

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université de 8 Mai 1945 -Guelma
Faculté des Mathématiques,d'informatique et des Sciences de la
matière
Département d'informatique



Mémoire de projet de fin d'étude
Filière :Informatique
Option :Informatique Académique

thème :

Optimisation d'allocation de Ressources
Limitées par un système Multi Agents

Encadré par :
Mme Benhamza Karima

Présenté par :
Aib Mohamed Nadji
Djellal Mouhssin

juin 2017

Résumé

L'optimisation d'allocation de ressources est un problème important qui se retrouve dans plusieurs applications de la vie courante. L'objectif recherché est de déterminer la meilleure attribution d'un ensemble de ressources à une population afin d'optimiser un but commun. Dans ce travail, nous abordons ce problème à travers une approche multi-agents qui est un axe de recherche très prometteur. Le choix d'une application s'est orienté vers un modèle (Acheteurs/vendeurs).

Mots clés : Allocation, Ressources, Optimisation, Système multi-agent (SMA).

Table des matières

I	 système Multi-Agents	11
1	Introduction	11
2	le concept d'agent :	11
2.1	Définitions :	11
2.2	Différence entre objet et agent [Ferber J, 1995] :	13
2.3	Caractéristique d'un agent [jarras I,2002] :	14
2.4	Classification des agents [Blangi P,2004/2005][Daily N,2007] [Mazyad H,2013]	14
3	système Multi-Agent	16
3.1	Définitions :	16
3.1.1	Définition 1[Jarras I,2002] :	16
3.1.2	Définition 2[Ferber j,1995] :	16
3.1.3	Définition 3 [Chaib D,2001] :	17
3.1.4	Définition 4 [Demazeau Y,1995] :	17
3.2	Les caractéristiques des SMA [Azaiez S,2007][Cernuzzi L,2002] [Benhawala F,2008] :	18
3.2.1	L'environnement	18
3.2.2	L'interaction dans un SMA [Ferber J ,1995][Bour- don F,1999/2000] [Belaqziz S,2014] :	19
3.2.3	L'organisation [Cernuzzi L,2005] :	20
3.2.4	La Communication entre les Agents [Ferber J ,1995] :	20
3.2.4.1	Communication par partage d'infor- mation [Azaiez S,2007] :	20
3.2.4.2	Communication par envoi de mes- sage [Daily N,2007]	21
3.3	Avantages d'une approche multi-agent	21
3.4	Problématiques des SMA [Chaib D,2001]	22
3.5	Domaine d'application des SMA [Daily N,2007][Demazeau Y,1995] :	23

3.6	Architecture des SMA (Bendahmane T,2015) :	24
3.6.1	Les systèmes centralisés (tableau noir) :	25
3.6.2	Les systèmes hiérarchiques	25
3.6.3	Les systèmes distribués	25
3.7	Les plates-formes orientées agents[Azaiez S, 2007]	26
3.7.1	La plate-forme MADKIT :	26
3.7.2	La plate-forme JADE	26
4	Conclusion	27
 II Optimisation D'allocation de Ressources Limitées		 28
1	Introduction	28
2	Concepts de base :	28
2.1	Tache :	28
2.2	Ressources :	28
2.3	Ressources limités [Medernach E,2011]	29
2.4	Ressources critique [Medernach E,2011] :	29
2.5	Caractéristique de ressources :	29
2.6	Exemples de ressources :	30
2.7	l'Allocation de ressources (Affectation) [Medernach E, 2011] [Vivek V, 2013] :	30
2.7.1	Critère d'allocation de ressources[Tan Z ,2007] :	31
2.7.2	L'interblocage dans l'allocation des taches aux ressources :	31
2.7.3	Règles d'allocation de ressources en temps réel :	32
2.8	L'Optimisation des ressources :	33
2.8.1	Domaines d'utilisation :	33
2.8.2	Critère d'optimisation :	34
2.8.2.1	Principe de l'équité [Madernach E,2011] [Young H,1994] :	34
2.8.2.2	Critères d'équité :	34
2.9	Travaux des SMA :	35
2.9.1	Un Système d'Information de Transport Multimodal (SITM) d'aide à la mobilité urbaine [Zgaya H, 2007] :	35
2.9.2	Un Système de santé [Daknou A ,2011] :	36
3	Conclusion	36
 III Conception et implémentation		 37
1	Introduction :	37

2	Choix de l'application :	37
3	Modélisation :	37
3.1	Modélisation d'agents :	37
3.2	Fonctionnement du système :	38
3.3	Scenario de flux d'information :	39
4	Implémentation :	40
4.1	Matériel utilisé :	40
4.2	Langage utilisé :	40
4.3	Outil utilisé :	41
4.4	Présentation de l'application :	42
4.4.1	Bibliothèque java utilisé :	42
4.4.2	Les classes utilisé :	42
4.5	Simulation et interactions entre les agents :	42
4.6	Communication entre les agents :	43
4.7	Lancement de l'application :	44
4.7.1	Autorisation d'accès :	44
4.7.2	Interface Menu principal :	44
4.7.3	Fenêtre d'histogramme :	46
4.7.4	Fenêtre de paramétrage :	47
4.8	Résultats obtenus :	48
4.8.1	Résultat comparatif :	48
4.8.2	Résultat histogramme :	49
4.8.3	Résultat fille d'attente :	49
5	Conclusion :	50

Table des figures

I.1	Représentation imagée d'un agent en interaction avec son environnement et les autres agents [Ferber J, 1995]	12
I.2	Représentation des tâches mise en oeuvre lors du fonctionnement de l'agent	13
I.3	Représentation de l'objet	13
I.4	Représentation de l'agent	13
I.5	Modèle d'agent réactif	15
I.6	Modèle d'agent cognitif	16
I.7	L'agent et l'environnement	17
I.8	La communication par partage d'information	21
I.9	La communication par envoi de message	21
I.10	Une classification des différents types d'application des systèmes multiagents	24
II.1	Affectation d'allocation	30
II.2	Le rangement des processus dans une liste circulaire	32
II.3	Représentation d'un exemple d'optimisation	33
III.1	Présentation des différentes relations entre agents	38
III.2	Schéma du flux d'information entre les différents agents.	40
III.3	Bibliothèque utilisée	42
III.4	Les classes utilisées	42
III.5	Fenêtre de Manipulation des Agents JADE	43
III.6	Autorisation d'accès	44
III.7	Message d'erreur	44
III.8	Présentation du Menu principal de l'application	45
III.9	Remplissage des valeurs.	45
III.10	Affichage de la file d'attente et le temps d'attente moyen de chaque acheteur	46
III.11	Histogramme avant exécution	46
III.12	Présentation de la fenêtre paramétrage	47
III.13	Résultat comparatif	48

III.14	Résultat histogramme	49
III.15	Résultat file d'attente	49

Liste des abreviations

1-SMA :Système Multi Agent.

2-AEIO :Agent Environnement Interaction Organisations .

3-IAD :intelligence artificiel distribué.

4-KS :knowledge source.

5-MADKIT :Multi Agent Developement Kit.

6-JADE :Java Agent Development framework.

7-FIFO :first in first out.

8-SJF :short job first.

9-STPT-LPT :Shortest Total Processing Time Largest ProcessingTim.

10-STPT :Shortest Total Processing Time.

11-SPT-Local :Shortest Total Processing Time Local.

12-SITM :Système d'Information de Transport Multimodal.

13-FIPA :Foundation for Intelligent Physical Agents.

14-ACL :Agent Communication Language.

REMERCIEMENT

Nous tenons à remercies en premier lieu Allah qui nous a donné vie et santé pour le parachèvement de ce modeste travail.

Nous remercions après de tout cœur notre encadreur «Mme Benhamza Karima » pour son soutien, sa sympathie, ses encouragements, la confiance qu'elle nous témoignée en acceptant de diriger ce travail et pour avoir mis à notre disposition ses conseils pour une meilleure maîtrise du sujet.

Nous remercions nos familles qui nous ont toujours donné la possibilité de faire ce que nous voulions durant nos études et qui ont toujours cru en nous.

En fin, nous remercions tous ceux qui ont contribué à ce travail par leurs remarques, leurs suggestions et leurs soutiens.

Aib Mohamed Nadji

Djellal Mohcen

Introduction Générale

L'accès incontrôlée à des ressource limitées conduit surement à une utilisation inutile ou moins puissante qui défavorise la file des utilisateurs. En effet, chaque individu essaie de maximiser son intérêt personnel et il aura tendance à s'approprier de manière exceptionnelle les ressources au désagrément des autres.

Afin de surmonter ce problème, on doit inclure des outils et des méthodes pour garantir une part équitable aux différents utilisateurs et assurer l'optimisation du système.

Dans ce travail, nous cherchons un cadre pour l'optimisation multi-agents à travers le problème d'allocation et d'ordonnancement de ressources.

Les systèmes multi-agents (SMA) propose des concepts notamment avantageux pour le développement de ces outils.

Organisation du mémoire

Ce mémoire est organisé en trois chapitres présentant respectivement, deux chapitres pour l'état de l'art, et un chapitre pour la modélisation et la réalisation de l'application.

- Le premier chapitre s'intéresse au domaine et la technologie agent et les systèmes multi-agents.
- Le deuxième chapitre passe en revue les notions liées à l'optimisation d'allocation de ressources limité.
- Le troisième chapitre contient les détails du modèle que nous avons proposé pour une approche basée agent, Il contient aussi l'implémentation du modèle proposé, une simulation des actes de communications entres agents et des aperçus montrant les différents interfaces de l'application.

Nous terminons ce mémoire par une conclusion générale qui résume l'apport essentiel de notre travail.

Chapitre I

ystème Multi-Agents

1 Introduction

Dans ce chapitre nous parlons sur Les Systèmes Multi Agents (SMA), ces systèmes rassemblent les travaux qui portent sur l'étude et la conception d'organisations d'agents autonomes, Le concept d'agent est aujourd'hui plus qu'une technologie efficace.

Tout d'abord On va présenter dans ce chapitre d'une manière générale le concept d'agent et les systèmes multi-agents, par la suite nous y expliquons ce qu'est un agent, un (SMA) ainsi que les interactions entre eux. Puis Les avantages d'une approche multi-agent, Nous présentons différentes problématiques des (SMA), Domaine d'application des Systèmes Multi-Agents, Enfin les dernières parties décrivent quelques Architecture des (SMA) et plates-formes orientées agents.

2 le concept d'agent :

2.1 Définitions :

Un agent est une entité réelle ou virtuelle, dont le comportement est autonome, évoluant dans un environnement, qu'il est capable de percevoir, sur lequel il est capable d'agir et d'interagir avec les autres agents.[Demazeau Y,1996]

Selon [Ferber J,1995] l'agent est : " une entité autonome physique ou abstraite qui est capable d'agir sur elle même et sur son environnement, qui dans un univers Multi Agents 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" .

A partir de cette définition, on comprend que l'agent est une entité située dans un environnement comprenant des objets passifs, et d'autres agents

où chacun cherche à satisfaire ses propres objectifs, pour cela il exécute un ensemble de tâches qu'il voit adéquates suivant sa situation courantes et suivant son répertoire d'action et le but qu'il désire atteindre.

Le processus de fonctionnement de l'agent comprend donc trois phases successives : [Chami D, 2010]

1. Une phase de perception qui permet d'élaborer une idée sur l'état actuel de l'environnement via un ensemble de capteurs.
2. Une phase de délibération qui permet de décider quelle action à exécuter suivant l'état de l'environnement et l'état interne de l'agent, c'est à ce niveau que le comportement de l'agent est décrit donc c'est la phase la plus importante.
3. Et finalement la phase d'action c'est l'exécution de l'action par l'actionneur correspondant, l'exécution qui va apporter des modifications sur l'état de l'environnement et sur l'état de l'agent.

Toutes ces définitions se fondent sur des notions semblables qui caractérisent l'agent, incluant l'autonomie et la capacité d'agir et de percevoir, et permettent de distinguer plusieurs types d'agents.

