples ont été volontairement négligées comme par exemple l'interface d'authentification, l'interface de paramétrage, etc.

Dans le chapitre suivant, je présenterai la conception UML détaillée de notre système.



# Chapitre 3

## Modélisation du MRMC

### 3.1 Choix d'UML

Comme le projet sera basé sur l'architecture J2EE, UML a été choisi comme langage de modélisation, car la notation UML est la plus appropriée pour des projets orienté objets. Ce choix peut être justifié également par plusieurs raisons :

- La notation UML facilite la compréhension et la communication d'une modélisation objet.
- UML est aujourd'hui un standard, adopté par les grands constructeurs de logiciel du marché.

Trois acteurs majeurs (Grady Booch, Ivar Jacobson et Jim Rumbaugh) se sont regroupés au sein de la société Rational pour définir le langage unifié UML (Unified Modeling Language) La notation UML a été conçue pour servir le langage de modélisation objet, indépendamment de la méthode de mise en œuvre.

MindWare utilise le logiciel Enterprise Architect pour la création des diagrammes UML. Ce logiciel lui confère une souplesse et un certain niveau de professionnalisme dans cette étape du projet.

La méthodologie de conception suivie par MindWare est très simple. En effet, elle se contente de modéliser tout système à analyser en créant trois diagrammes UML qui sont :

- Les diagrammes des cas d'utilisation.
- Les diagrammes de séquences.
- Les diagrammes de classes.

Dans mon l'analyse conceptuelle de mon projet j'ai suivi la méthodologie MindWare. Dans ce qui suit je vais essayer de présenter les diagrammes des cas d'utilisation avec une brève description de chaque cas, en suite je vais passer aux diagrammes de séquences avec les scénarios correspondants et en fin le diagramme de classes qui nous donnera une vue globale du système MRMC.

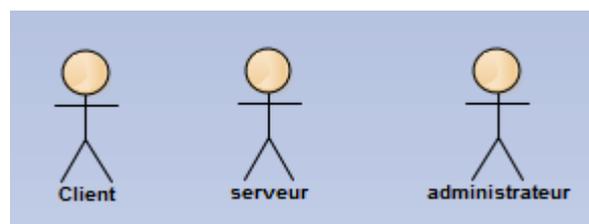
### 3.2 Les cas d'utilisation (Use case)

En génie logiciel et en ingénierie des systèmes, un cas d'utilisation définit une manière d'utiliser le système et permet d'en décrire les exigences fonctionnelles. D'après Bittner et Spence, « Un cas d'utilisation, défini simplement, permet de décrire une séquence d'événements qui, pris tous ensemble, définissent un système faisant quelque chose d'utile »<sup>1</sup>. Chaque cas d'utilisation contient un ou plusieurs scénarios qui définissent comment le système devrait interagir avec les utilisateurs (appelés acteurs) pour atteindre un but ou une fonction spécifique d'un travail. Un acteur d'un cas d'utilisation peut être un humain ou un autre système externe à celui que l'on tente de définir.<sup>13</sup>

MindWare utilise une méthode très simple pour la création des cas d'utilisation. En effet, sur la base de l'analyse préalable des besoins, elle recense les acteurs du système d'information et essaie de crée le cas ou les cas d'utilisation pour chaque acteur. En suite, un schéma des interactions entre les cas d'utilisation est dressé.

En suivant cette méthode, trois acteurs principaux appartiennent au système :

- Acteur client.
- Acteur serveur.
- Acteur administrateur.



<sup>13</sup> [http://fr.wikipedia.org/wiki/Cas\\_d%27utilisation](http://fr.wikipedia.org/wiki/Cas_d%27utilisation)

### 3.2.1 Acteur client

Cet acteur représente tout les équipements et machines à superviser. Les fonctionnalités offertes à cet acteur sont au nombre de trois.

#### a) Envoi de l'état de la machine

La tâche principale de cet acteur est l'envoi de l'état de la machine vers le serveur. Cette tâche est le noyau central de tout le système puisqu'elle l'alimente de toutes les informations techniques dont-il a besoin pour dresser l'état du réseau.

#### b) Téléchargement de l'agent client

La deuxième tâche consiste en le téléchargement de l'application cliente pour être capable de remonter les informations technique de la machine vers le serveur.

#### c) Mise à jour de l'agent client

La troisième tâche est la vérification de la disponibilité de mise à jour et la capacité de son installation. La mise à jour du jar est effectuée lorsque le numéro de version du jar en cour est différent de celui utilisé par l'administrateur. Cette vérification est réalisée à chaque envoi de l'état de la machine.

Le cas d'utilisation du client est présenté comme suit :

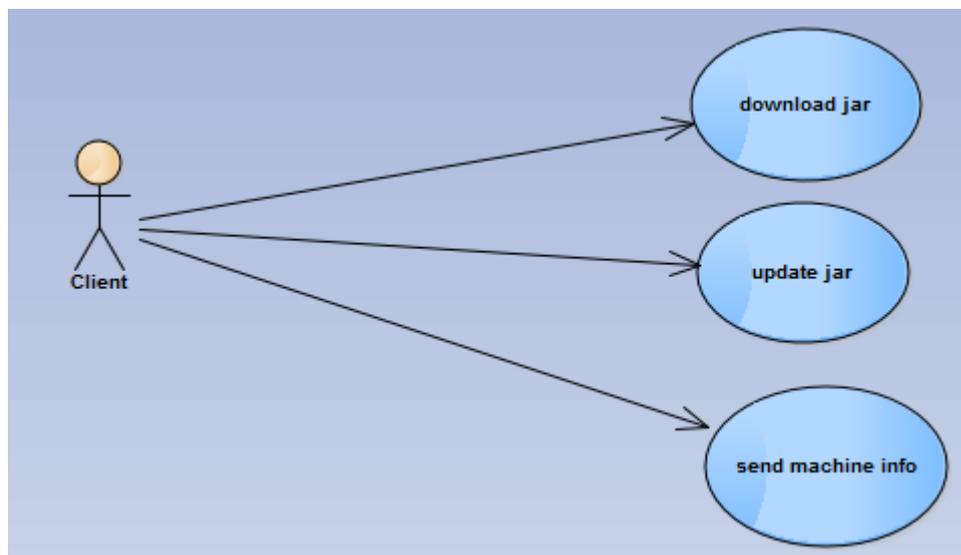


Figure 3.1: Cas d'utilisation du client

### **3.2.2 Acteur serveur**

Cet acteur représente le moteur d'application du système qui supervise la globalité du parc informatique. L'acteur serveur sera en charge des tâches suivantes: identifier les machines du parc, sauvegarder les informations de ces dernières ainsi que la détection des erreurs et des alertes du parc.

#### **a) identification des machines (identify machine)**

En fonction d'un intervalle paramétrable par l'acteur administrateur, le serveur doit identifier chaque machine supervisée et doit vérifier son état connecté ou non. Si la machine est connectée, le serveur pourra vérifier plusieurs paramètres comme par exemple la version de l'agent installé.

#### **b) Sauvegarde dans la base de données (store machine info into DB)**

Après que le client installe l'agent, il pourra envoyer ses données techniques au serveur pour que ces informations soient sauvegardées dans la base de données. Ces données constituent l'état de la machine dans un instant donné.

Avant chaque enregistrement de l'état d'une machine, une vérification de son existence préalable est réalisée. Si la machine a envoyé son état pour la première fois un nouveau enregistrement de machine est créé.

#### **c) Détection des alertes et des erreurs (detect warning and error)**

Selon un intervalle défini par l'acteur administrateur, le serveur doit détecter les alertes ainsi que les erreurs existantes dans les différentes machines du réseau. Ces alertes et erreurs sont configurées par l'administrateur et sont en relation avec l'état des composants constituant les machines surveillées.

Ces alertes et erreurs peuvent être détectées aussi lors de l'envoi de l'état des machines supervisées.

Le cas d'utilisation de la partie serveur est présenté ainsi :

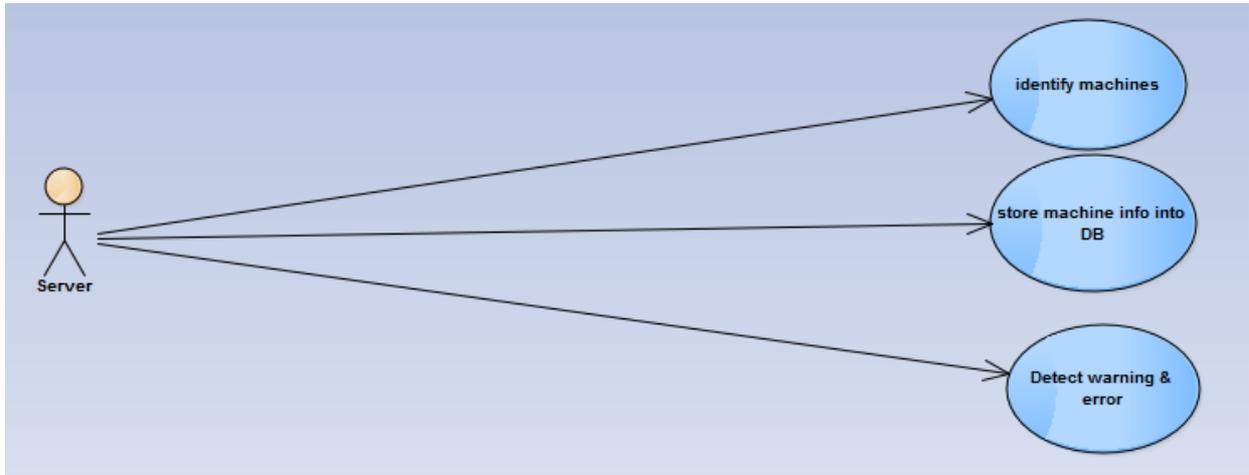


Figure 3.2: Cas d'utilisation du serveur

### 3.2.3 Acteur Administrateur :

L'acteur administrateur est représenté par tout utilisateur souhaitant utiliser la console d'administration de MRMC. Ces utilisateurs peuvent être des administrateurs avec les pleins pouvoirs sur l'application ou simplement des agents techniques avec des autorisations restreintes.

Cette partie explique les fonctions effectuées par l'administrateur, elle est présentée comme suit :

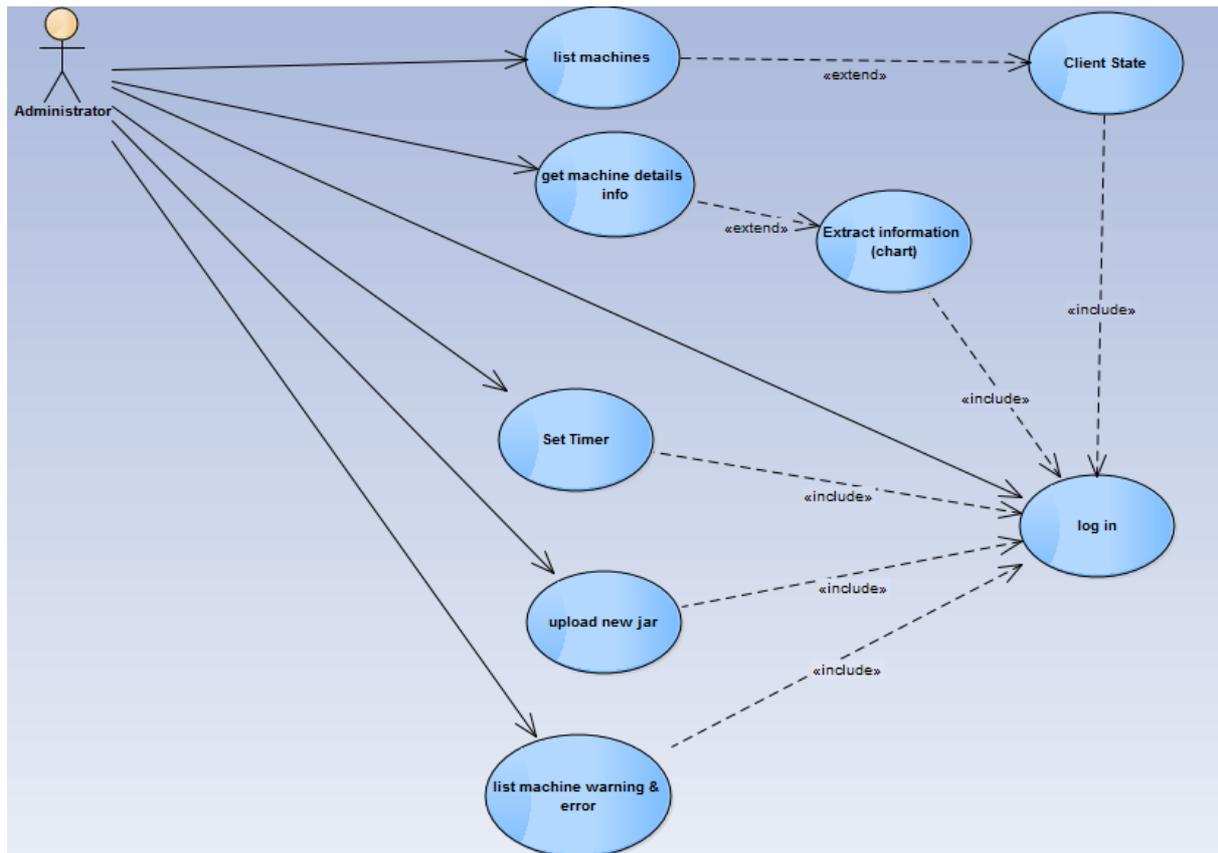


Figure 3.3: Cas d'utilisation d'administrateur

L'administrateur dispose de plusieurs fonctionnalités :

### a) Authentification (log in)

Avant d'accéder à n'importe quelle fonctionnalité l'administrateur doit s'authentifier pour avoir le droit de manipuler MRMC.

### b) Liste des machines (list machines)

Permet à l'administrateur d'avoir la liste des machines du réseau (avec des informations générales).

### c) Etat de la machine cliente (client state)

L'utilisateur peut à tout moment afficher l'état d'une machine supervisée comme l'état connecté ou non. Pour accéder à cette fonctionnalité l'utilisateur doit passer par l'action liste des machine précédemment décrite.

#### **d) Afficher l'état de la machine cliente (get machine detail info)**

Cette fonctionnalité donne la possibilité d'afficher les informations détaillées sur une machine du parc informatique.

#### **e) Extraire les informations de la machine (Extract information)**

Permet de retourner les informations de la machine en cours ces informations seront utilisées par l'action afficher l'état de la machine précédemment définie.

#### **f) Définir intervalle (Set timer)**

La fonctionnalité définir intervalle permet de préciser l'intervalle d'envoi des informations d'une ou plusieurs machines du réseau vers le serveur.

#### **g) Télécharger l'agent vers le client (upload new jar)**

Cette opération permet d'installer une nouvelle version de l'agent MRMC. Pour cela l'administrateur doit au paravent être authentifié et choisir un client déjà surveillé.

#### **h) Liste des alertes et erreurs des machines (list machine warning error)**

Comme son nom l'indique permet d'afficher la liste des machines qui présentent des alertes et/ou des erreurs. Pour cela l'administrateur devra au préalable configurer les alertes et les erreurs à détecter.

## 3.3 Diagrammes de séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation « Unified Modeling Language », on y met l'accent sur la chronologie des envois de messages.

La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent par le biais de messages.

### 3.3.1 Client

#### Scénario1

Dans ce scénario le client est sensé télécharger l'application cliente pour faire partie du réseau (pour que ce dernier soit surveillé par le serveur), une fois le téléchargement effectué, le serveur identifiera cette machine et vérifie son adresse MAC et IP, a` ce moment l'a l'extraction des données en local est activée.

Le diagramme de séquence de ce scénario se présente ainsi :

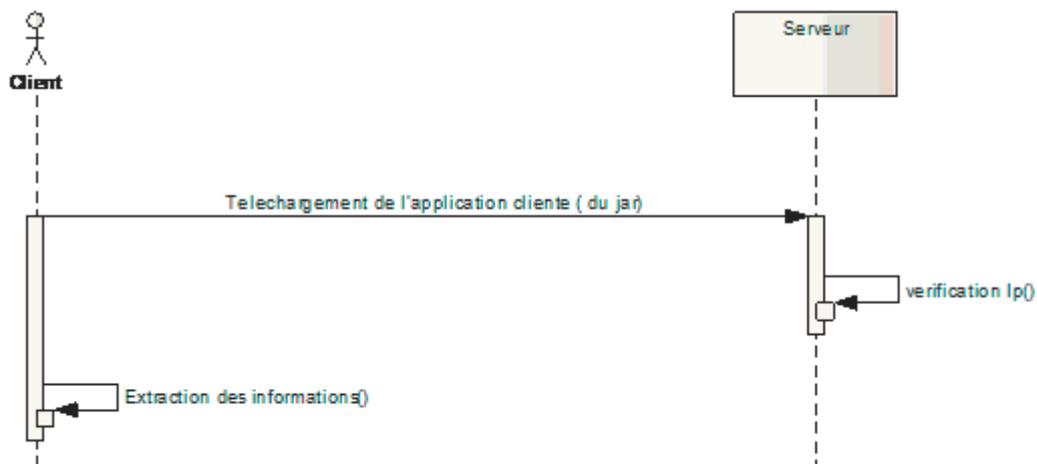


Figure 3.4: Premier téléchargement de l'application

#### Scénario2

Dans ce scénario le client dispose de l'application cliente, les informations relatives à ce dernier sont envoyées en permanence au serveur. Lors de chaque envoi, une comparaison est

effectuée entre la version du jar utilisé pour l'envoi (jar client) et celui utilisé dans le serveur, si les deux versions ne sont identiques, la dernière version sera téléchargée automatiquement dans la machine cliente.

Le diagramme de séquence relatif à cet effet se présente comme suit:

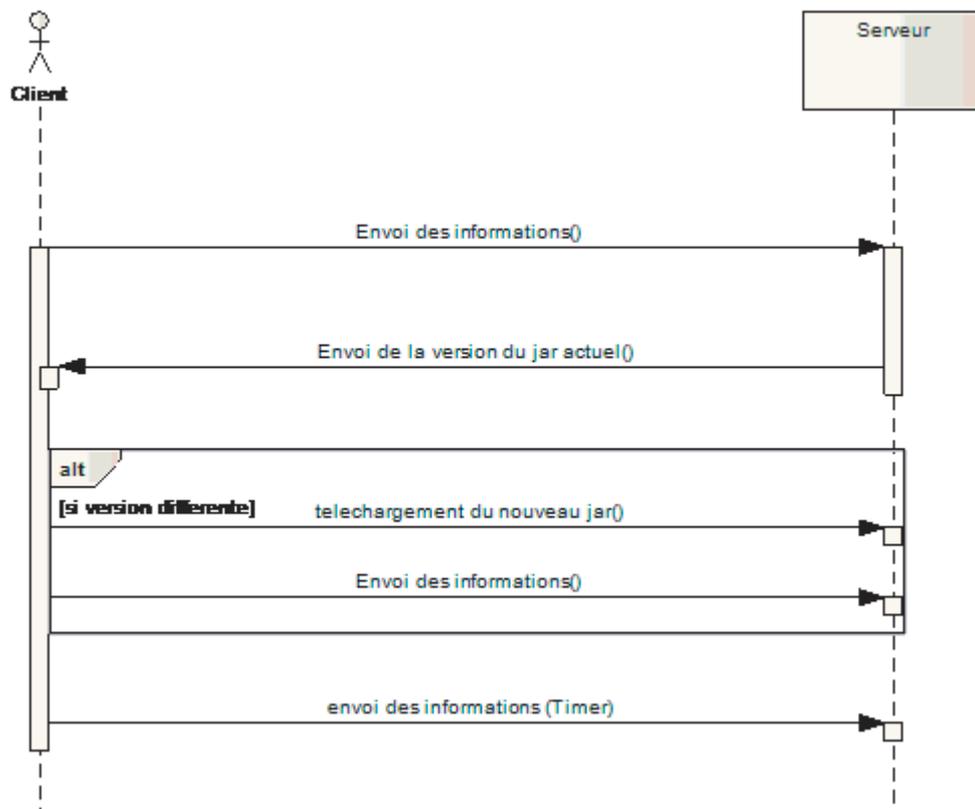


Figure 3.5: Mise à jour du jar

### 3.3.2 Serveur

Le serveur est toujours en attente des informations qui proviennent des clients du parc, une fois les données extraites (dans la machine cliente) elles sont envoyées vers le serveur et stockées dans une base de données hébergée dans ce dernier.

Le diagramme de séquence suivant explique cette partie :

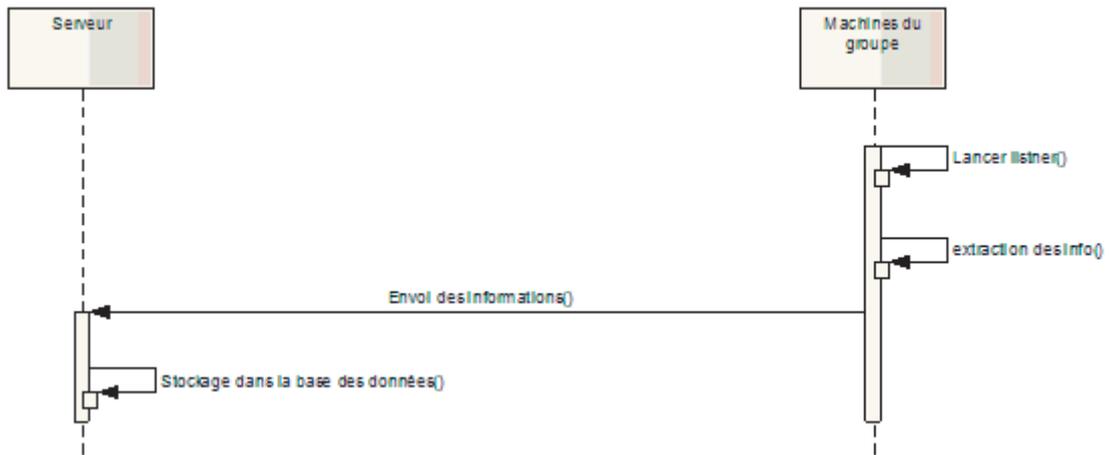


Figure 3.6: Récupération et stockages dans la base des données

### 3.3.3 Administrateur

Après l'authentification, l'administrateur aura la liste des machines du réseau, le diagramme de séquence suivant est relatif au paramétrage appliqué sur toutes les machines clientes.

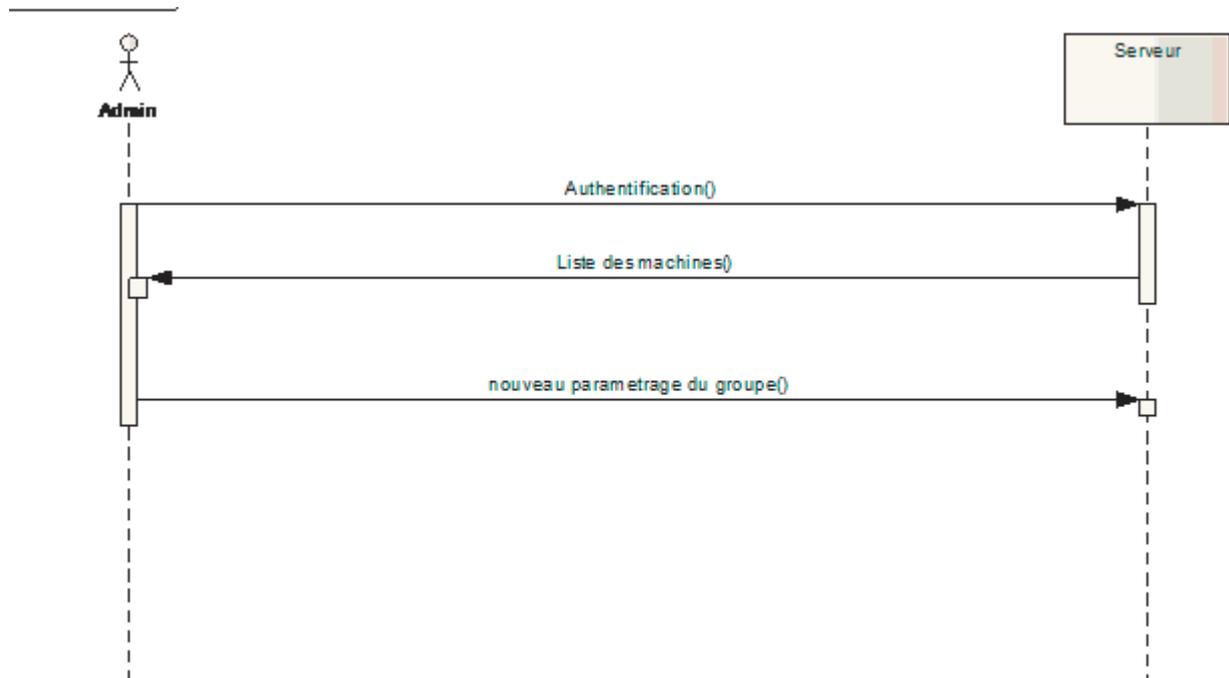


Figure 3.7: Paramétrage par défaut aux machines du réseau

## 3.4 Diagramme des classes

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

Dans la méthodologie de conception suivie par MindWare, le diagramme de classes est le plus important et c'est le modèle sur lequel se base toute la partie développement. Pour le projet MRMC, le diagramme de classes a été divisé en trois parties, la première c'est le package machine, la deuxième partie c'est le package surveillance (monitoring), la troisième c'est le package assistance (helpdesk). La figure suivante illustre ce découpage :

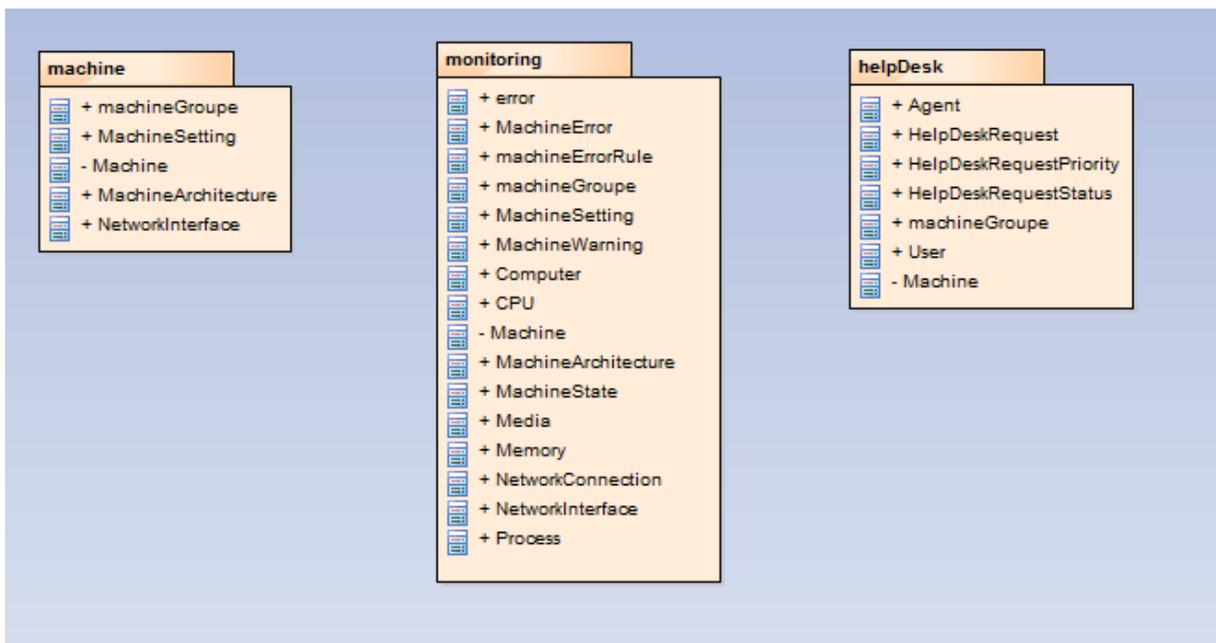


Figure 3.8 : packages du diagramme de classes MRMC

Maintenance je vais essayer de détailler les trois packages et décrire les différentes classes de chacun.

