Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: *Réseaux et systèmes distribués (R.S.D)*

Thème

Mise en œuvre d'une application SMA Dans les réseaux P2P (Supervision system)

Réalisé par :

- **Mlle. KADDOUR Halima**

Présenté en Juin 2015 devant le jury composé de MM.

- *Mr. Messabihi Mohamed* *(Président)*
- *Mr. BENZIAN Yeghmoracène* *(Encadreur)*
- *Mr. MOUFFOK Benattou* *(Examinateur)*
- *Mr. Tadlaoui Mohamed* *(Examinateur)*

Année universitaire : 2014-2015

Table des matières

Dédicace

Je dédie ce travail à :
Ceux que personne ne peut compenser les sacrifices qu'ils ont consentis pour mon éducation et mon bien être :

A mes chers parents.

A mon frère, et mes sœurs.
A tout ma famille.
A mes chers(es) amis(es).

Toute personne qui de prés ou de loin, a participé à ma formation.

Remerciements

Louage à dieu, Bienfaiteur miséricordieux.

Paix et bénédiction sur son prophète, Mohamed, ultime envoyé.

Je tiens à remercier DIEU qui m'accordé la santé, la possibilité ainsi que la volonté d'entamer er de continuer mes études.

Je remercie profondément mon encadreur Mr. BENZIAN, pour ces conseils avisés, et pour la qualité de son encadrement et sa disponibilité.
Je remercie l'ensemble des jurés d'avoir accepté d'examiner et d'évaluer mon travail.

Je remercie tous ceux qui ont contribué de prés ou de loin, pour leur soutiens moral ou matériels, dans la réalisation de mon projet.

Je m'adresse tous d'abord mes grands remerciements à mes parents et toute ma famille (merci pour vos encouragements permanents et pour votre confiance).

Introduction Générale :

Les utilisateurs suivent plusieurs démarches pour surveiller et assurer la bonne gestion de leurs systèmes. Ces dernières peuvent nécessiter beaucoup de personne et d'argent pour gérer des systèmes éloignés à l'aide des systèmes d'exploitation différents.

Pour cela et dans le cadre de ce mémoire, nous avons fixé l'objectif de concevoir et de réaliser un système de monitoring system, qui assurant la vérification de la santé d'un serveur coté ressources matérielles

Plus précisément, nous suivrons l'approche d'agents mobiles qui est une nouvelle architecture dédiée à la réalisation des systèmes distribués. Cette dernière modélise les applications à travers un ensemble d'agents mobiles.

Nous avons choisi la plateforme JADE pour le de développement des agents et JAVA comme langage d'implémentation.

Le document est organisé comme suite :

Le chapitre I : présente la technologie p2p, au premier temps nous représentons cette technologie puis , nous traiterons l'architecture des réseaux p2p. Nous finirons par décrire les avantages et les inconvénients de cette technologie. .

Le chapitre II : présente les notions de base du concept d'agent, les agents mobiles et les systèmes multi-agent.

Le chapitre III : est consacré à la partie réalisation et implémentation de l'application, au premier partie il représente la notion de supervision informatique, on définit ses fonctionnalités et niveaux d'information, ses aspects et ses rôles, puis en deuxième partie il explique en détaille l'architecture et le fonctionnement de l'application réalisée. Finalement, nous concluons la présente du PFE par un résume de l'ensemble de notre travail, et présentons quelques perspectives.

I. Introduction :

L'informatique paire à pair se définit comme le partage des ressources et des services par échange direct entre systèmes. Ces échanges peuvent porter sur les informations, les cycles de traitement, la mémoire cache ou encore le stockage sur disque des fichiers.

Contrairement au modèle client / serveur, chaque système est une entité réseau complète qui remplit à la fois le rôle de serveur et celui de client.

Le pair-à-pair (en anglais, *peer-to-peer*, abrégé *P2P*) est un modèle de réseau informatique proche du modèle client-serveur. À la différence du modèle client-serveur où un seul gros ordinateur (serveur) dessert de l'information à de nombreux terminaux (clients), dans le modèle P2P, chaque client est aussi un serveur. Tous les ordinateurs récupèrent de l'information et resservent cette l'information obtenue.

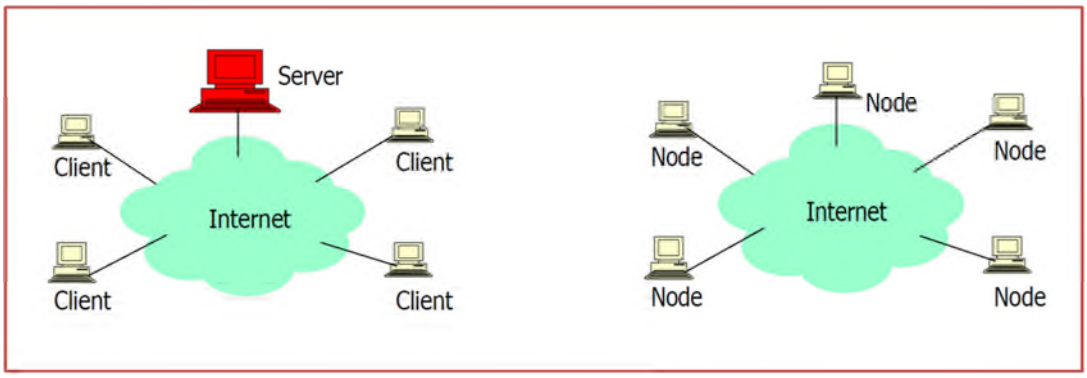

Figure I.1 : Les deux modèles de réseaux (client/serveur, Peer to Peer).

II.Propriétés des réseaux Pair à Pair :

- Pas de coordination centralisée.
- Pas de bases de données centralisées.
- Aucun nœud n'a une vision globale du système.
- Le comportement global émerge à partir des interactions locales.
- Tous les services et données sont accessibles depuis n'importe quel nœud.
- Les nœuds sont autonomes.
- Les nœuds et connexions sont non fiables.

III.Caractéristiques des architectures peer-to-peer :

Les réseaux P2P présentent trois caractéristiques importantes :

- l'auto-organisation des pairs ou nœuds est assurée par un processus de découverte qui permet le regroupement logique des nœuds.
- la communication symétrique est assurée par les nœuds qui jouent un rôle identique dans l'offre et la demande de services. C'est la différence majeure

avec le modèle client/serveur dans lequel le client et le serveur ont des rôles différents.

- dans le contrôle distribue, il n'existe pas de contrôleur central qui dicte le comportement de certains nœuds. Ainsi, chaque nœud détermine son niveau de participation au réseau selon ses besoins. Un nœud décide, lui-même, quand accéder au réseau, quand le quitter et quelles ressources partager et consommer. Ceci nécessite une distribution du contrôle entre tous les nœuds.

IV.Classification (L'architecture) des réseaux P2P :

Il est également possible de regrouper ces architectures en 3 grands groupes souvent utilisées pour classifier les réseaux P2P :

IV.1.P2P centralisé :

Il existe un serveur central, gérant l'identité des utilisateurs, l'indexation des partages, les recherches et la mise en relation des pairs : le serveur donne à chaque client l'adresse des pairs possédant le fichier recherché. Le client se connecte alors à ce pair pour échanger directement les données. Les fichiers ne passent donc pas par le serveur. L'index est donc centralisé, mais les fichiers sont décentralisés.
Ce type de P2P est considéré comme l'architecture la plus faible car il suffit de s'attaquer au serveur central pour faire tomber le réseau.

Avantage :

- Le confort d'utilisation est grand, puisqu'il n'y a pas de soucis de connexion au bon serveur.
- La recherche de document est facilitée. En effet, le serveur est omniscient : si un fichier est disponible sur le réseau, on est en mesure de le savoir systématiquement.

Inconvénients :

- La sécurité est le talon d'Achille de ce type d'architectures : il suffit de supprimer le serveur pour que l'intégralité du réseau soit inactive.
- Sensible au partitionnement du réseau (serveur inatteignable) et aux attaques.
- Aucun anonymat n'est possible, puisque chaque utilisateur est identifié sur le serveur. Il est alors on ne peut plus facile d'élaborer des profils utilisateurs .
- Le risque de surcharge de l'annuaire qui peut devenir un point de défaillance du système.