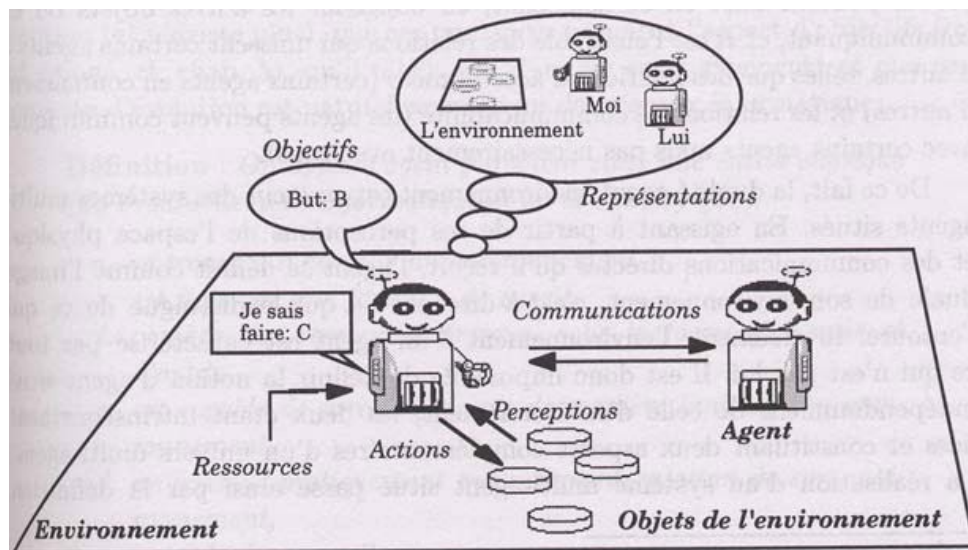


FIGURE I.1 – Représentation imagée d'un agent en interaction avec son environnement et les autres agents [Ferber J, 1995]

En va représenter un autre schéma qui facilite bien la compréhension du fonctionnement d'un agent :

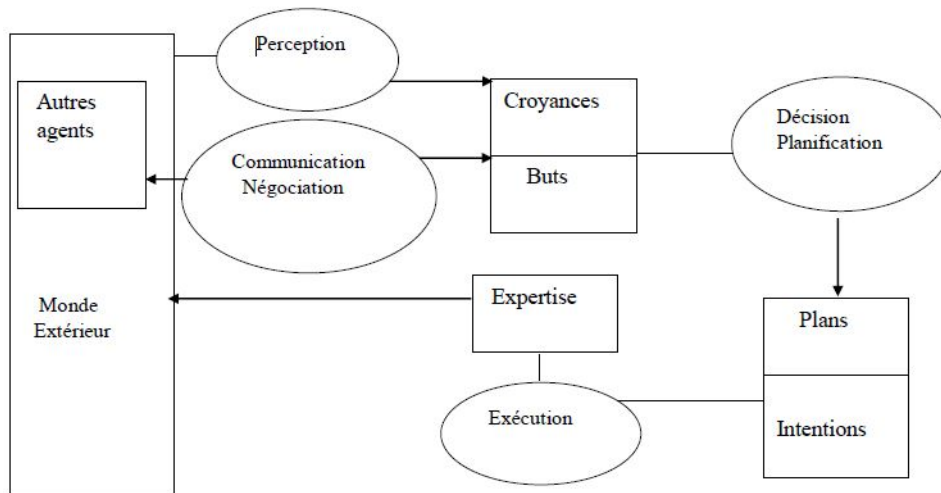


FIGURE I.2 – Représentation des tâches mise en oeuvre lors du fonctionnement de l'agent

2.2 Différence entre objet et agent [Ferber J, 1995] :

Il ya une différence entre un agent et un objet en commence notre comparaison par dire que l'objet est une entité passive (ou réactive). Si personne ne demande la valeur d'un attribut ou n'active une méthode de l'objet, alors il ne se passe rien.

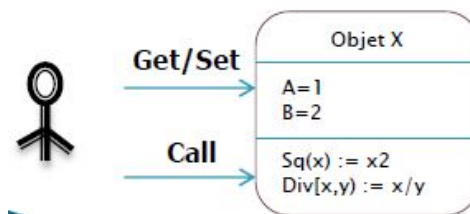


FIGURE I.3 – Représentation de l'objet

Contrairement à l'agent qui possède, en plus des attributs et méthodes, des tâches internes qui fonctionnent même en l'absence de sollicitations externes. Un agent peut donc agir même si personne ne lui demande rien.

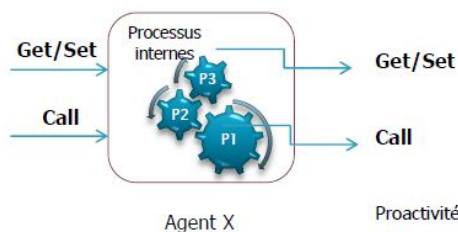


FIGURE I.4 – Représentation de l'agent

2.3 Caractéristique d'un agent [Jarras I,2002] :

Les différentes caractéristiques de l'entité agent peuvent être résumées dans la définition suivante : « un agent est un système informatique situé dans un environnement, et qui agit de façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu ». En partant des définitions citées, on peut identifier les caractéristiques suivantes pour la notion d'agent.

(a) **Autonome :**

L'agent est autonome s'il prend des décisions motivées par son état interne sans intervention extérieure.

(b) **Réactif :**

On dit qu'un agent est réactif dans son environnement s'il est capable de le percevoir et de réagir aux changements intervenant par ses actions.

(c) **Social :**

Un agent est dit social s'il est capable d'interagir avec d'autres agents.

(d) **Proactif :**

C'est-à-dire qu'il n'agit pas uniquement en réponse à son environnement mais, il est également capable de prendre l'initiative au bon moment.

2.4 Classification des agents [Blangi P,2004/2005][Daily N,2007][Mazyad H,2013]

Les agents sont généralement classés selon trois courants de pensée : l'école réactive, l'école cognitive et l'école hybride.

(a) **Les agents réactifs :**

L'école « réactive » prétend au contraire qu'il n'est pas nécessaire que les agents soient intelligents individuellement pour que le système ait un comportement global intelligent. Les agents réactifs sont des agents simples qualifiés non intelligents, ils répondent d'une façon opportune aux modifications de leurs environnements résultant des stimuli externes (leurs actions sont provoquées et non pas choisies), ils agissent en fonction de ces dernières sans nécessité de compréhension de leurs univers ni de leurs buts.

Les sociétés d'agents réactifs sont caractérisées par le nombre important des agents qui sont capables ensemble de produire des actions évoluées qu'on appelle l'émergence de l'intelligence, mais dont les individus pris séparément ne possèdent qu'une représentation faible de leur environnement et des buts globaux. Il y a deux modèles qui peuvent servir à la conception d'agent réactif :

1 Agent à réflexes simples

Ce type d'agent agit uniquement en se basant sur ses perceptions courantes. Il utilise un ensemble de règles prédéfinies, du type Si condition alors action, pour choisir ses actions.

2 Agent conservant une trace du monde

Le type d'agent qui a été décrit précédemment, ne peut fonctionner que si un tel agent peut choisir ses actions en se basant uniquement sur sa perception actuelle.

En va expliquer mieux cela en donnant un schéma explicatif sur l'agent réactif :

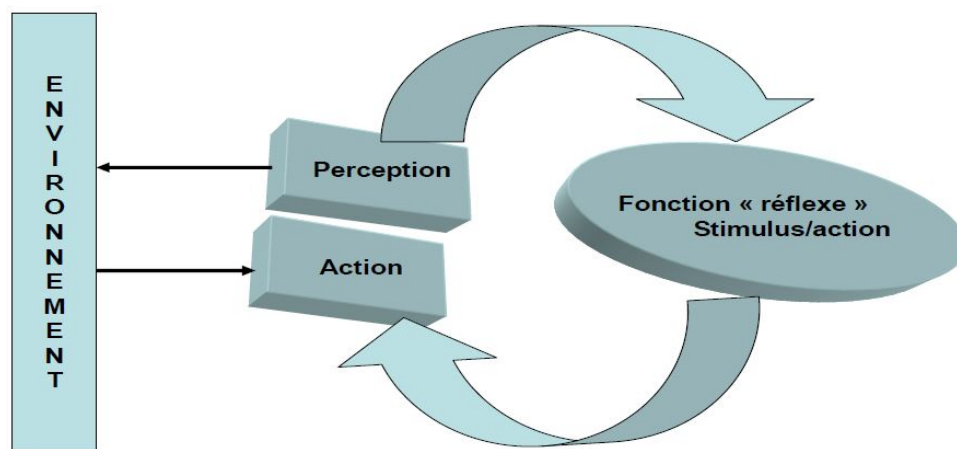


FIGURE I.5 – Modèle d'agent réactif

(b) Les agents cognitifs [Benhawala F,2008][weerd M,2009] :

Ils sont parfois dits "intentionnels", leur caractéristique fondamentale est la volonté de communiquer et de coopérer et la mémorisation du passé. Ils possèdent des buts à atteindre à l'aide d'un plan explicite.

Les sociétés d'agents cognitifs contiennent communément un petit groupe d'individus qui sont des agents intelligents où chacun possèdera une base de connaissances comprenant l'ensemble des informations et des savoir-faire nécessaires à la réalisation de sa tâche et à la gestion des interactions avec son environnement et les autres agents.

Un agent cognitif raisonne sur ses croyances qui représentent sa compréhension du monde dans lequel il évolue. Il raisonne aussi sur ses désirs et intentions en relation avec ses croyances et capacités afin de prendre des décisions auxquelles sont associés les plans qu'il va accomplir pour agir dans le monde .

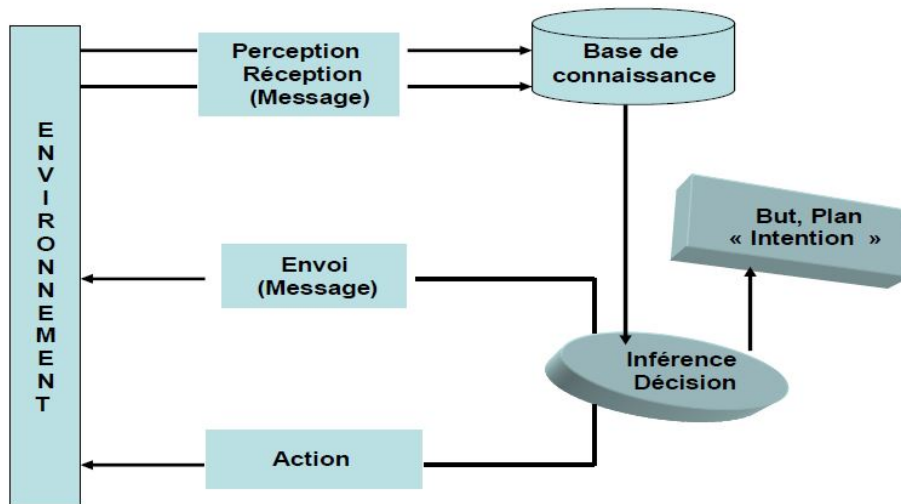


FIGURE I.6 – Modèle d’agent cognitif

(c) **Agent hybride**

Les agents hybrides sont des agents ayant des capacités cognitives et réactives. Ils conjuguent en effet la rapidité de réponse des agents réactifs ainsi que les capacités de raisonnement des agents cognitifs.

3 système Multi-Agent

3.1 Définitions :

Plusieurs auteurs s’accordent généralement pour définir un SMA comme un système composé d’agents qui communiquent et collaborent pour achever des objectifs spécifiques personnels ou collectifs, en va donc citer quelque définitions pour bien comprendre c’est quoi un SMA

3.1.1 Définition 1[Jarras I,2002] :

Les systèmes Multi-agents sont des systèmes distribués conçus idéalement comme un ensemble d’agents interagissant le plus souvent selon des modes de coopération, de coordination et de négociation et de coexistence .

3.1.2 Définition 2[Ferber j,1995] :

Les systèmes Multi-Agents sont des systèmes composés des éléments suivants :

- (a) Un environnement : C’est-à-dire un espace disposant généralement d’une métrique.