### 3.4.1 package « machine »

Dans ce diagramme on trouve toutes les classes en relation avec la gestion des machines et dont les instances sont statiques et ne changent pas de valeurs une fois créés.

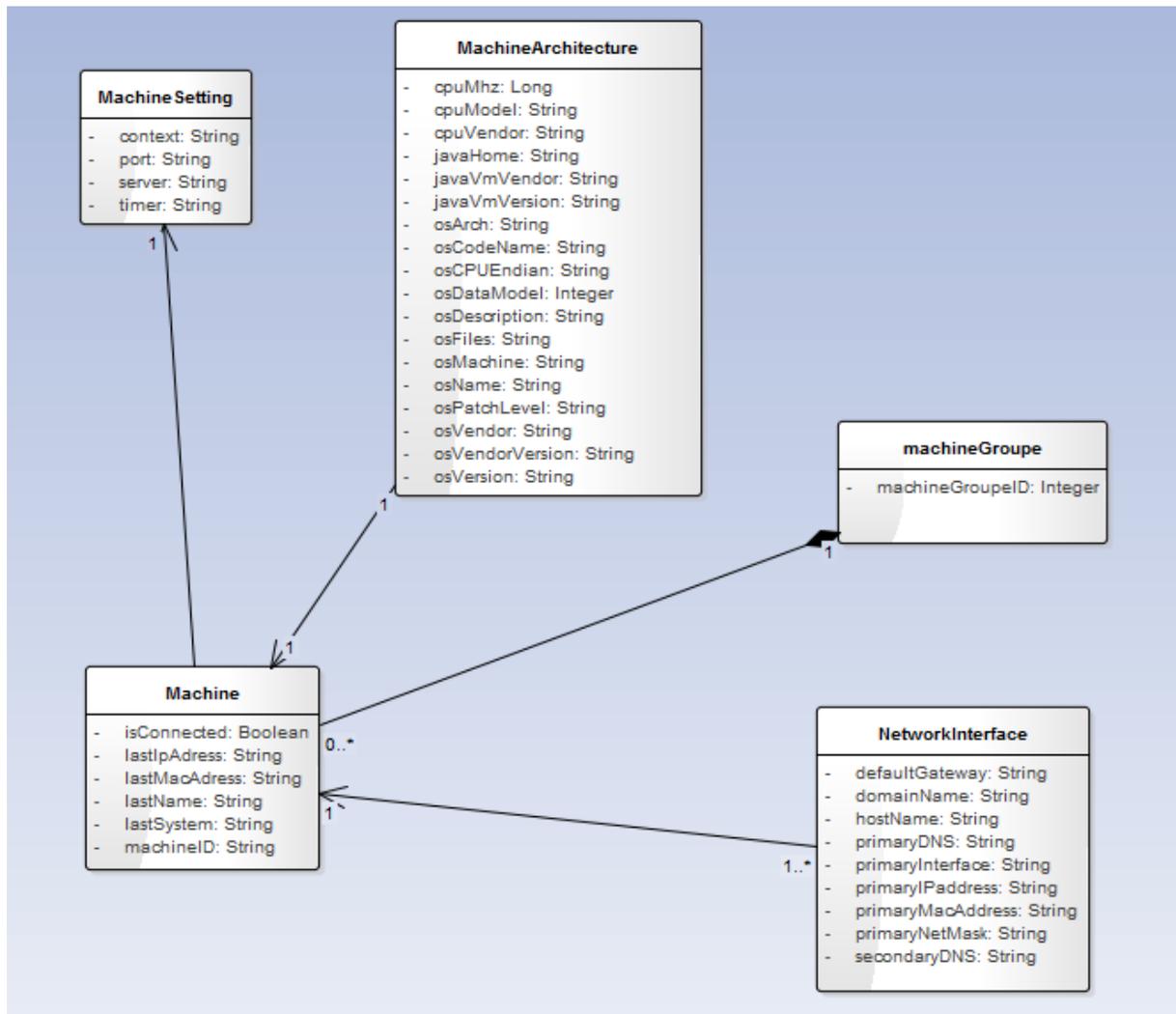


Figure 3.9 : diagramme de classes gestion des machines

Dans ce diagramme les classes suivantes existent :

- **Machine**

C'est la classe principale du package et du projet. C'est cette classe qui permet d'identifier les équipements surveillés. Les machines sont caractérisées par plusieurs attributs dont « machineID » qui constitue l'identifiant de la machine à superviser. Les autres attributs comme « isConnected » sont des informations sur son l'état actuel.

- **Architecture de la machine (MachineArchitecture)**

Cette classe contient toute les informations technique sur l'architecture matérielle et logicielle de la machine. Ces données donne une vue statique sur l'architecture du micro processeur, le système d'exploitation et la version de java.

- **Interface réseau (NetworkInterface)**

Comme une machine peut avoir une seule ou plusieurs cartes réseau la multiplicité de ce côté est « 1..\* ». Une interface est identifiée par son adresse MAC et contient des adresses IP pour sa passerelle et ses serveurs DNS.

- **Groupe machine (machineGroup)**

Pour des raisons de facilité de gestion des machines, la classe « machineGroup » a été ajoutée et chaque machine peut être ajoutée à un et un seul groupe. Chaque groupe est identifié par « machineGroupID ».

- **Paramétrage machine (machinesetting)**

Cette classe sert à configurer chaque machine en spécifiant son serveur de surveillance ainsi que son port et son délai d'envoi des données technique. Cette classe n'est utilisée que si les ces paramètre sont différents des paramètres généraux de l'application MRMC.

### **3.4.2 Package surveillance**

Ce package regroupe toutes les classes en relation avec la partie surveillance de MRMC. En effet la partie surveillance est le noyau de l'application qui l'alimente avec toutes les données techniques susceptibles de guider l'administrateur à avoir une idée générale sur l'état du parc informatique. De ce fait, ce package englobe la grande partie des classes utilisées dans MRMC.

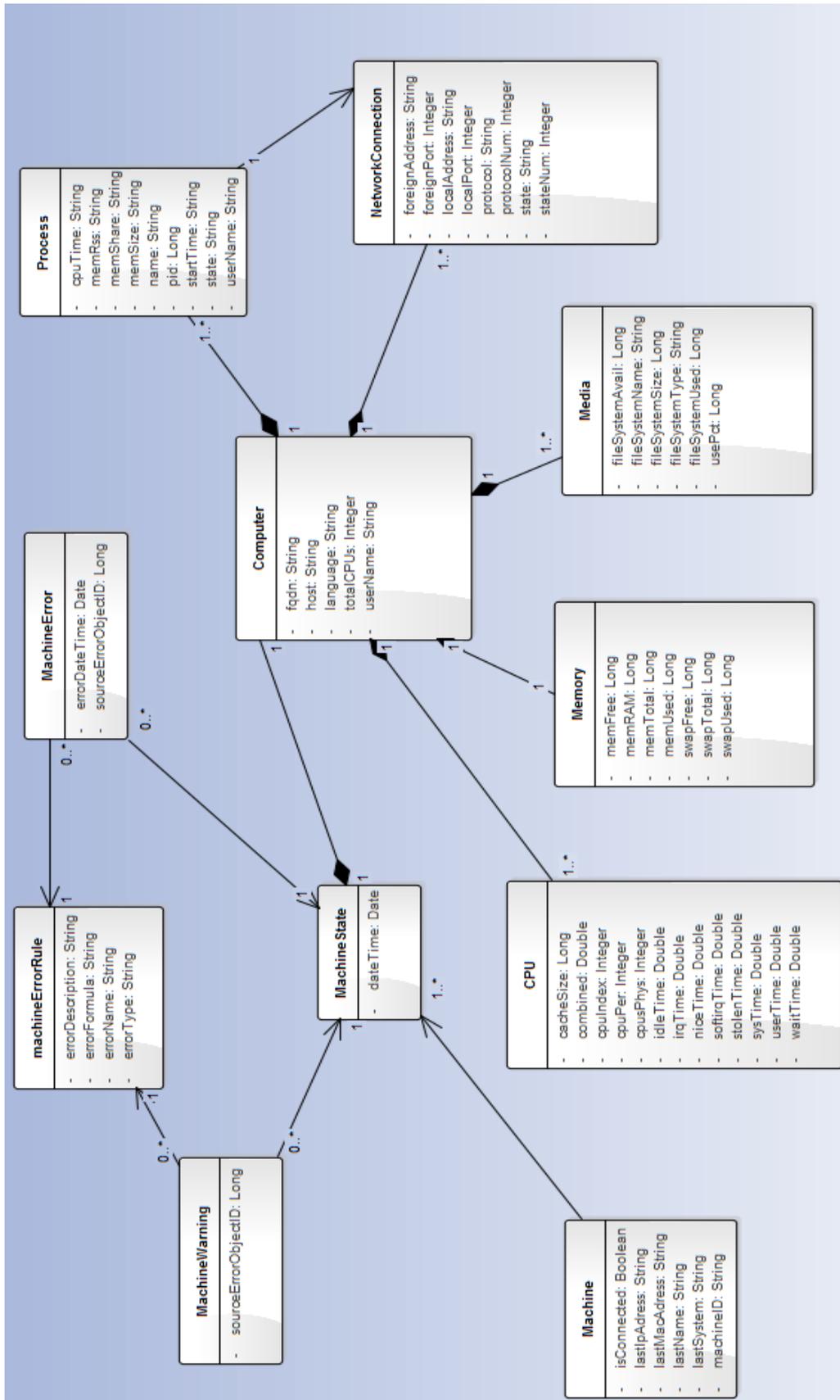


Figure 3.9 : diagramme de classes surveillance (monitoring)

Dans ce package on trouve les classes suivantes :

- **Ordinateur (computer)**

MRMC est une application destinée à superviser tous les types d'équipements installés dans un réseau informatique. Mais dans cette première version, l'application se contentera de surveiller les ordinateurs et c'est pour cette raison que la classe « computer » a été créée. Cette classe contient des informations spécifiques à tout ordinateur comme son nom, la langue utilisée, le total des microprocesseurs, etc.

- **Etat de machine (MachineState)**

Une machine est surveillée tant qu'elle est présente dans le parc informatique et que l'agent MRMC est exécuté. A un intervalle bien défini, la machine envoie ses données techniques au serveur MRMC pour qu'il les enregistre dans sa base de données et à chaque sauvegarde un nouveau état de machine est créé en spécifiant la date et l'heure de création.

- **Processus (Process)**

Classe relative aux processus qui sont en cours d'exécution dans une machine donnée.

- **Microprocesseur (CPU)**

Contient toutes les données relatives au microprocesseur tel que son horloge, le pourcentage de son exploitation à un moment donné...

- **Mémoire (Memory)**

Corresponds aux informations relatives à la mémoire (taille totale de la ram, mémoire utilisée, mémoire libre...).

- **Media**

Cette classe nous informe sur les médias de la machine (nombre et noms des disques durs, l'espace occupé ainsi que l'espace libre dans ces derniers)

- **Connexion réseau (NetworkConnection)**

Cette classe nous informe sur la nature du réseau établi ainsi que sur l'état de ce dernier sur la machine en question.

- **Règles des erreurs des machines (machineErrorRule)**

Pour que les alertes et les erreurs soient détectées dans les machines surveillées, l'administrateur doit les paramétrer en spécifiant leurs noms et leurs types, soit erreur ou alerte, et en donnant la formule de leurs détections.

- **Erreur des machines (machineError)**

Cette classe permet d'enregistrer les erreurs de la machine surveillée en fonction du paramétrage spécifié dans les règles des erreurs et des alertes.

- **Alerte des machines (machineWarning)**

Cette classe permet d'enregistrer les alertes de la machine surveillée en fonction du paramétrage spécifié dans les règles des erreurs et des alertes.

### 3.4.3 Package assistance « helpDesk »

Ce diagramme de classes contient toutes les classes en relation avec la gestion de l'assistance aux utilisateurs du parc informatique. Le service d'assistance est assuré par un ensemble d'agents qui sont responsables de la maintenance des machines ou des groupes de machine. Pour chaque intervention, un ticket est créé (helpDeskRequest) qui a une priorité et un statut.

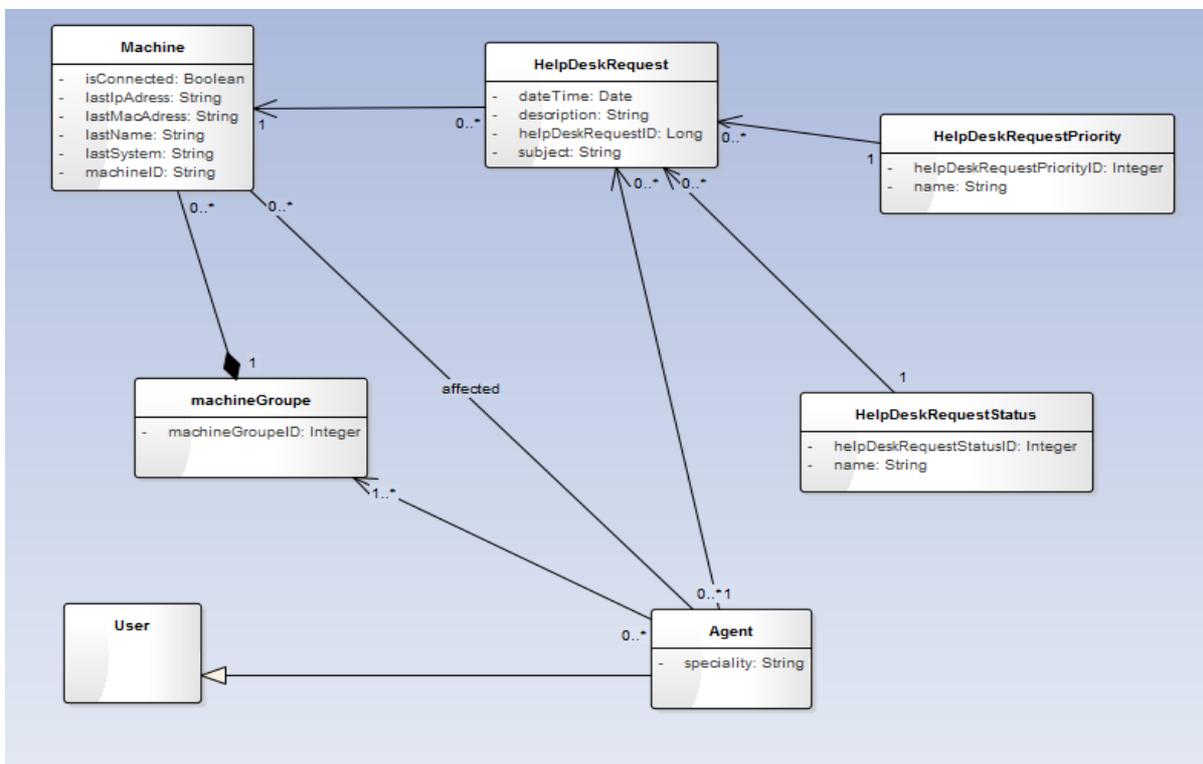


Figure 3.10 : diagramme de classes de gestion d'assistance (helpdesk).

- **Agent**

Le service d'assistance est assuré par les agents techniques. La classe « Agent » est une spécialisation de la classe « User » à fin qu'il puisse utiliser l'application MRMC pour le suivi de ses tickets d'assistance.

- **Ticket d'assistance (HelpDeskRequest)**

Pour chaque intervention, un ticket est crée pour la machine défailante et il affecté à un agent. Chaque ticket à un sujet, une date de création, une priorité et un statut.

- **Priorité du ticket (HelpDeskRequestPriority)**

Chaque ticket a une priorité en fonction de son degré d'importance. Ce degré d'importance est fixé par l'administrateur de MRMC.

- **Statut du ticket (HelpDeskRequestStatus)**

Chaque ticket a un statut de départ et change de statut en fonction du degré de finalisation de l'intervention d'assistance.

## **3.5 Conclusion**

Ce chapitre a été consacré à la présentation de la phase de conception UML de MRMC. Pour des raisons de lisibilité le système a été découpé en plusieurs parties en fonction des acteurs pour les cas d'utilisation et les diagrammes de séquences et en fonction des services offerts pour le diagramme de classes

Le chapitre suivant, sera consacré à l'étude technique qui présentera les différents outils utilisés pour l'élaboration du projet.



# Chapitre 4

## Etude technique du projet

Après la validation de la conception de MRMC par l'équipe projet et avant de passer au développement proprement dit de la solution, un autre type d'analyse est réalisée par MindWare, c'est l'étude technique.

MindWare entend par étude technique une analyse de tous les aspects en relation avec les technologies à utiliser par l'équipe de développement, commençant par l'éditeur jusqu'au langage de programmation en passant par toutes les bibliothèques et API à utiliser. Pendant cette étude l'équipe de projet mènent de longues discussions par fois houleuses dans le but de choisir les bons outils.

La grande importance de cette partie vient du fait que les outils choisis seront utilisés tout au long de la partie développement et qu'un mauvais choix technologique et la tentative de le corriger fera subir au projet de grands préjudices en terme de délais de réalisation et de pertes financières assez importantes.

### 4.1 Outils, langages et technologies utilisés

Pour le développement le choix des technologies à utiliser est très important et nécessite une grande expérience et une connaissance des tendances techniques. Dans la plus part des cas, ce choix est réalisé mutuellement par une équipe de projet après de longues discussions et d'échange d'idées.

Dans MindWare, plusieurs critères déterminent le choix d'une technologie ou une autre, mais parmi les plus importants on peut citer la licence de l'outil, sa stabilité et sa place dans les tendances techniques mondiale.

En effet, MindWare privilégie les outils opensource d'une part pour leur code ouvert et d'autre part pour la communauté libre qui est très active. Aussi, le développement d'une application stable nécessite obligatoirement l'utilisation de bibliothèques stables.

Dans ce chapitre on va détailler les outils, langages ainsi que les technologies utilisées pour la réalisation du Projet MRMC et pour chaque outil je vais essayer de justifier son choix. Citons maintenant quelques technologies.

#### **4.1.1 Plateforme de développement : JEE**

Comme dit précédemment, MRMC est une application full web destinée à être intégrée dans un ERP que MindWare a commencé son développement et compte finaliser dans les années à venir. De ce fait, le choix de la plateforme de développement était déjà réalisé dans l'analyse technique du dit ERP et c'est la plateforme JAVA JEE qui a été choisie.

JEE est une plateforme de développement créée pour étendre JAVA STANDARD EDITION la plateforme de base de JAVA. L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications.

Le choix de MindWare c'est porté sur JEE pour plusieurs raisons. En premier, la portabilité des applications développées en JAVA donne à la société une très grande souplesse dans le choix des plateformes d'hébergement opensource et diminue ainsi le coût de la solution finale. La sécurité et la stabilité reconnues des applications java sont d'importants arguments commerciaux et marketing dans toute campagne de vente. On peut ajouter à tout cela, le grand nombre de bibliothèques opensource et gratuit ainsi que la communauté immense dans laquelle on trouvera facilement des solutions à nos problèmes de développement les plus complexes.

#### **4.1.2 Editeur de développement : Eclipse**

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels dont l'objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et Framework) mais aussi d'AGL recouvrant modélisation, conception, test, gestion de configuration, création de rapport. Son IDE (éditeur de développement intégré), partie intégrante du projet, vise notamment à supporter tout langage de programmation<sup>14</sup>

MindWare étant une société favorisant l'utilisation des produits opensource, le choix d'Eclipse était des plus logiques.

---

<sup>14</sup> [http://fr.wikipedia.org/wiki/Eclipse\\_%28projet%29](http://fr.wikipedia.org/wiki/Eclipse_%28projet%29)

### 4.1.3 Le serveur d'application Tomcat

Tomcat est un conteneur de Servlet et un serveur web opensource développé par la fondation apache. Tomcat implémente plusieurs technologies JEE incluant java Servlet, Java Server Page (jsp), Websocket, etc.<sup>15</sup>

L'architecture de Tomcat a été créée en s'appuyant sur plusieurs composants, parmi lesquelles :

- Le moteur Catalina qui est le conteneur de servlet implémentant les spécifications de SUN Microsystem pour les Servlet et les JSP.
- Coyote qui est un conteneur Web supportant le protocole http 1.1. C'est ce composant qui permet à Tomcat de jouer un rôle de serveur web.
- Jasper le moteur JSP de Tomcat qui lui permet de compiler les pages JSP pour produire des classes java qui peuvent être exécutées par Catalina. Jasper peut aussi détecter toute modification dans les pages JSP afin de les recompiler.

Le but principal de Tomcat est de fournir des solutions serveur basées sur la plate-forme Java, de qualité identique aux applications commerciales mais développées avec l'esprit d'ouverture du logiciel libre.

Plusieurs solutions de serveur d'application étaient disponibles pour le développement de MRMC. Mais pour des raisons de facilité MindWare a choisi Tomcat. Aussi, comme l'équipe de développement n'envisageait pas l'implémentation des EJB, l'utilisation des grands serveurs d'application comme JBoss ou GlassFish n'était pas nécessaire.

Le cheminement d'une requête http peut être modélisé comme suit :

---

<sup>15</sup> [http://en.wikipedia.org/wiki/Apache\\_Tomcat](http://en.wikipedia.org/wiki/Apache_Tomcat)

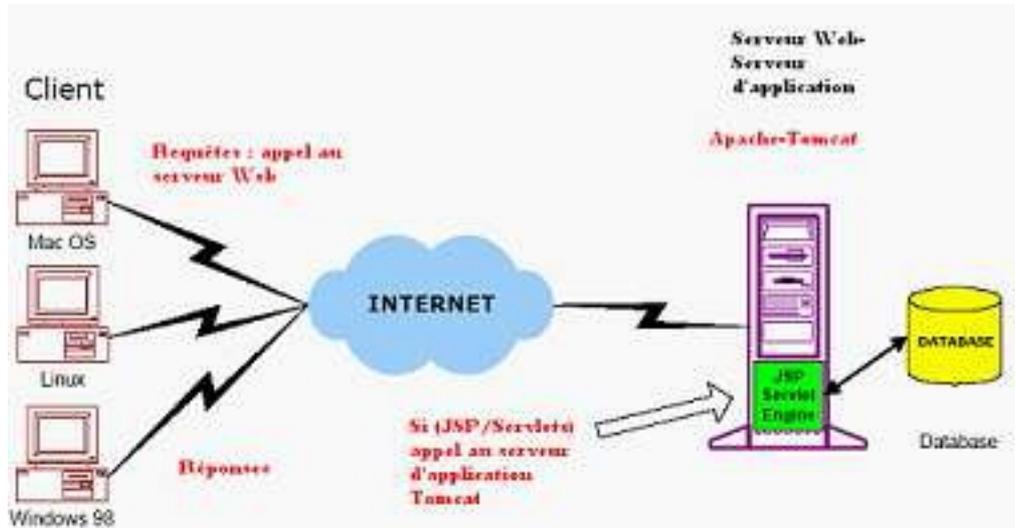


Figure 4.1 : traitement des requêtes http par Tomcat

#### 4.1.4 Le langage JAVA

JAVA est un langage de programmation développé par Sun Microsystems, inspiré de C++ fonctionnant sur le principe de machine virtuelle, il peut être utilisé par n'importe quelle plateforme logiciel ou matériel. Les programmes Java peuvent être appelés depuis des documents HTML ou de manière autonome. Lorsqu'ils s'exécutent à partir d'une page Web, on les appelle des applets Java. Lorsqu'ils s'exécutent sur un serveur Web, on les dénomme servlet. Java est portable : le code compilé peut être exécuté sur de nombreux processeurs à partir du moment où le système d'exécution est présent.