IV.2.P2P Hybride :

Ces réseaux utilisent des nœuds spéciaux appelés "super-peer" réalisant les indexations des fichiers partagés et servant d'intermédiaire pour les nœuds qui leur sont rattachés. Ils sont choisis en fonction de leur puissance de calcul, leur bande passante…

- Les nœuds disposant d'une bonne bande passante sont organisés en P2P. Ce sont les super- Peers.
- Les nœuds avec une faible bande passante sont reliés en mode client/serveur à un super Peer.
- Les super-Peers disposent d'un index des ressources de leur cluster.

Ce type d'architecture est notamment utilisé dans les réseaux

Inconvénients : Le risque de défaillance d'un super-nœud.

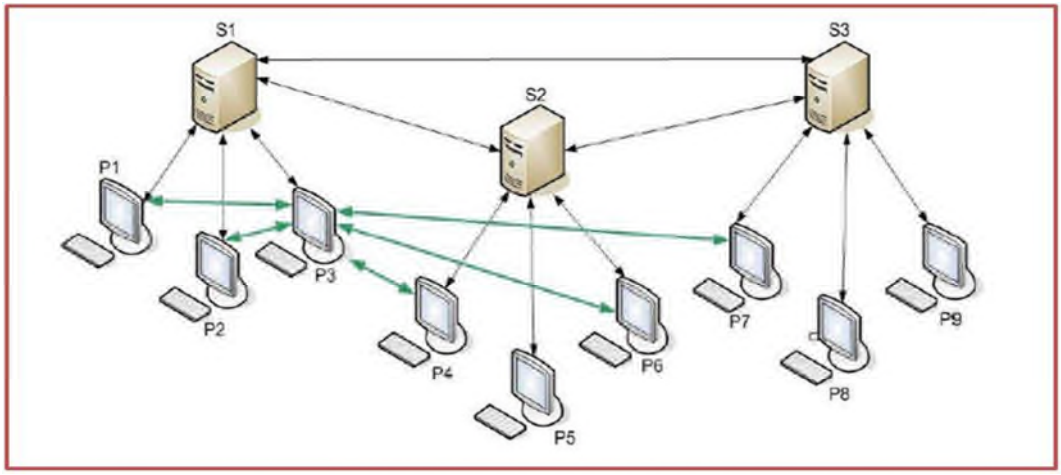

Figure I.2 : Exemple de réseau super-Peer.

IV.3.P2P "purs" :

Les réseaux totalement décentralisés sont dits "pures". Dans ce type d'architecture, l'ensemble des nœuds sont égaux et jouent le même rôle. Chaque pair gère donc les recherches des ressources et le partage, sans passer par un serveur central ou des super-peer, en utilisant en général des techniques de "flood" ou des algorithmes spécifiques.

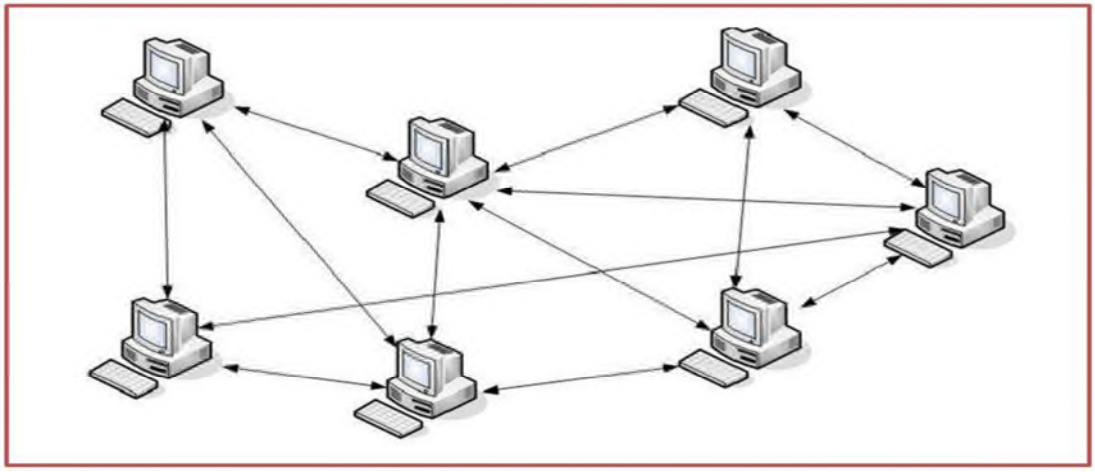

Figure I.3 : Exemple d'architecture P2P purs.

Avantages :

- La taille d'un réseau est théoriquement infinie, contrairement aux autres architectures dont le nombre de clients dépend du nombre et de la puissance des serveurs.
- L'utilisation d'un réseau est anonyme, c'est à dire qu'il est impossible d'y repérer quelqu'un volontairement.
- Réseau très tolérant aux fautes.
- S'adapte bien à la dynamique du réseau (allées et venus des pairs).

Inconvénients :

- Gros consommateur de Bande passante, temps de recherche important.
- Pas de garantie de succès, ni d'estimation de la durée des requêtes.
- Pas de sécurité, ni de réputation (pas de notion de qualité des pairs, ni des données fournies).
- Problème du free-Riding (personnes ne partageant pas de fichiers).

V.Application des réseaux P2P :

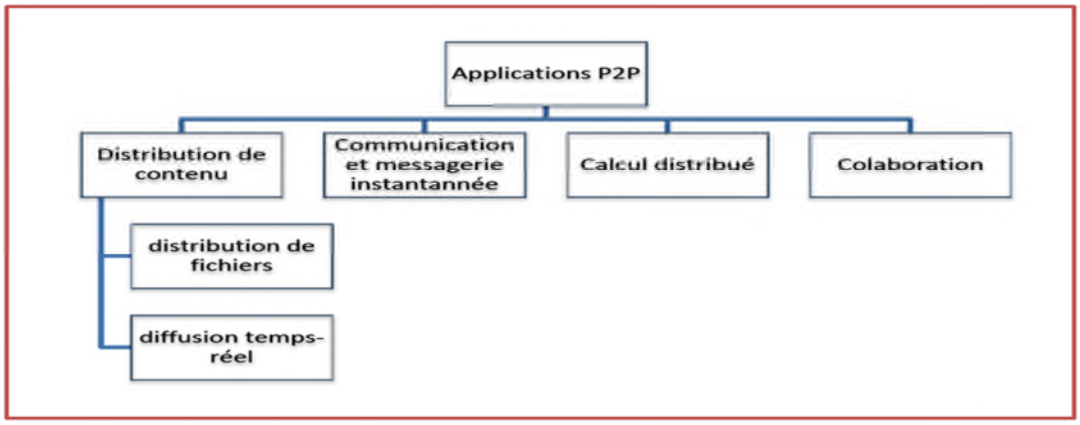

Figure I.4 : classification des applications P2P.

Applications de distribution de contenus :

Le but est la distribution massive de contenu pour alléger la surcharge des serveurs prévus initialement à cette tâche, comme Napster…

Applications de communication et de messagerie instantanée :

Se développent largement sur internet grâce aux facilites apportées par les réseaux P2P en termes de groupement d'utilisateurs, recherche distribuée, et efficacité de distribution. Comme le Skype…

Applications de calculs distribues :

Le principe consiste à subdiviser une tâche complexe, qui nécessiter un temps de calcul considérable.

Applications de collaboration.

Permet à plusieurs utilisateurs de travailler sur le même document en même temps grâce aux réseaux P2P.

VI.Les avantages et les inconvinients des Peer-to-Peer :

VI.1. Les avantages :

Les P2P se sont distingués des architectures traditionnelles grâce à leurs différents avantages. En effet, ils sont souvent assimilés, par le grand public, aux logiciels de partage de fichiers, cependant leur application ne s'arrête pas là. Ils permettent de mutualiser les ressources de milliers d'ordinateurs dont les capacités sont sous-exploitées par leurs utilisateurs :

- La répartition de la charge : les échanges sont gérés directement par les paires.

Les P2P ne posent pas le problème de congestion des réseaux, Chaque pair gère ses propres données et répond aux requêtes des autres pairs.

- La capacité de stockage est ainsi infiniment supérieure à celle d'un serveur traditionnel.
- La puissance de calcul : pour les utilisateurs moyens, chaque ordinateur utilise moins de 20 % de sa puissance de calcul. Une application répartie, s'activant en général avec l'écran de veille des ordinateurs, basée sur la technologie Peer-to-Peer peut donc permettre de mutualiser cette puissance non utilisée pour des recherches demandant des capacités considérable, qu'un serveur isolé ne peut posséder.
- Outils de travail collaboratif (gestion d'agenda, etc.).
- Résistance aux pannes (sauvegardes croisées) : les données étant présentes sur de nombreux postes différents.
- Extensibilité : auto-configuration des pairs. Il est très facile de rajouter de nouveaux pairs. Ceux-ci sont gérés dynamiquement.