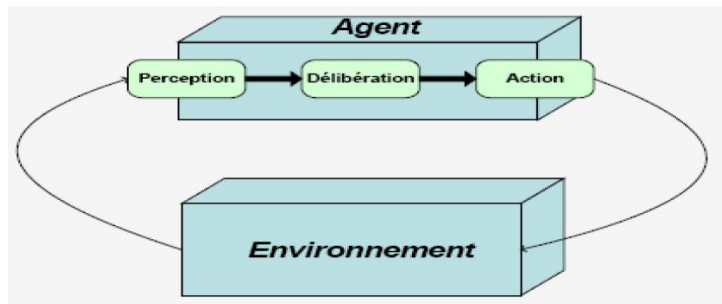


FIGURE I.7 – L'agent et l'environnement

- (b) Un ensemble d'objets : Situés dans cet environnement cela signifie que pour tout objet il est possible à un moment donné d'associer une position, ces objets sont passifs, c'est-à-dire qu'ils peuvent être créés, détruits, manipulés et perçus par les agents.
- (c) Un ensemble d'agents : Ce sont des objets particuliers, ils représentent les entités actives du système.
- (d) Un ensemble de relations : Qui unissent les objets entre eux.
- (e) Un ensemble d'opérations : Ce sont les différents types de manipulation qu'appliquent les agents sur les objets du système et qui sont en générale : perception production, consommation, transformation... etc.
- (f) Un ensemble d'opérateurs : Chargés de représenter l'application de ces opérations et la création du monde à cette tentative de modification, que l'on appellera les lois de l'univers. "

3.1.3 Définition 3 [Chaib D,2001] :

Un système multi-agents est un système distribué composé d'un ensemble d'agents interagissant selon des modes de coopération, de concurrence ou de coexistence.

3.1.4 Définition 4 [Demazeau Y,1995] :

Un SMA est un système composé d'agents qui évoluent dans un environnement et interagissent de manière organisé(AEIO). Ainsi, un SMA est défini selon l'équation ci-dessous :

$$SMA = Agents + Environnement + Interaction + Organisations \quad (I.1)$$

3.2 Les caractéristiques des SMA [Azaiez S,2007][Cernuzzi L,2002][Benhawala F,2008] :

Les systèmes multi-agents sont issus du domaine de l'intelligence artificielle distribuée. Ainsi, dans un SMA, les traitements et les données sont distribués mais aussi les compétences, les rôles et les buts des agents. Ces agents ont chacun un point de vue partiel et il n'y a aucun contrôle global du système. En plus, ils sont en activité permanente et prennent leurs propres décisions en fonction de leurs objectifs et leurs connaissances et sont capables de communiquer entre eux selon des langages plus ou moins élaborés .

Les agents sont situés dans un environnement où ils interagissent en vue de réaliser conjointement une tâche ou d'atteindre conjointement un but particulier. Ainsi, c'est la notion d'interaction qui distingue un SMA d'une collection d'agents indépendants Par ailleurs, cette notion est une caractéristique principale d'un SMA en plus de l'organisation qui est produite par l'interaction entre les agents.

3.2.1 L'environnement

Dans un système multi-agents, on appelle environnement l'espace commun aux agents du système. Un environnement peut être :

(a) **Déterministe**

Ou non, selon que l'état futur de l'environnement ne soit, ou non, fixé que par son état courant et les actions de l'agent.

(b) **Episodique**

Si le prochain état de l'environnement ne dépend pas des actions réalisées par les agents.

(c) **Statique**

Si l'état de l'environnement est stable pendant que l'agent réfléchit. Dans le cas contraire, il sera qualifié de dynamique.

(d) **Discret**

Si le nombre des actions faisables et des états de l'environnement est fini. Les caractéristiques de l'environnement influencent la façon dont on conçoit un agent car il faut tenir compte de l'évolution de l'environnement, de la capacité de l'agent de saisir cette évolution et de sa capacité à décider en conséquence.

3.2.2 L'interaction dans un SMA [Ferber J ,1995][Bourdon F,1999/2000] [Belaqziz S,2014] :

Un système Multi-agents est un système cohérent et intelligent constitué de plusieurs agents qui se chargent de réaliser un but commun, mais cette cohérence et cette intelligence généralement ne vient pas de l'intelligence des agents qui le composent mais de leurs interactions. L'interaction est définie comme une mise en relation dynamique entre deux ou plusieurs agents par le biais d'un ensemble de relations réciproques .

Les interactions ne sont pas seulement la conséquence des actions effectuées par plusieurs agents en même temps, mais aussi l'élément nécessaire à la constitution d'organisations sociales.

(a) **La coopération**

Chaque agent dispose des compétences qui lui permettent de résoudre certains problèmes ou effectuer certaines tâches, mais parfois ses capacités ne seront pas suffisantes devant des situations complexes, donc l'agent aura besoin de l'intervention des autres agents qui vont l'aider à faire évoluer le système vers ses objectifs. Donc la coopération est la participation de plusieurs agents pour satisfaire un but individuel ou commun.

(b) **La négociation**

Les buts des agents dans un système Multi-agents peuvent être incompatibles et leurs demandes sont parfois contradictoires, donc il faut trouver un moyen qui permet à chaque agent de poursuivre son travail, autrement dit les agents doivent négocier la solution.

On définit la négociation comme le processus d'améliorer les accords (en réduisant les inconsistances et l'incertitude) sur des points de vue communs ou des plans d'action grâce à l'échange structuré d'informations pertinentes . On distingue trois types de cardinalités dans un processus de négociation .

1 **Négociation un-à-un** :Un agent négocie avec un autre

2 **Négociation un-à-plusieurs** :Un seul agent négocie avec plusieurs autres agents. C'est le type de cardinalité dans le système étudié dans ce travail, où un agent superviseur négocie avec plusieurs agents agriculteurs

3 **Négociation plusieurs-à-plusieurs** :Plusieurs agents négocient avec plusieurs d'autres agents en même temps.

(c) **La coordination**

La coexistence des agents dans un environnement où les ressources sont limitées, et où les objectifs locaux des différents agents peuvent se contredire provoquant l'apparition de situations de conflits qui influencent sur le rendement global de tout le système, et ainsi diminuent les avantages de coopération. Pour vaincre ces situations de conflits et d'en sortir ou d'en éviter, les agents sont amenés à exécuter des actions supplémentaires (or que les actions productives), ces dernières sont dites actions coordinatrices. Le concept de coordination regroupe l'ensemble d'outils et méthodes qui peuvent être employés pour résoudre des conflits (dus aux ressources partagées limitées ou aux objectifs incompatibles) ou pour optimiser des comportements (éliminer des actions redondantes et inutiles) et plus généralement pour assurer un tout cohérent.

3.2.3 L'organisation [Cernuzzi L,2005] :

L'organisation d'un système multi-agents est la manière dont le groupe d'agent est constitué, à un instant donné, pour pouvoir fonctionner. Cette organisation peut être statique ou dynamique. Son dynamisme est le fait que le groupe se réorganise à chaque fois en fonction de la tâche à accomplir.

3.2.4 La Communication entre les Agents [Ferber J ,1995] :

La communication est l'un des concepts pertinents dans les systèmes multi-agents. Comme chez les humains, la communication est à la base de l'interaction et de l'organisation sociale, elle permet aux agents de coopérer, négocier, échanger des informations et effectuer des tâches en commun. Sans communication, on distingue essentiellement deux modèles de communication :

1. Communication par partage d'information.
2. Communication par envoi de messages.

3.2.4.1 Communication par partage d'information [Azaiez S,2007] :

La communication entre les différents agents du système est réalisée par partage d'information lorsque ceux-ci disposent d'une zone de données commune dans laquelle ils rangent les conclusions qu'ils ont pu tirer. Outre ces résultats partiels, cette zone renferme Les données du problème initial. Les agents peuvent ainsi y puiser les informations dont ils ont besoin pour résoudre une

partie du problème globale. Ce type de communication correspond à ce que la littérature désigne communément sous le nom de modèle du blackboard (tableau noir).

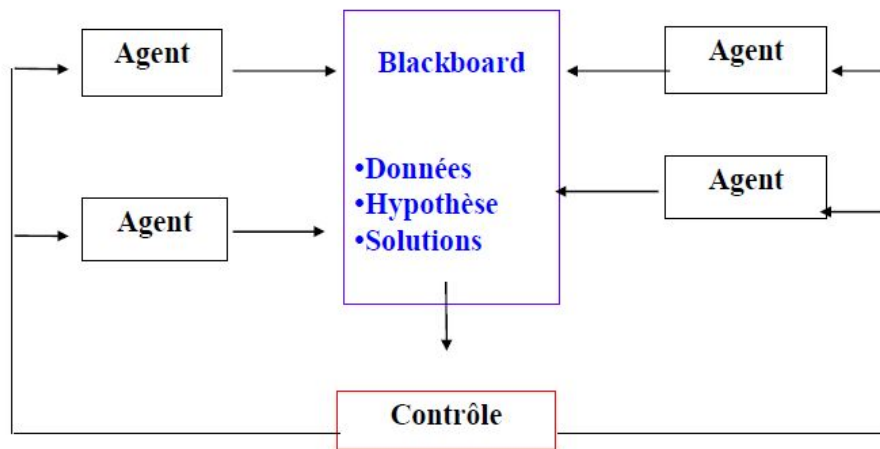


FIGURE I.8 – La communication par partage d’information

3.2.4.2 Communication par envoi de message [Daily N,2007] : Les SMA fondés sur la communication par envoi de messages se caractérisent par le fait que chaque agent possède une représentation propre et locale de l’environnement qui l’entoure. Chaque agent va alors interroger les autres agents sur cet environnement ou leur envoyer des informations sur sa propre perception des choses.

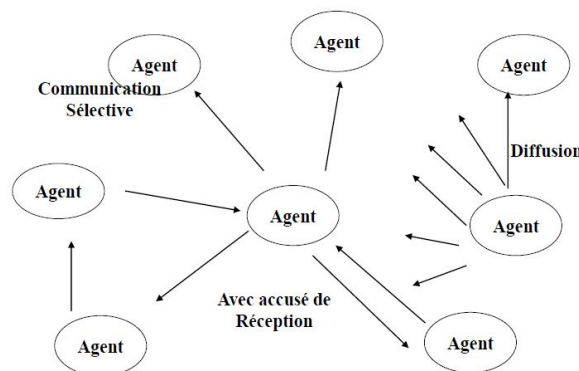


FIGURE I.9 – La communication par envoi de message

3.3 Avantages d’une approche multi-agent

Le système multi-agent présente les avantages suivants par rapport à un système d’un seul agent ou approche centralisée[Ferber J,1995] :

- (a) Le système multi-agent distribue les ressources et les capacités de calcul à travers un réseau des agents interconnectés, considérant qu'un système centralisé peut être en proie à des limitations de ressources ou les pannes critiques, un système multi-agent est un système décentralisée et donc ne souffre pas d'une point unique de défaillance "single point of failure" problème associé à des systèmes centralisés.
- (b) Un système multi-agent permet l'interconnexion et l'interopérabilité des multiples systèmes existants par construit un agent encapsuler.
- (c) Un système multi-agent résoudre les problèmes en termes d'interaction autonomes des composants des agents, qui s'avère être un moyen plus naturel de représenter l'allocation des tâches, la planification d'équipe, les préférences des utilisateurs, les environnements ouverts, et ainsi de suite.
- (d) Un système multi-agent récupère efficacement et filtrer des informations coordonnées à partir de sources qui sont spatialement distribués.
- (e) Un système multi-agent propose des solutions dans des situations où l'expertise est spatialement et temporellement distribuée.
- (f) Un système multi-agent améliore les performances globales du système, spécifiquement sur les dimensions de l'efficacité de calcul, la fiabilité, l'extensibilité, la robustesse, la maintenabilité, la réactivité, la flexibilité et la réutilisation.

3.4 Problématiques des SMA [Chaib D,2001]

On peut relever cinq problématiques principales lors de la création de systèmes multi-agents :

- (a) D'abord, la problématique de l'action : comment un ensemble d'agents peut agir de manière simultanée dans un environnement partagé, et comment cet environnement interagit en retour avec les agents ?
Les questions sous-jacentes sont entre autres celles de la représentation de l'environnement par les agents, de la collaboration entre agents, de la planification multi-agent.
- (b) Ensuite ,la problématique de l'agent et de sa relation au monde, qui est représentée par le modèle cognitif dont dispose l'agent. L'individu d'une société multi-agent doit être capable de mettre en œuvre les actions qui répondent au mieux à ses objectifs. Cette capacité à la décision est liée à un "état mental" qui reflète les perceptions, les représentations, les croyances et un certain nombre de paramètres "psychiques" (désirs,

tendances. . .) de l'agent. La problématique de l'individu et de sa relation au monde couvre aussi la notion d'engagement de l'agent vis-à-vis d'un agent tiers.