Le choix de java est très naturel du fait de l'utilisation de la plateforme JEE.

#### 4.1.5 La technologie Java Server Pages (JSP)

Les JSP sont des pages contenant du code Java imbriqué dans du HTML. Cette approche est similaire à celle de l'intégration PHP/HTML.

JSP est une technologie Java qui permet la génération de pages dynamiques sur le web. JSP consiste en une page HTML incluant du code Java qui s'exécutera soit sur le serveur Web, soit sur le serveur d'application. Le langage HTML décrit la manière dont s'affiche la page, le code Java servant à effectuer un traitement, par exemple récupérer les informations nécessaires pour effectuer une requête dans une base de données.

#### 4.1.6 Les Servlet

C'est un programme Java qui s'exécute dynamiquement sur le serveur Web et permet l'extension des fonctions de ce dernier, typiquement : accès à des bases de données, transactions d'e-commerce, etc. Un servlet peut être chargé automatiquement lors du

démarrage du serveur Web ou lors de la première requête du client. Une fois chargés, les servlet restent actifs dans l'attente d'autres requêtes du client.

#### **4.1.7 Le Framework Struts2**

Apache Struts2 est un Framework de développement d'applications web de seconde génération implémentant le design pattern modèle, vue, contrôleur (MVC).

Apache Struts2 est élégant, extensible pour la création d'applications web Java. Contrairement à ce que peut indiquer le nom, Struts2 n'est pas une continuation de Struts1 mais une fusion entre les Framework Struts1 et WebWork. Struts2 est crée pour répondre aux faiblesses de Struts1.

Plusieurs Framework pouvaient être choisis pour le développement de MRMC comme JSF, Play ou même spring. Mais le choix de cette bibliothèque était déjà fait lors du début du développement de l'ERP de MindWare et de ce fait son utilisation pour MRMC était très logique.

#### **4.1.8 JPMapper**

JPMapper (Java persistant Mapper) est une couche java intelligente et rapide de la persistance de donnée. Cette solution a été crée par la société MindWare pour offrir un service ORM à ses application JEE, il crée une relation logique entre les systèmes de bases de données relationnelles et les objets Java. La technique JPMapper est axé sur les points suivants :<sup>16</sup>

- **Elégance:** JPMapper interroge le système de gestion de base de données et convertit automatiquement les colonnes d'attributs de l'objet et aux données de base de données types de types de données Java.
- **Simplicité :** JPMapper permet d'interagir rapidement avec la base de données. pour la récupération, (par exemple), d'un objet de base de données, les lignes utilisées dans les méthodes classiques, peuvent maintenant être remplacé par 2 lignes au maximum.
- **Flexibilité :** JPMapper vous donne le plein contrôle de type connexions (transactionnels ou non-transactionnel), les caractéristiques JDBC (connexion, déclarations...).
- **Rapidité :** Nous essayons toujours de diminuer le temps de réponse, JPMapper utilise des algorithmes rapides en réduisant au minimum possible l'utilisation du contrôle de flux et les fichiers de configuration qui décrivent les objets et les relations entre eux.

---

<sup>16</sup> <http://jpmapper.sourceforge.net/>

### 4.1.9 La bibliothèque SIGAR

Dans la plus part des logiciel de surveillance réseau, l'utilisation d'un agent de supervision à installer dans la machine cliente facilite beaucoup l'extraction et l'exploitation des données techniques. Pour MindWare, deux solutions étaient possible soit le développement d'un agent ou l'utilisation d'une bibliothèque déjà crée et après de longues discussions c'est la deuxième solution qui a été choisie.

La tâche de recherche d'un agent m'a été confiée et après la comparaison entre plusieurs bibliothèques j'ai opté pour SIGAR. Les critères qui ont déterminé ce choix sont principalement sa licence Apache version 2, son noyau développé en java et surtout sa documentation très bien faite.

Cette partie correspond à l'axe pivot du projet c'est la raison pour laquelle elle va être détaillée un peu plus que les autres parties.

SIGAR (System Information Gatherer and Reporter) a été conçu et mis en application par Doug MacEachern chez technologies covalentes à la fin de l'année 2002 et son développement a continué chez Hyperic comme composant de noyau du produit de QG. SIGAR. Cette bibliothèque est une tentative qui fourni une plate-forme api (Application Programing Interface) pour collecter des informations du système.<sup>17</sup>

Le SIGAR api fournit une interface portative pour recueillir des informations de système comme : La mémoire système, le fichier d'échange, unité centrale de traitement, ouvertures mémoire des processus, environnement de travail, dossiers ouverts, détection et métrique de système de fichiers, détection d'interface réseau, information de configuration et métrique des tables TCP et d'UDP, table de route réseau. Ces informations sont disponibles dans la plupart des systèmes d'exploitation, mais chaque système a ses propres manières pour la fournir.

SIGAR fournit aux développeurs un api pour accéder à ces informations indépendamment de la plate-forme logiciel et matériel de la machine. Le noyau api est développé en C pur avec des attaches actuellement mises en application pour Java, Perl, rubis, python, Erlang, PHP et C#.

Les plates-formes suivantes sont actuellement soutenues :

---

<sup>17</sup> <https://support.hyperic.com/display/SIGAR/Home>

b

Operating System	Architecture	Versions	Binary Included
Linux	x86	2.2, 2.4, 2.6 kernels	✓
Linux	amd64	2.6 kernel	✓
Linux	ppc	2.6 kernel	✓
Linux	ppc64	2.6 kernel	✓
Linux	ia64	2.6 kernel	✓
Linux	s390	2.6 kernel	✗
Linux	s390x	2.6 kernel	✓
Windows	x86	NT 4.0, 2000 Pro/Server, 2003 Server, XP, Vista, 2008 Server, 7	✓
Windows	x64	2003 Server, Vista, 2008 Server, 7	✓
Solaris	sparc-32	2.6, 7, 8, 9, 10	✓
Solaris	sparc-64	"	✓
Solaris	x86	8, 9, 10	✓
Solaris	x64	"	✓
AIX	ppc	4.3, 5.1, 5.2, 5.3, 6.1	✓
AIX	ppc64	5.2,5.3,6.1	✓
HP-UX	PA-RISC	11	✓
HP-UX	ia64	11	✓
FreeBSD	x86	4.x	✗
FreeBSD	x86	5.x, 6.x	✓
FreeBSD	x64	6.x	✓
FreeBSD	x86,x64	7.x,8.x	✗
OpenBSD	x86	4.x,5.x	✗
NetBSD	x86	3.1	✗
Mac OS X	PowerPC	10.3, 10.4	✓
Mac OS X	x86	10.4, 10.5, 10.6	✓
Mac OS X	x64	10.5, 10.6	✓

Figure 4.2: Plate formes soutenue par Sigar

SIGAR est certifié pour les distributions Linux suivantes :

Distribution	Versions
Red Hat	6.2, 7.3, 8.0, 9.0
RHEL	3, 4, 5, 6
CentOS	3, 4, 5
Fedora	2, 3, 4, 5, 6, 7, 8, 9, 10
SuSE	8, 9, 10, 11
Ubuntu	6.06, 8.04, 8.10, 9.04
Debian	2.6, 3.0, 3.1, 3.2, 4.0, 5.0
VMware ESX	2.x, 3.0
XenServer	3.1, 3.2, 4.0, 4.1, 5.0
Slackware	10, 11
Mandrake	10
Scientific Linux	5
Gentoo	

Figure 4.3: Versions des distributions certifiées

#### 4.1.10 Le langage XML

XML est l'abréviation de « Extensible Markup Language ». Ce langage est le fruit des travaux du World Wide Web (<http://www.w3.org>). XML permet de créer de façon très souple des documents structurés ayant une ressemblance superficielle avec HTML mais dont l'objectif est tout autre. Cette ressemblance est due à l'origine commune des deux langages, hérités d'une norme de définition de langage appelée SGML (Standard Generalized markup Language).

La différence essentielle entre XML et HTML concerne la façon dont ils sont dérivés de SGML : XML est un sous ensemble de SGML, c'est à-dire une version allégée destinée à faciliter son utilisation sur le Web et à faire un format d'échange de données. Ces simplifications facilitent l'écriture d'applications XML, en revanche, HTML est une application de SGML c'est à-dire un langage spécifique qui obéit à la norme SGML.

Puisque HTML a été conçu pour échanger des données sur le Web, donc pourquoi créer un nouveau langage ? En vérité, XML et HTML ont des objectifs différents.

Un document HTML comporte des balises relatives à la mise en forme du texte mais rien sur son contenu réel. En d'autres termes, HTML concerne la présentation et non pas la nature du contenu. XML tente de distinguer le contenu de la présentation. Sa structure est proche de celle de HTML. La différence majeure est que l'on peut définir la structure des données à l'aide de balises personnalisées. Supposons que nous ayons un document contenant des informations bibliographiques. En HTML, il se présenterait ainsi :

```
<html>
  <head>
    <title>Catalogue</title>
  </head>
  <body>
    <h1>Catalogue</h1>
    <br><b>Titre</b>livre1
    <br><b>Auteur</b>auteur1
    <br><b>Prix</b>200 dh
    <br><b>Titre</b>livre2
    <br><b>Auteur</b>auteur2
    <br><b>Prix</b>185 dh <br>
  </body>
</html>
```

Supposons que nous souhaitons réaliser une application qui extraie la liste des ouvrages écrits par l'auteur1 du document HTML si dessus. La tâche est réalisable mais difficile à faire.

L'application doit appréhender la structure de la page HTML pour l'analyser.

Il serait difficile d'interpréter son contenu texte car la page ne comporte pas d'informations concernant la structure des données.

En XML, le même document se présente ainsi :

```

<?xml version="1.0" ?>
  <liste_des_livres>
    <livre>
      <titre>livre1</titre>
      <auteurs>
        <auteur>auteur1</auteur>
      </auteurs>
      <prix>200 dh</prix>
    </livre>

    <livre>
      <titre>livre2</titre>
      <auteurs>
        <auteur>auteur2</auteur>
      </auteurs>
      <prix>185 dh</prix>
    </livre>
  </liste_des_livres>

```

La présentation du fichier XML dans le navigateur Internet Explorer sera comme suit

:

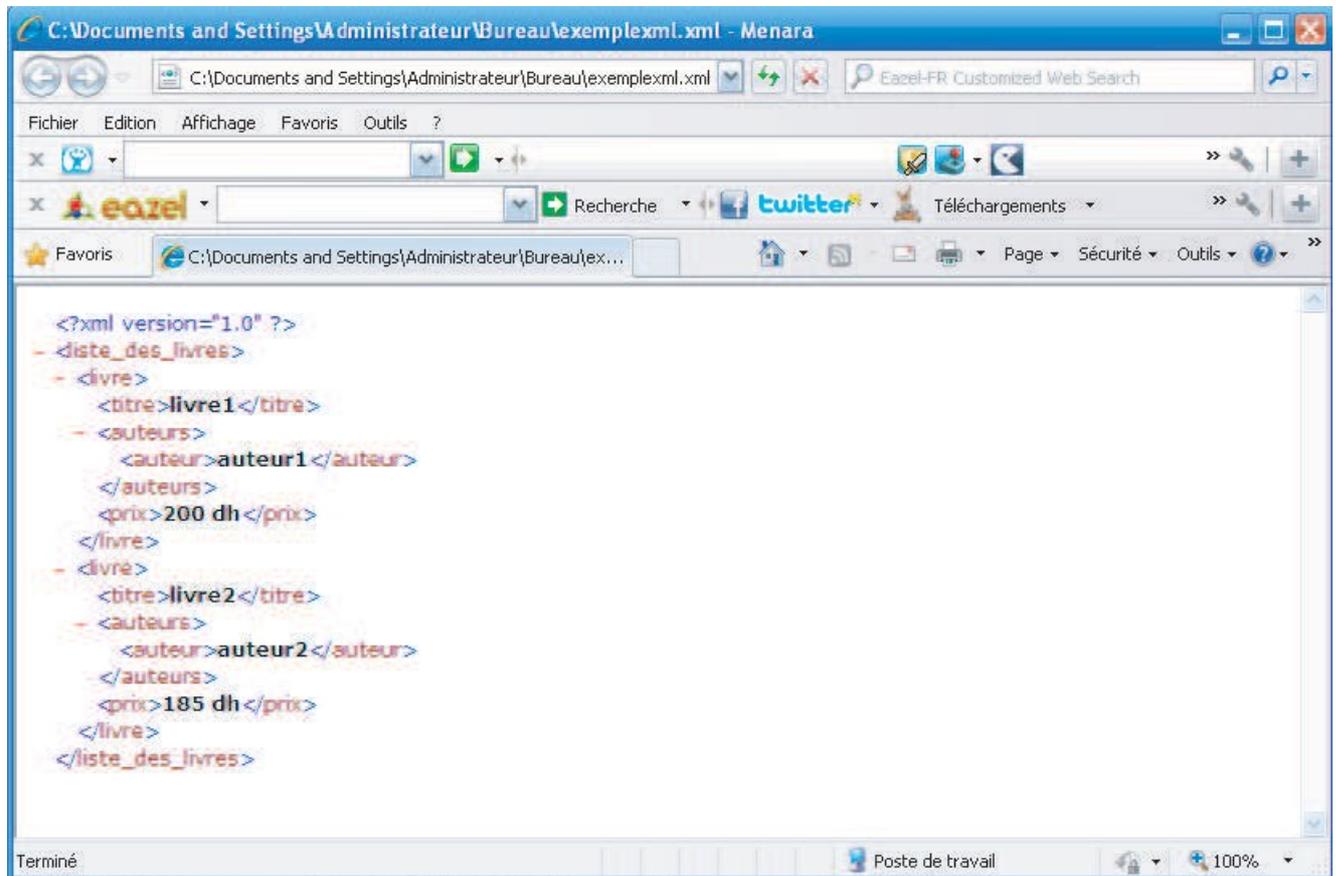


Figure 4.4: Exemple d'un fichier XML affiché par IE

L'extraction de la liste des ouvrages écrits par l'auteur1 requiert encore quelques efforts mais il est plus aisé de mettre en œuvre une telle application car la structure des données parle d'elle-même.

Les principaux avantages d'XML par rapport à HTML sont :

- L'extensibilité : XML fournit une description plus précise du contenu du document en acceptant un ensemble de tags extensibles. Les auteurs XML peuvent définir leurs propres tags et ainsi la précision dépendra de l'auteur et plus du langage.
- Hiérarchisation : XML peut spécifier des documentations lourdes et complexes avec des hiérarchisations.
- Evolution des hypertextes : XML a plus de spécifications pour implémenter des hypertextes (liens bidirectionnels, nommage dynamique, etc.)
- Recherche : La recherche avancée est prise en charge par XML, c'est beaucoup plus facile car la structure et le sens du document sont identifiés. La recherche peut par exemple se faire sur des noms de tags, etc.

#### **4.1.11 Dom4j**

Dom4j est un Framework (API) open source pour manipuler des données XML, XSL et Xpath. Il est entièrement développé en Java et pour Java. Il permet de lire, écrire, naviguer, créer ou modifier un document XML et exécuter des requêtes XPATH.

Dom4j propose un modèle de représentation d'un document XML et une API pour en faciliter l'utilisation.

Pour obtenir une telle représentation, dom4j utilise soit SAX, soit DOM. Comme il est compatible JAXP, il est possible d'utiliser toute implémentation de parser qui implémente cette API.

#### **4.1.12 Le langage XPATH**

XPATH est un langage de requêtes simple permettant de spécifier un ensemble de chemins à travers un document et donc de désigner un sous ensemble du document, sa syntaxe de base est semblable à l'adressage dans un système de fichier.

##### **Exemple :**

Soit le fichier annuaire.xml :

```
< ?xml version='1.0' ?>
<annuaire type='pages blanches'>
<entree>
<nom>BELKASMI</nom>
<telephone>0661251220</telephone>
</entree>
<entree>
<nom>ZEAARAOU</nom>
<telephone>0661672978</telephone>
</entree>
</annuaire>
```

Pour accéder par exemple à la liste des nœuds ” nom ” de l’annuaire, la syntaxe de la requête est : /annuaire/entree/nom

Un interpréteur XPath retournera la liste des noeuds répondant à cette requête, à savoir:

```
<nom>Belkasmi</nom>
<nom>ZEAARAOU</nom>
```

XPath permet aussi de gérer des requêtes plus complexes. Par exemple, pour rechercher le numéro de téléphone de Belkasmi dans le document, on utilise la syntaxe suivante :

```
/annuaire/entree[nom="Belkasmi"]/telephone
```

On utilise des expressions entre crochets pour indiquer des conditions sur les résultats obtenus. Dans l’exemple précédent, on a mis comme condition sur les entrées obtenues d’avoir un fichier dont le contenu de l’élément nom est égal à ” Redouane”.

Enfin, XPATH permet aussi une recherche par prédicat et fournit un ensemble de fonctions pour des requêtes plus complexes.

### 4.1.13 Le langage JavaScript

JavaScript est un langage de script basé sur la norme ECMAScript. Il s’insère dans le code HTML d’une page web ou dans un fichier à part et permet d’en augmenter le spectre des possibilités.

Ce langage de Programmation Orientée Objet (POO), faiblement typé, est exécuté côté client.

### La technologie AJAX

Asynchronous Javascript And Xml (AJAX) désigne un nouveau type de conception de pages Web permettant l’actualisation de certaines données d’une page sans procéder au rechargement total de cette page.

L'avantage principal d'AJAX est la vitesse de réponse aux actions de l'utilisateur.

### **La bibliothèque JQuery**

JQuery est une bibliothèque JS développée principalement par John Resig. Par bibliothèque on entend donc un ensemble cohérent de fonctions permettant de s'affranchir des tâches répétitives de façon uniforme sur les navigateurs les plus courants, elle est sous licence GPL et MIT, et donc complètement réutilisable sur des travaux professionnels. Son poids Compressé est de 14 Ko, ce qui est tout à fait raisonnable sur la plupart de nos projets.

Les navigateurs supportés sont :

- Firefox 1.5+
- Internet Explorer 6+
- Safari 2.0.2+
- Opera 9+

### **La librairie jqPlot**

jqPlot est un plugin JQuery pour générer des graphiques pure JavaScript côté client dans des pages web.

#### **4.1.14 Bibliothèque bureau à distance (jrDesktop)**

JrDesktop est une bibliothèque totalement écrite en java dont le but est d'offrir aux applications JEE un API de bureau à distance multiplateforme Windows, Linux et Mac. Mise a part le contrôle à distance, elle offre plusieurs fonctionnalités dont la compression de données, connexion sécurisée avec SSL/TLS, transfert de fichier, authentification, etc.

Le choix de jrDesktop a été fait pour plusieurs raison, le plus important étant son statut multiplateforme et sa licence GPL.

Le schéma ci-dessous illustre le fonctionnement de jrDesktop :

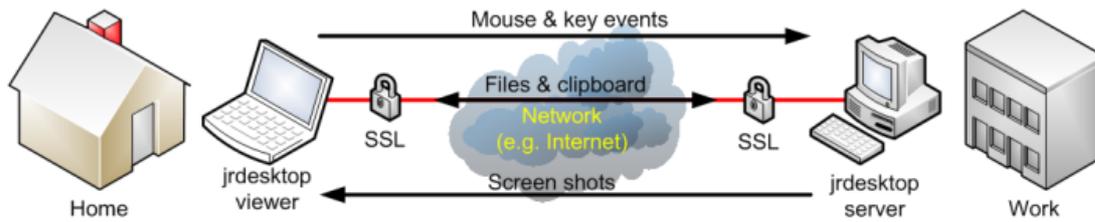


Figure 4.5 : les fonctionnalités de JrDesktop

#### 4.1.15 Méthodologie du codage : Technique MVC

La méthodologie MVC (Model View Controller) est une architecture pour construire des applications qui séparent les données (modèle) à partir de l'interface utilisateur (vue) et le traitement (contrôleur).

Cette technique permet de Fournir une interface de programmation entre les données et le traitement. MVC est largement utilisé dans des cadres d'application Web.

Le schéma suivant explique ce mécanisme :

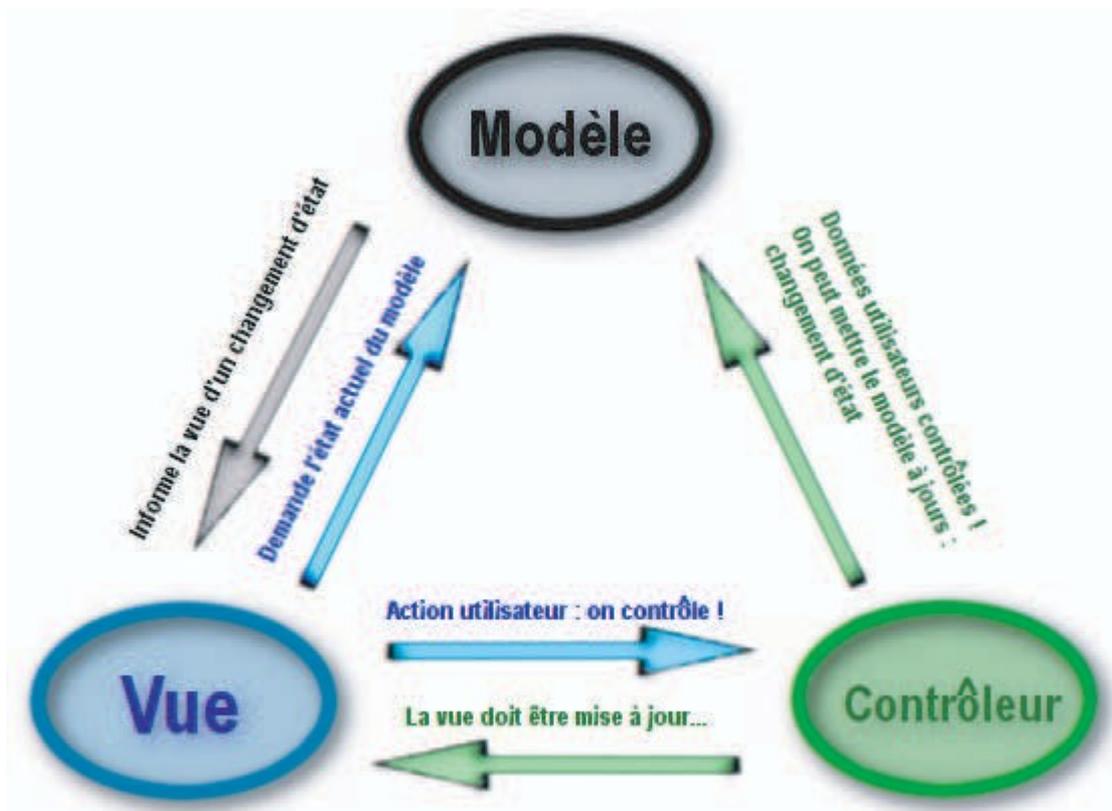


Figure 4.6: Méthodologie MVC

#### **4.1.16 SVN**

Subversion (en abrégé SVN) est un logiciel de gestion de versions, distribué sous licence Apache et BSD. Il a été conçu pour remplacer CVS. Ses auteurs s'appuient volontairement sur les mêmes concepts (notamment sur le principe du dépôt centralisé et unique) et considèrent que le modèle de CVS est bon, seule son implémentation est perfectible.<sup>18</sup>

Pour utiliser SVN, MindWare a mis en place un serveur Linux sur lequel a été installé le serveur SVN pour que les développeurs y uploadent leurs documents ou programmes. Ensuite un dépôt a été mis en place portant le nom du projet MRMC avec un architecture spéciale de dossier que MindWare utilise pour tous ses projets de développement, sur ce dépôt un ensemble d'autorisation a été crée pour les développeurs.

Pour faire un upload, le développeur doit utilisateur doit disposer d'un logiciel SVN client, celui utilisé par MindWare est Tortoise-svn.

## **4.2 Utilisation des technologies dans MRMC**

Dans ce qui suit, je vais essayer de faire une démonstration de l'utilisation de chaque outil ou technologie cité ci-dessous et pour chacun donner un exemple d'utilisation. Le classement que je vais suivre dans cette démonstration est basé sur la chronologie de développement. Et pour finir la démonstration, je présenterai les phases de développement de l'agent MRMC ce qui donnera une idée générale sur la méthodologie de développement d'application JEE suivie par MindWare.

Cette démonstration utilisera beaucoup de concepts techniques et nécessite une connaissance assez approfondie des technologies JEE.

### **4.2.1 Le dépôt SVN**

La première étape dans le développement de MRMC est la création du dépôt qui est réalisée par le responsable de projet M. Adil ZEAARAOUI sur le serveur SVN. Après quelque commande Linux le dépôt ressemble à cela :

---

<sup>18</sup> [http://fr.wikipedia.org/wiki/Apache\\_Subversion](http://fr.wikipedia.org/wiki/Apache_Subversion)

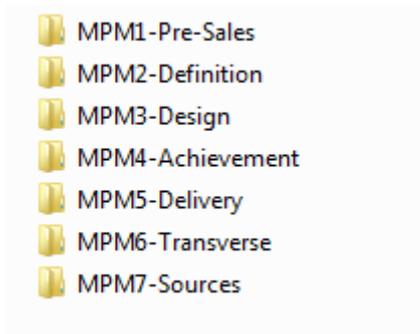


Figure 4.7 : Arborescence du dépôt SVN

Deux dossiers sont très utilisés par les développeurs. Le dossier « Design » dans lequel les fichiers des modèles UML et base de données sont créés (cas d'utilisation, diagramme de séquences, diagramme de classes et le modèle conceptuel de données). Le deuxième répertoire « Sources » est utilisé pour la structure du projet JEE de MRMC où seront enregistrés les fichiers de développement.