VI.2. Les inconvénients:

- Les systèmes d'échanges de fichiers peer-to-peer peuvent facilement souffrir de l'anarchie générale. L'anonymat aidant, les membres sont parfois tentés d'avoir des comportements malveillants. Sans un minimum de contrôle, on peut voir apparaître des activités comme la diffusion de virus ou le freeloading.
- Les virus : Un virus de type cheval de Troie s'attaque aux réseaux de type peer- to-peer , le réseau Gnutella a été infecté par un virus de type Worm. Il était capable de surveiller les connexions Internet et répondait à toutes les requêtes quelles qu'elles soient. Il simulait la présence d'un fichier correspondant à la requête et si ce fichier était téléchargé, c'était le ver qui était transmis et infectait la nouvelle machine. Ainsi, les nœuds devenaient surchargés car ils répondaient à toutes les requêtes.

VII. Conclusion :

Le P2P est une technologie qui a un principe d'exploiter toutes les capacités de stockage et de traitement des ordinateurs connectés, cette architecture donne toute son autonomie au client dans le réseau en réduisant le rôle des serveurs, pour cela, les réseaux pair-à-pair ont démontré qu'il est possible d'organiser des ressources à travers des terminaux hétérogènes.

I. Introduction :

La technologie des agents offre une nouvelle méthode pour analyser, concevoir et implémenter des applications car ils font partie du domaine IAD . Le terme agent est associé à plusieurs application informatique , ayant des buts à atteindre. Il est utilisé, depuis plusieurs années.

II. Technologie des agents :

II.1. Notion d'agent :

Plusieurs définitions d'agent ont été proposées. Ferber (1995) propose : « Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents ».

II.2. Caractéristiques d'un agent :

Un agent présente un certain nombre de propriétés, dont certaines sont plus importantes que d'autres selon le type d'application. Voici les principales propriétés **(Wooldridge and Jennings, 1995; Jennings et al, 1998) :**

- **Situation :** l'agent est capable de recevoir des entrées sensorielles de son environnement et peut effectuer des actions qui changent cet environnement (par exemple : systèmes de contrôle de processus, démons logiciels).
- **Autonomie :** l'agent est capable de prendre des décisions et d'agir sur son état interne sans l'intervention directe d'un tiers (humain ou un autre agent) et contrôle ses propres actions ainsi que son propre état interne.
- **Réactivité :** l'agent est capable de percevoir son environnement et de répondre dans le temps requis aux changements qui apparaissent dans cet environnement.
- **Proactivité :** l'agent ne doit pas agir seulement en réponse à son environnement, mais il doit exhiber un comportement orienté but et opportuniste avec la capacité de prendre l'initiative au bon moment.
- **Sociabilité :** l'agent doit être capable d'interagir, quand la situation l'exige, avec les autres agents et les humains via une sorte de langage de communication, afin d'accomplir ses tâches ou aider les autres dans leurs activités.
- **Mobilité :** fonctionnalité supplémentaire, elle permet à l'agent de se déplacer de manière autonome à travers le réseau.

- **Apprentissage :** l'agent est capable de changer son comportement en fonction des expériences passées **(Franklin and Graesser, 1996).**

II.3. Typologie des agents :

Les architectures d'agents sont regroupées en trois classes:

II.3.1. Agent réactif :

1- Agents à réflexes simples :

-Ce type d'agent agit en se basant uniquement sur ses perceptions courantes.

-Utilise un ensemble de règles prédéfinies pour choisir ses actions,du type: SI *condition* ALORS *action.*

-L'agent exécute l'action qui correspond à la règle activée par ses perceptions. Il présente l'avantage d'être fort simple mais en pratique il est très limité. En effet, un tel agent ne peut fonctionner correctement que s'il peut prendre la bonne décision sur la base du seul percept courant détecté, c'est à dire seulement si l'environnement est entièrement observable.

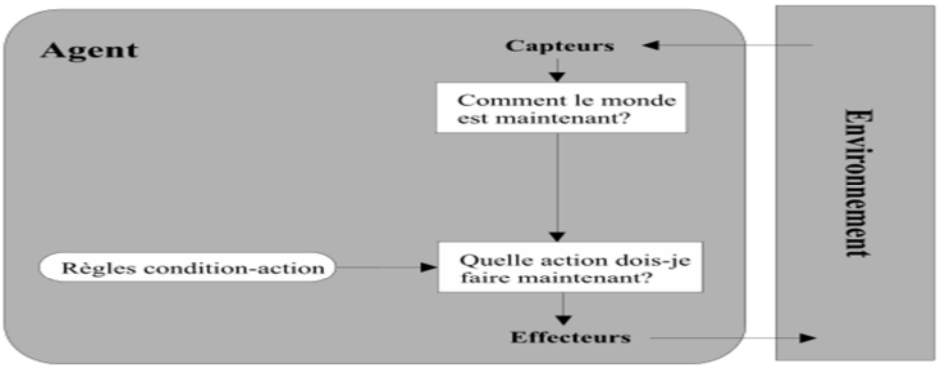

Figure II.1.Schéma d'un Agent à réflexe simple.

Les limites des agents à réflexes simples :

- L'agent peut choisir ses actions en se basant uniquement sur sa perception actuelle.
- L'agent peut avoir deux perceptions identiques mais qui sont en réalité différentes.
- « Les capteurs » de l'agent ne fournissent pas une vue complète sur l'état du monde.
- Manque de flexibilité

☞ L'agent doit maintenir des informations internes sur l'état de l'environnement.

2- Agents conservant une trace du monde :

Ce type d'agents Utilisent ses informations internes pour mettre à jour ses perceptions actuelles à savoir:

a. L'état précédent de l'environnement.
b. L'évolution de l'environnement.
c. L'impact de ses actions.

Choisissent leurs actions en se basant sur une perception « amélioré » de l'environnement.

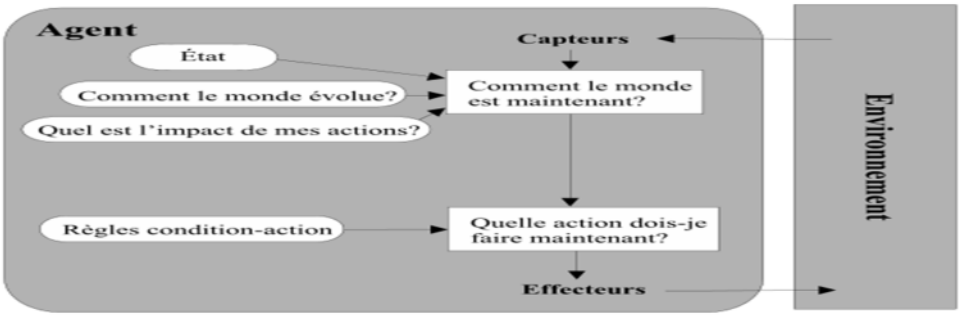

Figure II.2.Schéma d'un Agent conservant une trace du monde

Les Limites des agents conservant une trace du monde :

* Les agents utilisent seulement leurs connaissances sur l'état de l'environnement pour choisir leurs actions.
* Absence de but explicite.
* Manque de flexibilité.

II.3.2. Agent délibératif :

1- Les agents ayant des buts :

Ce type d'agent possède:

* Une description de l'état actuel de son environnement.
* Des informations décrivant ses buts.
* Une projection sur le future.
* Beaucoup plus de flexibilité.

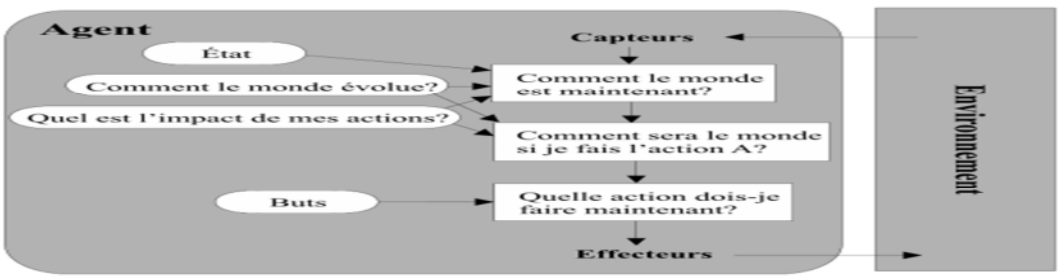

Figure II.3. Schéma d'un agent ayant des buts.