- (c) Les systèmes multi-agents passent aussi par l'étude de la nature des interactions, comme source de possibilités d'une part et de contraintes d'autre part. La problématique de l'interaction s'intéresse aux moyens de l'interaction (quel langage ? quel support ?), et à l'analyse et la conception des formes d'interactions entre agents. Les notions de collaboration et coopération (en prenant coopération comme collaboration + coordination d'actions + résolution de conflits) sont ici centrales.
- (d) On peut évoquer ensuite la problématique de l'adaptation en termes d'adaptation individuelle ou apprentissage d'une part et d'adaptation collective ou évolution d'autre part.
- (e) Enfin, il reste la question de la réalisation effective et de l'implémentation des SMA, en structurant notamment les langages de programmation en plusieurs types allant du langage de type L5, ou langage de formalisation et de spécification, au langage de type L1 qui est le langage d'implémentation effective. Entre les deux, on retrouve le langage de communication entre agents, de description des lois de l'environnement et de représentation des connaissances.

3.5 Domaine d'application des SMA [Daily N,2007][Demazeau Y,1995] :

Les domaines d'application des SMA sont particulièrement riches, On peut considérer qu'il existe cinq grandes catégories d'applications des systèmes multi-agents : la résolution de problèmes au sens large, la robotique distribuée, la simulation multi-agent, la construction de mondes hypothétiques et la conception génétique de programmes.

voici donc des définitions de quelques catégories d'application :

- (a) **La résolution des problèmes** : Qui concerne toutes les situations dans lesquelles des agents logiciels, purement informatiques, accomplissent des tâches utiles aux êtres humains.
- (b) **La robotique distribuée** : Porte sur la réalisation d'un ensemble de robots qui coopèrent pour accomplir une mission et utilise des agents concrets qui se déplacent dans un environnement réel.
- (c) **La simulation Multi-agents** : La simulation est la démarche scientifique qui consiste à réaliser une reproduction artificielle, appelée modèle,

d'un phénomène réel que l'on désire étudier, à observer le comportement de cette reproduction lorsqu'on en fait

Varié certains paramètres, et à en induire ce qui se passerait dans la réalité sous l'influence de variations analogues.

Voici un schéma explicatif qui représente ces grandes catégories d'application des SMA :

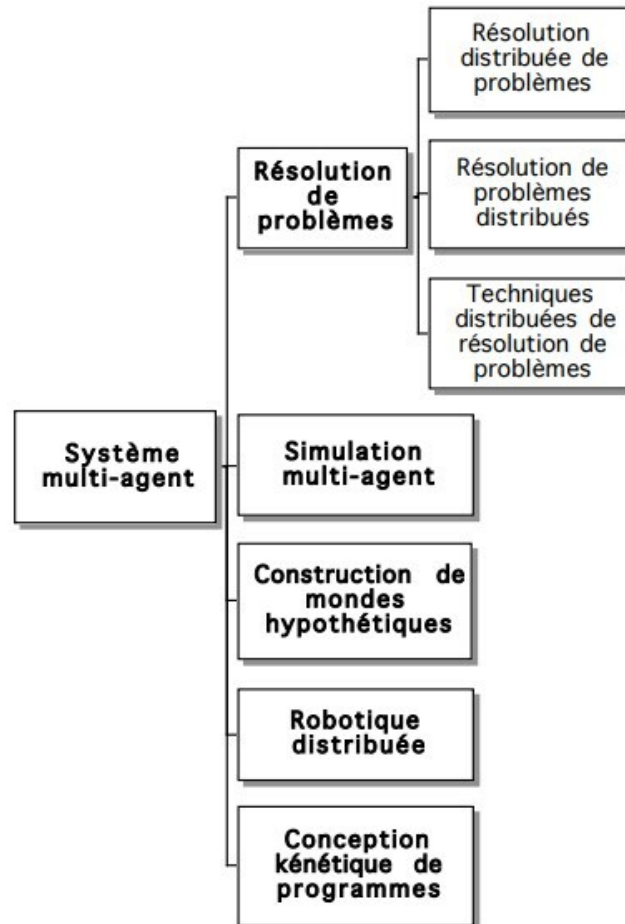


FIGURE I.10 – Une classification des différents types d'application des systèmes multiagents

La réussite des applications fondées sur les SMA développés dans plusieurs domaines d'application comme la simulation, la gestion d'informations, le commerce électronique démontre que les SMA paraissent être l'approche la plus adaptée à un nombre important de domaines.

3.6 Architecture des SMA (Bendahmane T,2015) :

Les systèmes multi agents existent sous plusieurs formes à savoir :

3.6.1 Les systèmes centralisés (tableau noir) :

L'architecture de tableau noir est l'une des plus utilisée dans les systèmes multi agents cognitifs, elle s'est rapidement imposée en IAD comme une architecture suffisamment souple et puissante pour pouvoir implémenter les mécanismes de raisonnement et de calculs intervenant à l'intérieur des agents. Le modèle de tableau noir est fondé sur un découpage en modules indépendants appelés source de connaissances (KS) qui interagissent indirectement en partageant leurs informations sur un espace de travail

(tableau noir) qui contient les états partiels d'un problème en cours de résolution et toutes les informations que s'échangent les KS. Et pour éviter tout conflit d'accès à cette ressource unique, un dispositif de contrôle gère tous les conflits d'accès entre les KS.

L'architecture d'un système à base de tableau noir comprend trois sous-systèmes

- (a) Les sources de connaissance (KS).
- (b) La base partagée (le "tableau" proprement dit) qui comprend toutes les informations que s'échangent les KS.
- (c) Un dispositif de contrôle qui gère les conflits d'accès entre les KS.

3.6.2 Les systèmes hiérarchiques

Basés sur une structure où les agents répondent à leur supérieur hiérarchique en terme organisationnel, comme par exemple dans une usine. Ainsi, le système est divisé en plusieurs niveaux, chaque groupe d'agent d'un niveau inférieur interagissent entre eux dans le même niveau pour aboutir à un but global qui est le but de l'agent chef du Niveau supérieur, et ainsi jusqu'à l'arrivée au dernier niveau où le but global du système est émergent.

Le peu de tolérance aux fautes et les limites imposées par les capacités des supérieurs hiérarchiques représentent l'inconvénient de ce type de système.

3.6.3 Les systèmes distribués

La distribution est un atout pour les SMA, elle permet de contrôler la complexité des problèmes à résoudre en décomposant le système en sous-systèmes constitués d'un ou de plusieurs agents et effectuant chacun une partie du travail ce qui conduit à une meilleure adaptation aux changements de l'environnement extérieur.

L'approche distribuée procure de grands avantages comme la très grande tolérance aux fautes, le partage équitable du travail entre les sous-systèmes et/ou agents, l'utilisation plus uniforme des ressources, l'indépendance et l'autonomie des sous-systèmes, la répartition des tâches, l'efficacité du modèle concurrent et/ou parallèle, leur grande extensibilité, etc.

3.7 Les plates-formes orientées agents [Azaiez S, 2007]

Il existe une multitude de plates-formes multi-agents dédiées à différents modèles d'agent.

Les plates-formes fournissent une couche d'abstraction permettant de facilement implémenter les concepts des systèmes multi-agents. D'un autre côté, elle permet aussi le déploiement de ces systèmes. Ainsi, elles constituent un réceptacle au sein duquel les agents peuvent s'exécuter et évoluer. En effet, les plates-formes sont un environnement permettant de gérer le cycle de vie des agents et dans lequel les agents ont accès à certains services.

Dans ce qui suit, nous présentons quelques plates-formes.

3.7.1 La plate-forme MADKIT :

La plate-forme MadKit (Multi-Agents Development kit) est développée à l'Université de Montpellier II. Bien qu'elle puisse supporter le développement de divers systèmes, elle semble bien adaptée pour les applications de simulation.

La plate-forme MADKIT est basée sur les concepts agent, groupe et rôle. Cette implémentation correspond à la conception réalisée au niveau de la méthodologie AALAADIN. à partir des concepts AGR, cette méthodologie définit une démarche de développement axée sur la spécification du cadre organisationnel des applications multi-agents. Cette démarche définit l'ensemble des rôles possibles, spécifie les interactions, et décrit les structures abstraites de groupe.

3.7.2 La plate-forme JADE

La plate-forme JADE (Java Agent Development framework) est certainement celle qui est la plus utilisée par la communauté des systèmes multi-agents. JADE permet de développer et d'exécuter des applications distribuées basées sur le concept d'agents et d'agents mobiles.

JADE n'offre pas de méthodologie, par contre plusieurs méthodologies la prennent comme plate-forme cible lors de la génération de code. L'implémen-

tation de Jade est basée sur Java. La plate-forme peut être répartie sur un ensemble de machines et configurée à distance. La configuration du système peut évoluer dynamiquement puisque la plate-forme supporte la mobilité des agents.

4 Conclusion

On a présenté dans ce chapitre, une vue détaillée sur un approche qui est actuellement un domaine de recherche très actif, l'approche de concepts inhérents aux agents et aux systèmes multi agents, comme les méthodologies de développement, les problématiques des systèmes multi-agent (SMA), les domaines d'application des SMA, et les plates-formes orientées agents.

Dans le prochain chapitre nous allons décrire la simulation d'allocation de ressources limitées et la résolution du problème d'allocation de ses ressources.

Chapitre II

Optimisation D'allocation de Ressources Limitées

1 Introduction

Dans ce chapitre nous allons présenter les concepts de bases liés à la problématique d'optimisation d'allocation de ressources limitées, qu'ils se retrouvent souvent dans la vie courante, L'objectif est de déterminer la meilleure manière de distribuer ou bien d'affecter un ensemble de tâches au nombre de ressources qui sont limités.

Nous poursuivrons notre recherche sur la question de comment répartir les ressources entre les agents et sur quels critères nous baser. Nous y expliquons ce qu'est une tâche, une ressource, ressource limités, ressource critique, ainsi l'allocation de ressource puis l'optimisation de ressources, Nous présentons les différents critères d'optimisation, Enfin on va donner quelques travaux sur l'optimisation d'allocation de ressources limitées.

2 Concepts de base :

2.1 Tache :

Une tâche est une entité élémentaire d'activité qui est localisée dans le temps par une date de début et une date de fin (ou par une de ces dates et sa durée). Pour être réalisée, une tâche utilise une ou plusieurs ressources avec une intensité qui est habituellement considérée comme invariable pendant l'exécution de la tâche. [Kocsis T, 2011]

2.2 Ressources :

Selon [Kocsis T, 2011] une ressource est un moyen technique ou humain, disponible en quantité limitée, qui est requis pour la réalisation d'une tâche.

La durée d'une tâche n'est pas toujours connue à priori, mais peut être en fonction de la quantité et de la performance de la ressource utilisée.

Exemple :

Selon [Ferrié J, 1997] Pour s'exécuter, une tâche a besoin d'un certain nombre d'éléments : Des éléments matériels comme de la mémoire, un processeur, des périphériques (clavier et écran par exemple) et des éléments logiciels, fichiers, tables. Tous ces éléments matériels et logiciels sont appelés des ressources.

2.3 Ressources limités [Medernach E,2011]

Ce sont des ressources utilisables dont la quantité est limitée. La rareté des ressources veut dire que les ressources sont produites en quantité limitée.

2.4 Ressources critique [Medernach E,2011] :

Lorsque les ressources sont à accessibilité limitée on peut dire que ces ressources sont critiques. Par exemple une ressource d'ordinateur ne pouvant être utilisée que par un seul processus à la fois.

Exemple :

L'unité centrale dans un système monoprocesseur est une ressource critique. Lorsqu'un processus demande une ressource donnée, toute unité disponible de ce type peut lui être attribuée.

2.5 Caractéristique de ressources :

On peut dire qu'une ressource tout ce qui est nécessaire à l'avancement d'une tâche, on va donc la caractériser par les caractéristiques suivantes :

- (a) Un défaut de ressource peut provoquer la mise en attente d'une entité.
- (b) la tâche demande l'accès à une ressource.
- (c) Certaines demandes sont implicites ou permanentes.
- (d) Le système alloue une ressource à une tâche.
- (e) On dit qu'une ressource est en mode d'accès exclusif si elle ne peut être allouée à plus d'une tâche à la fois.