#### **4.2.2 Arborescence du projet Agent/JEE/ MVC**

Pour chacun de ses projets de développement, MindWare crée une structure de répertoire spéciale. Cette structure lui permet de faciliter la maintenance de l'application par les différents développeurs de l'équipe de projet. Cette structure est créée en utilisant l'EDI Eclipse.

Deux structures sont créées, la première pour le projet Agent MRMC qui est un projet JAVA classique avec quelque interface SWING et basé sur la bibliothèque « Sigar » et « jrDesktop » et la deuxième pour la console d'administration web qui est une application JEE/Struts2.

Dans le projet JEE/Struts2, deux dossiers sont très importants dans sa structure. Le dossier SRC qui contient le package JAVA de tous les contrôleurs et modèles utilisés conformément aux orientations du design-pattern MVC.

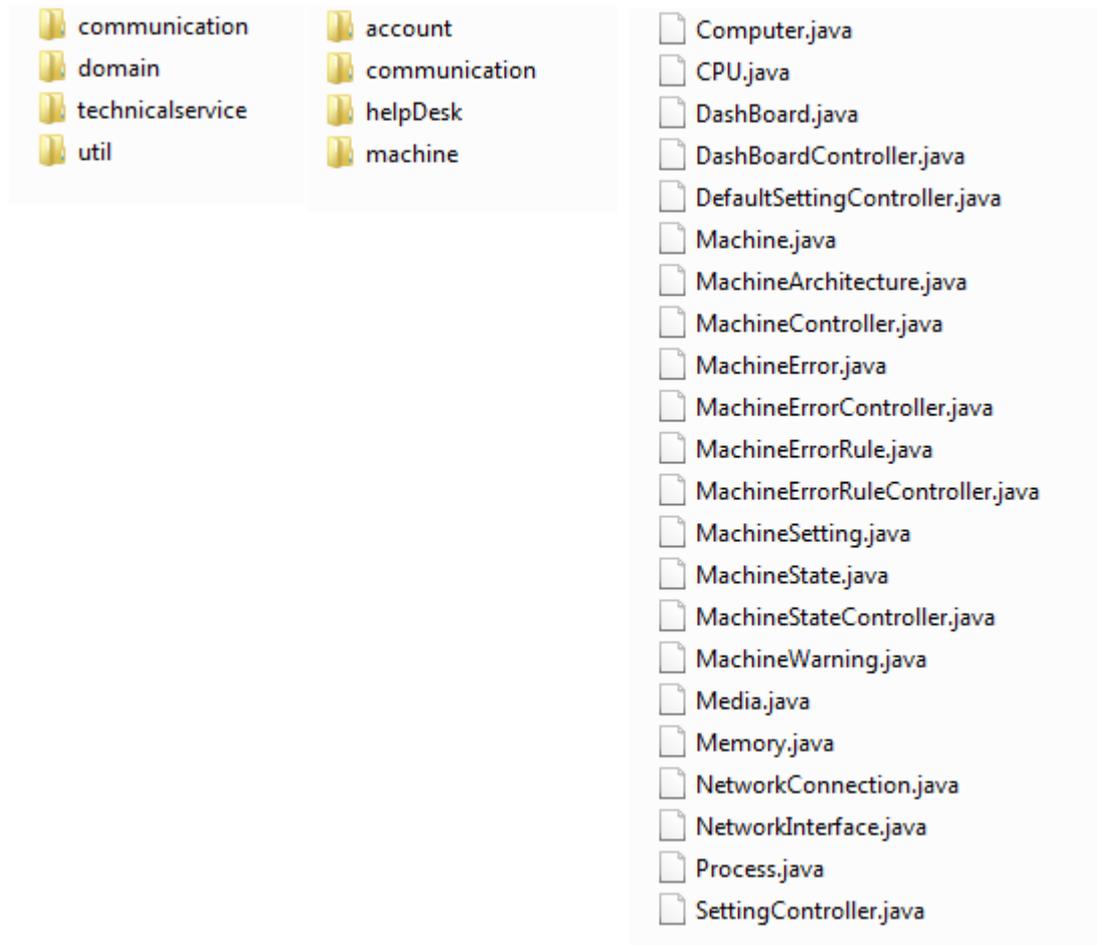


Figure 4.8 : Les modèles et contrôleurs MRMC

Le deuxième dossier est « webContent » qui contient tous les fichiers en relation avec la partie vue de MVC (HTML, jsp, js, images, css ...).

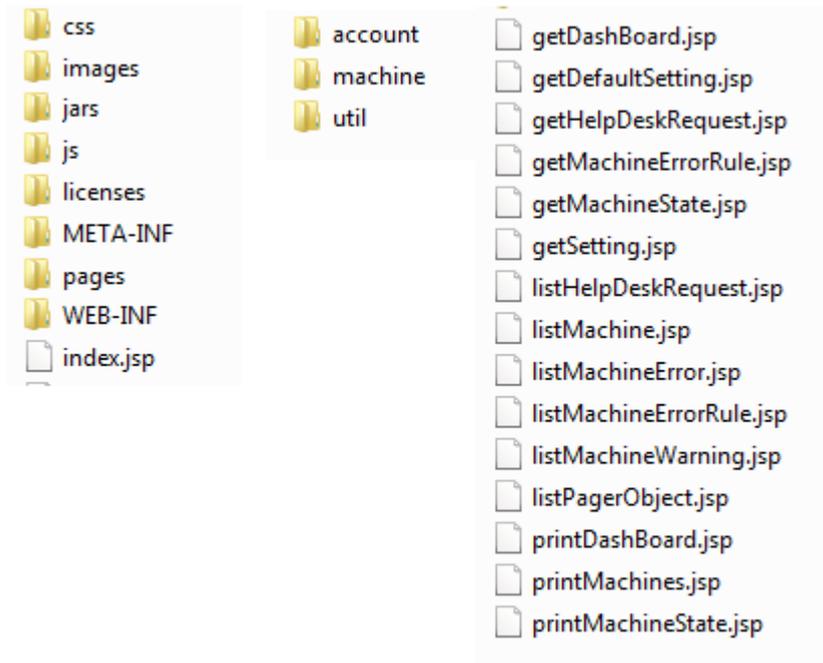


Figure 4.9 : les vues MRMC

### 4.2.3 Agent MRMC

L'agent MRMC est un programme à installer sur les machines clientes. Les fonctionnalités que doit offrir cet agent sont la surveillance de la machine sur laquelle il est installé en utilisant la bibliothèque « Hyperic Sigar », le contrôle à distance avec l'aide de la bibliothèque « jrDesktop », un service de chat textuel et un autre d'audioconférence développé entièrement, tous les deux, par l'équipe de projet MRMC.

Pour la fonctionnalité de surveillance et de bureau à distance il fallait décortiquer le code source des bibliothèques « jrDesktop » et « Hyperic Sigar » et extraire les fonctions nécessaires pour les intégrer dans l'agent, chose qui n'était pas facile vu la documentation compliquée et incomplète des dites bibliothèques.

Une fois cette tâche réalisée, le tout devait être packagé dans un seul fichier java exécutable. Pour cela, il était nécessaire d'utiliser un fichier XML descripteur « build.xml » et une bibliothèque nommée « one-jar ».

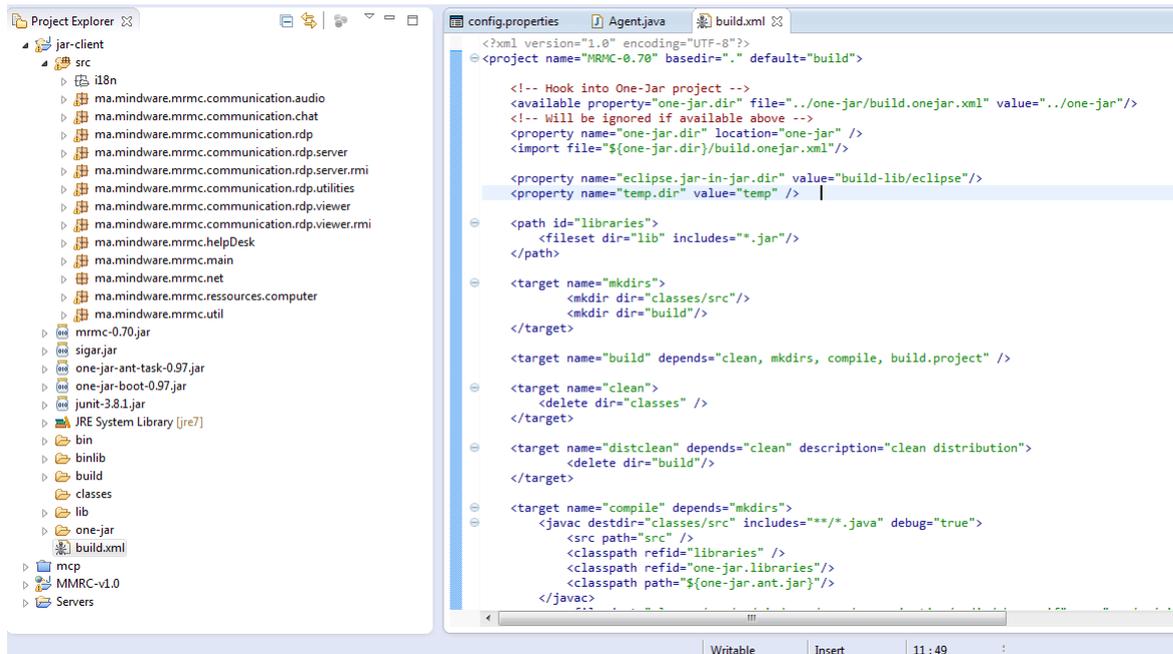


Figure 4.10 : projet Eclipse de l'agent MRMC

Après la création de fichier java exécutable, il fallait mettre au point un installateur pour faciliter l'intégration et le lancement de l'agent sur les machines clientes. Pour ce faire, il fallait utiliser un empaqueteur de déploiement et le choix c'est porté sur « InnoSetup » qui est un utilitaire opensource et gratuit. InnoSetup utilise un script pour la création de l'auto-installateur comme illustré dans la figure suivante :

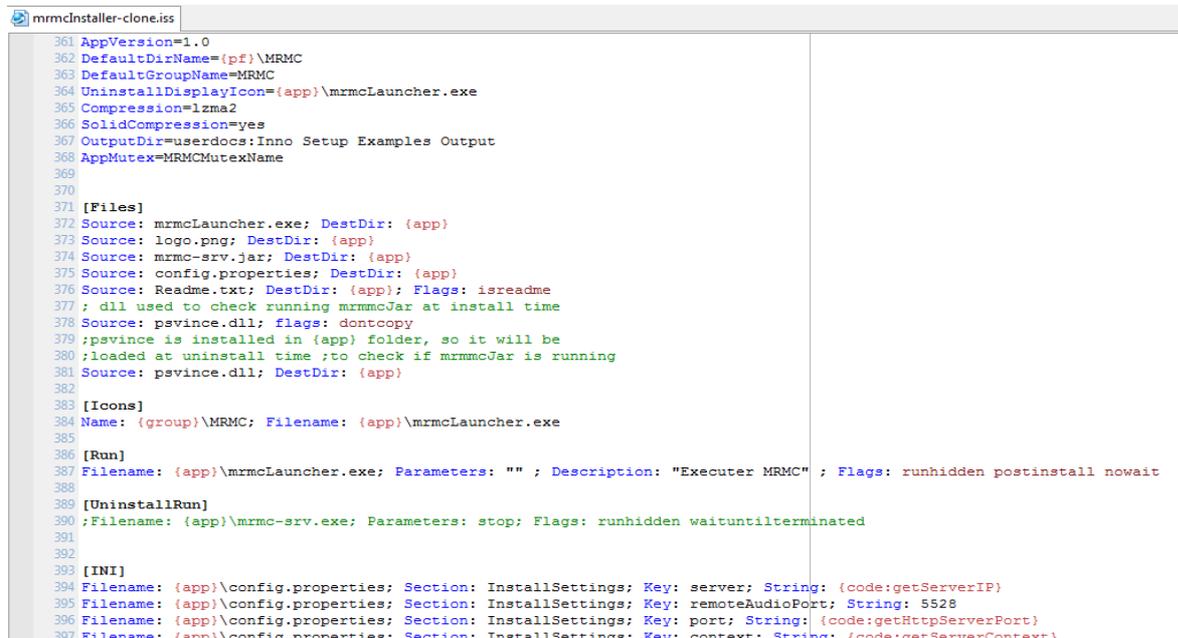


Figure 4.11 : script de l'installateur de l'agent MRMC

Les bibliothèques (les JAR) et les classes JAVA sont compressés dans un seul fichier qui servira d'installateur de l'agent MRMC, le plus souvent ce fichier est nommé « Setup.exe ».

#### 4.2.4 Console d'administration web

Dans cette partie du projet où a été utilisé le plus grand nombre de technologie et de bibliothèque. La présentation des technologies suivra un classement par ordre d'importance.

- **Apache Tomcat**

Pour utiliser le serveur Tomcat il suffit de l'ajouter dans le projet web dynamique. Les fichiers de configuration sont créés et paramétrés automatiquement par Eclipse. Cette configuration est réalisée avec des assistants Eclipse mais dans la phase d'hébergement de l'application dans un mode production plusieurs méthodes peuvent être utilisées. Pour héberger ses applications, MindWare crée un fichier de déploiement appelé « fichier WAR » qui est une sorte d'archive où sont enregistrés toutes les classes JAVA, les pages JSP, les bibliothèques, etc. Ce WAR sera déployé en utilisant « Tomcat Manager ».

- **Struts2/MVC**

Struts2 est le noyau central de la console d'administration. Pour utiliser ce Framework MVC il faut simplement configurer le fichier « web.xml » et ajouter les bibliothèques nécessaires au fonctionnement de Struts2.

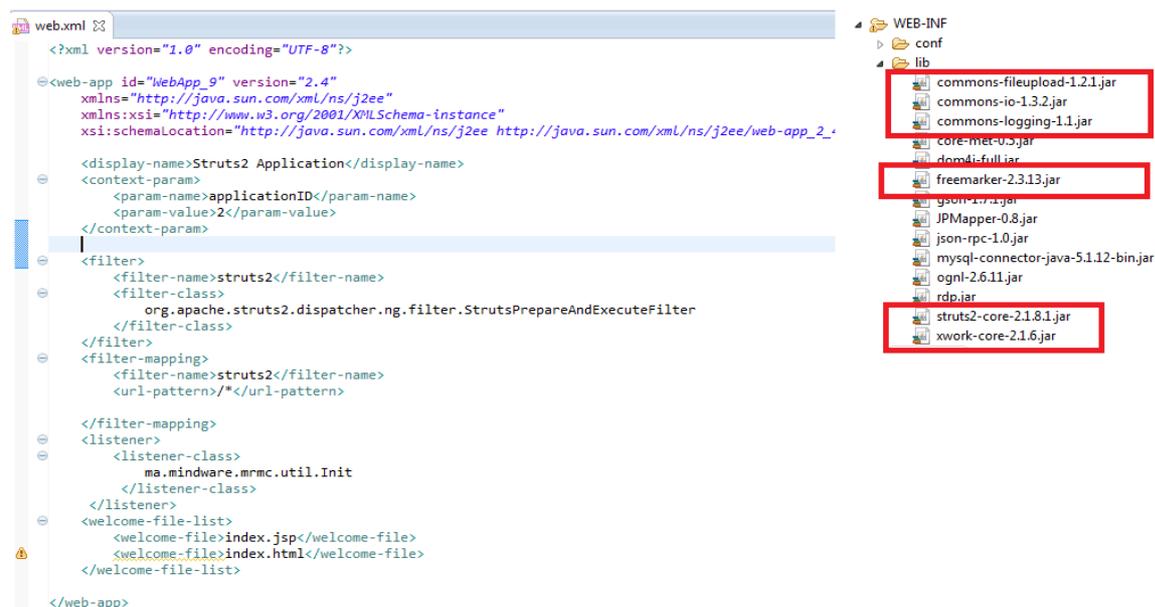
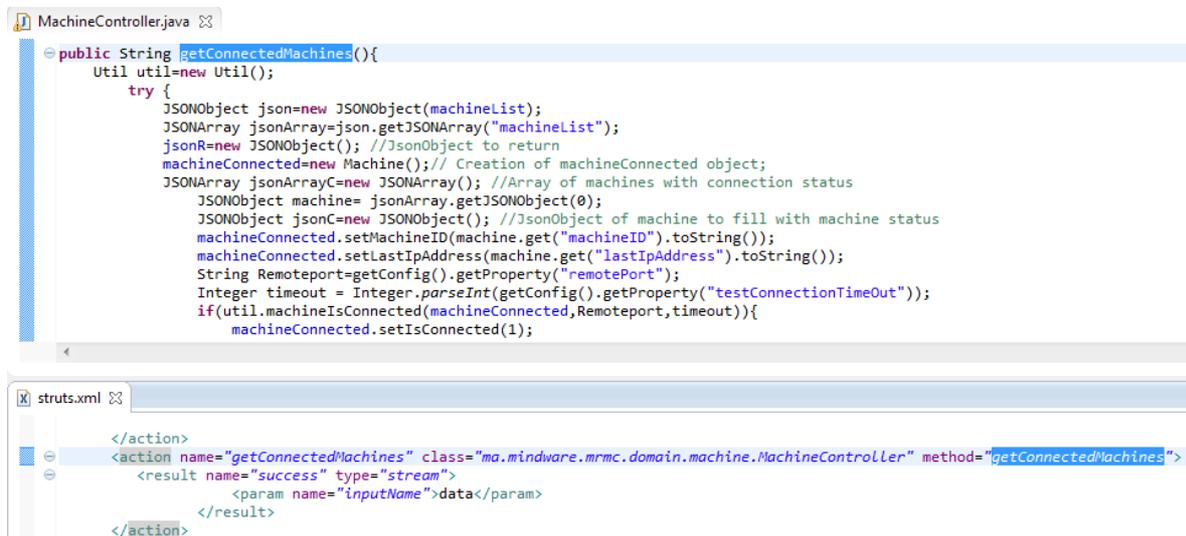


Figure 4.12 : Utilisation de Struts2

Struts2 est basé sur le concept d'actions qui sont toutes les interactions entre l'application MRMC et les utilisateurs. Ces actions sont cataloguées dans le fichier « Struts.xml » est sont aussi définies dans les différents IHM du projet. Chaque action est traitée par une méthode d'un contrôleur.



```
MachineController.java
public String getConnectedMachines(){
    Util util=new Util();
    try {
        JSONObject json=new JSONObject(machineList);
        JSONArray jsonArray=json.getJSONArray("machineList");
        jsonR=new JSONObject(); //JsonObject to return
        machineConnected=new Machine();// Creation of machineConnected object;
        JSONArray jsonArrayC=new JSONArray(); //Array of machines with connection status
        JSONObject machine= jsonArray.getJSONObject(0);
        JSONObject jsonC=new JSONObject(); //JsonObject of machine to fill with machine status
        machineConnected.setMachineID(machine.get("machineID").toString());
        machineConnected.setLastIpAddress(machine.get("lastIpAddress").toString());
        String Remoteport=getConfig().getProperty("remotePort");
        Integer timeout = Integer.parseInt(getConfig().getProperty("testConnectionTimeout"));
        if(util.machineIsConnected(machineConnected,Remoteport,timeout)){
            machineConnected.setIsConnected(1);
        }
    }
}

struts.xml
</action>
<action name="getConnectedMachines" class="ma.mindware.mrmc.domain.machine.MachineController" method="getConnectedMachines">
    <result name="success" type="stream">
        <param name="inputName">data</param>
    </result>
</action>
```

Figure 4.13 : présentation des actions Struts2

- **ORM JPMapper**

JPMapper est un ORM très facile à utiliser, il ne demande pas beaucoup de configuration, il suffit de créer le fichier « dbInformation.properties » qui sert à lier l'application avec un SGBDR et de mapper les modèles avec les tables. Ce mappage est réalisé avec la notion d'annotation JAVA.

```
#####
### DB Information ###
#####
|
dbName mcmc
dbUser root
dbPassword
host localhost
port 3306
dbFieldDelimiter `

#####
### Data Type Mapping ###
#####
db2java.CHAR String
db2java.VARCHAR String
db2java.TEXT String
db2java.INT Integer
db2java.TINYINT Integer
db2java.SMALLINT Integer
db2java.MEDIUMINT Integer
db2java.BIGINT Long
db2java.FLOAT Float
db2java.DOUBLE Double
db2java.DATE Date
db2java.TIMESTAMP Timestamp
db2java.DATETIME Timestamp

#####
### Data Type with or without delimiter###
#####
dbFieldValueDelimiter.CHAR `
dbFieldValueDelimiter.VARCHAR `
dbFieldValueDelimiter.TEXT `
dbFieldValueDelimiter.INT
dbFieldValueDelimiter.TINYINT
dbFieldValueDelimiter.SMALLINT

package ma.mindware.mrmc.domain.machine;
import java.util.ArrayList;

public class Machine {
    @ID
    private String machineID;
    private Integer isConnected;
    private String lastIpAddress;
    private String lastMacAddress;
    private String lastName;
    private String lastSystem;
    private String jarVersion;

    @FOREIGN
    private MachineSetting machineSetting=new MachineSetting();
    @FOREIGN
    private MachineArchitecture machineArchitecture=new MachineArchitecture();

    private ArrayList<MachineError> machineErrorList;
    private ArrayList<MachineWarning> machineWarningList;
    private ArrayList<NetworkInterface> networkInterfaceList;
    private ArrayList<MachineState> machineStateList;

    public Machine(){
    }

    public ArrayList<MachineState> getMachineStateList() {
        return machineStateList;
    }

    public void setMachineStateList(ArrayList<MachineState> machineStateList) {
        this.machineStateList = machineStateList;
    }

    public String getJarVersion() {
```

Figure 4.14 : utilisation de JPMapper

- XML

Le langage XML est utilisé tout au long du projet MRMC. Il est utilisé dans les fichiers de configuration comme ceux du serveur Tomcat et de Struts2. XML est aussi utilisé dans le développement de la console d'administration et surtout dans la partie envoi de l'état de la machine cliente surveillée. En effet, toutes les valeurs techniques envoyées au serveur sont sous forme de fichier XML.

```

MRMCFacade.java
public void getComputerInfo(){
    StringBuffer xml=new StringBuffer("<?xml version='1.0' encoding='UTF-8'?>");
    Computer computer=new Computer();
    computer.initialize();
    SimpleDateFormat simpleDateFormat=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    String ip="";
    if(computer.getNetInterfaceList()!=null){
        for(int j=0;j<computer.getNetInterfaceList().size();j++){
            if(computer.getNetInterfaceList().get(j).getPrimaryIpAddress()!=null){
                ip=computer.getNetInterfaceList().get(j).getPrimaryIpAddress();
                break;
            }
        }
    }
    xml.append("<Machine>");
    xml.append("<jarVersion><![CDATA["+version+"]]></jarVersion>");
    xml.append("<machineID><![CDATA["+this.machineID+"]]></machineID>");
    xml.append("<lastIpAddress><![CDATA["+ip+"]]></lastIpAddress>");
    xml.append("<dateTime><![CDATA["+simpleDateFormat.format((new Date()+))]></dateTime>");
    xml.append("<MachineSetting>");
    xml.append("<server><![CDATA["+this.server+"]]></server>");
    xml.append("<port><![CDATA["+this.port+"]]></port>");
    xml.append("<context><![CDATA["+this.context+"]]></context>");
    xml.append("<listeningPort><![CDATA["+this.listeningPort+"]]></listeningPort>");
    xml.append("<remoteAudioPort><![CDATA["+this.remoteAudioPort+"]]></remoteAudioPort>");
    xml.append("<timer><![CDATA["+this.timer+"]]></timer>");
    xml.append("<timerLastIp><![CDATA["+this.timerLastIp+"]]></timerLastIp>");
    xml.append("</MachineSetting>");
    xml.append("<Computer>");
    xml.append("<host><![CDATA["+computer.getHost()+"]]></host>");
    xml.append("<fqdn><![CDATA["+computer.getFqdn()+"]]></fqdn>");
    xml.append("<userName><![CDATA["+computer.getUserName()+"]]></userName>");
    xml.append("<cpusPhys><![CDATA["+computer.getCpusPhys()+"]]></cpusPhys>");
    xml.append("<corePerCpu><![CDATA["+computer.getCorePerCpu()+"]]></corePerCpu>");
    if(computer.getLanguage()!=null){
        xml.append("<language><![CDATA["+computer.getLanguage()+"]]></language>");
    }
}

```

Figure 4.15 : utilisation de XML

## 4.3 Développement de la solution MRMC

Dans cette partie je vais détailler techniquement l'utilisation des technologies et bibliothèques dans le projet MRMC. Pour formuler correctement et aisément cette démonstration je vais me contenter de présenter la fonctionnalité la plus importante du projet celle de l'envoi, réception et enregistrement des données reçues d'une machine cliente.

### 4.3.1 Développement de l'agent MRMC

Le développement de l'agent MRMC devait nécessairement et obligatoirement commencer en premier vu la dépendance de la console d'administration et du serveur MRMC aux données et informations techniques envoyées par les machines clientes.