Les limites des agents ayant des buts :

- Les buts ne sont pas suffisants pour générer un comportement de haute qualité.
- L'agent raisonne seulement sur ses buts et n'a pas de moyen pour choisir une action de « qualité ».
- L'agent doit être capable de « préférer » un état à un autre.
- L'agent a besoin de reconnaître pour chacun des états son degré de satisfaction.

2- Les agents utilisant une fonction d'utilité :

Dans plusieurs situations, les buts ne sont pas suffisants pour générer un comportement de haute qualité. Par exemple, s'il y a plusieurs chemins possibles pour atteindre le port, certains seront plus rapides et d'autres plus dangereux. Dans cette situation, l'agent raisonnant uniquement sur ses buts n'a pas de moyens pour choisir le meilleur chemin, son seul but étant de se rendre à destination. En effet, les buts ne procurent qu'une simple distinction entre les états où l'agent est satisfait et les états où il ne l'est pas. En fait, l'agent doit plutôt s'appuyer sur une manière plus fine d'évaluer les états pour être en mesure de reconnaître pour chacun des états son degré de satisfaction. Pour cela, on dit que l'agent va *préférer* un état à un autre si son utilité est plus grande dans le premier état que dans le deuxième.

Généralement, l'utilité est une fonction qui attribue une valeur numérique à chacun des états. Plus l'état a une grande valeur, plus il est désirable pour l'agent.

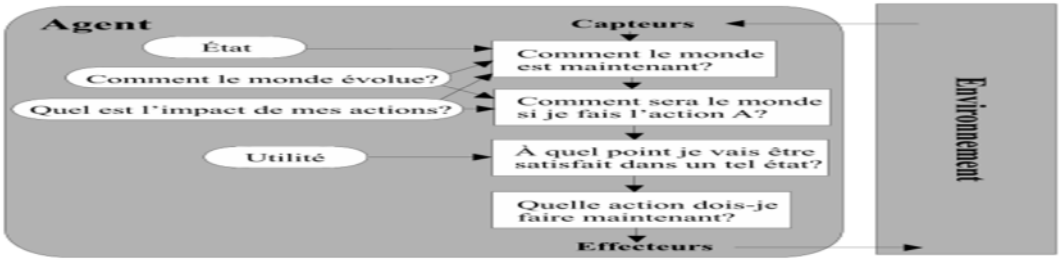

Figure II.4. Schéma d'un agent utilise une fonction d'utilité.

3- Agents BDI :

Les agents se basent sur trois aspects pour choisir leurs actions:

* Les croyances qui représentent un ensemble d'informations que l'agent possède sur son environnement.
* Les désirs qui représentent les options disponibles à l'agent.
* Les intentions qui représentent les buts envers lesquels il s'est engagé.

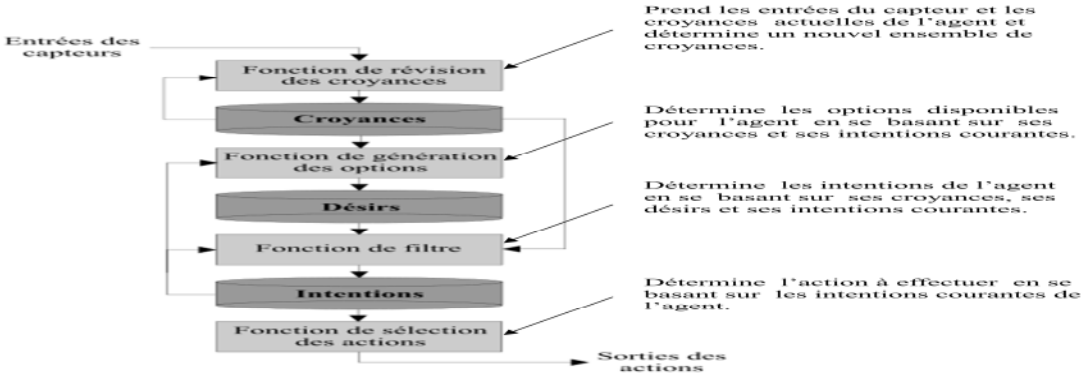

Figure II.5. Schéma d'un agent BDI.

II.3.3 Les agents hybrides :

L'architecture hybride est une architecture composée d'un ensemble de modules organisés dans une hiérarchie, chaque module étant soit une composante cognitive, ou une composante réactive. Pour cela, le comportement proactif de l'agent, dirigé par les buts, est combiné avec un comportement réactif aux changements de l'environnement.

Dans les architectures hybrides , on retrouve une couche purement réactive, cette couche prend ses décisions en se basant sur des données brutes . La couche intermédiaire fait abstraction des données brutes et travaille avec une vision qui se situe au niveau des connaissances de l'environnement. Finalement, la couche supérieure se charge des aspects sociaux de l'environnement, c'est à dire du raisonnement tenant compte des autres agents.

II.4. L'apprentissage pour les agents :

Les perceptions de l'agent ne devraient pas être utilisées seulement pour choisir des actions mais elles devraient être aussi utilisées pour améliorer l'habilité de l'agent à agir dans le futur.

- L'apprentissage de l'agent lui permet d'évoluer, de s'adapter et de s'améliorer. Plus l'agent effectue des taches similaire plus il devient plus rapide.
- Le comportement de l'agent passe graduellement d'un état délibératif à un état réactif.

III. Les agents mobiles :

Les agents mobiles ont été lancés sur l'initiative de General Magic avec le système d'agents mobiles Telescript (White, 1995). On définit un agent mobile comme étant un agent capable de se déplacer, sous son propre contrôle, d'un site à un autre dans un réseau hétérogène pour accomplir la tâche du client. A chaque site visité, l'agent mobile interagit avec d'autres agents et les ressources présentes sur le site. En général, l'agent mobile revient à son site initial une fois sa mission accomplie (**Gray et al, 1996 ; Rothermel and Schwehm, 1998)**.

III.1. Processus de migration des agents mobiles :

La migration permet à un agent de se déplacer d'un site à un autre à travers le réseau. Elle comporte les étapes suivantes (El Falou, 2006) :

- La sérialisation du contexte de l'agent et l'inclure avec son code dans un message.
- Le processus de l'agent et ses communications sont suspendus et l'agent est détruit.
- L'envoi du contexte de l'agent et son code vers la machine de destination.
- Dès l'arrivée sur la machine distante, l'état de l'agent est restauré et un processus léger est créé pour la poursuite de l'exécution.
- Les communications en cours sont redirigées vers la machine de destination et les messages seront traités après la réactivation de l'agent.
- La réactivation de l'agent sur la machine de destination et la migration prend fin.

III.2. Domaines d'application des agents mobiles :

Plusieurs applications bénéficient particulièrement du paradigme d'agents mobiles. Parmi ces applications :

- **Le commerce électronique :** les agents mobiles sont bien adaptés pour le commerce électronique. Une transaction commerciale peut nécessiter un accès en temps réel aux ressources distantes. Les agents mobiles représentent leurs propriétaires et peuvent agir et négocier en leurs noms en utilisant différents stratégies pour atteindre leurs buts.
- **Recherche d'information distribuée :** dans cette recherche, l'agent est transféré vers les sources d'informations où il crée localement des indexes et les renvoie à son propriétaire s'il est disponible. Même si le propriétaire est déconnecté, La recherche peut être poursuivie.

* **Services des réseaux de télécommunication :** la gestion de ces services de est caractérisée par la reconfiguration dynamique du réseau et la personnalisation des utilisateurs. Dans ce cas, L'utilisation de la technologie des agents mobiles maintient ces systèmes flexibles et efficaces en dépit des exigences strictes dans lesquelles ils opèrent.
* **Surveillance et notification:** c'est des applications qui montre la nature asynchrone des agents mobiles. Ces agents m sont capables de surveiller la source d'information et son emplacement. Dans ce cas, la durée de vie des agents mobiles est indépendante du processus qui les crée.
* **Calcul parallèle :** les agents mobiles peuvent utiliser pour administrer des tâches parallèles en créant une cascade de clones dans le réseau. Ainsi, un calcul qui nécessite une grande puissance de traitement, peut être distribué dynamiquement via une infrastructure d'agents mobiles.