Exemple :

Un disque est une ressource à accès exclusif (un seul accès simultané), une

zone mémoire peut être à accès partagé. Le mode d'accès à une ressource dépend largement de ses caractéristiques technologiques. Deux ressources sont dites équivalentes si elles assurent les mêmes fonctions vis à vis de la tâche demandeur. Les ressources équivalentes sont groupées en classes afin d'en faciliter la gestion par l'Ordonnanceur.

2.6 Exemples de ressources :

Dans notre recherche on a trouvé plusieurs exemple de ressources on va citer celle que nous avons trouvé intéressant :

- (a) Le processeur (dont la référence est implicite) c'est-à-dire Le système doit allouer l'unité centrale au processus pour que l'exécution puisse se faire.
- (b) La mémoire : Le processus a toujours besoin de la mémoire soit pour y stocker ses instructions soit pour ses données. La mémoire peut être allouée à l'exécution.
- (c) Les fichiers

2.7 l'Allocation de ressources (Affectation) [Medernach E, 2011] [Vivek V, 2013] :

Avant d'utiliser une ressource, une tâche doit la demander, explicitement ou implicitement, il doit la libérer après utilisation. Les opérations de demande et de restitution d'une ressource sont appelées l'allocation et la libération.

Une ressource est susceptible de réquisition (« préemption ») si le système peut la retirer au tâche auquel elle était allouée pour l'affecter à une autre tâche.

Voici un schéma qui représente un modèle de décomposition du problème d'allocation

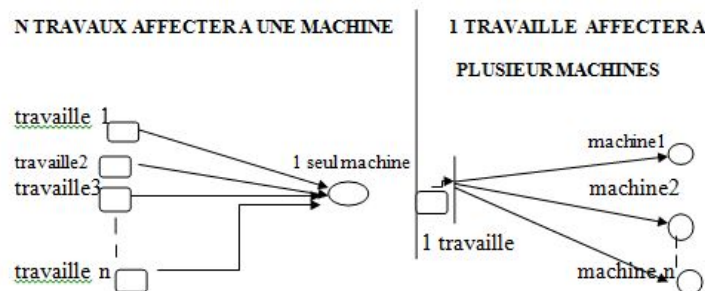


FIGURE II.1 – Affectation d'allocation

2.7.1 Critère d'allocation de ressources[Tan Z ,2007] :

Pour une allocation efficace des ressources, il devrait éviter les situations suivantes :

1. Sur-provisioning : se pose lorsque le fournisseur alloue pour le consommateur des ressources plus que la demande.
2. Sous-provisioning : se produit lorsque l'allocation des ressources est moins que la demande.
3. Situation conflictuelle de ressource : se pose lorsque deux applications tentent d'accéder à la même ressource en même temps.
4. Le manque en ressources : se pose lorsque les ressources sont limitées.
5. La fragmentation des ressources : cette situation se pose lorsque les ressources sont isolées. [Il y a suffisamment de ressources, mais l'allocation est impossible.]

Des ressources doivent être allouées de façon permanente à des utilisateurs. Ces ressources sont considérées comme générales et peuvent être diverses comme des espaces mémoire, des partitions de disque, de la bande passante réseau, des bandes de fréquences [Ahmed M,2009], des orbites satellitaires [BouveretS, 2006], voire l'approvisionnement en eau [Wang L, 2004].

2.7.2 L'interblocage dans l'allocation des taches aux ressources :

Selon [Ferrié J,Mossière J,1972] Un ensemble de taches sont interbloqués si chacun d'eux est bloqué en attente d'une ressource qui a été allouée à l'un des autres taches. Plusieurs conditions sont requises pour qu'il puisse y avoir interblocage :

- (a) Les ressources allouées doivent être à accès exclusif.
- (b) Une tache qui possède déjà des ressources peut faire de nouvelles demandes d'allocation et donc se bloquer en attente de l'obtention de la nouvelle ressource demandée.
- (c) La réquisition des ressources est impossible.
- (d) L'attente doit être circulaire. Les taches peuvent être rangés dans une liste circulaire, chaque tache attendant une ressource possédée par le suivant.

On peut distinguer des règles pour résoudre le problème d'affectation ou bien d'allocation de ressources en temps réel et on va les citer ci-dessous.

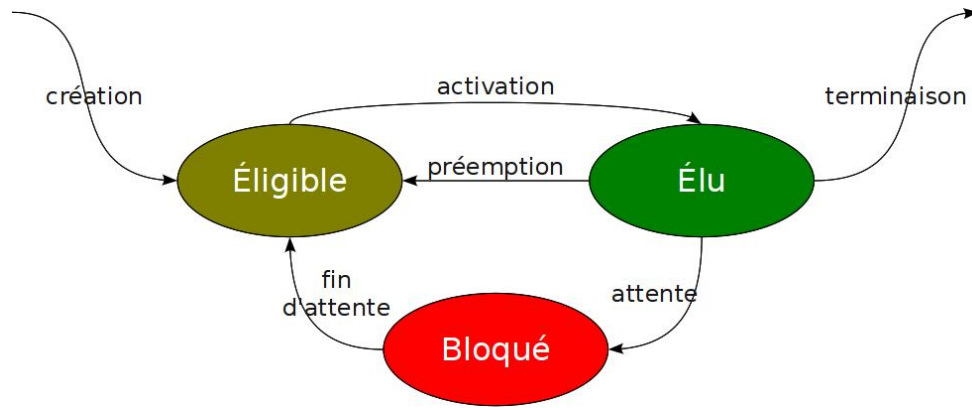


FIGURE II.2 – Le rangement des processus dans une liste circulaire

2.7.3 Règles d'allocation de ressources en temps réel :

Selon [Hennet J, 1997] Si l'on connaît les caractéristiques stationnaires ou quasi-stationnaires des lois d'arrivées, on peut calculer les taux moyens optimaux d'affectation, de façon exacte ou approximative selon les hypothèses retenues.

Deux types de techniques sont alors envisageables pour résoudre le problème d'affectation de ressources en temps réel :

- (a) **Les techniques statiques** : Basées uniquement sur des règles de priorité ou d'affectation automatique.
- (b) **Les techniques dynamiques** : Qui utilisent des informations disponibles sur l'état de chaque ressource, et sur les nombres d'opérations déjà en attente ou planifiées sur cette ressource.

En outre, la prise en compte, comme paramètres de référence, des taux moyens optimaux d'affectation, donne naissance à plusieurs algorithmes d'allocation dynamique des ressources, parmi lesquels on peut mentionner les suivants :

- (a) **Répartition de Bernoulli** : Affectant les jobs selon un tirage aléatoire respectant en moyenne des taux moyens optimaux d'affectation.
- (b) **Affectation cyclique** : Approchant les taux moyens optimaux d'affectation par la répétition périodique de la même séquence d'affectations.
- (c) **Affectation pseudo-cyclique** : Réalisant avec une certaine tolérance les taux moyens optimaux grâce à un organe de commande représenté par un réseau de Petri cyclique value.

Pour une allocation efficace de ressources on a besoin d'une optimisation, cette optimisation cherche à apporter des réponses à un type général de

problèmes qui consistent à sélectionner le meilleur élément parmi plusieurs appartenant au même ensemble. On présente ci-dessous tout ce qui concerne l'optimisation des ressources.

2.8 L'Optimisation des ressources :

Le mot optimiser se définit comme la recherche de l'optimum (minimum ou maximum) d'une fonction f donnée, appelée fonction objectif ou fonction de coût ou encore critère d'optimisation. Ce besoin d'optimiser découle de la nécessité de fournir à l'utilisateur un système qui puisse répondre au mieux à ses exigences. C'est donner les meilleures conditions d'utilisation, de fonctionnement, de rendement.

2.8.1 Domaines d'utilisation :

Dans notre recherche on a trouvé plusieurs domaines concernant l'optimisation de ressources qui sont très importante dans notre vie quotidienne tel que le militaire, le transport comme les aéroports, les routes les trajets, aussi les contrôles des réseaux comme les infrastructures et les distributions, ainsi que le domaine d'informatique qui facilite nos tâche comme la gestion de processus et la gestion de mémoire.

Exemple (problème d'optimisation) : On à 3 machines et 8 taches, chaque tache utilise une unité de temps, l'objectif c'est comment affecter les taches aux machines de manière à minimiser le temps utilisé. Le schéma à gauche explique que les 8 taches sont accomplies au bout de 7 unité de temps sur 3 machines, et le schéma à droite signifié que les 8 tâches sont accomplies au bout de 6,5 unités de temps, don il y a m^n possibilité

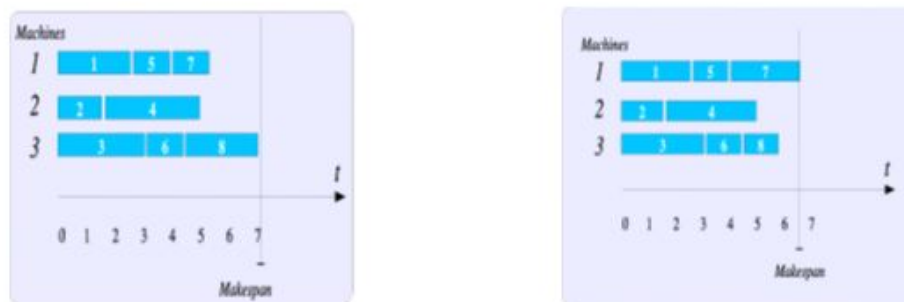


FIGURE II.3 – Représentation d'un exemple d'optimisation

2.8.2 Critère d'optimisation :

Il Ya plusieurs critère d'optimisation dans les quelle elle peut arriver à des résultats satisfaisantes, on a choisis comme critère le critère de l'équité qu'on va l'expliquer brièvement ci-dessous.

2.8.2.1 Principe de l'équité [Madernach E,2011] [Young H,1994] :

La question de l'équité se divise en deux groupes distincts :

Tout d'abord celui de la répartition où plusieurs utilisateurs font des demandes à un gestionnaire de ressources qui décide de la répartition à donner à chacun. La fonction du gestionnaire est de répartir les ressources entre utilisateurs de la façon la plus équitable possible.

Un autre groupe de problème est celui du partage où plusieurs fournisseurs indépendants approvisionnent en ressources plusieurs clients. De même on souhaite que la part donnée par les fournisseurs ainsi que la part reçue par les clients soit répartie de la façon la plus équitable possible.

Les besoins des utilisateurs ne peuvent pas être totalement prévus. Toutefois un planning prévisionnel de l'utilisation des ressources où différents scénarios sont envisagés, est établi. Ce planning renseigne sur l'évolution des demandes. Lorsqu'un groupe d'utilisateurs se partage une ressource commune, ce type de problème survient.

2.8.2.2 Critères d'équité : Dans notre recherche on a trouvé plusieurs critère d'équité qu'on va les citer si dessous :

- (a) **Équité en Débit de calcul [Bender M , 1998] :** Ce critère est équivalent à une vitesse d'exécution. Dans la perspective qu'un utilisateur qui obtient un débit de calcul faible aura tendance à changer pour un site offrant un débit plus élevé.
- (b) **Équité en Attente :** La comparaison s'effectue sur le temps d'attente des tâches, défini comme la différence entre la date de début d'exécution et la date de disponibilité de la tâche.
- (c) **Équité en Temps de séjour :** On considère avec ce critère de comparaison que deux utilisateurs sont une qualité de service équivalente lorsque les temps entre l'arrivée et l'achèvement sont identiques.

Ce principe d'évaluation peut être une agrégation de différents critères selon les besoins des utilisateurs ou des différents domaines d'application utilisés. L'important étant de pouvoir comparer le niveau de service obtenu par ces utilisateurs et ainsi de pouvoir décider quel est le moins bien servi.