L'agent MRMC est responsable de trois fonctionnalités importantes, l'extraction et l'envoi des données au serveur, le chat textuel et la demande d'assistance.

## A) Extraction et envoi de données

L'extraction des données est faite automatiquement en fonction de l'intervalle défini lors de l'installation de l'agent. Cette tâche est réalisée avec l'aide de la bibliothèque « Sigar » et génère un fichier XML qui sera envoyé au serveur pour être traité et enregistré. Ce fichier sera décortiqué et constituera un objet d'état d'une machine « MachineState » qui sera enregistré dans la base de données avec l'aide de la bibliothèque « jpMapper ».

Le schéma suivant explique ce mécanisme :

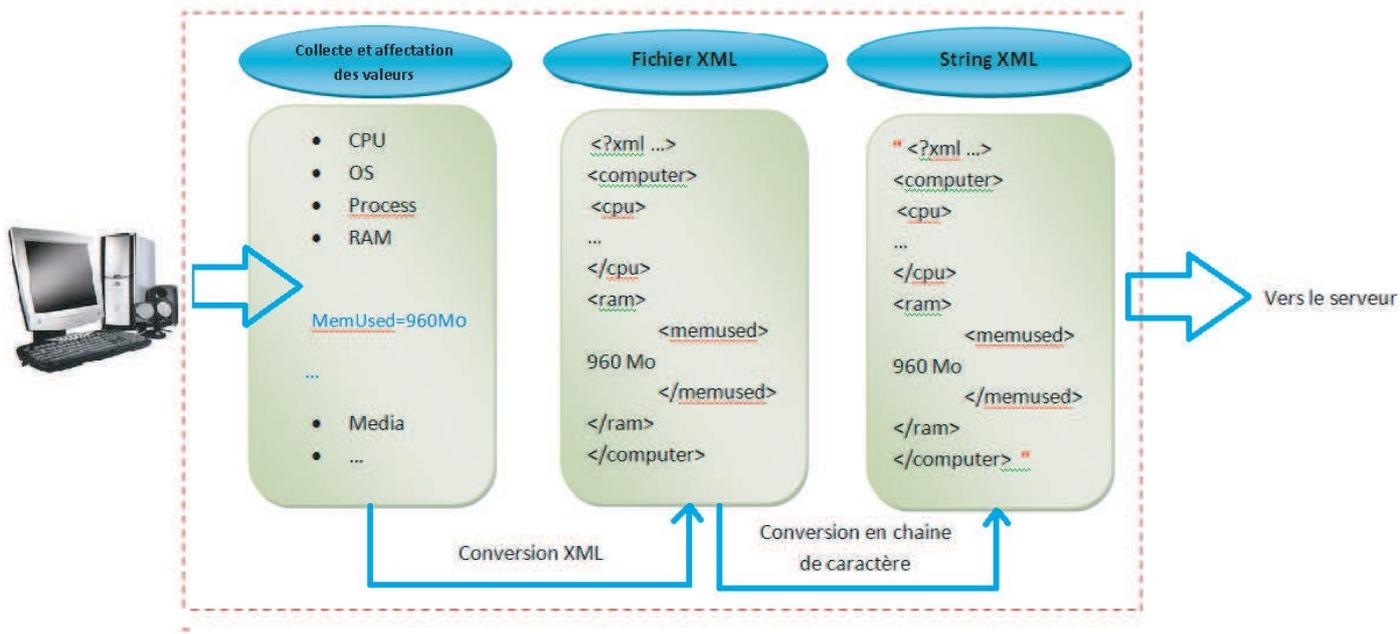


Figure 4.16: Traitement côté client

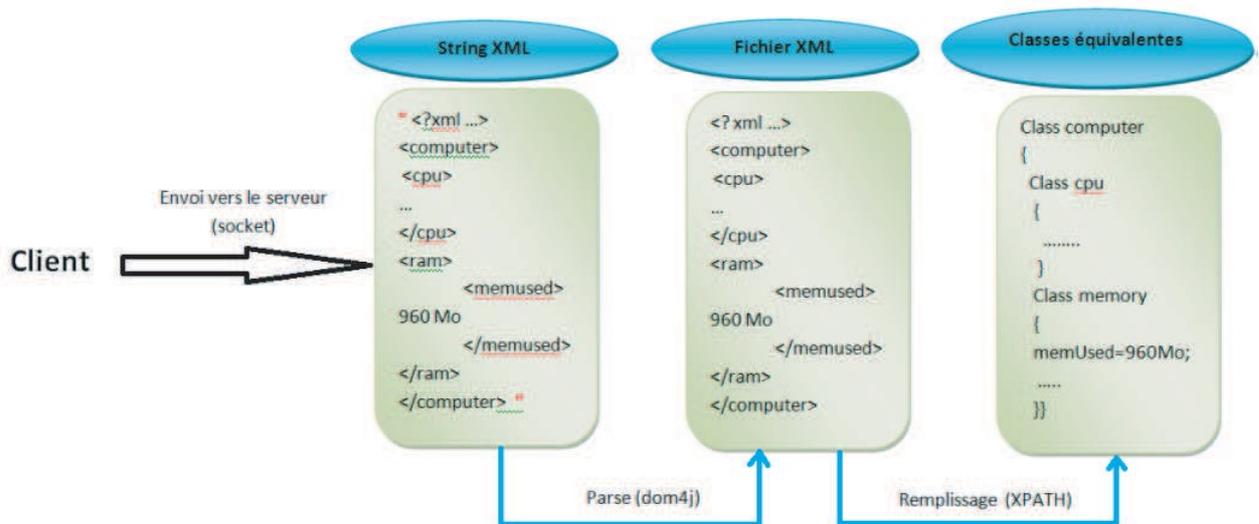


Figure 4.17: Traitement côté serveur

### a. Etape1: pré-extraction

Avant d'entamer l'étape d'extraction j'ai réalisé des classes qui correspondent aux différentes composantes d'un pc, les noms ainsi que les variables de ces dernières étaient choisis en se basant sur les noms qui existent dans la bibliothèque SIGAR et du diagramme de classes UML.

Les différentes classes sont présentées ainsi :



Figure 4.18: Présentation des classes

La classe computer.java englobe toutes les autres classes relatives aux composantes d'un ordinateur qui sont CPU.java, Media.java, Memory.java, NetworkConnection.java, NetworkInterface.java, OS.java et Process.java.

Prenons à titre d'exemple la classe Memory.java, elle est composée des attributs suivants :

```
Memory.java
public class Memory {
    private long memTotal;
    private long memUsed;
    private long memFree;
    private long swapTotal;
    private long swapUsed;
    private long swapFree;
    private long memRAM;
    private double memUsedPct;
    private double swapUsedPct;

    public Memory() {
    }

    public void initialize() {
        Sigar sigar = new Sigar();
        Mem mem = null;
        Swap swap = null;
        try {
            mem = sigar.getMem();
            this.memTotal = mem.getTotal();
            this.memUsed = mem.getUsed();
            this.memFree = mem.getFree();
            this.memRAM = mem.getRam();
            if(this.memTotal==0L){
                this.memUsedPct=0;
            }else{
                this.memUsedPct=(new Double(this.memUsed)/new Double(this.memTotal));
            }
            swap = sigar.getSwap();
            this.swapTotal = swap.getTotal();
            this.swapUsed = swap.getUsed();
            this.swapFree = swap.getFree();
            if(this.swapTotal==0L)this.swapUsedPct=0;
            else
                this.swapUsedPct=(new Double(this.swapUsed)/new Double(this.swapTotal));
        }
    }
}
```

Figure 4.20 : Extraction des informations de la mémoire

Les champs cités précédemment portent les mêmes noms des méthodes existantes dans la bibliothèque SIGAR, le but est de faire la relation entre les variables de la classe et les valeurs de retours des méthodes de l'API

### b. Etape 2 : Structuration des données

Les données collectées précédemment doivent être structurées sous un format XML puis envoyées sous forme de fichier XML. Pour chaque objet computer des nœuds sont ajoutés au fichier XML, La structuration est réalisée comme suit :

```

MRMCFacade.java
public void getComputerInfo(){
    StringBuffer xml=new StringBuffer("<?xml version=\"1.0\" encoding=\"UTF-8\"?>");
    Computer computer=new Computer();
    computer.initialize();
    SimpleDateFormat simpleDateFormat=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    String ip="";
    if(computer.getNetInterfaceList()!=null){
        for(int j=0;j<computer.getNetInterfaceList().size();j++){
            if(computer.getNetInterfaceList().get(j).getPrimaryIPAddress()!=null){
                ip=computer.getNetInterfaceList().get(j).getPrimaryIPAddress();
                break;
            }
        }
    }
    xml.append("<Machine>");
    xml.append("<jarVersion><![CDATA["+version+"]></jarVersion>");
    xml.append("<machineID><![CDATA["+this.machineID+"]></machineID>");
    xml.append("<lastIpAddress><![CDATA["+ip+"]></lastIpAddress>");
    xml.append("<dateTime><![CDATA["+simpleDateFormat.format(new Date())+"]></dateTime>");
    xml.append("<MachineSetting>");
        xml.append("<server><![CDATA["+this.server+"]></server>");
        xml.append("<port><![CDATA["+this.port+"]></port>");
        xml.append("<context><![CDATA["+this.context+"]></context>");
        xml.append("<listeningPort><![CDATA["+this.listeningPort+"]></listeningPort>");
        xml.append("<remoteAudioPort><![CDATA["+this.remoteAudioPort+"]></remoteAudioPort>");
        xml.append("<timer><![CDATA["+this.timer+"]></timer>");
        xml.append("<timerLastIp><![CDATA["+this.timerLastIp+"]></timerLastIp>");
    xml.append("</MachineSetting>");
    xml.append("<Computer>");
    xml.append("<host><![CDATA["+computer.getHost()+"]></host>");
    xml.append("<fqdn><![CDATA["+computer.getFqdn()+"]></fqdn>");
    xml.append("<userName><![CDATA["+computer.getUserName()+"]></userName>");
    xml.append("<cpusPhys><![CDATA["+computer.getCpusPhys()+"]></cpusPhys>");
    xml.append("<corePerCpu><![CDATA["+computer.getCorePerCpu()+"]></corePerCpu>");
    if(computer.getLanguage()!=null){
        xml.append("<language><![CDATA["+computer.getLanguage()+"]></language>");
    }else{
        xml.append("<language><![CDATA["+computer.getLanguageString()+"]></language>");
    }
}

```

Figure 4.21 : Création du fichier XML

### c. Etape 3 : Transformation des données

Cette étape consiste à mettre le contenu du fichier XML dans une variable de type chaîne de caractère, le but de cette étape est d'alléger le trafic réseau lorsque ce dernier est composé de plusieurs machines.

```

xml.append("</Computer>");
xml.append("</Machine>");
NetworkFacade networkFacade = new NetworkFacade(this.server,this.port,this.context,"createMachineState");
String xmlStr=xml.toString();
xmlStr=xmlStr.replaceAll("%", " ");
networkFacade.sendTextData("xml="+xmlStr);

```

Figure 4.22 : Envoi du fichier XML

### d. Etape 4 : Envoi et réception des données

Après la structuration des informations techniques sous forme de fichier XML, l'envoi est réalisé en utilisant la classe Socket qui sont des objets permettant la gestion des flux de données en utilisant le protocole TCP/IP.

Le schéma suivant visualise le rôle des sockets dans cette partie :

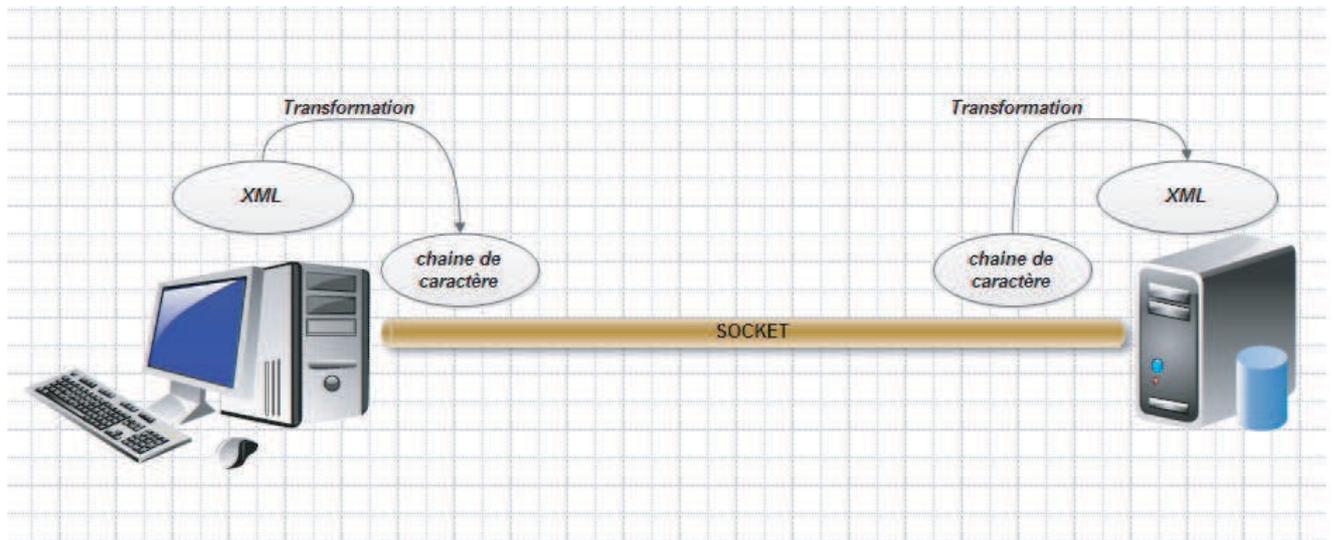


Figure 4.23: Envoi client/serveur

```

public String sendTextData(String text){
    /**
     * connect to server (http)
     */
    String cnt="";
    try {
        URL url = new URL("http://" + this.host + ":" + this.port + this.context + "/" + action);
        URLConnection conn = url.openConnection();
        conn.setConnectTimeout(8000);
        conn.setDoOutput(true);
        OutputStreamWriter writer = new OutputStreamWriter(conn.getOutputStream());
        //write parameters
        writer.write(text);
        writer.flush();
        writer.close();
        // Get the response
        InputStream is=conn.getInputStream();
        byte[] contentByte=new byte[is.available()];
        is.read(contentByte, 0, is.available());
        String content=new String(contentByte);
        cnt=content;
        return content;
    } catch (MalformedURLException e) {
        e.printStackTrace();
        return null;
    } catch (IOException e) {
        e.printStackTrace();
        System.out.println(cnt);
        return null;
    }
}
}

```

Figure 4.24 : Création du Socket pour l'envoi du fichier XML

Après l'envoi des données, les informations sont reçues dans le serveur sous format chaîne de caractère et transformées en un fichier XML en utilisant le parseur dom4j. Après cette étape des objets sont créés pour chaque classe et alimentés avec les informations relatives aux machines clientes. Pour se faire, le fichier XML obtenu est parcouru en utilisant le

langage XPATH (outils) pour obtenir les informations relatives aux différents nœuds du fichier.

```
private void setMachineFromXml(){
    machine.setMachineID(document.selectSingleNode("/Machine/machineID").getText()); // ajouter dans l xml
    machine.setIsConnected(1);
    String jarVersion = document.selectSingleNode("/Machine/jarVersion").getText();
    String lastIpAddress = document.selectSingleNode("/Machine/lastIpAddress").getText();
    String lastName = document.selectSingleNode("/Machine/Computer/host").getText();
    machine.setJarVersion(jarVersion);
    machine.setLastIpAddress(lastIpAddress);
    machine.setMachineSetting(machineSetting);
    machine.setLastName(lastName);
    SimpleDateFormat sdfSource = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    String strDate=document.selectSingleNode("/Machine/dateTime").getText();
    try {
        Date date = sdfSource.parse(strDate);
        machineState.setDateTime(new Timestamp(date.getTime()));
    } catch (ParseException e) {
        e.printStackTrace();
    }
}

private void setMachineSettingFromXml(){
    machineSetting.setServer(document.selectSingleNode("/Machine/MachineSetting/server").getText());
    machineSetting.setPort(document.selectSingleNode("/Machine/MachineSetting/port").getText());
    machineSetting.setContext(document.selectSingleNode("/Machine/MachineSetting/context").getText());
    machineSetting.setTimer(document.selectSingleNode("/Machine/MachineSetting/timer").getText());
    machineSetting.setListeningPort(document.selectSingleNode("/Machine/MachineSetting/listeningPort").getText());
    machineSetting.setTimerLastIp(document.selectSingleNode("/Machine/MachineSetting/timerLastIp").getText());
    machine.setMachineSetting(machineSetting); // verifier !! et l'objet machineState ( inexistant ds le fichier Xml )
}

private void setComputerFromXml(){
    computer.setHost(document.selectSingleNode("/Machine/Computer/host").getText());
    //String host = document.selectSingleNode("//host").getText();
    computer.setFqdn(document.selectSingleNode("/Machine/Computer/fqdn").getText());
    computer.setUserName(document.selectSingleNode("/Machine/Computer/userName").getText());
    computer.setLanguage(document.selectSingleNode("/Machine/Computer/language").getText());
}
}
```

Figure 4.25 : parse du fichier XML

**e. Etape 5 : Enregistrement des données**

Les informations relatives aux différentes machines du parc sont stockées dans une base de données MySQL, les tables et les champs de cette dernière sont basés sur le diagramme des classes présenté précédemment.

Le schéma suivant explique ce mécanisme :

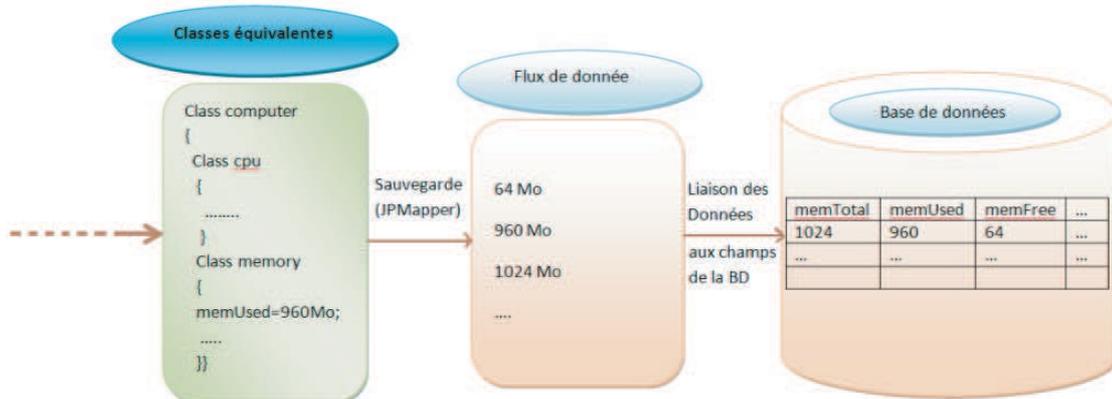


Figure 4.26: Enregistrement des données

A l'aide de PhpMyAdmin, la base de données a été créée ainsi :



Figure 4.27: Base de données dans MySQL

Avec JPMapper, il y a plusieurs façons d'enregistrer des données dans MySQL. C'est en fonction de la complexité de la tâche de sauvegarde ou même de lecture qu'il faut choisir la méthode adéquate. En effet, si la tâche est simple on utilise un accès direct à la base de données, et on utilise les méthodes de base de JPMapper (put, get, list, delete, etc), mais si la tâche est complexe on utilise le mode transactionnel ou même on pourra créer une classe DBService.

Prenons l'exemple d'enregistrement de l'état d'une machine. Ainsi pour enregistrer les données reçues d'une machine cliente, il faut simplement créer un objet Etat de la machine « MachineState » et l'alimenter des différentes informations techniques reçues lors de l'étape précédente. En suite, en utilisant la bibliothèque JPMapper il faut créer une connexion au serveur MySQL et sauvegarder l'objet créé dans la table correspondante.

```
machineStateInstance=util.getMachineStateFormXml();
machineStateInstance.getMachine().setLastIpAddress(clientIP);
try {
    MachineState machineState=(MachineState) getDBService().run(MachineStateDBService.class, serviceParameters, "create", this.machineStateInstance);
    Boolean lg= (Boolean) service.run(MachineErrorDBService.class, serviceParameters, "create", this.machineStateInstance);
```

Création de l'objet machineState

Enregistrement dans la BD en utilisant un DBService

Figure 4.28 : création de l'objet « MachineState » et appel du « DBService »

```

MachineStateDBService.java
public MachineState create(MachineState machineState) throws TechnicalServiceException{
    Transaction transaction = null;
    try {
        transaction=this.jpMapperFacade.getNewTransaction();
        transaction.begin();
        //Save Memory
        Long memoryID = transaction.put(machineState.getComputer().getMemory());
        //Save computer
        machineState.getComputer().getMemory().setMemoryID(memoryID);
        Long computerID=transaction.put(machineState.getComputer());
        machineState.getComputer().setComputerID(computerID);
        //save CPU
        for(int i=0;i<machineState.getComputer().getCpuList().size();i++){
            CPU cpu=machineState.getComputer().getCpuList().get(i);
            cpu.getComputer().setComputerID(computerID);
            Long cpuID=transaction.put(cpu);
            cpu.setCpuID(cpuID);
        }
        //Save Medias
        for(int i=0;i<machineState.getComputer().getMediaList().size();i++){
            Media media=machineState.getComputer().getMediaList().get(i);
            if(!media.getFileSystemName().equalsIgnoreCase("null"))
            {
                media.getComputer().setComputerID(computerID);
                Long mediaID=transaction.put(media);
                media.setMediaID(mediaID);
            }
        }
        //Save Process
        //HashMap<Long, Long> prc=new HashMap<Long, Long>();
        for(int i=0;i<machineState.getComputer().getProcessList().size();i++){
            Process process=machineState.getComputer().getProcessList().get(i);
            process.getComputer().setComputerID(computerID);
            // il faut creer la var networkConnectionID et la remplir
            Long processID=transaction.put(process);
            process.setProcessID(processID);
            //prc.put(process.getPid(), processID);
        }
    }
}

```

Figure 4.29 : « DBService » avec mode transactionnel

## B) Lecture et présentation des données

Pour afficher n'importe quelle donnée enregistrée dans la base, il faut utiliser le modèle MVC. Toujours avec le même exemple, pour afficher l'état d'une machine, il faut passer par le contrôleur pour accéder à la base, extraire les informations, créer l'objet correspondant du modèle, alimenter l'objet avec les valeurs extraites et en fin faire passer cet objet à la vue, généralement une page JSP, pour qu'elle l'affiche .