IV. Les systèmes multi agents SMA :

IV.1 Définition :

On appelle système multi-agents (ou SMA), un système composé des éléments suivants:

1. Un environnement **E**, c'est-à-dire un espace disposant généralement d'une métrique.
2. Un ensemble d'objets **O** situé dans **E.** Ces objets peuvent être perçus, créés, détruits et modifiés par les agents.
3. Un ensemble **A** d'agents, représentent les entités actives du système.
4. Un ensemble de relations **R** qui relient des agents entre eux.
5. Un ensemble d'opérations **Op** permettant aux agents de percevoir, produire, consommer, transformer et manipuler des objets.
6. Des opérateurs chargés de représenter l'application de ces opérations et la réaction de l'environnement envers les tentatives de modification.

Figure II.6 **:** Architecture d'un SMA.

IV.2 Caractéristique d'un SMA :

Généralement, un SMA possède les caractéristiques suivantes :

- Les agents agissent et travaillent indépendamment les uns des autres.
- Chaque agent est une partie du système.
- Chaque agent travaille dans le but d'accomplir ses tâches.
- Chaque agent est capable de communiquer et d'interagir avec d'autres agents.
- Un agent coopère avec les autres agents lorsque nécessaire.
- Un agent est capable de coordonner ses activités avec les autres agents pour accéder à des ressources et à des services partagés dont il a besoin (pour réaliser ses buts).
- Les agents ont un but commun.
- Chaque agent a une vue partielle du SMA.

IV.3. Type des SMA :

On peut différencier les SMA selon le nombre d'agents, la nature et la complexité de ces agents. Ainsi, un SMA peut être soit :

- **Ouvert** : les agents y entrent et en sortent librement (par exemple: l'utilisation d'un SMA pour développer une application de commerce électronique).
- **Fermé** : où l'ensemble d'agents reste le même.
- **Homogène** : dont tous les agents sont construits sur le même modèle.
- **Hétérogène** : dont les agents de modèles sont différents, c'est-à-dire de granularités différentes.

IV.4. Intérêts des SMA :

Les principaux intérêts se résument ainsi:

- Ils s'adaptent bien à la réalité dans la mesure où de nombreux problèmes sont de nature distribuée ;
- L'utilisation de nombreux agents résolvant le même problème de façon différente, produit généralement des solutions de meilleure qualité en termes de complétude et de précision ;
- Ils permettent d'intégrer des connaissances diverses et complexes;
- Les connaissances insuffisantes peuvent être complétées suivant la résolution ;
- Ils permettent de résoudre des problèmes de taille et de complexité telle qu'il n'est pas réaliste d'essayer de les résoudre avec un seul agent ;
- La *modularité* : la complexité d'un système d'IA croit avec la taille de sa base de connaissances. Partitionner ce système en N agents réduit sa complexité par un

facteur parfois plus grand que N, et la configuration résultante se trouve plus facile à développer, à tester et à maintenir ;

IV.5.Domaines d'application des SMA :

Les systèmes multi-agents ont déjà montré des promesses particulières dans une variété de systèmes technico-sociaux comme: la gestion du trafic aérien, la gestion des ressources, les applications boursières, la télémédecine, l'environnement de robotique cognitive ou les systèmes de commande et de contrôle en temps réel, etc.

Aussi, la technologie agent a trouvé sa place dans les systèmes dynamiques, où l'action autonome, la flexibilité et l'intelligence sont requises quand les quantités de données à traiter dépassent l'envergure de traitement sur une seule machine. Cette approche constitue un bon moyen de résolution et d'appréhension des problèmes. Cela, explique l'utilisation actuelle des systèmes multi-agents dans divers domaines, dont nous avons cité quelques-uns brièvement :

- le web sémantique et la gestion documentaire sur le web ;
- les réseaux tels que Peer to Peer (P2P) ;
- les systèmes distribués concurrents et les bases de données reparties ;
- le commerce électronique sur le net et la gestion des processus d'affaires ;
- les systèmes d'information coopératifs ;
- les télécommunications ;

IV.6.Avantages et inconvénients des SMA :

IV.6.1. Les avantages :

Plusieurs avantages sont attribuables aux SMA :

- ✓ Si plusieurs agents peuvent travailler en parallèle, les SMA améliorent leur performance par un travail parallèle des agents.
- ✓ La facilité de « changement d'échelle d'une architecture » car plusieurs agents peuvent s'ajouter ou se retirer dynamiquement d'un système.
- ✓ La configuration automatique d'un SMA
- ✓ Plus un logiciel est modulaire, plus la complexité et les coûts de développement diminuent.
- ✓ Flexibilité des systèmes : les systèmes sont composés de plusieurs agents qui peuvent donc résoudre différents problèmes.

- ✓ Les SMA reflètent la réalité : la majorité des problèmes sont distribués qui s'adapte facilement aux SMA.
- ✓ Diversité : Les SMA peuvent avoir parmi les agents les constituants une grande diversité, ce qui donne la possibilité aux concepteurs d'intégrer différents agents (réactifs, cognitifs …etc.)
- ✓ Coopération : Les systèmes entre eux peuvent coopérer pour la résolution de problèmes plus complexes.
- ✓ A chaque agent sa façon de résoudre les problèmes, donc un même problème peut avoir différentes solutions selon les agents.

IV.6.2. Les inconvénients :

Malgré tous les avantages des SMA ils ont quelques inconvénients tels que :

- La sécurité des applications : les agents peuvent communiquer sans aucun contrôle .
- L'exécution des agents s'effectuant en parallèle ,il est encore plus difficile de comprendre leur fonctionnement à partir de leur code, du fait du non-déterminisme inhérent au parallélisme.
- Enfin, et surtout, les systèmes multi-agents sont des logiciels complexes, difficiles à appréhender et à concevoir. Il est donc nécessaire de réduire leur complexité en dégageant leur structure et en analysant séparément leurs différents composants sans pour autant perdre de vue l'organisation générale.

V. Conclusion :

Les systèmes multi-agents ont apporté une nouvelle vision des choses et une nouvelles façon de concevoir des solutions à des problèmes qui au pare-avant était impossible ou extrêmement difficiles à résoudre. Dans notre choix de la plateforme d'agents nous avons choisi JADE (Java Agent Development Framework). Ce choix est justifié par le fait que JADE est une plate-forme multi-agents développée en Java, dotée d'une interface graphique et qui peut être répartie sur plusieurs serveurs et le fait d'être open source va faciliter l'implémentation.

I. Introduction :

Ce chapitre sera consacré à l'implémentation et mise en pratique de notre système. Nous commençons par décrire la notion de la supervision informatique et les outils de développement utilisés, puis nous présentons l'interface utilisateur de notre système.

II. La supervision informatique :

II.1. Définition :

En informatique, la supervision est une technique, qui permet de surveiller, analyser, rapporter et d'alerter les fonctionnements normaux et anormaux des systèmes informatiques. D'un point de vue théorique, Eric D'HEM explique : « le Monitoring s'agit de répéter de manière régulière un processus de test ou de surveillance d'une personne ou d'un bien. Le but étant d'obtenir très rapidement et simplement une vision précise des événements ou anomalies sur la période analysée» Entre outre, La supervision informatique consiste à indiquer et/ou commander l'état d'un serveur, d'un équipement réseau ou d'un service software pour anticiper les plantages ou diagnostiquer rapidement une panne.

II.2. Les fonctionnalités de la supervision :

les moniteurs de supervision regroupent les fonctionnalités suivants:

Supervision réseau : elle porte sur la surveillance de manière continue de la des services en ligne, du fonctionnement du réseau, des débits et bande passante, de la sécurité, etc.

Supervision système : elle permet de vérifier l'état de santé d'un serveur du côté matériel (mémoire, CPU, disque dur, etc.).

Exécution de commandes : permet de surveiller les actions ou les programmes lancés automatiquement.

Envoi d'alertes : elle a pour principe d'émettre des messages d'alerte sous forme sonore, visuelle ou par e-mail.

Cartographie : cette fonctionnalité vise l'ensemble d'architecture informatique surveillé.

Rapports d'activité (reporting) : ce sont les activités comme les tableaux de bord et les histogrammes.

II.3. Les niveaux d'informations :

Les niveaux d'informations sont répartis en plusieurs catégories :

Informations sur les systèmes : elle permet d'analyser les informations sur du système comme l'utilisation du CPU, l'occupation de la mémoire physique, l'espace libre des disques dures, etc.

Informations sur les réseaux : Ce type de surveillance va permettre de diagnostiquer la disponibilité d'un équipement physique connecté à un réseau. Tels que : les commutateurs, les routeurs et les serveurs ainsi que les onduleurs.