Evidement selon le critère retenu pour cette comparaison différents choix d'allocation seront possibles. Il est possible de distinguer plusieurs types d'ordonnanceurs. Leurs principales fonctions sont les suivantes [Motwani R, 1993] [Medernach E , 2005] :

- (a) **Premier arrivé, premier servi (FIFO)** : Exécute les tâches dans l'ordre de leur arrivée.
- (b) **Plus court d'abord(SJF)** : Cette politique suppose que la quantité de calcul totale est connue pour chaque utilisateur, sans que les durées des tâches ne le soient. (SJF) exécute les tâches à l'instant t dans l'ordre croissants
- (c) **"STPT-LPT" (Shortest Total Processing Time Largest Processing Time)** : Compare les utilisateurs présents selon l'ordre des temps croissants et lorsqu'une machine se libère exécute la tâche de plus grande durée pour l'utilisateur sélectionné.
- (d) **STPT ("Shortest Total Processing Time")** : Elle place les tâches des utilisateurs selon l'ordre des quantités croissants. Cette politique place en premier toutes les tâches de l'utilisateur qui a la quantité de calcul la plus petite, puis continue sur l'utilisateur suivant.
- (e) **SPT-Local ("Shortest Total Processing Time Local")** : C'est une politique de liste dans laquelle on maintient une file d'attente par utilisateur : à chaque instant t où une machine se libère, cette politique exécute, parmi l'ensemble des premières tâches reçues pour chaque utilisateur et non exécutées, celle qui minimise la quantité. Chaque utilisateur voit donc ses tâches s'exécuter dans l'ordre de leur arrivée.

2.9 Travaux des SMA :

Plusieurs travaux d'optimisation d'allocation de ressource se basent sur le système multi-agent existant, on peut citer :

2.9.1 Un Système d'Information de Transport Multimodal (SITM) d'aide à la mobilité urbaine [Zgaya H, 2007] :

Un système qui se base sur une approche multi-agent en intégrant des algorithmes d'optimisation. L'architecture du SITM est ouverte, dynamique et distribuée, composée de sociétés d'agents. Ce système permet d'optimiser la gestion des flux de données pour faire face à un nombre important de requêtes simultanées. L'optimisation intègre le temps de réponse des requêtes,

le coût de l'information, la taille des données transférées et l'état du réseau informatique, ce qui fait que le problème résolu est multi-contraintes multi-objectifs.

2.9.2 Un Système de santé [Daknou A ,2011] :

Les professionnels de santé doivent s'engager dans de nouvelles actions destinées à la modernisation de l'offre de soin. Les exigences de maîtriser les dépenses tout en garantissant une meilleure qualité des soins poussent à adopter de nouvelles approches permettant de résoudre les problématiques liées à l'organisation du processus de prise en charge des patients et au dimensionnement des moyens humains et matériels.

Ils s'intéressent aux solutions visant la prise en charge rapide et efficace des patients se pose sur le paradigme Système Multi-Agent marie avec l'optimisation permettant ainsi l'amélioration de l'utilisation des ressources

3 Conclusion

Dans ce chapitre, on a vu un ensemble de concepts inhérents à la problématique d'optimisation d'allocation de ressources limité, on a compris aussi c'est quoi une tâche, une ressource, ressource limité, ressource critique, ainsi que l'allocation de ressource et l'optimisation de cette ressource, et on a vu plusieurs travaux concernant l'optimisation d'allocation de ressource limité avec les SMA. Dans le chapitre suivant, nous allons présenter notre démarche de conception et de l'implémentation d'une architecture multi-agents pour la réalisation d'une bonne optimisation d'allocation d'une ressource.

Chapitre III

Conception et implémentation

1 Introduction :

Dans ce chapitre, nous allons nous focaliser sur un problème classique et connu qui est la minimisation du temps d'attente d'un acheteur qui fait la queue dans une supérette afin d'avoir une liste des produits auprès d'un vendeur. Nous présentons les différentes étapes réalisées durant l'implémentation et la modélisation de notre application, ensuite nous définissons la fonctionnalité de notre application et enfin nous discutons les résultats obtenus.

2 Choix de l'application :

Notre application comporte deux parties la première consiste à représenter le cas d'un seul vendeur qui répond aux demandes de plusieurs acheteurs et la deuxième consiste à représenter le cas de deux vendeurs. Le nombre minimum de vendeurs est choisi pour souligner et valider l'optimisation de ressources limitées par les SMA. Notre objectif dans les deux cas est d'optimiser le temps d'attente moyen de tous les acheteurs. Les résultats obtenus sont comparés à la politique du premier venu premier servi (par défaut).

3 Modélisation :

3.1 Modélisation d'agents :

Comme toute application multi agents celle-ci comporte plusieurs agents selon leur rôle. Notre modèle est constitué de trois types d'agents associés aux différents acteurs de notre superette représentés ci-dessous :

- (a) **Agents acheteurs** : Ce sont ceux qui viennent à notre superette pour satisfaire leurs achats.

- (b) **Agents vendeurs** :Ce sont ceux qui servent les clients en leur donnant les produits demandés.
- (c) **Agents portail (coordinateur)** :Composé de deux agents coordinateurs .leur fonctionnalité consiste sur la coordination entre les acheteurs et les vendeurs afin d'organiser la vente.

3.2 Fonctionnement du système :

Dans notre application nous traitons le problème d'optimisation entre acheteurs vendeurs tel que le résultat à obtenir est minimisé le temps d'attente moyen des acheteurs. Pour arriver à ce résultat on a utilisé un algorithme qui teste au début le temps d'arrivé et le nombre de produits de chaque acheteur ensuite il va les trier dans la file d'attente. L'acheteur qui a un nombre de produit et un temps d'arrivé minimal va être affecter au vendeur si il est libre dans le cas d'un seul vendeur, dans le cas de deux vendeurs, les deux acheteurs qui arrivent au premier s'affectent directement aux deux vendeurs libre, les autre vont la libération d'un vendeur.

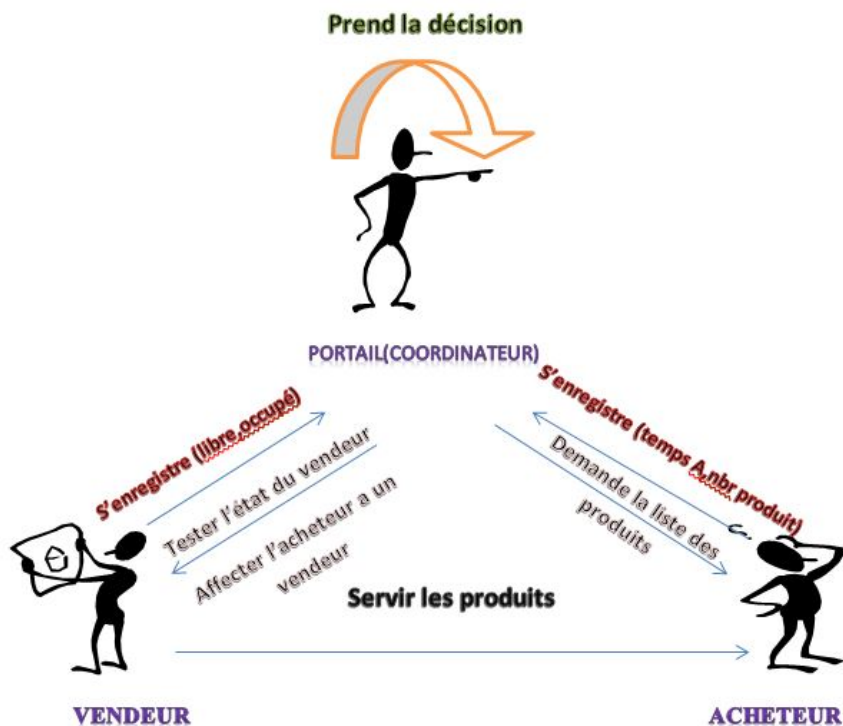


FIGURE III.1 – Présentation des différentes relations entre agents

3.3 Scenario de flux d'information :

Cette application va contenir plusieurs types d'agents raison pour laquelle nous nous sommes penchés sur un diagramme de flux d'information.

Nous avons créé trois catégories d'agents. Les agents appartenant au même système ne peuvent communiquer entre eux :

- (a) Les agents du système acheteur vont :
 - 1 -Entrer dans une queue.
 - 2 -S'enregistrer selon les temps d'arrivés et le nombre de produits achetés.
- (b) Pour le système portail (coordinateurs), nous avons deux agents .
 - 1 -L'agent coordinateur1 va veiller sur les requêtes suivantes :
 - i. Demande des produits auprès des acheteurs, Il reçoit leur réponse (la liste des produits) .
 - ii. Demande de l'Etat des vendeurs.
 - iii. Application d'algorithme d'optimisation
 - iv. Affectation de l'acheteur au vendeur libre.
 - 2 -L'agent coordinateur2 enregistre les différents acteurs de l'application.
- (c) Les agents vendeurs distribuent les produits demandés.

Le scénario de flux d'information est schématisé dans la figure suivante.(Figure III.2)

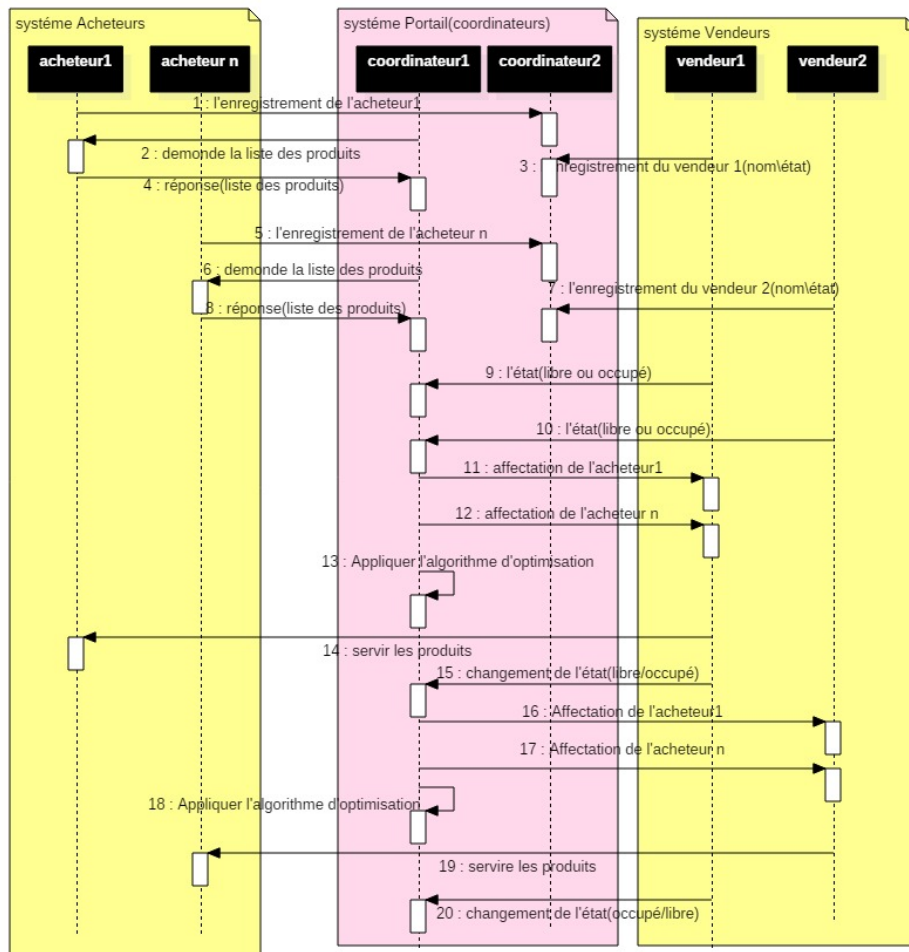


FIGURE III.2 – Schéma du flux d'information entre les différents agents.

4 Implémentation :

4.1 Matériel utilisé :

Nous avons utilisé un ordinateur Intel (R) core 5 duo cadencés à 2.00 MHz avec une RAM de 4.00 Go et un système Windows 8.1 édition Intégrale.

4.2 Langage utilisé :

Nous avons utilisé le langage java avec la plateforme JADE .La raison pour laquelle on a choisit java est que les agents développés sous Jade sont entièrement écrits en JAVA.

JAVA est un langage multiplateforme qui permet aux concepteurs, selon le principe <write once, run every where >, d'écrire un code capable de fonctionner dans tous les environnements (quel que soit le système d'exploitation).