Cette partie est schématisée comme suit :

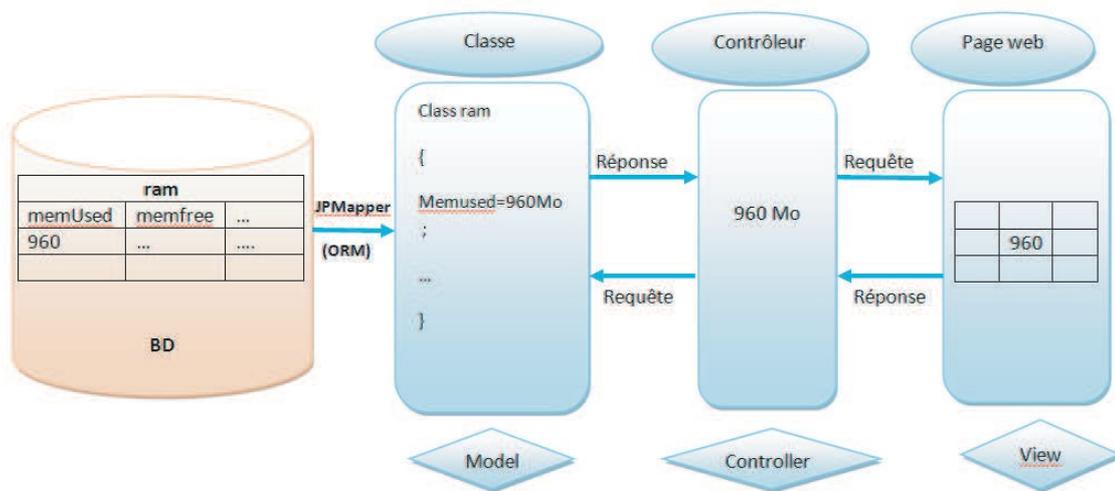


Figure 4.30: Présentation des données avec MVC

```
MachineStateController.java
public String getMachineState() {
    this.getDBService();
    HashMap<String, Object> lastHash=new HashMap<String, Object>();
    ArrayList<MachineError> machineError=new ArrayList<MachineError>();
    ArrayList<MachineWarning> machineWarning=new ArrayList<MachineWarning>();
    Machine machine=new Machine();
    lastMachineState=new MachineState();
    Gson gson=new Gson();
    machine.setMachineID(machineID);
    try {
        lastHash=(HashMap<String, Object>) service.run(MachineStateDBService.class, serviceParameters, "getLast", machine);
        if(lastMachineState!=null){
            lastMachineState=(MachineState) lastHash.get("lastMachineState");
            machineError=(ArrayList<MachineError>) lastHash.get("machineError");
            machineWarning=(ArrayList<MachineWarning>) lastHash.get("machineWarning");

            JSONObject jsonObject=new JSONObject(gson.toJson(lastMachineState));
            jsonObject.put("sizeDisplayFormat", getConfig().getProperty("sizeDisplayFormat"));
            jsonObject.put("decimal", getConfig().getProperty("decimal"));

            String machineErrorString = gson.toJson(machineError);
            JsonParser parser = new JsonParser();
            JSONArray errorJSONArray = parser.parse(machineErrorString).getAsJSONArray();

            String machineWarningString = gson.toJson(machineWarning);
            //JsonParser warningParser = new JsonParser();
            JSONArray warningJSONArray = parser.parse(machineWarningString).getAsJSONArray();

            jsonObject.put("machineErrorList", errorJSONArray);
            jsonObject.put("machineWarningList", warningJSONArray);
            data=new StringBufferInputStream(jsonObject.toString());
            //data=new StringBufferInputStream((gson.toJson(lastMachineState)));
        }
    } catch (TechnicalServiceException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

Extraction des données



Passer l'objet à la page JSP sous format JSON

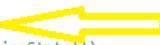


Figure 4.31 : Extraction et passage des données