Informations sur les applications et services : ce type de supervision vérifie les applications exécutées et leurs informations retournent.

Pour les deux derniers niveaux la supervision réseau permet de vérifier la connectivité d'un serveur Web, qui rend les sites Web accessible. et la supervision d'application vérifie le contenu de ces sites Web.

II.4. Le rôle de la supervision :

L'objectif de la supervision est de surveiller le système et garantir sa disponibilité même s'il anomalie :

- Tenter de prévenir en cas de défaillances matérielles ou l'interruption des services et garantir une augmentation d'information rapide;
- Récupérer les applications et les services sans perte de données en une durée d'intervention minimale.

II.5.Les aspects de la supervision :

Les aspects de supervision peuvent classer dans trois catégories principales qui sont :

II.5.1. La Fiabilité : Il s'agit de l'utilisation la plus courante du monitoring informatique. Une surveillance permanente de la disponibilité de l'équipement est effectuée, et ce pour détecter la moindre anomalie et de la signaler à l'administrateur.

II.5.2. La Performance: Le but de performance est de retourner des informations sur le rendement d'un équipement ou d'un service comme par exemple le temps de résolution DNS, le temps de connexion, et le temps de récupération de la page dans le cas d'une page Web. Grâce à cette analyse, la bande passante va surdimensionné.

II.5.3. Le Contenu : c'est-à-dire on détecte le contenu les informations retournées par les éléments déjà surveillés et analysés, par exemple, le cas de la suppression d'un fichier sur un serveur FTP ou la modification d'une page Web.

III. Domaine d'application :

Notre projet se déroule dans un réseau P2P qui est actuellement le plus répandus surtout pour résoudre les problèmes de la gestion et management des machines.

IV. Les outils utilisés pour notre application :

Nous avons utilisé deux Lap tops qui sont reliés entre eux avec une liaison TCP/IP en attribuant chacun une adresse IP différente.

Figure IV.1 : Les Peers du réseau.

V. Environnement de développement :

Dans la figure suivante, nous illustrons le choix en termes d'environnement de développement.

Figure IV.2 : Environnement de développement.

Le choix des outils de développement joue un rôle très important et décisif dans le développent des applications. Plusieurs critères devront être définis et respectés. Parmi ces critères on site :

- **Le temps de développement** : nous devrons utiliser les outils qui minimisent le temps de développement, et facilitent l'implémentation et la maintenance de l'application.
- **La stabilité** : les outils utilisés devront être les plus stables possibles pour faciliter le débogage lors du développement.
- **La portabilité** : les outils devront nous permettre de réaliser des applications compatibles avec un grand nombre de plateformes matérielles et logicielles.
- **Simplicité** : facilité d'utilisation et disponibilité de la documentation.
- **Les licences** : notre préférence va aux outils libres de droits qui n'imposent pas de restriction sur le développement et la diffusion de l'application.

Nous allons présenter dans ce qui suit les outils utilisés et les raisons du choix de chaque outil.

V.1. Plateforme JADE pour le développement des SMA :

Pour développer des agents, il est pratique de choisir une plate-forme. A l'heure actuelle, plusieurs outils ont été développés pour la réalisation d'agents. Si ces plateformes abondent en quantité, elles ne répondent pas toutes aux critères que nous nous sommes fixés. Cette partie présente notre démarche dans le choix d'une plateforme de développement d'agents.

La plateforme devra répondre aux contraintes suivantes :

- La plateforme doit répondre à plusieurs fonctionnalités et offrir une large gamme de bibliothèques.
- IL faut tenir compte de la nature de l'interface de développement vu le temps court réserver au développement des agents.
- La plateforme doit être répandue. Elle doit avoir été utilisée dans plusieurs projets de développement de systèmes multi-agents pour avoir un maximum d'exemple et de tutoriaux.
- Le langage de programmation sous-jacent et la nature des messages échangés sont aussi un point important. Un langage connu et répandu favorisera une meilleure compréhension des codes sources.

Nous avons choisis JADE (Java Agent Developpement Framework) pour les raisons suivantes:

- ✓ Facilité d'installation.
- ✓ Documentation détaillée.
- ✓ Utilise un langage puissant et stable (JAVA).
- ✓ Intégration avec d'autres outils de développement (Intégration dans Eclipse via les plugins EJIP et EJADE).
- ✓ Licence libre (LGPL: (pour GNU Lesser General Public License) ; c'est une licence utilisée par certains logiciels libres).

V.2 Environnement Eclipse pour programmer en JAVA :

JAVA fut le langage de notre choix vu que nous avons choisi préalablement la plateforme JADE, et aussi grâce à sa portabilité et à son aspect Orienté Objet qui facilite la migration d'une conception d'objet vers une implémentation de classe.

Pour Eclipse, c'est un environnement de développement intégré (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et

polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse est principalement écrit en Java, et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse vient du fait de son architecture totalement développée autour de la notion de plug-in: toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in. Plusieurs logiciels commerciaux sont basés sur ce logiciel libre.

Figure IV.3 : Interface d'Eclipse.

VI. Les points essentiels de notre travail :

Pour mettre en place cette application on a besoin d'un :

- *Peer superviseur :* c'est le Peer qui sera contacté par les autres Peers soit local ou distant, *l'agent_ superviseur* réside dans ce Peer.
- *Peer client:* (local ou distant), il demande à *l'agent_Peer* d'exécuter et renvoyer les résultats au Peer maître.

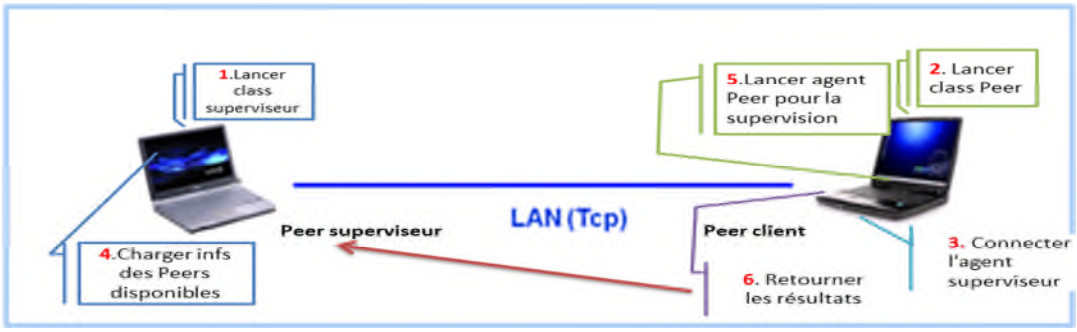

Figure IV.4 : les étapes de l'application

Diagramme de classes de l'application

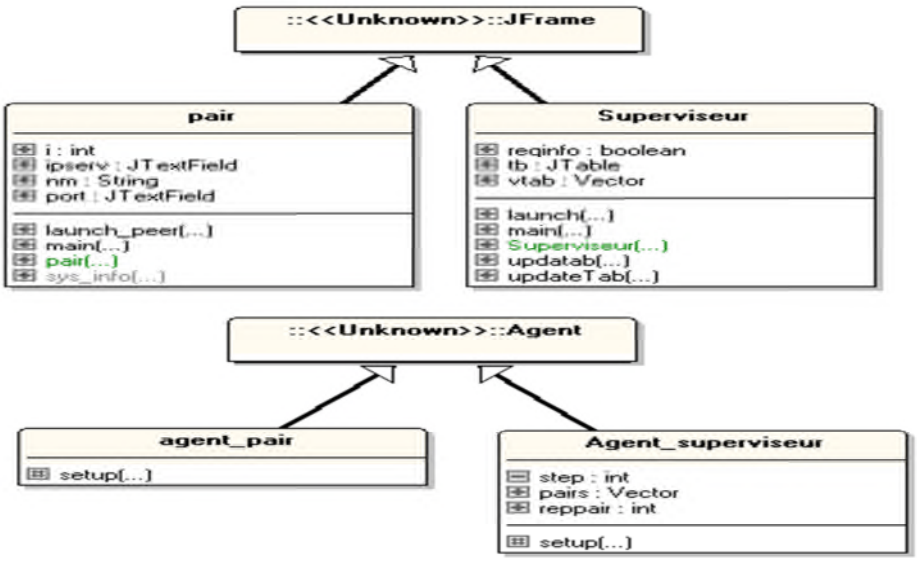

Figure IV.5: Diagramme des classes.