JAVA est un langage orienté objet possède les caractéristiques suivantes :

- (a) **Orienté objet** : Veut dire un langage possède les propriétés d'encapsulation de l'héritage et polymorphisme, classe abstraites et interfaces.
- (b) **Multitâche** : Il permet l'utilisation des threads qui sont des unités d'exécution isolées.
- (c) **Interprété** : Le code source est compilé en pseudo code puis exécuté par un interpréteur java.
- (d) **L'indépendance de toute plate-forme** : La possibilité d'exécuter les programmes java sous tous les environnements qui possède une machine virtuelle java.
- (e) **Il est simple** : L'utilisation des pointeurs et la gestion de la mémoire est automatique.

Nous avons utilisé JADE car les agents développer sous la plate-forme Jade, sont entièrement écrits en Java. Jade assure une communication transparente par échange de messages dans le langage normalise FIPA-ACL, elle gère le cycle de vie des agents de leurs naissances jusqu'à leurs morts .

4.3 Outil utilisé :

On a utilisé le NetBeans IDE 8.2,c'est un environnement de développement intégré (EDI),placé en open source par Sun en juin 2000 sous licence CDDL, et GPLv2 (common Development and Distribution License).

En plus de java ,NetBeans permet également de supporter Différents autres langage(Python,C,C++,JavaSCRIPT,XML,Ruby ,PHP et HTML).

Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projet multi-langage,refactoring,éditeur graphique d'interface et de pages web) Conçu en java , NetBeans est disponible sous Windows ,Linux,MacOS X...ect

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)).L'IDENetbeans s'appuie sur cette plateforme.

4.4 Présentation de l'application :

Dans notre application on va présenter les interfaces graphiques associées aux différentes fonctionnalités de notre système. Les interfaces graphiques sont celles d'authentification, menu principale, Conception multi agents, et finalement les résultats comparatifs(Histogramme).

4.4.1 Bibliothèque java utilisé :



FIGURE III.3 – Bibliothèque utilisé

4.4.2 Les classes utilisé :

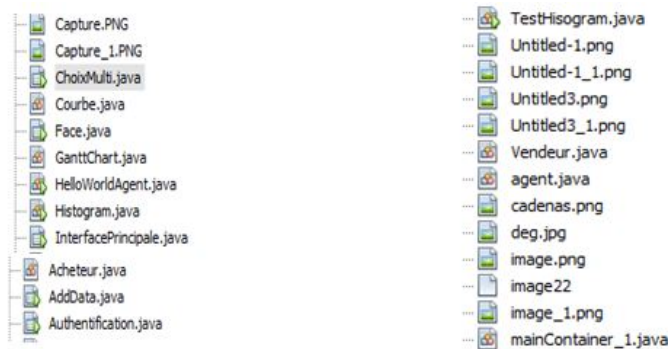


FIGURE III.4 – Les classes utilisé

4.5 Simulation et interactions entre les agents :

L'interaction entre agents définit par l'aptitude d'échanger des informations et des connaissances qui peuvent être mutuellement comprise. Comme l'interaction entre les personnes, l'interaction entre les agents a besoin d'un langage de communication commun, une capacité d'échanger les connaissances et une compréhension mutuelle des connaissances échangées [Barceinas, 1998]

4.6 Communication entre les agents :

Au début les agents (portail, vendeurs, acheteurs) vont être lancé, ensuite l'agent portail envoie des messages aux agents acheteurs pour demander la liste de produit de chaque acheteur afin d'envoyer cette liste aux agents vendeur celle-ci qui va satisfaire les besoins des agents acheteurs en envoyant des messages de remerciement. Enfin l'agent portail libère l'agent vendeur pour une autre commande.

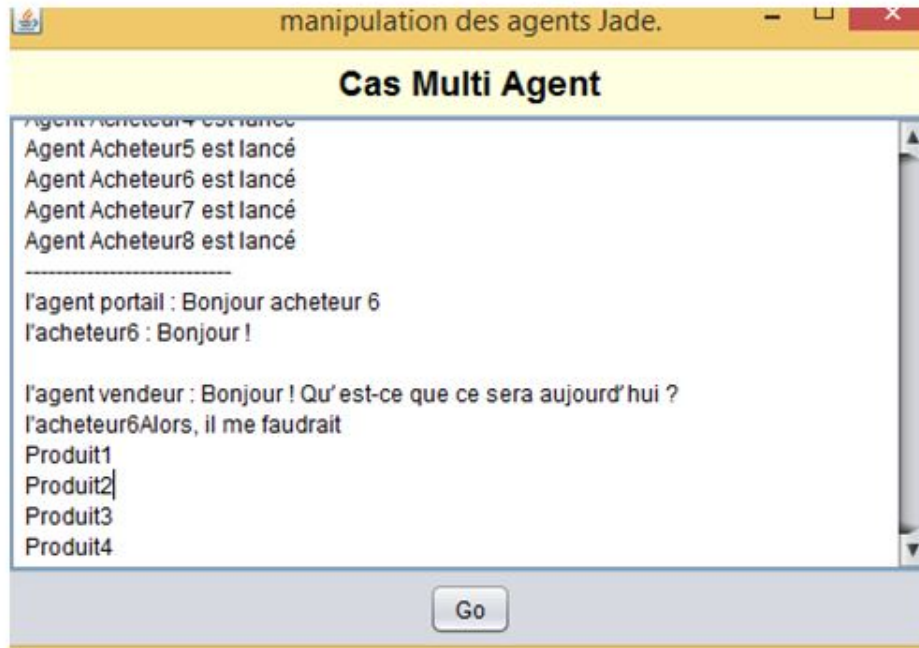


FIGURE III.5 – Fenêtre de Manipulation des Agents JADE

4.7 Lancement de l'application :

4.7.1 Autorisation d'accès :

Après avoir lancé l'application, une fenêtre apparaîtra qui demande de saisir le pseudonyme et le mot de passe puis cliquer sur le bouton « valider » afin d'accéder au Menu principal de l'application.

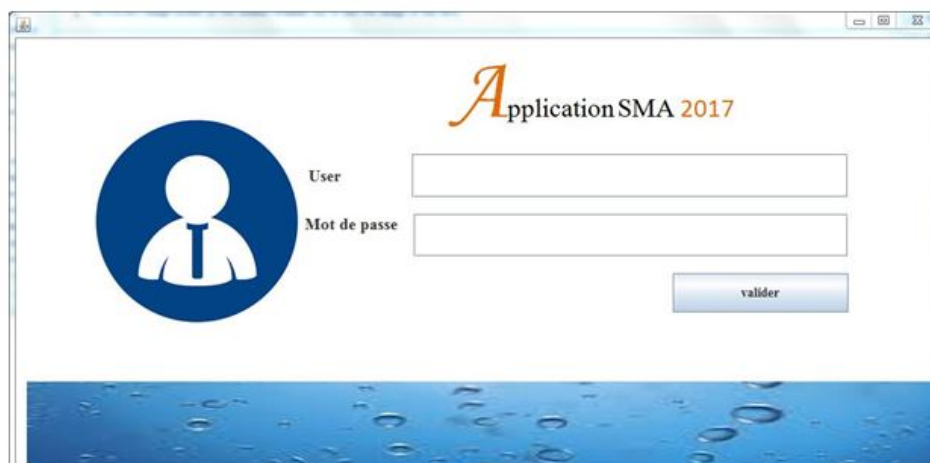


FIGURE III.6 – Autorisation d'accès

Si le mot de passe ou/et le pseudonyme est incorrecte (s), un message d'erreur apparaîtra (erreur) .



FIGURE III.7 – Message d'erreur

Après avoir saisi la combinaison juste (mot de passe-pseudo) on tombe sur la fenêtre principale de l'application

4.7.2 Interface Menu principal :

Notre système vérifie par la suite la combinaison, et si elle est juste une autre fenêtre de Menu principale va être affichée.

On a conçu une fenêtre de paramétrage avec laquelle l'utilisateur peut gérer le nombre d'acheteurs (de 1 à 8) et le nombre de vendeurs (1 ou bien 2)

Il nous reste juste à noter que notre projet se focalise sur le cas d'un seul vendeur ou de deux vendeurs, ainsi si l'utilisateur choisissait l'algorithme d'optimisation c'est-à-dire l'acheteur qui a un nombre de produits minimal servi par le premier vendeur libre, si les deux vendeurs sont occupés à la fois il se met en attente en respectant son temps d'arrivée. Si l'utilisateur choisissait l'algorithme par défaut c'est-à-dire l'acheteur qui vient au premier lieu va affecter au premier vendeur libre.

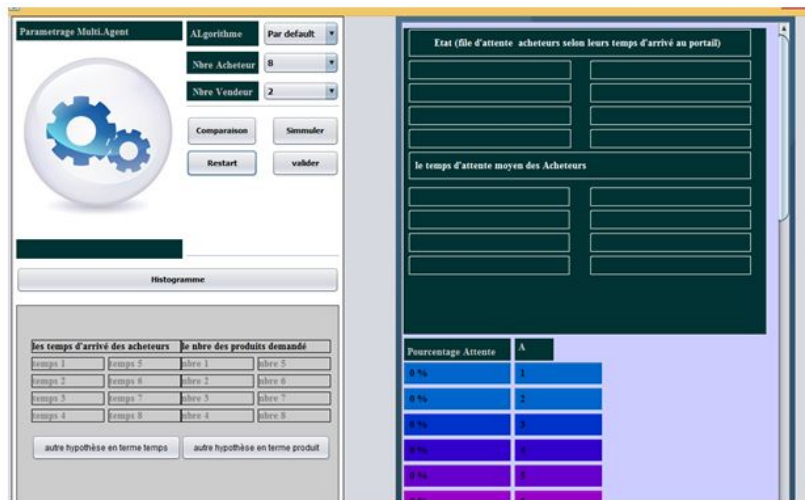


FIGURE III.8 – Présentation du Menu principal de l'application

Dans cette fenêtre nous pouvons constater toute une palette des boutons .en plus fenêtre de données .La première comporte le paramétrage aléatoire de notre problème (le temps d'arrivé de chaque acheteur et le nombre de produit compté demander par l'acheteur)

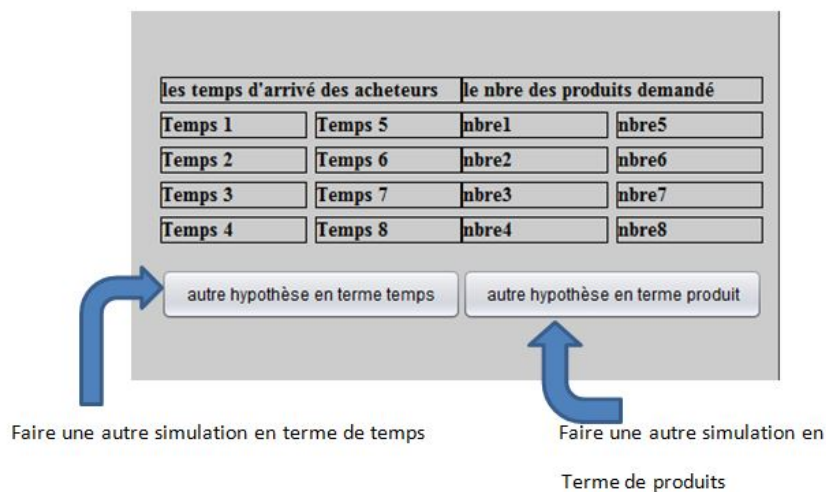


FIGURE III.9 – Remplissage des valeurs.

La deuxième affiche la file d'attente qui trie les acheteurs selon leur temps d'arrivé et leur nombre de produit ainsi le temps d'attente moyen de chaque acheteur et le temps moyen global.