```

getMachineState.jsp
<%@ taglib prefix="s" uri="/struts-tags"%>
<%@ page import="java.util.Properties;" %>
<%@ page language="java" contentType="text/html; charset=utf-8"
pageEncoding="utf-8"%>
<!--[if lt IE 9]><script language="javascript" type="text/javascript" src="js/lib/jqPlot/excanvas.js"></script><![endif]-->
<link href="css/lib/jqPlot/jquery.jqplot.min.css" rel="stylesheet" type="text/css" />
<script language="javascript" type="text/javascript" src="js/lib/jqPlot/jquery.jqplot.min.js"></script>
<script language="javascript" type="text/javascript" src="js/lib/jqPlot/jqplot.pieRenderer.min.js"></script>
<script language="javascript" type="text/javascript" src="js/lib/jqPlot/jqplot.barRenderer.min.js"></script>
<script language="javascript" type="text/javascript" src="js/lib/jqPlot/jqplot.categoryAxisRenderer.min.js"></script>
<script language="javascript" type="text/javascript" src="js/lib/jqPlot/jqplot.logAxisRenderer.min.js"></script>
<script language="javascript" type="text/javascript" src="js/lib/jqPlot/jqplot.canvasTextRenderer.min.js"></script>
<script language="javascript" type="text/javascript" src="js/lib/jqPlot/jqplot.canvasAxisLabelRenderer.min.js"></script>
<script language="javascript" type="text/javascript" src="js/lib/jqPlot/jqplot.canvasAxisTickRenderer.min.js"></script>
<script language="javascript" type="text/javascript" src="js/lib/jqPlot/jqplot.dateAxisRenderer.min.js"></script>
<script language="javascript" type="text/javascript" src="js/lib/jqPlot/jqplot.enhancedLegendRenderer.min.js"></script>
<script type="text/javascript" src="js/lib/jqPlot/jqplot.pointLabels.min.js"></script>
<input id="currentDiv" value="machineDiv" type="hidden"/>
<input id="machineID" value="" type="hidden"/>
<input id="machineIP" value="" type="hidden"/>
<div>
<ul class="tabs">
<li class="current" id="td_machineDiv"><a id="lk_machineDiv" href="#"><s:property value="getText('index.machine')"/></a></li>
<li id="td_cpuDiv"><a id="lk_cpuDiv" href="#"><s:property value="getText('index.cpu')"/></a></li>
<li id="td_memoryDiv"><a id="lk_memoryDiv" href="#"><s:property value="getText('index.memory')"/></a></li>
<li id="td_mediaDiv"><a id="lk_mediaDiv" href="#"><s:property value="getText('index.media')"/></a></li>
<li id="td_networkInterfaceDiv"><a id="lk_networkInterfaceDiv" href="#"><s:property value="getText('index.netInterface')"/></a></li>
<li id="td_netStatusDiv"><a id="lk_netStatusDiv" href="#"><s:property value="getText('index.netStatus')"/></a></li>
<li id="td_processDiv"><a id="lk_processDiv" href="#"><s:property value="getText('index.process')"/></a></li>
<li id="td_errorDiv"><a id="lk_errorDiv" href="#"><s:property value="getText('index.error')"/></a></li>
<li class="noBorder" id="td_warningDiv"><a id="lk_warningsDiv" href="#"><s:property value="getText('index.warning')"/></a></li>
</ul>
</div>
<div class="tools">
<div class="iconTools">
<span class="print" id="printMachineState"></span>

```

Figure 4.32 : Présentation des données avec une page JSP

## 4.4 Conclusion

Dans cette étude technique j'ai essayé de présenter la majorité des technologies utilisées dans le développement de MRMC et dans laquelle j'ai suivi un classement chronologique et par ordre d'importance. Dans le chapitre suivant je vais détailler les fonctionnalités du projet final MRMC.



# Chapitre 5

## Présentation de MRMC

Dans ce chapitre je vais essayer de présenter les fonctionnalités développées dans la solution MRMC. Dans cette démonstration je vais débiter par la partie Agent de surveillance et ensuite passer à la console d'administration.

Pour la console d'administration, je vais détailler la partie supervision, le paramétrage des alertes et erreurs, la gestion de l'assistance et enfin le tableau de bord.

### 5.1 Présentation de l'agent

Pour l'agent, la présentation mettra la lumière sur l'installation de l'agent, l'extraction et la collecte des informations des différentes machines du réseau, leur enregistrement et la gestion des demandes d'assistance.

#### 5.1.1 Installation

Pour pouvoir surveiller une machine du réseau, il faut installer l'agent MRMC. Cette installation est réalisée par un auto-installeur créé avec un logiciel empaqueteur nommé « InnoSetup ». Durant l'installation, plusieurs paramètres doivent être renseignés.

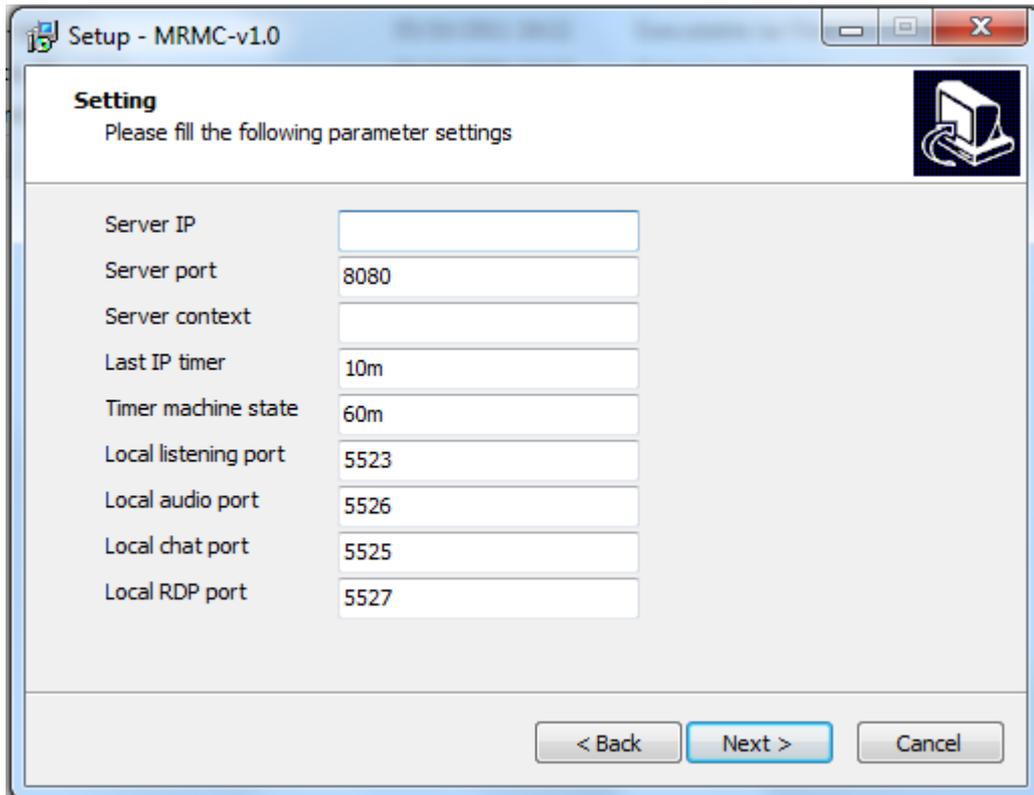


Figure 5.1 : Paramétrage de l'agent MRMC

Le premier paramètre à renseigner est le nom ou l'adresse IP du serveur de surveillance, le port du serveur TomCat qui est par défaut 8080 ainsi que le contexte de l'application MRMC installée sur le serveur en question. Un serveur MRMC peut avoir un ou plusieurs contextes et chaque contexte peut être affecté à un groupe de machines, cette option est à utiliser dans les grands parcs informatiques mais dans les petits un seul contexte suffit largement.

Les autres ports sont utilisés dans la partie assistance à distance. Le port audio pour la conversation audio, le port chat pour la conversation textuel et le port RDP est pour la prise de contrôle. Si les paramètres par défaut lors de l'installation de l'application serveur MRMC n'ont pas été modifié, elle faut laisser ces valeurs à leurs valeurs initiales.

Après l'installation de l'agent, la machine cliente est ajoutée à la liste des équipements surveillés et un icône est ajouté dans la barre de tâche. Cet icône servira de point de contrôle de l'application cliente.

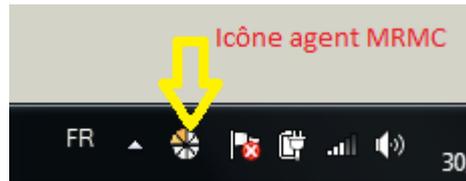


Figure 5.2 : Agent MRMC installé

## 5.1.2 Demande d'assistance

L'agent MRMC dispose d'un menu contextuel qui offre deux options pour demander une assistance, la première étant la création d'un ticket et la deuxième le chat textuel avec un agent de l'équipe technique.

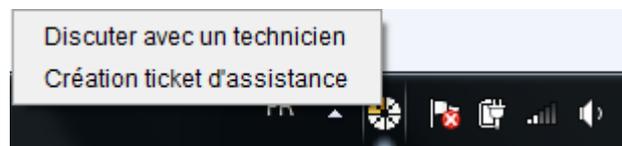


Figure 5.3 : Menu Contextuel de l'agent MRMC

### a) Création de ticket

Cette fonctionnalité est accessible en activant le menu textuel de l'agent MRMC en cliquant avec le bouton droit de la souris et en suite choisir création ticket d'assistance. L'utilisateur de la machine cliente doit renseigner plusieurs informations pour que le ticket soit validé, il doit donner un sujet et description précise de l'incident, définir une priorité et choisir un agent parmi ceux disponible. La date et le statut sont définis automatiquement et ne peuvent être modifiés par l'utilisateur. Après la création du ticket l'agent est informé lors de son accès à la console d'administration

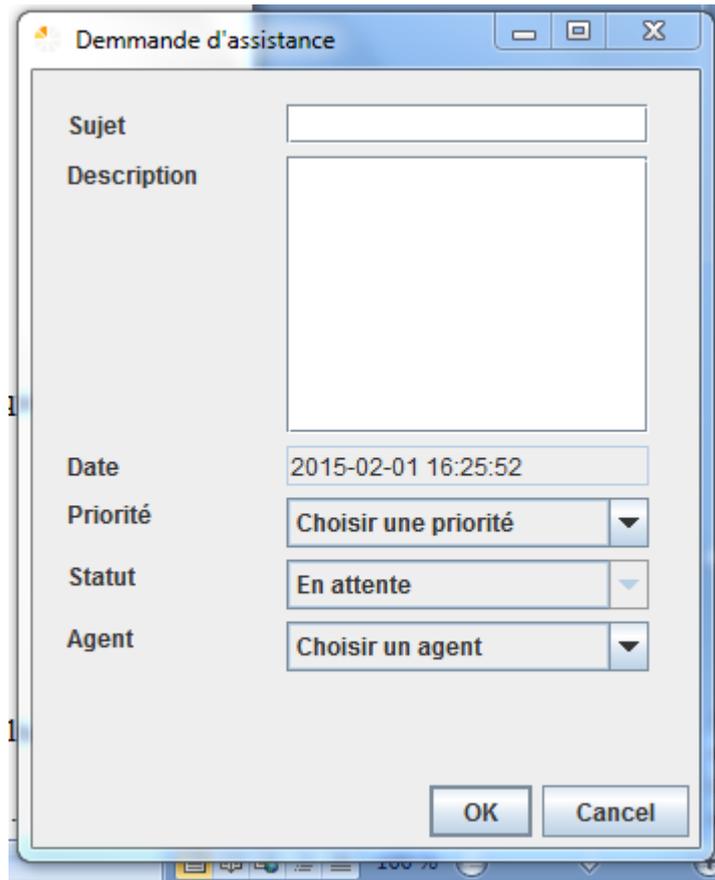


Figure 5.4 : Création de ticket d'assistance

### b) Chat textuel

Pour accéder à cette option, l'utilisateur doit choisir l'option « Discuter avec un technicien », en suite une fenêtre est afficher permettant le chat avec un technicien disponible.

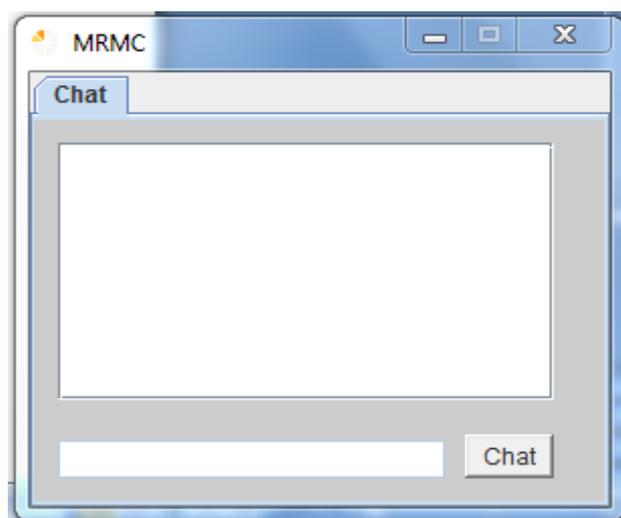


Figure 5.5 : Chat avec un technicien

## 5.2 Présentation de la Console d'administration

Dans cette partie, je vais présenter les fonctionnalités offertes par la console d'administration qui constitue l'outil de contrôle mis à la disposition de tout administrateur ou technicien du parc informatique.

En effet, après que l'utilisateur de l'application soit authentifié, la console d'administration offre plusieurs fonctionnalités importantes et qui peuvent être regroupées en trois groupes principaux, le premier étant les outils de supervision, la gestion de l'assistance aux utilisateurs, et le troisième outil est le tableau de bord.

### 5.2.1 Authentification

Dès l'exécution de la console d'administration, une fenêtre d'authentification est affichée, après que l'utilisateur ait saisi correctement son « login » et son « mot de passe », l'application est affichée en mettant à la disposition de l'administrateur ou de l'agent les fonctionnalités qui lui sont autorisées.

Il faut signaler que MRMC étant un module d'un ERP général, la partie de création des utilisateurs et de l'affectation des droits et autorisation est accessible avec un autre module de l'ERP nommé « MindWare Control Panel » ou « MCP ».

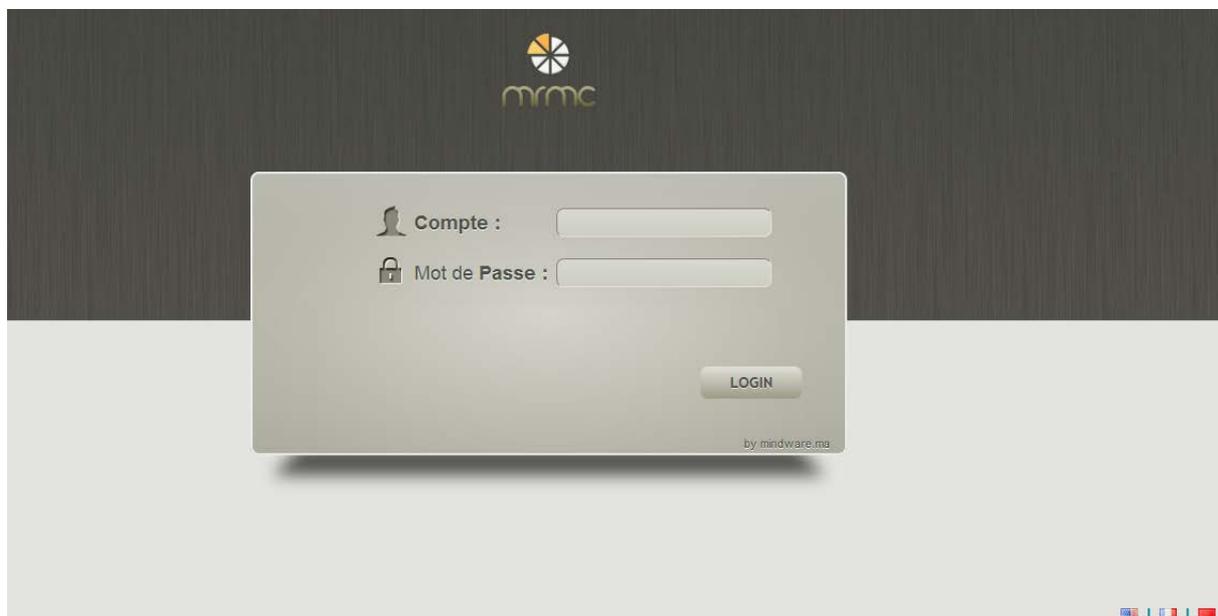


Figure 5.6 : Fenêtre d'authentification

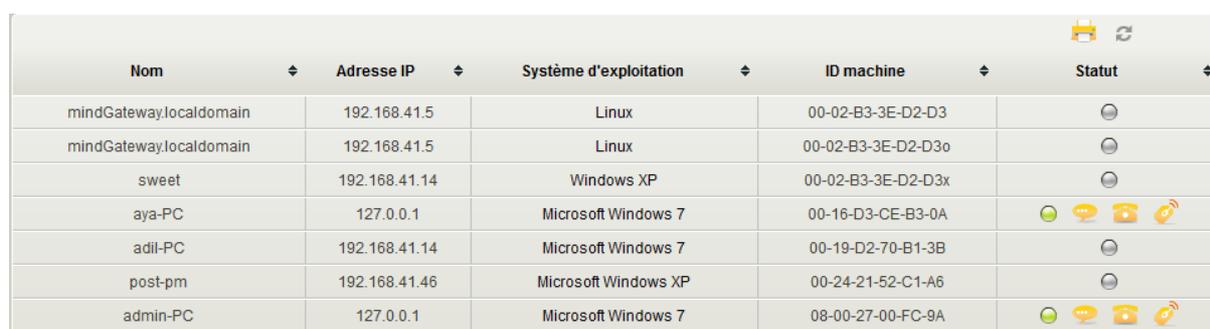
## 5.2.2 Supervision du parc

La console d'administration dispose de plusieurs outils pour superviser le réseau parmi lesquels on peut citer la liste des machines surveillées, la liste des alertes et erreur et en fin le paramétrage de détection des incidents.

### a) Liste des machines

La première interface affichée après l'authentification de l'utilisateur est la liste des machines surveillées, elle liste tous les équipements surveillés sur lesquels l'agent MRMC est installé. Cette liste est affichée sous forme de tableau paginé avec plusieurs colonnes et une possibilité d'impression et de rafraîchissement.

La colonne statut permet de donner l'état actuel de la machine, si elle est connectée un cercle vert est affiché sinon c'est un cercle gris qui est affiché. La même colonne offre d'autres fonctionnalités qui seront détaillées plus tard.



Nom	Adresse IP	Système d'exploitation	ID machine	Statut
mindGateway.localdomain	192.168.41.5	Linux	00-02-B3-3E-D2-D3	●
mindGateway.localdomain	192.168.41.5	Linux	00-02-B3-3E-D2-D3o	●
sweet	192.168.41.14	Windows XP	00-02-B3-3E-D2-D3x	●
aya-PC	127.0.0.1	Microsoft Windows 7	00-16-D3-CE-B3-0A	● 🗨️ 📶 📡
adil-PC	192.168.41.14	Microsoft Windows 7	00-19-D2-70-B1-3B	●
post-pm	192.168.41.46	Microsoft Windows XP	00-24-21-52-C1-A6	●
admin-PC	127.0.0.1	Microsoft Windows 7	08-00-27-00-FC-9A	● 🗨️ 📶 📡

Figure 5.7 : Liste des machines surveillées

### b) Etat d'une machine

Pour avoir plus de détail sur une machine précise, l'utilisateur doit cliquer sur la ligne correspondante dans la liste des machines. Après avoir fait son choix, l'application affichera un état complet de la machine avec des informations statiques et d'autres dynamiques.

Cette interface est devisée en plusieurs onglets :

**L'onglet machine** : affichant une description de la machine surveillé avec des informations statiques sur le système d'exploitation installé, l'architecture du microprocesseur, le nom complet, la langue, etc.

Machine	Processeur	Memoire	Media	Interface réseau	Statut réseau	Processus	Erreur	Avertissement
Nom de la machine	admin-PC							
Date	Feb 1, 2015 11:53:32 PM							
Marque	Intel							
Modèle	Core(TM)2 Duo CPU U7600 @ 1.20GHz							
Nom complet	admin-PC.Home							
Utilisateur	admin							
langue	1036:French (France)							
Nombre de microprocesseur	2							
Description	Microsoft Windows 7							
Service Pack	Service Pack 1							
Architecture	x64							
Version JAVA	Oracle Corporation							
JAVA_HOME	C:\Program Files\Java\jre7							

Figure 5.8 : description de la machine surveillée

**L'onglet processeur** : affichant le nombre de microprocesseur logique de la machine surveillé avec des indicateurs de charge détectés lors de sa dernière supervision.

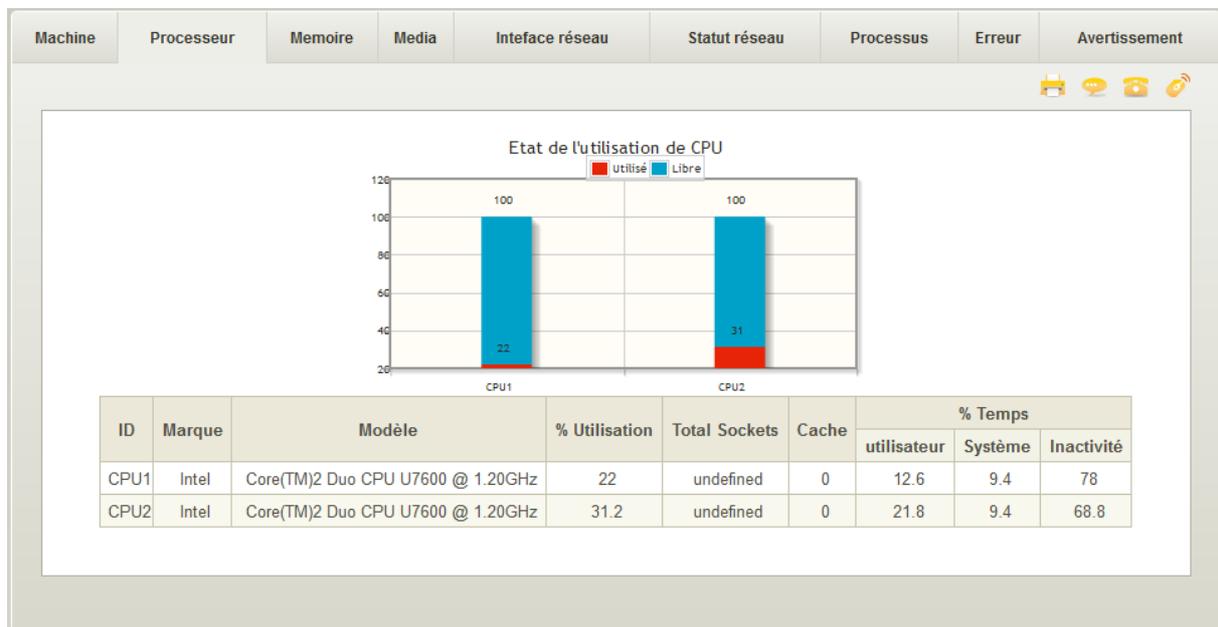


Figure 5.9 : description des microprocesseurs

**L'onglet mémoire** : affichant la taille de la mémoire vive et virtuelle installée avec des indicateurs de chargement de la mémoire détectés lors de sa dernière supervision.

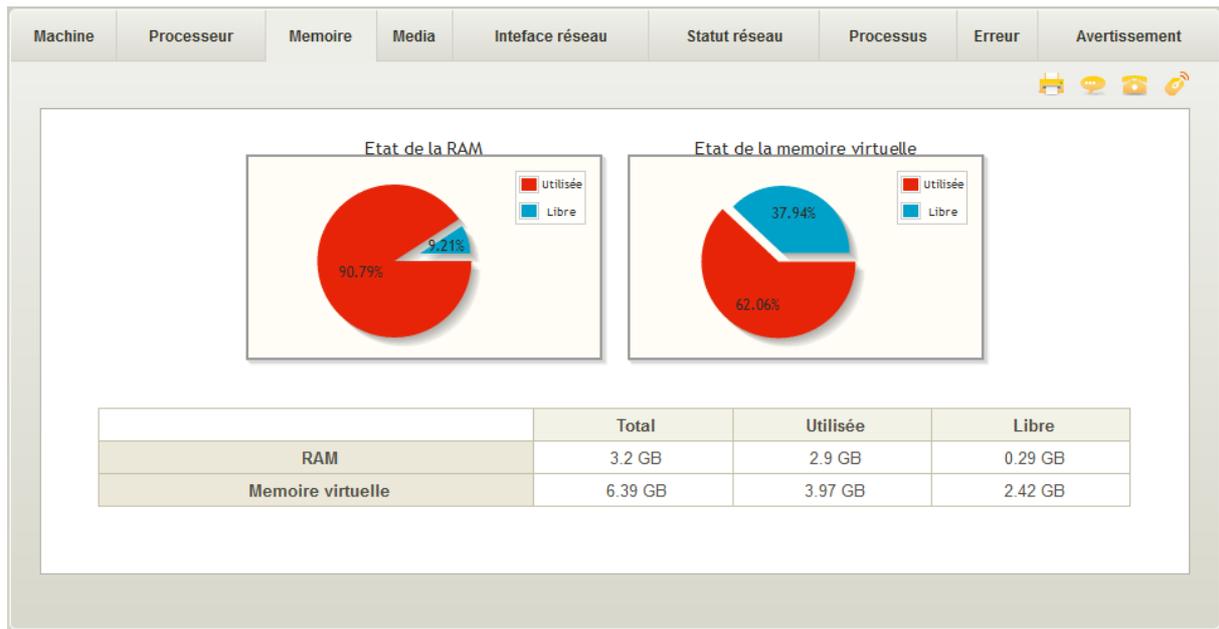


Figure 5.10 : Etat de la mémoire

**L'onglet média** : affichant la liste des disques durs de la machine surveillée avec des indicateurs des capacités et de système de fichier des différentes partitions et leur niveau d'espace libre et plein.



Figure 5.11 : Etat des médias

**L'onglet interfaces réseau** : affichant les différentes cartes réseau avec leur nom, adresse IP et MAC et leur statut

Machine	Processeur	Memoire	Media	Interface réseau	Statut réseau	Processus	Erreur	Avertissement
Nom	Nom complet			@ IP	@ MAC	MTU	Active	
eth0	WAN Miniport (IPv6)			0.0.0.0	78-95-20-52-41-53	1500	1	
eth1	WAN Miniport (Network Monitor)			0.0.0.0	78-95-20-52-41-53	1500	1	
eth2	Connexion r?seau Intel(R) 82566MM Gigabit - VirtualBox Bridged Networking Driver Miniport			0.0.0.0	00-1F-29-BD-38-F6	1500	0	
eth3	Connexion r?seau Intel(R) 82566MM Gigabit - VirtualBox Bridged Networking Driver Miniport-QoS Packet Scheduler-000			0.0.0.0	00-1F-29-BD-38-F6	1500	0	
eth4	WAN Miniport (IP)			0.0.0.0	78-95-20-52-41-53	1500	1	
eth5	Connexion r?seau Intel(R) 82566MM Gigabit - VirtualBox Bridged Networking Driver Miniport-WFP LightWeight Filter-000			0.0.0.0	00-1F-29-BD-38-F6	1500	0	
eth6	Connexion r?seau Intel(R) 82566MM Gigabit			0.0.0.0	00-1F-29-BD-38-F6	1500	0	
eth7	Connexion r?seau Intel(R) 82566MM Gigabit - VirtualBox Bridged Networking Driver Miniport-SCF packet filter-000			0.0.0.0	00-1F-29-BD-38-F6	1500	0	
eth8	VirtualBox Host-Only Ethernet Adapter-QoS Packet Scheduler-0000			0.0.0.0	08-00-27-00-FC-9A	1500	1	
eth9	P?riph?rique Bluetooth (r?seau personn			0.0.0.0	00-1E-37-03-FF-6C	0	0	
eth10	VirtualBox Host-Only Ethernet Adapter-WFP LightWeight Filter-0000			0.0.0.0	08-00-27-00-FC-9A	1500	1	
eth11	VirtualBox Host-Only Ethernet Adapter-SCF packet filter-0000			0.0.0.0	08-00-27-00-FC-9A	1500	1	
eth12	WAN Miniport (IPv6)-QoS Packet Scheduler-0000			0.0.0.0	78-95-20-52-41-53	1500	1	
eth13	WAN Miniport (IPv6)-SCF packet filter-0000			0.0.0.0	78-95-20-52-41-53	1500	1	
eth14	WAN Miniport (IP)-QoS Packet Scheduler-0000			0.0.0.0	78-95-20-52-41-53	1500	1	
eth15	Intel(R) Wireless WiFi Link 4965AG - VirtualBox Bridged Networking Driver Miniport			0.0.0.0	00-13-E8-AB-C8-6D	0	0	
eth16	WAN Miniport (IP)-SCF packet filter-0000			0.0.0.0	78-95-20-52-41-53	1500	1	

Figure 5.12 : liste des interfaces réseau

**L'onglet statut réseau** : affichant les différentes connexions actives dans la machine surveillée avec des indicateurs sur le protocole utilisé et les ports ouverts

Machine	Processeur	Memoire	Media	Interface réseau	Statut réseau	Processus	Erreur	Avertissement
								
Protocol	@ locale	Port local	@ distante	Port distant	Etat			
tcp	127.0.0.1	49861	127.0.0.1	3306	TIME_WAIT			
tcp	127.0.0.1	49860	127.0.0.1	8080	ESTABLISHED			
tcp	127.0.0.1	49857	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49856	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49846	127.0.0.1	8080	ESTABLISHED			
tcp	127.0.0.1	49845	127.0.0.1	49844	ESTABLISHED			
tcp	127.0.0.1	49844	127.0.0.1	49845	ESTABLISHED			
tcp	127.0.0.1	49841	127.0.0.1	49842	TIME_WAIT			
tcp	127.0.0.1	49833	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49832	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49759	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49758	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49749	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49748	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49747	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49746	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49745	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49744	127.0.0.1	3306	ESTABLISHED			
tcp	127.0.0.1	49743	127.0.0.1	3306	ESTABLISHED			

Figure 5.13 : liste des connexions réseau

**L'onglet processus** : affichant la liste des processus en cours d'exécution sur la machine surveillée avec des indicateurs sur la taille mémoire occupée, la charge du processeur, etc.

PID	Utilisateur	heure de début	Memoire Utilisée (MB)	Etat	Temps CPU	Nom
4	unknown	NA	5.49	R	NA	System
280	Système	-394835600:-8:-23	4.12	R	0:2:4	SystemRoot\System32\smss.exe
408	Système	-394835602:-58:-26	49.66	R	0:17:55	SystemRoot\system32\csrss.exe
464	Système	-394835603:-14:-33	53.77	R	0:3:6	wininit.exe
476	Système	-394835603:-14:-48	219.71	R	3:36:18	SystemRoot\system32\csrss.exe
524	Système	-394835603:-33:-47	61.27	R	0:55:7	C:\Windows\system32\services.exe
548	Système	-394835603:-35:-52	62.73	R	0:9:21	winlogon.exe
556	Système	-394835603:-35:-52	59.75	R	1:8:6	C:\Windows\system32\sass.exe
564	Système	-394835603:-35:-52	38.35	R	0:2:36	C:\Windows\system32\lsme.exe
696	unknown	-394835604:-3:-57	58.74	R	2:7:23	C:\Windows\system32\svchost.exe
760	Système	-394835604:-9:-24	56.39	R	0:1:48	C:\Windows\SysWOW64\svchost.exe
780	Système	-394835604:-18:-15	69.51	R	0:29:7	C:\Windows\SysWOW64\svchost.exe
812	Système	-394835604:-20:-51	50.27	R	0:0:46	C:\Program Files (x86)\Hewlett-Packard\File Sanitizer\HPFSService.exe
864	unknown	-394835604:-39:-3	62.91	R	0:35:52	C:\Windows\system32\svchost.exe
956	unknown	-394835604:-10:-26	97.4	R	0:45:29	C:\Windows\System32\svchost.exe

Figure 5.14 : liste des processus

L'onglet **erreur** : affichant les erreurs détectées sur la machine avec leur proportion comparé au total des erreurs du parc

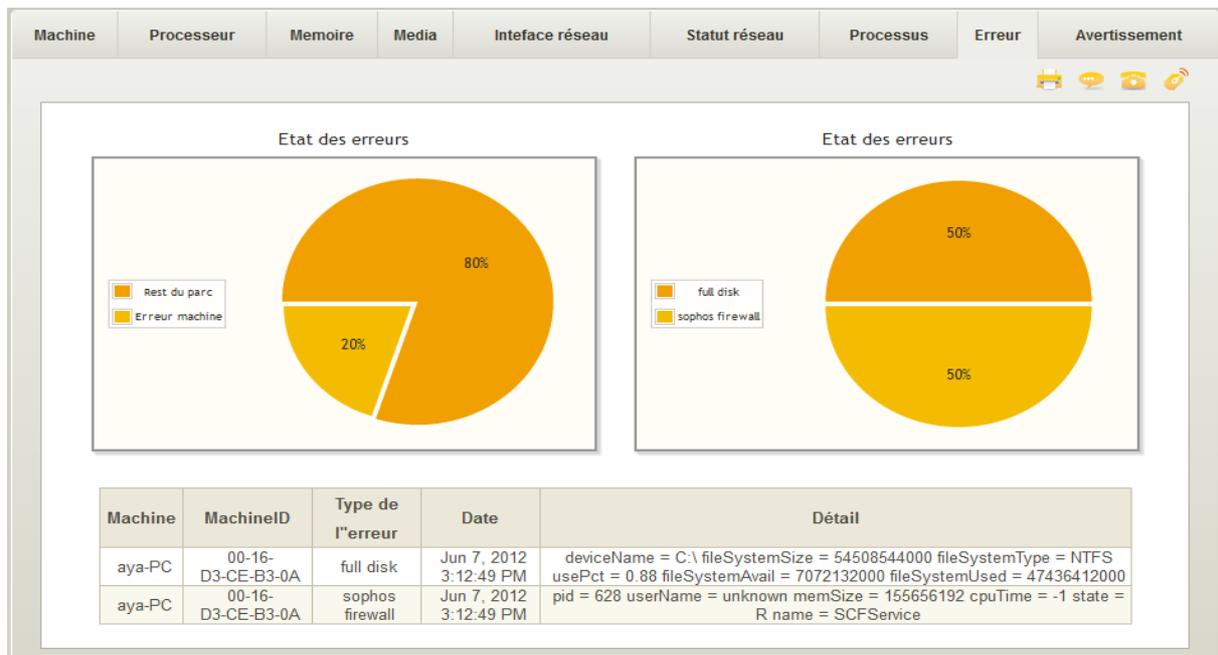


Figure 5.15 : liste des erreurs

**L'onglet des alertes** : affichant la liste des alertes détectées dans la machine surveillées avec des indicateurs de proportion comparé au reste du parc.

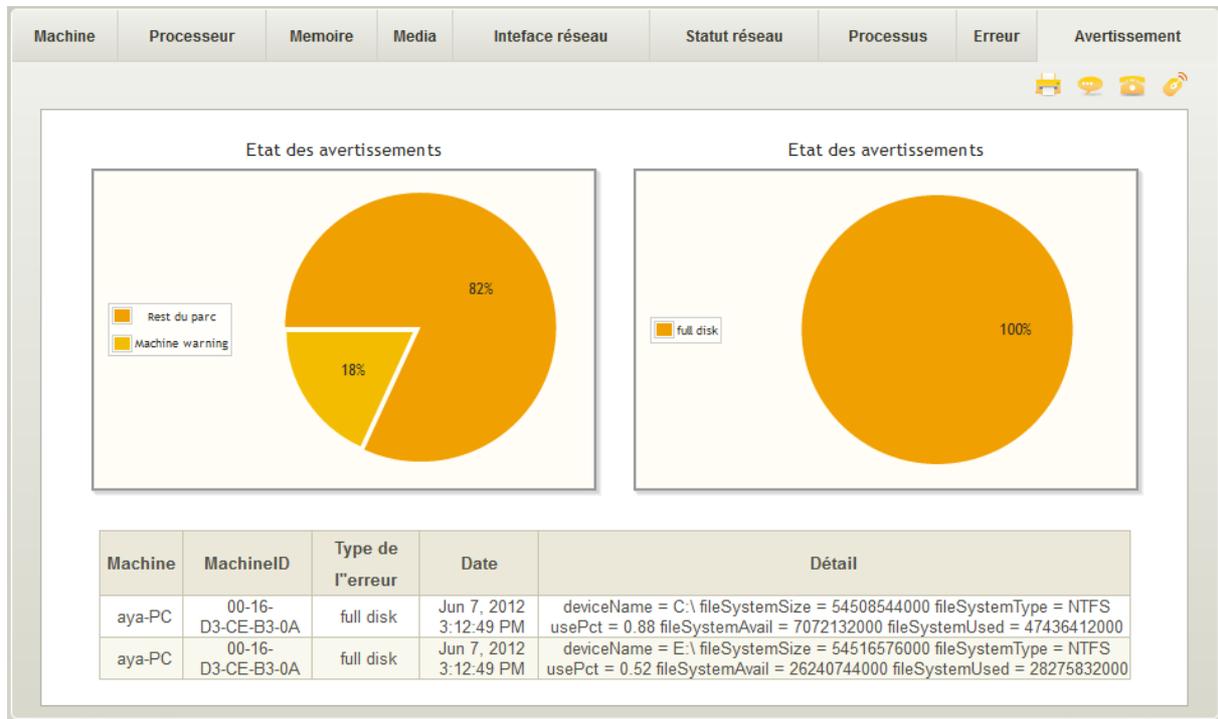


Figure 5.16 : Liste des alertes

### 5.2.3 Paramétrage des alertes et erreurs

MRMC permet d'automatiser la détection des erreurs ou alertes dans les machines clientes. Pour cela, il suffit simplement de choisir la source de l'erreur (Média, processeur, mémoire, etc.) et de spécifier le seuil de l'alerte et celui de l'erreur.

Prenons un exemple, si on veut détecter le niveau d'espace libre d'un disque des machines du parc et affecter un seuil de 50% pour les alertes et 70% pour les erreurs, il suffit de créer le paramétrage suivant :

Figure 5.17 : paramétrage des alertes et erreurs

Après la détection des alertes et erreurs, elles sont affichées en haut de la console d'administration avec un indicateur du nombre de machines concernées



Figure 5.18 : Nombre d'alerte et erreurs

Si l'utilisateur veut avoir plus de détail sur les erreurs détectées sur le parc, il suffit qu'il clique sur le lien « ERREUR » de la figure précédente. Le détail des erreurs comporte plusieurs indicateurs de date de détection, la machine concernée, le type et le détail de l'erreur.

Machine	MachineID	Type de l'erreur	Date	Détail
post-pm	00-24-21-52-C1-A6	full disk	Dec 9, 2011 4:59:01 PM	deviceName = C:\fileSystemSize = 40960048000 fileSystemType = NTFS usePct = 0.83 fileSystemAvail = 7251980000 fileSystemUsed = 33708068000
post-pm	00-24-21-52-C1-A6	full disk	Dec 9, 2011 4:59:01 PM	deviceName = E:\fileSystemSize = 74360128000 fileSystemType = NTFS usePct = 0.71 fileSystemAvail = 21823264000 fileSystemUsed = 52536864000
post-pm	00-24-21-52-C1-A6	sophos firewall	Dec 9, 2011 4:59:01 PM	pid = 600 userName = SYSTEM memSize = 157564928 cpuTime = 602827 state = R name = C:\Program Files\Sophos\Sophos Client Firewall\SCFService.exe
adil-PC	00-19-D2-70-B1-3B	full disk	Jan 2, 2012 3:42:53 PM	deviceName = C:\fileSystemSize = 14868124000 fileSystemType = NTFS usePct = 0.98 fileSystemAvail = 395008000 fileSystemUsed = 14473116000
adil-PC	00-19-D2-70-B1-3B	full disk	Jan 2, 2012 3:42:53 PM	deviceName = D:\fileSystemSize = 18257840000 fileSystemType = NTFS usePct = 0.95 fileSystemAvail = 972980000 fileSystemUsed = 17284860000
adil-PC	00-19-D2-70-B1-3B	full disk	Jan 2, 2012 3:42:53 PM	deviceName = E:\fileSystemSize = 84092208000 fileSystemType = NTFS usePct = 0.84 fileSystemAvail = 14155888000 fileSystemUsed = 69936320000
adil-PC	00-19-D2-70-B1-3B	full disk	Jan 2, 2012 3:42:53 PM	deviceName = F:\fileSystemSize = 2974202000 fileSystemType = UDF usePct = 1 fileSystemAvail = 0 fileSystemUsed = 2974202000
		sophos	Jan 2, 2012	

Figure 5.19 : Liste des erreurs détectées

### 5.2.4 Gestion de l'assistance aux utilisateurs

La console d'administration offre aux agents techniques de suivre les tickets d'assistance qui lui sont affectés et une fois l'incident résolu ils peuvent changer le statut du ticket. La liste des tickets offre plusieurs indicateurs comme la machine concernée, la date du ticket, une description du problème, etc.

Référence	Sujet	Date	Priorité	Statut	Utilisateur en charge	Machine
1	r	07/06/12 14:13:27.000	Bas	En attente	admin	00-16-D3-CE-B3-0A
5	o	07/06/12 14:23:12.000	Bas	En attente	admin	00-16-D3-CE-B3-0A
8	s	07/06/12 14:23:45.000	Elevée	En attente	admin	00-16-D3-CE-B3-0A
2	y	07/06/12 14:13:45.000	Moyenne	En attente	adom	00-16-D3-CE-B3-0A
4	i	07/06/12 14:23:03.000	Moyenne	En attente	adom	00-16-D3-CE-B3-0A
7	q	07/06/12 14:23:34.000	Moyenne	En attente	admin	00-16-D3-CE-B3-0A
3	u	07/06/12 14:22:50.000	Normale	En attente	admin	00-16-D3-CE-B3-0A
6	p	07/06/12 14:23:24.000	Normale	En attente	adom	00-16-D3-CE-B3-0A
9	d	07/06/12 14:23:55.000	Normale	En attente	admin	00-16-D3-CE-B3-0A
10	f	07/06/12 14:24:05.000	Normale	En attente	admin	00-16-D3-CE-B3-0A

Figure 5.20 : liste des tickets d'assistance

Pour modifier le statut du ticket il suffit de cliquer sur sa ligne correspondante et l'interface détail du ticket est affichée :

Figure 5.21 : détail du ticket

L'agent pour lequel a été affecté le ticket dispose de trois options pour assister l'utilisateur de la machine. La première option est de démarrer une session de chat, la deuxième est de démarrer une session d'audio conférence et la troisième est la prise de contrôle à distance de la machine concernée. Ces trois options sont accessibles via l'interface « Liste des machines » comme illustré dans la figure suivante :

Nom	Adresse IP	Système d'exploitation	ID machine	Statut
mindGateway.localdomain	192.168.41.5	Linux	00-02-B3-3E-D2-D3	🔇
mindGateway.localdomain	192.168.41.5	Linux	00-02-B3-3E-D2-D3o	🔇
sweet	192.168.41.14	Windows XP	00-02-B3-3E-D2-D3x	🔇
aya-PC	127.0.0.1	Microsoft Windows 7	00-16-D3-CE-B3-0A	🟢 🗨️ 📞 📡
adil-PC	192.168.41.14	Microsoft Windows 7	00-19-D2-70-B1-3B	🔇
post-pm	192.168.41.46	Microsoft Windows XP	00-24-21-52-C1-A6	🔇
admin-PC	127.0.0.1	Microsoft Windows 7	08-00-27-00-FC-9A	🟢 🗨️ 📞 📡

Chat →  
Audio conférence →  
Prise de contrôle →

Figure 5.22 : options d'assistance

### 5.2.5 Tableau de bord

MRMC offre à l'équipe technique un tableau de bord synthétisant tout les indicateurs techniques du parc informatique sous forme de graphe. Cette synthèse peut être affichée pour une durée et/ou pour une machines précise.

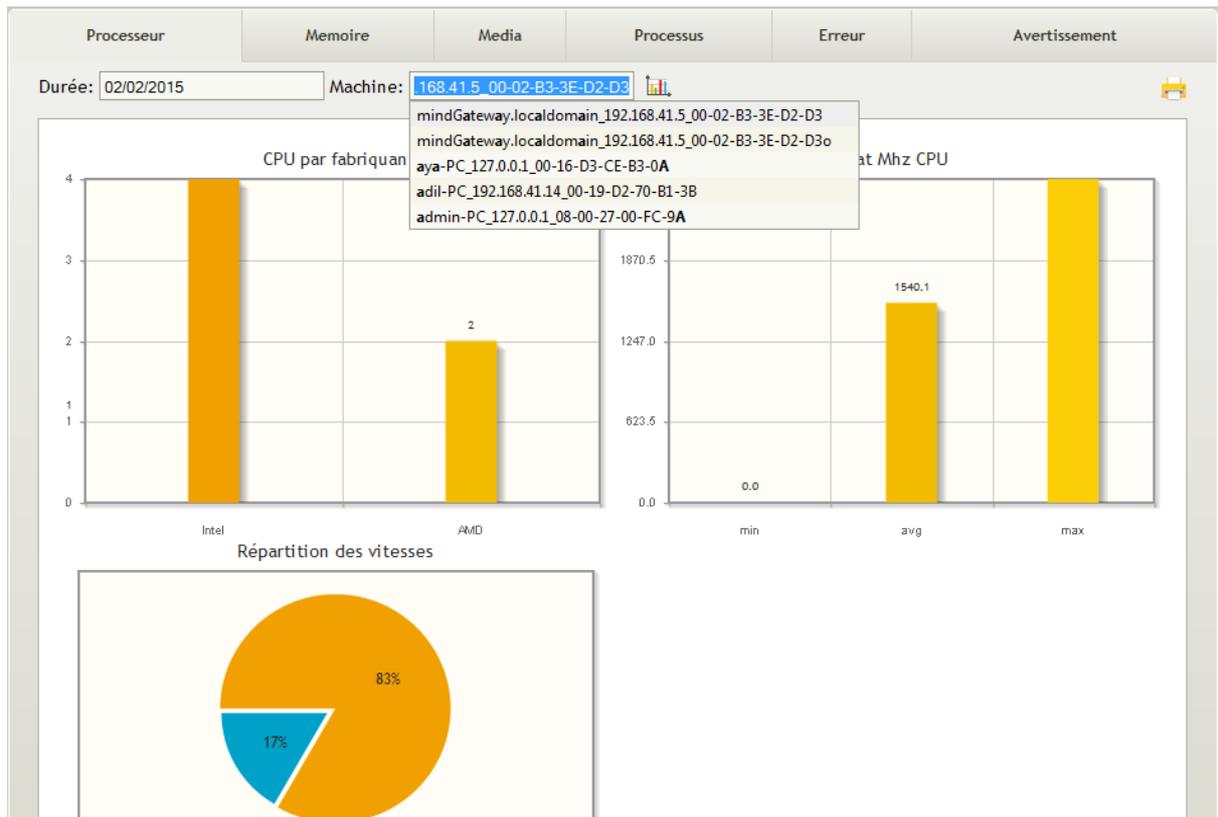


Figure 5.23 : Tableau de bord

Le tableau de bord présente une multitude d'indicateurs et de graphes offrant une vue globale sur l'état du parc informatique pendant une durée bien déterminée. Ces indicateurs concernent tous les objets d'une machine que ça soit l'état de la mémoire, des médias, des erreurs, etc.

## 5.3 Critique et perspective de la solution

Le projet MRMC est en continuelle modification à fin d'optimiser sa stabilité et l'efficacité de ses fonctionnalités. Dans ce chapitre j'ai présenté les fonctionnalités les plus importantes de la solution MRMC développée par la société MindWare.

En effet, Plusieurs fonctionnalités manquent à MRMC que je pourrai grouper en deux parties, fonctionnalités de l'agent et de la console d'administration MRMC.

### 5.3.1 Fonctionnalités de l'agent

L'agent MRMC est un outil de supervision en premier lieu qui doit être installé sur les machines clientes. Pour satisfaire les attentes de l'équipe de développement et surtout les attentes des utilisateurs, je pense qu'il faudrait ajouter un certain nombre de fonctionnalités, parmi lesquelles :

### **a) Surveillance en temps réel**

Jusqu'à présent l'agent MRMC ne permet que la supervision des machines en intervalle fixe, et ne permet pas la l'auto détection d'incident. Ainsi MRMC peut facilement rater la détection d'un problème qui s'est déclenché avant le lancement de sa routine de constitution d'état de la machine (machineState).

Pour résoudre ce problème je pense qu'il faudrait ajouter une fonctionnalité de détection en temps réel ce qui permettrait au administrateur de suivre le fonctionnement d'une machine suspecte.

### **b) Installation sur Linux**

MRMC procède un installateur sur Microsoft Windows chose inexistante jusqu'à présent sur Linux ce qui rend son utilisation assez difficile sur les systèmes du monde libre et demande une intervention en Shell sur la machine sur laquelle on veut l'installer.

A fin de faciliter l'utilisation de MRMC sur Linux il faudrait créer un script d'installation.

### **c) Suivi du statut des tickets**

Avec l'agent MRMC, l'utilisateur de la machine surveillé peut créer facilement un ticket d'assistance mais ne peut pas malheureusement modifier ou même simplement suivre son statut.

Je pense qu'il serait souhaitable de donner à l'utilisateur la possibilité de modifier ou de suivre le statut d'un ticket dont-il est le propriétaire en ajoutant dans le menu contextuel de l'agent un lien vers la liste des tickets classés par ordre chronologique.

### **d) Fonctionnement sur un réseau étendu (NAT Traversal)**

L'agent MRMC est outil qui n'autorise la prise de contrôle de la machine supervisée que dans un réseau local et ne peut être utilisé dans un réseau étendu comme Internet. Ce problème vient tout simplement de la présence de passerelle entre les différents réseaux locaux et de l'utilisation de la technologie NAT (Network Address Translation) ou traduction d'adresse réseau qui fait fait correspondre les adresses IP internes non-uniquees et souvent non routables d'un intranet à un ensemble d'adresses externes uniques et routables. Ce mécanisme permet notamment de faire correspondre une seule adresse externe publique visible sur Internet à toutes les adresses d'un réseau privé, et pallie ainsi l'épuisement des adresses IPv4.<sup>19</sup>

---

<sup>19</sup> [http://fr.wikipedia.org/wiki/Network\\_address\\_translation](http://fr.wikipedia.org/wiki/Network_address_translation)

Pour résoudre ce problème il faudrait utiliser une solution pour outrepasser les passerelles connue sous le nom de « NAT Traversal ». Plusieurs solutions existent mais la plus adaptée est « ICE4J ».

### **5.3.2 Fonctionnalités de la console d'administration**

La console d'administration est l'outil qui permet aux administrateurs réseau d'avoir une idée claire sur l'état du parc informatique et ainsi diminuer le plus les interventions de maintenance corrective. MRMC offre un nombre important de fonctionnalités dans ce sens mais il lui manque d'autre pour optimiser son rendement.

#### **a) Détection automatique des machines**

MRMC ne peut gérer que les machine enregistrées sur sa base et dans le cas ou une nouvelle machine est intégrée dans le réseau, MRMC n'a aucun moyen de la détecter et de l'ajouter dans sa base de données.

Pour palier à ce problème, il faudrait ajouter une fonctionnalité de détection automatique des nouvelles machine en réalisant des recherches dans le parc en se basant sur les plages d'adresse IP existantes dans MRMC.

#### **b) Installation à distance**

Pour qu'une machine soit surveillée par MRMC il faut y installer son agent. Malheureusement jusqu'à présent l'installation n'est possible qu'en local c'est-à-dire que le technicien doit se déplacer vers la machine en question chose assez difficile dans les grands parcs informatiques.

La solution est d'ajouter une fonctionnalité d'installation à distance d'une machine ou de groupe de machine en se basant sur les adresses IP et les comptes utilisateurs avec pouvoir.

#### **c) Utilisation du protocole SNMP**

La surveillance d'une machine dans MRMC est basée exclusivement sur l'installation de son agent. Or un certain nombre d'équipement réseau n'autorise pas l'installation d'un tel programme mais offre en contre partie d'autre solution comme l'utilisation de technologie comme SNMP ou protocole simple de gestion de réseau qui est un protocole de communication qui permet aux administrateurs réseau de gérer les équipements du réseau, de superviser et de diagnostiquer des problèmes réseaux et matériels à distance.<sup>20</sup>

L'utilisation du protocole SNMP élargira les gammes d'équipement réseau pouvant être supervisé par MRMC comme les routeurs, imprimantes, etc.

---

<sup>20</sup> [http://fr.wikipedia.org/wiki/Simple\\_Network\\_Management\\_Protocol](http://fr.wikipedia.org/wiki/Simple_Network_Management_Protocol)

## **5.4 Conclusion**

MindWare a investi beaucoup de moyens financiers et humains pour le développement de MRMC, elle a finalisé un nombre important de fonctionnalité mais d'autres sont encore en phase de test ou même de développement. Dans ce chapitre j'ai essayé de présenter les fonctionnalités les plus importantes de MRMC et j'ai cité quelque une que je trouve manquantes.



# Chapitre 6

## Conclusion générale

MRMC est un projet très important et très large dont le développement a été lancé par MindWare depuis l'année 2011 et qui est toujours en continuelle optimisation et de maintenance. Ce projet répond à une nécessité que tout administrateur réseau veut satisfaire qui est la gestion de parc informatique de façon professionnelle, structurée et extensible.

Le développement de cette application a été initié pour répondre à plusieurs problématiques internes à MindWare se résumant dans la nécessité de création d'un module de gestion de parc informatique s'intégrant dans son ERP ainsi que son intention de contribuer au prolifération de l'utilisation des bibliothèque, outil et application opensource.

La réalisation de MRMC est passée par plusieurs phases débutant par une étude de marché et concurrentielle des différentes solutions du marché. En suite, une étude fonctionnelle a été menée pour fixer les différentes tâches que MRMC doit accomplir après laquelle une modélisation UML s'est achevée par la génération du diagramme de classes.

Avant de commencer la programmation de MRMC, l'équipe de projet commence par la création des différentes maquettes répondant aux fonctionnalités de la future application. Après cela, le chef de projet divise le travail sur l'ensemble de l'équipe et réalise un suivi hebdomadaire pour vérifier et valider les livrables de chaque membre.

La première version de MRMC a vu le jour à l'année 2013 et son optimisation et maintenance est toujours à l'ordre du jour. Néanmoins, la version 1.0 a été testée et validée par l'équipe de projet et a été présentée à plusieurs clients dont la majorité ont exprimé des avis favorable et ont formulé quelques critiques.

Parmi les critiques les plus importantes qui ont été exprimées celle de l'incapacité de MRMC de gérer des machines clientes dans un réseau étendu revenait tout le temps. En effet, cette lacune prive MindWare d'un nombre important de grand client dont le parc informatique dépasse les centaines d'équipement et dont le réseau est découpé en plusieurs sites géographiques.

MindWare a commencé le développement d'une nouvelle version qui prendra en compte toutes les critiques exprimées par ses clients ainsi que d'autres fonctionnalités formulées par son équipe de développement. Une version beta est cours de test et de validation et verra le jour prochainement.

MRMC a constitué pour moi le premier projet réel auquel je contribue de façon professionnelle ce qui m'a permis de concevoir une idée claire sur les procédures de réalisation des logiciels ainsi que la complexité du travail en groupe. Aussi, ce stage m'a offert la chance de nouer des relations et contacts avec des personnes de qualité que ça soit au niveau professionnel que social.

Au cours de la période de mon projet de fin d'études, j'ai eu l'opportunité de mettre en application les différentes connaissances acquises durant ma formation. De plus, j'ai eu l'occasion de développer un sens d'auto-formation qui m'a beaucoup aidé à maîtriser de nouvelles technologies.

Les difficultés que j'ai rencontrées résident essentiellement dans l'ampleur du projet et la nouveauté des technologies avec lesquelles j'ai travaillé. Aussi, l'importance et des fois la complexité des tâches dont j'étais responsable ainsi que le respect des dates de livraison ont constitué de grands défis auxquels j'ai été confronté tout au long de ma période de stage.

Concernant ma réalisation, j'ai contribué à la création de l'agent MRMC surtout la partie extraction des données techniques et la génération du fichier XML. Aussi, dans la partie console d'administration j'ai accompli la majorité des modules prévus pour la surveillance et le contrôle de parc informatique et j'estime que l'outil développé pourra être utilisé facilement dans tous les systèmes « Helpdesk ».

Finalement, j'espère que ce rapport sera utile pour toute autre personne qui va le découvrir.

# Références

- [1] <http://www.francetop.net/dictionnaire/synonymes/definition/Monitoring>
- [2] [www.mindware.ma](http://www.mindware.ma)
- [3] [http://www.hyperic.com/press-releases/9\\_22\\_2004](http://www.hyperic.com/press-releases/9_22_2004)
- [4] <http://dom4j.org/download.html>
- [5] [http://www.javafr.com/codes/EXEMPLE-PARSEUR-DOM\\_41491.aspx](http://www.javafr.com/codes/EXEMPLE-PARSEUR-DOM_41491.aspx)
- [6] <http://www.experts-exchange.com/Programming/Languages/Java/Q20649856.html>
- [7] <http://www.java2s.com/Code/Java/Collections-Data-Structure/Iterate.htm>
- [8] <http://jpmapper.sourceforge.net/>
- [9] <http://www.jqplot.com/tests/>
- [10] <http://www.webapart.fr/blog/tutoriauxjqplot>
- [11] <http://struts.apache.org/download.cgi#struts218>

- [12] <http://kmdkaci.developpez.com/tutoriels/java/bien-debuter-avec-struts2/>
- [13] <http://www.scribd.com/doc/56634436/49/Definition>
- [14] <http://www.journaldunet.com/encyclopedie/definition/158/34/20/java.shtml>
- [15] <http://www.siteduzero.com>
- [16] <http://www.developpez.net/forums/d259570/java/generaljava/apis/java-util/modifier-fichier-properties/>
- [17] <http://www.zdnet.fr/telecharger/logiciel/enterprise-architect-11008543s.htm>
- [18] <http://www.dicodunet.com/defieb/mysql.htm>
- [19] <http://cougart.wordpress.com/2009/01/26/pencil-logiciel-gratuit-danimation-2d>
- [20] <http://struts.apache.org/download.cgi/struts218>

# **Annexes**

## **Struts2**

### **Installation et Configuration**

Pour utiliser Struts2 il faut le télécharger via <http://struts.apache.org/download.cgi#struts218> et décompresser le contenu du répertoire « lib » dans un répertoire différent.

Pour configurer Struts2 on procède comme suit :

Ajouter les fichiers « jar » relatifs à Struts2 dans le répertoire « lib » :

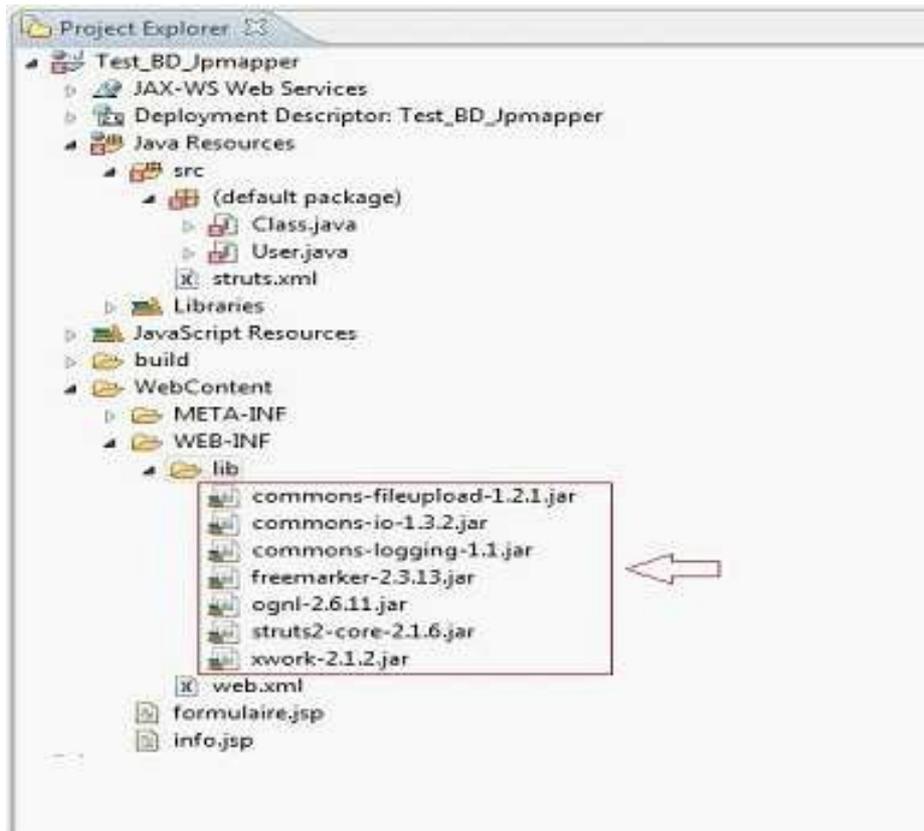


Figure 7.1: Configuration de Struts 2

Le fichier de configuration web.xml se présente ainsi:

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID"
version="2.5">

  <display-name>Java / MYSQL</display-name>
  <!-- contient les info sur la première page à consulter il sera utiliser 1
  fois( au début ) -->
  <filter>
    <filter-name>struts2</filter-name>
    <filter-class>org.apache.struts2.dispatcher.FilterDispatcher</filter-class>
  </filter>

  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  <welcome-file-list>
    <welcome-file>formulaire.jsp</welcome-file>
  </welcome-file-list>
</web-app>