L'architecture générale de l'application:

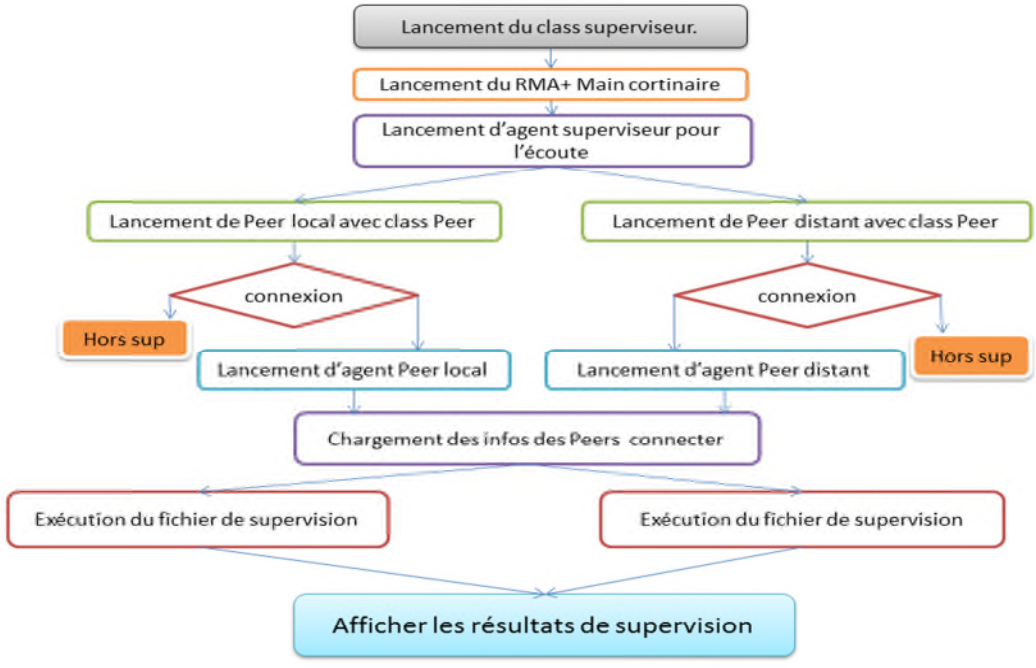

Figure IV.6 : L'architecture générale.

Le déroulement de l'application :

- **Lancement de l'interface principale de l'application SMAS**

Présentation de l'interface de SMAS (System Multi Agent Superviseurs) :

L'environnement de l'application est intuitif et très simple d'usage, il permet d'avoir une vue globale sur le fonctionnement des services considérés sur les nœuds supervisés.

Figure IV.7 : L'interface utilisateur SMAS.

Appelle RMA de jade :

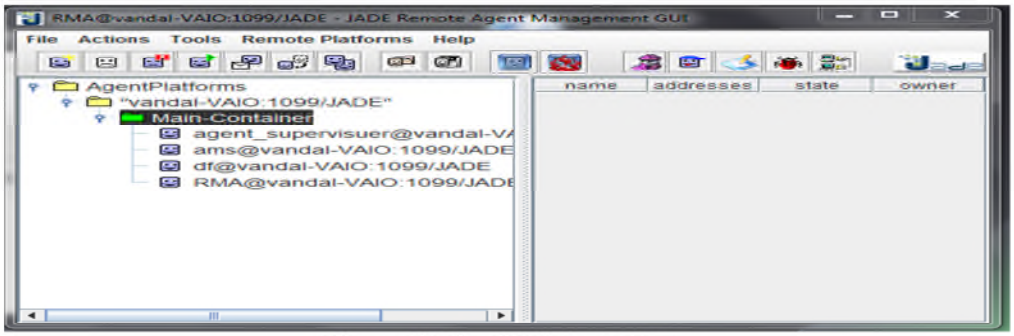

Figure IV.8 : GUI de JADE

- **Mise en pratique :**

La figure suivante montre l'application exécutée sur les machines qui vont être supervisées, pour le champ ou l'adresse doit être entrée, il s'agit de l'adresse IP du serveur de supervision ou se trouve le conteneur principal avec lequel sera relié le conteneur lancé par l'application cliente (Figure IV.9).

Figure IV.9 : Les clients de supervision.

- **Connexion de Pair local**

Figure IV.10 : charger l'@IP de pair local.

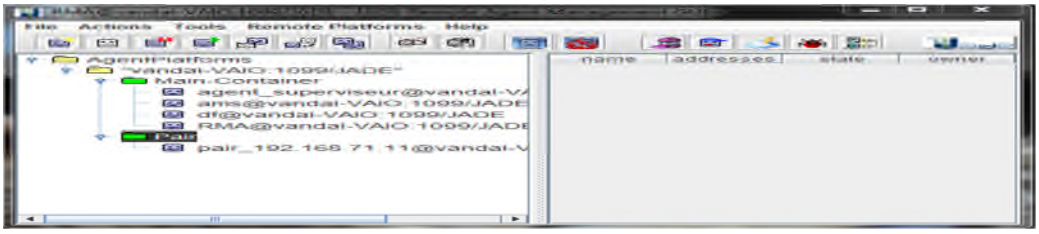

Figure IV.11 : Création de l'agent pair local.

Figure IV.12 :L'agent sniffer.

- **Connexion de Pair distant :**

Figure IV.13 : charger l'@IP de pair distant.

- **Charger les informations des pairs :**

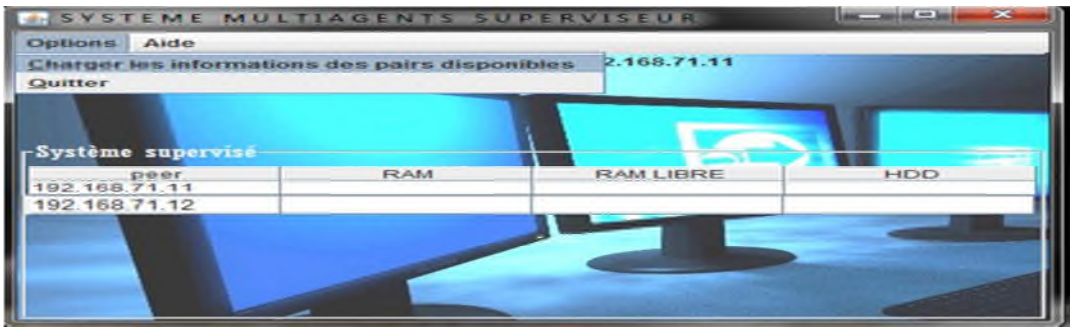

Figure IV.14 : Commande pour charger les informations.

Résultats :

Quel que soit le résultat des tests, l'agent remonte un message au serveur contenant un rapport détaillé. Après analyse du rapport des résultats envoyé, le serveur affiche les résultats.

Figure IV.15 : tableau de résultats.

VII. Conclusion :

Dans ce chapitre, nous avons présenté les différents aspects liés à l'implémentation de notre système. Cette partie nous a permis de toucher un peu au côté pratique et nous a émergé dans les nouvelles technologies telles que les SMA et la supervision informatique, ce qui nous a permis de comprendre le fonctionnement de ces technologies

Conclusion Générale :

Dans ce travail nous avons relié deux domaines très vastes de l'informatique actuelle. D'un côté la supervision informatique et d'un autre l'intelligence artificielle avec les systèmes multi-agents précisément sur L'approche d'agents mobiles qui constitue un paradigme de programmation puissant et efficace pour les applications sur les réseaux P2P . L'objectif était de donner une vue générale sur ce paradigme et de mettre en pratique nos connaissance théorique

La réalisation de ce projet nous a pu acquis une expérience pratique en développement une petite application qui se compose d'un ensemble d'agent qui surveillent le bon fonctionnement du système informatique.

La mise en œuvre de cette application nous a permis :

- faire la connaissance de l'environnement Eclipse pour le développement des applications Java
- L'utilisation pratique d'une plateforme multi agents « Jade » pour l'implémentation des SMA.

Et bien sûr, il y a toujours place pour l'amélioration de notre architecture, comme perspectives nous pouvons penser à :

- ❖ Rajouter d'autres services pour enrichir et compléter les modules de supervision afin d'être plus fiables et plus complets comme l'approche SNMP pur la supervision sur réseaux
- ❖ Élaborer d'autres versions pour d'autres systèmes d'exploitation comme linux.

.

[Arcangeli et Leriche, 2006] : J. P. Arcangeli et S. Leriche, ≪Un Framework a Composants et agents pour les applications repartis ≫ Dans : Revue des sciences et Technologies de l'information, série L'Objet, Lavoisier, Vol. 12, N. 4, p. 103- 132, décembre 2006.

[Bourron] :T. Bourron. Application des systèmes multi-agents dans les télécommunications : États de l'art, enjeux et perspectives.

[CAGLAYAN,1998]: A. CAGLAYAN, C. HARRISON « Les Agents, Applications bureautiques, *Internet, Intranet* ». InterEditions, 1998.

[Caillou et al, 2001] :P. Caillou, S. Aknine, et S. Pinson, ≪Méthode de formation et de restructuration dynamique de coalitions d'agents fondée sur L'optimum de Pareto , Dans Proc. 9emes Journées Francophones d'Intelligence Artificielles et Systèmes Multi-Agents (JFIAD), 2001.

[Chaib-draa et al, 2001] : B. Chaib-draa, I. Jarras et B. Moulin, ≪Systèmes Multi-Agents Principes généraux et applications ≫Editions J.P. Briot, Y. Demazeau, Agent et systèmes multi agents, Hermes, 2001.

[CHAIB, 02]: B. CHAIB-DRAA, I. JARRAS, « *Aperçu sur les systèmes multi agents*», Série Scientifique, Montréal, 2002.

[Cohen, 2003] : B. Cohen, Incentives Build Robustness in BitTorrent . Proceedings of the 1st Worshop on Economics of Peer-to-Peer Systems, Berkeley, USA, 2003.

[DEMAZEAU, 1995]: Y. DEMAZEAU, « *From Interactions to Collective Behaviour in Agent- Based Systems* », Proceeding of the First European Conference on Cognitive Science, Saint-Malo, p. 117-132, 1995.

[DIENG,1990]: R. Dieng. Relations Linking Cooperating Agents. In Proceedings of the 2nd European Workshop MAAMAW'90, pages 185-202, Saint-Quentin en Yvelines-France, August 1990.

[DURFEE, 1990]: E. H. Durfee and T. A. Montgomery. A Hierarchical Protocol for Coordinating Multi-agent Behaviors. In Proceedings of the 8th National Conference on Artificial Intelligence, pages 86-93, Cambridge, July- August 1990. Volume One.

[Durfee, 2001]: E.H. Durfee, Distributed Problem Solving and Planning. Pages 118–149, 2001.

[El-Fallah et al, 1996]: A. El-Fallah Seghrouni et S. Haddad, ≪A coordination algorithm for multi-agent planning Dans: Proceedings of International Conference on Multi-agent Systems-1996 (ICMAS'96), AAAI, Press, Kyoto, Japan, 1996.

[ETZIONIA,1994] : Etzioni O., Weld D., An Softbot-based interface to the Internet. *Communication of the ACM*, 1994, 37, 7, pp. 72-76.

[FERBER, 1997]: J. FERBER, « *Les Systèmes Multi-Agents : vers une Intelligence Collective* » Inter-éditions, 1997.

[FEREBR,1995] : J.Ferber, Les Systèmes Multi-Agents. Vers une intelligence collective. Inter Éditions, 1995

[FININ, 1993]: T. FININ, G. WIEDERHOLD, "A*n Overview of KQML: A nowledge Query and Manipulation Language*", Department of Computer Science, Stanford University, 1993.

[FININ, 1994]: T. FININ, R. FRITZON, D. MCKAY, R. MCENTIRE, "*KQML as an agent communication language*", In Proceedings of the Third International Conférence on Information and Knowledge Management (CIKM'94), ACM Press, 1994.

[FININ, 1994b]: T. FININ, R. FRITZON, D. MCKAY, R. MCENTIRE. "*KQML A Language and Protocol for Knowledge and Information Exchange*", (Technical Report No. CS-94-02). University of Maryland, Department of Computer Science, 1994.

[FININ, 1994c]: T. FININ, J. WEBER, G. WIEDERHOLD, M. GENESERETH R. FRITZON, J. McGuire, S. Shapiro, D. McKay, R. Pelavin, C. Beck, "*Specification of the KQML Agent Communication Language plus example agent policies and architectures (DRAFT Report)*", DARPA Knowledge Sharing Initiative External Interface Working Group, 1994.

[Milojicic et al, 2002] : D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins et Z. Xu, Peer-to-Peer Computing , HP Laboratories Palo Alto, report HPL-2002-57.

[NWANA,1996]: Nwana H. S., Software Agents: An Overview. In Knowledge Engineering Review, Vol. 11(3), 1996.

[GRAY,1997]: Rus D., Gray R., Kotz D., Transportable Information Agent. *Journal of Intelligent Information systems.* 9, 3, pp. 215-238, 1997

[Rho00]: Rhodes J., Just-In-Time Information Retrieval. Ph.D. Thesis, MIT Media Lab,

May 2000.

[Ripeanu et al, 2002] : M. Ripeanu, A. Iamnitchi et I. Foster, ≪Mapping the Gnutella Network ≫Internet Computing, IEEE, Vol.6, Issue 1, p. 50-57, 2002.

[SEC, 03]: Y. SECQ, « *RIO : Rôles, Interactions et Organisations une méthodologie pour les systèmes multi-agents ouverts* », Thèse Doctorat, Université des Sciences et Technologies de Lille, France, 2003.

[Shehory et al, 1998]: O. Shehory et S. Kraus, ≪Methods for Task Allocation via Agent Coalition Formation ≫Artificial Intelligence, 101(1–2):165–200, 1998.

[TrEsp99] : Tranvouez E., Espinasse B., Protocoles de coopération pour le réordonnancement d'atelier. Ingénierie des Systèmes Multi-Agent (Actes des Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents), (Novembre 1999 ; Saint Gilles), JFIADSMA'99, Ed. par Marie ierre Gleizes, Paris : Hermes, 1999.

[Von Martial, 1990]: F. von Martial, ≪Interactions among autonomous planning agents ≫ Dans Decentralized AI - Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi- Agent Worlds (MAAMAW-89). Elsevier Science Publishers B.V.: Amsterdam, Pays Bas, 1990.

LISTE DE FIGURES :

Liste des abréviations

ACC :	Agent Communication Chanel
ACL :	Agent communication Language
AID :	Agent Identifier
AMS :	Agent Management System
API :	Application Programming Interface
AP :	Agent Provider
AP_F	Agent Provider Facilitator
CPU :	Central Unit Processing
DF :	Directory Facilitator
ED :	Environnement de Développement
FIPA:	Foundation For Intelligent Physical Agent
J2SE :	Java 2 Standard Edition
JADE :	Java Agent DEvelopement framework
JDK :	Java Development Kit
JRE :	Java Runtime Environment
JVM :	Java Virtual Machine
SMA :	Système Multi Agents
TILAB :	Telecom Italia LABoratories
GUI :	Graphical User Interface
JAR :	Java ARchive
LEAP	Light Extensible Agent Platform
RMA	Remote Monitoring Agent

Résumé :

Avectous les nouveaux équipements et les nouvelles technologies apparues et qui évoluent chaque jour, les réseaux se complexifient et la tâche de la surveillance devient de plus en plus difficile, pour cela nous avons construit un système
Multi-agent qui se base sur les réseaux pair à pair qui propose un cadre de réponse à deux enjeux: autonomie et organisation. Les agents du système développé sont programmés en JAVA et sous la plate forme JADE.

Mots clés : Pair à pair, Système Multi-Agents, agent mobile, système de supervision.

Abstract :

With all the new equipment and new technologies emerged and which are changing every day, the networks have become more complex and the task of supervision becomes more and more difficult, for that we have built a multi-agent systembased on peer to peer networks which proposes a framework for response to two issues: autonomy and organization. The agents of our system are developed in JAVA language under the JADE platform.

Key words: P2P, Multi-agent System, mobile agent, supervision system.

ملخص:

مع جميع المعدات والتكنولوجيات الجديدة التي تتطور يوميا, تزداد مهمة حماية و مراقبة الشبكات صعوبة ،و لهذا أنشأنا نظاما متعدد العملاء الذي يستند بدوره على أنظمة الشبكات الثنائية و الذي يقترح الرد على موضوعين: الاستقلال الذاتي والتنظيم ،حيث أن العملاء مبرمجون باستخدام لغة البرمجة JAVA عن طريق الأرضية JADE.

الكلمات الرئيسية:الشبكات الثنائية، متعدد العملاء، العميل المتنقل، نظام الرقابة