```

Les éléments les plus importants dans le fichier « web.xml » sont « filtre » et « mapping » utilisés pour la configuration du « FilterDispatcher » qui est la servlet de base du Framework Struts 2, il permet de traiter toutes les requêtes en entrée, il permet l'accès aux éléments de base. (ConfigurationManager, ActionMapper, ObjectFactory ) du Framework pour traiter les requêtes.

## Le fichier xml Struts.xml

Le fichier struts.xml permet de configurer le mapping entre l'URL et l'implémentation de l'action. Ce fichier doit être dans le dossier « src », il se présente comme suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
<package name="default" extends="struts-default">
<action name="ajout" class="Class" method="detail">
    <result name="ok">info.jsp</result>
</action>
</package>
</struts>
```

## Les actions

La seule contrainte pour les actions, avec Struts 2, est le fait qu'elles doivent implémenter au moins une méthode du type « no-argument » qui doit retourner un « String » ou un objet « Result », sinon on peut utiliser la méthode « execute() ».

Avec Struts 2, il n'est pas obligatoire d'hériter d'une action générique, mais les actions peuvent hériter de la classe générique « ActionSupport » qui implémente la méthode « execute() ».

La classe « ActionSupport » implémente plusieurs interfaces telles que « Action », « Validateable ».

Ci-après un exemple d'action Struts2 :

## Class.java

```
import com.opensymphony.xwork2.ActionSupport;
public class Class extends ActionSupport{
    private String nom;
    private String prenom;
    public String detail() {
        System.out.println("Bonjour");
        User user = new User();
        user.setNom(this.nom);
        user.setPrenom(this.prenom);
        return prenom;
    }
    public String getNom() {
        return nom;
    }
}
```

## User.java

```
public class User {
    private int iduser;
    private String nom;
    private String prenom;
    public int getIduser() {
        return iduser;
    }
    public void setIduser(int iduser) {
        this.iduser = iduser;
    }
    public String getNom() {
        return nom;
    }
    public void setNom(String nom)
    {
        this.nom = nom;
    }
    public String getPrenom() {
        return prenom;
    }
    public void setPrenom(String prenom) {
        this.prenom = prenom;
    }
}
```

Enfin on crée la page Jsp :

## Le fichier JSP « formulaire.jsp »

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="s" uri="/struts-tags" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>title </title>
</head>
<body>
<h1>Connexion à la Base de donnée</h1>
<s:form action = "ajout" >
<s:textfield name = "Nom" label="Nom" ></s:textfield>
<s:textfield name = "Prenom" label="Prénom" ></s:textfield>
<s:submit value = "Enregistrer"></s:submit>
</s:form>
</body>
</html>
```

Le schéma suivant résume le concept de Struts2 :

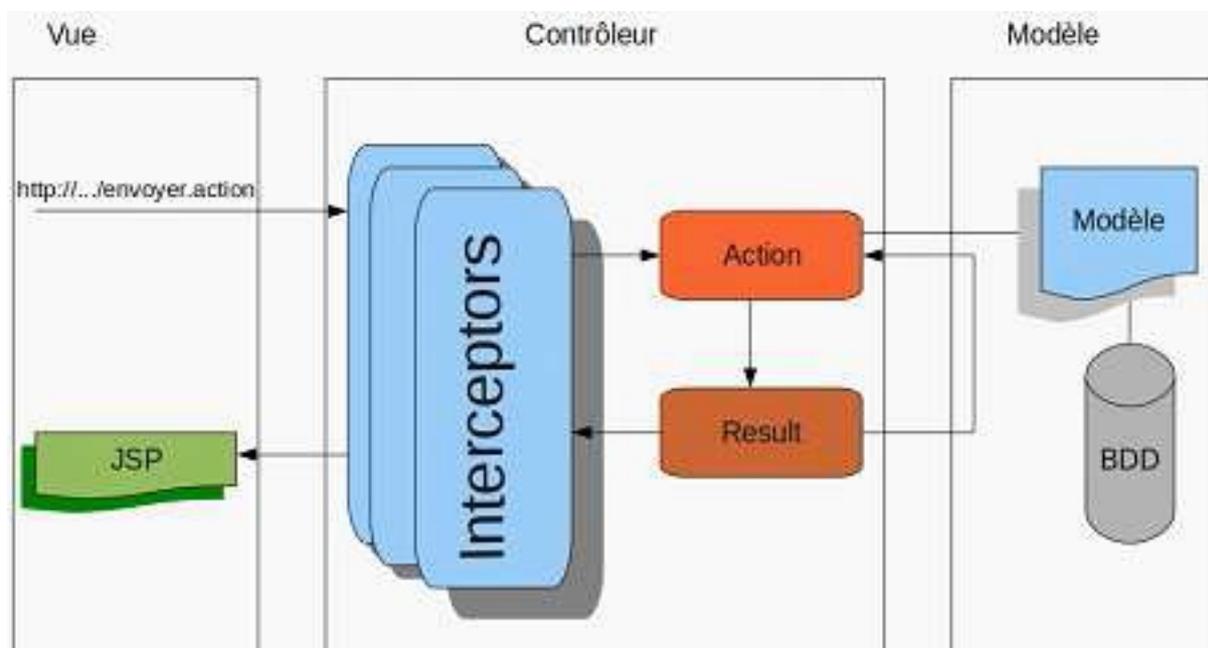


Figure 7.2: Principe de Struts2

## JPMapper

### Installation et Configuration

Pour configurer « JPMapper » il suffit d'ajouter les deux fichiers « jar » relatifs à « JPMapper » dans le répertoire « lib » :

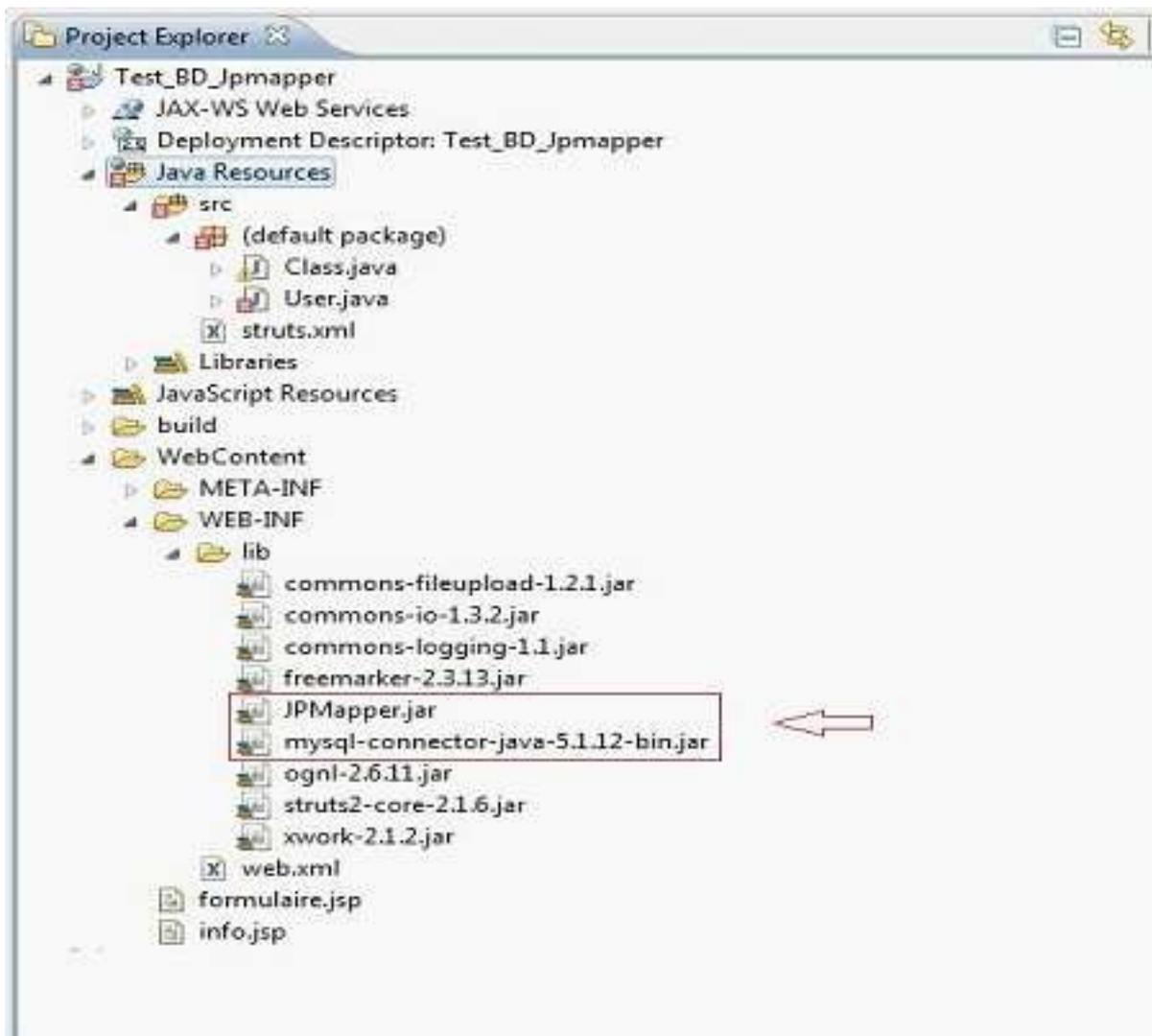


Figure 7.3: Configuration de JPMapper

Puis on crée un objet de type « JPMapperFacade » auquel on va affecter le chemin: du fichier « Dbinformation.properties » qui contient toutes les informations de la Base de donnée. On peut prendre à titre d'exemple une base de données nommée « mrmc » à la quelle on souhaite se connecter :

Une fois les fichiers « .jar » qui correspondent à « JPMapper » sont mis dans le dossier « lib », on crée dans le contrôleur un objet de type JPMapper et on affecte le chemin du fichier « Dbinformation.properties » à ce dernier.

Le code suivant illustre ce mécanisme :

```
JPMapperFacade jpmapper=null;
jpmapper =
JPMapperFacade.getInstance("C://FichierProperties/dbInformation.properties");
```

### Remarque:

Le chemin : « C : //FichierProperties/dbInformation.properties » est le chemin du fichier qui contient les informations relatives à la base de données.

Le nom de la base de données sera mentionné à l'intérieur de ce dernier (dbInformation.properties) :

```
#####
### DB Information ###
#####
dbms mysql
dbName mcmc
dbUser root
dbPassword
host localhost
port 3306
'dbFieldDelimiter '
#####
```



Figure 7.4: Contenu du fichier « dbInformation.properties »

## Dom4j

### Installation et Configuration

Télécharger la dernière version à l'url : <http://dom4j.org/download.html>.

Il suffit de décompresser le fichier téléchargé. Ce dernier contient de nombreuses bibliothèques (ant, Xalan, Xerces, crimson, junit,) ainsi que la documentation.

La méthode la plus simple pour utiliser rapidement dom4j est de copier le fichier « dom4jfull.jar » dans le répertoire « lib ».

## **MySQL**

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde<sup>1</sup>, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

Son nom vient du prénom de la fille du cocréateur Michael Widenius, My. SQL fait allusion au Structured Query Language, le langage de requête utilisé.

## **Enterprise Architect**

Enterprise Architect est un outil visuel de génie logiciel assisté par ordinateur (CASE) s'appuyant sur le langage de modélisation UML (Unified Modeling Language). Il comporte toutes les fonctions nécessaires à la conception et au développement de systèmes logiciels orientés objets ainsi qu'à leur documentation et gestion. Enterprise Architect combine la puissance des spécifications de la version 2.0 d'UML avec une interface intuitive pour créer des modélisations complètes et avancées.

## **Pencil**

Pencil est un logiciel de graphisme et d'animation 2D (en anglais) distribué sous licence GPL. Il permet l'utilisation de calques (bitmap ou vectoriel) et est doté de nombreux outils de création graphique de base tel que le crayon, le pinceau, la gomme ...

## Résumé

Tout système de surveillance de parc informatique doit être capable de superviser des centaines de serveurs et des milliers de services. « MindWare Remote Monitoring and Control » ou « MRMC » a été créé pour cette finalité par la société MindWare qui est une société Marocaine d'édition logiciel et d'ingénierie réseau. MindWare a lancé le développement de cette application depuis l'année 2011 en réponse à des besoins exprimés par ses clients et pour l'intégrer dans son PGI.

La Création de MRMC a débuté par une étude préalable centrée sur une analyse des besoins et s'est terminée par l'étape de programmation et test en passant par une étude de marché, une analyse conceptuelle basée sur la méthodologie UML, une étude fonctionnelle où toutes les fonctionnalités à développer ont été cataloguées et distribuées sur les membres de l'équipe de développement.

Dès le départ et durant toutes ces étapes, j'ai été intégré à l'équipe de projet de MRMC avec qui j'ai contribué à la conception et au développement des différentes fonctionnalités de la solution. A la fin de ces phases, le résultat final est une application web développée avec différentes technologies basées sur la plateforme JEE et fortement Ajaxisée.

Le présent rapport est une synthèse de ce travail et présente ma modeste contribution à fin d'accomplir la mission qui m'a été confiée.

## Abstract

All system monitoring enterprise machines must be able to supervise hundreds of servers and thousands of services. "MindWare Remote Monitoring and Control" or "MRMC" was created for this purpose by Mindware Company; Mindware is a Moroccan company specialized in software and network engineering. Mindware launched the development of this application since 2011 responding the needs of the customer, and to be included in ERP.

The development of MRMC was started by a preliminary study focused on needs analysis and finished by programming and test; this was by a market research, analysis and design using UML, a functional study where all features to develop were cataloged and distributed over team members.

From the starting and during all steps, i was integrated to the development team, i contributed to the design and the development of the application's features. At the end the resulted application is a web based application developed using JEE platform with Ajax.

This report is a synthesis of this work and presents my modest contribution to this in order to fulfill the mission entrusted to me.

**Mots clés:** MindWare, MRMC, PGI, JEE, UML, Ajax.

**Keywords:** MindWare, MRMC, ERP, JEE, UML, Ajax.