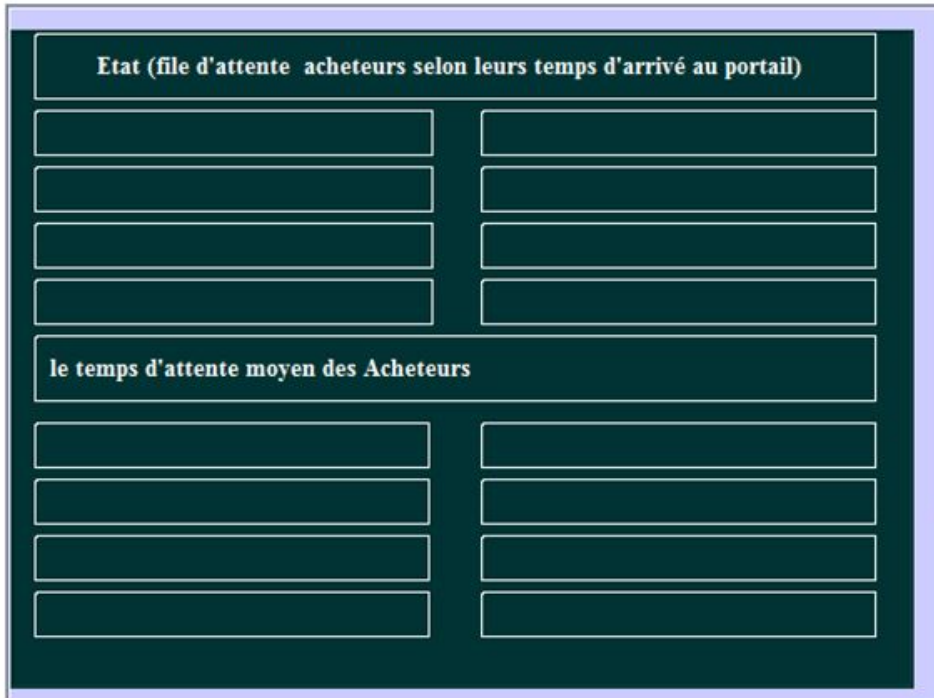


FIGURE III.10 – Affichage de la file d'attente et le temps d'attente moyen de chaque acheteur

4.7.3 Fenêtre d'histogramme :



FIGURE III.11 – Histogramme avant exécution

Voici l'histogramme qui affiche les résultats de temps d'attente de chaque acheteur qui lui convient avec le pourcentage. Ceci permettra à l'utilisateur

de juger quel algorithme est meilleur.

4.7.4 Fenêtre de paramétrage :

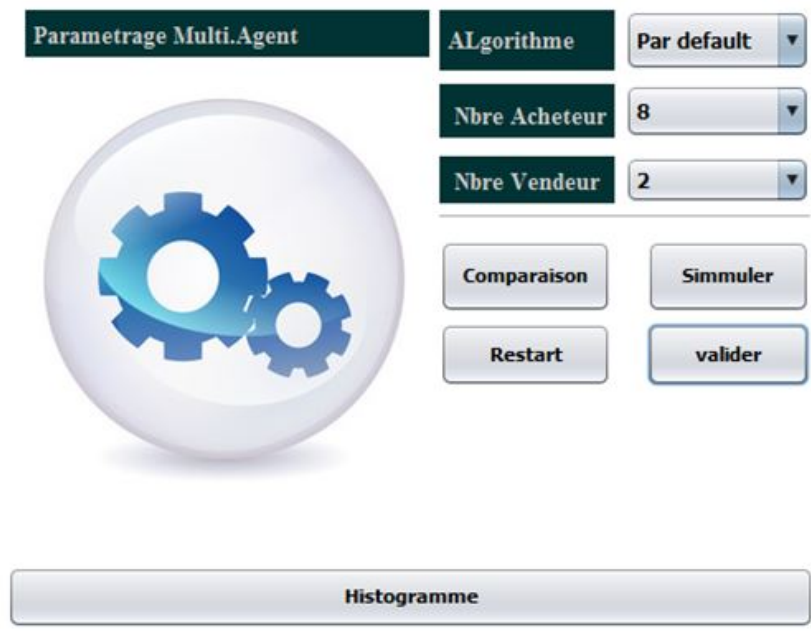


FIGURE III.12 – Présentation de la fenêtre paramétrage

Cette fenêtre de paramétrage avec laquelle l'utilisateur peut gérer le nombre d'acheteurs (de 1 à 8) et le nombre de vendeurs (1 ou bien 2) ainsi le choix de deux algorithmes (algorithme par défaut ou bien algorithme d'optimisation).

La fenêtre contient plusieurs boutons tels que :

- (a) **Le bouton comparaison ou Histogramme :** Ce bouton nous emmène a une fenêtre la ou il s'affiche un histogramme de comparaison entre les deux algorithmes utilisé au sein de cette application (l'algorithme par défaut et celui de l'optimisation) ; alors la l'utilisateur peut avoir un très bon perspective vis-à-vis le temps d'attente de chaque acheteur, il peut juger à partir de cette présentation graphique qui est le meilleure Algorithme.

A noter seulement qu'on a utilisé la classe Rectangle 2D pour tracer leurs bâtons (de l'histogramme) et la hauteur de ce rectangle s'accorde avec le temps d'attente de l'acheteur qui lui convient.

- (b) **Le bouton Simmuler :** Il permet d'affecter des valeur au temps d'arriver des acheteur ainsi le nombre de produits demandé

- (c) **Le bouton valider** : Il permet d'exécuter le programme et affiche le temps d'attente moyen de chaque acheteur ainsi le temps d'attente moyen global
- (d) **Le bouton Restart** : Il permet de refaire l'exécution avec des nouvelles valeurs

4.8 Résultats obtenus :

Les résultats obtenus s'affiche par un histogramme comparatif, un histogramme et le choix d'algorithme ainsi par les temps d'attentes moyen pour chaque acheteur et le temps moyen global.

4.8.1 Résultat comparatif :

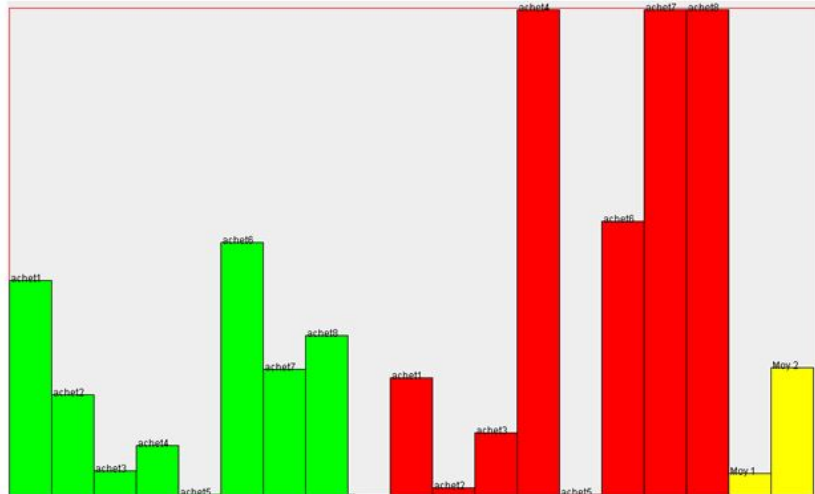


FIGURE III.13 – Résultat comparatif

On a dessiné l'histogramme en se basant sur une classe java le Rectangle 2D qui figure parmi les classes du package geom java.

Par la suite on a enregistré les temps d'attente de chaque acheteur dans un tableau (structure de donnée) et on a créé une instance de la classe Rectangle2D en lui spécifiant à chaque fois la hauteur (height) en fonction du temps d'attente de chaque acheteur.

Pour distinguer les bâtons qui conviennent à l'algorithme d'optimisation de ceux de l'autre Algorithme on a utilisé deux couleurs d'efférentes, ainsi on pourra juger facilement le résultat.

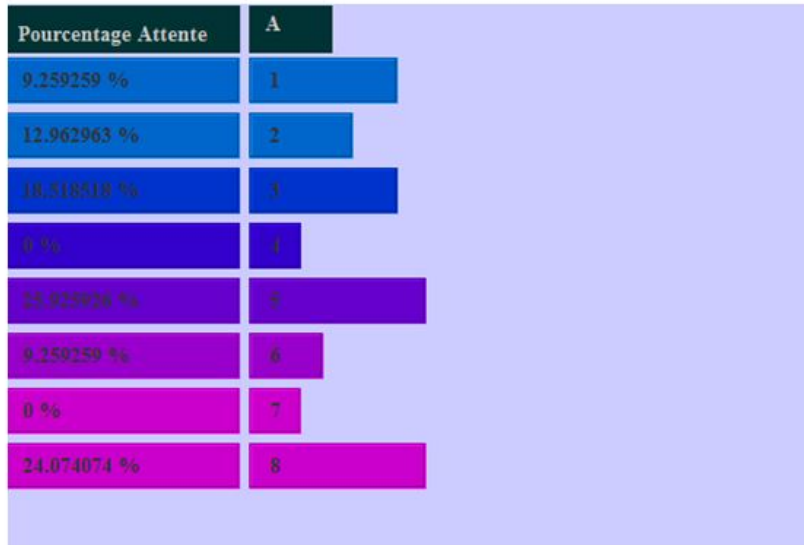


FIGURE III.14 – Résultat histogramme

4.8.2 Résultat histogramme :

Cette fenêtre affiche un histogramme pour les deux algorithmes utilisés au sein de cette application (l'algorithme par défaut et celui de l'optimisation). Donc l'utilisateur peut voir le temps d'attente de chaque acheteur selon l'algorithme choisi. À noter seulement qu'on a utilisé le pourcentage.

4.8.3 Résultat file d'attente :

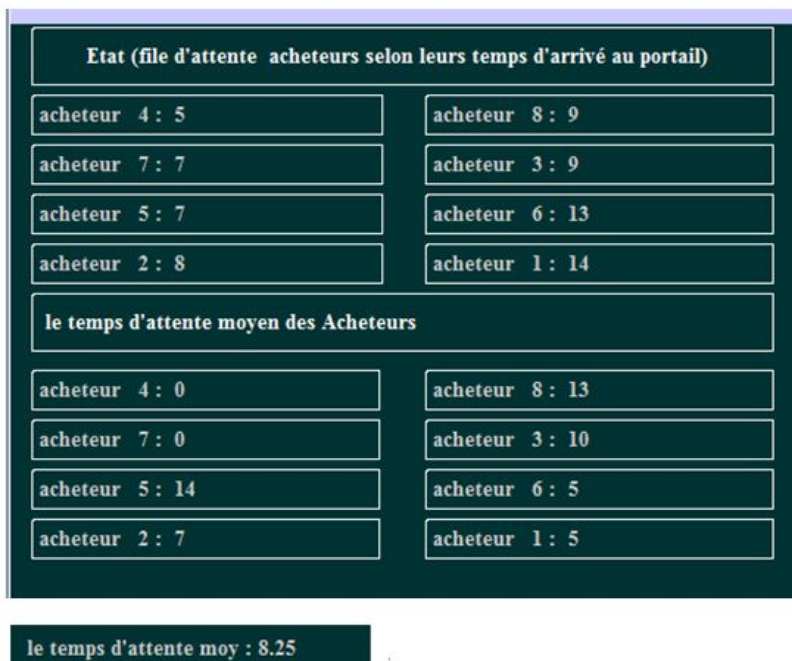


FIGURE III.15 – Résultat file d'attente

On a calculé le temps d'attente moyen de chaque acheteur et le temps d'attente moyen global. Donc l'utilisateur peut voir le résultat des temps d'attente moyen pour chaque acheteur et le temps d'attente moyen globale ainsi l'ordre des acheteurs dans la file d'attente et tout ça après avoir validé la simulation

5 Conclusion :

Dans ce chapitre, nous avons essayé de mettre en œuvre l'ensemble des idées qui caractérisent l'architecture proposée en se concentrant sur l'implémentation de sous la plateforme JADE. Nous avons présenté une simulation de communication entre les agents du système suivi d'une présentation des interfaces et des captures d'écrans de notre application. Les résultats obtenus en appliquant la politique d'optimisation proposée donne un meilleur résultat par rapport celui de la politique par défaut.

Conclusion général

Dans ce travail, nous avons abordé le problème d'allocation de ressources. Nous avons utilisé une approche multi-agents pour le partage équitable de ressources entre les membre d'une population . L'approche SMA offre plusieurs avantage qui sont mi en évidence dans l'application choisie (vendeur, Acheteur).

En adoptant le paradigme "agent", nous avons modélisé le processus d'allocation des ressources dans une supérette (client/vendeur). Le modèle comporte trois types d'agent : Agents acheteurs : ce sont ceux qui viennent pour satisfaire leurs achats. Agents vendeurs : ce sont ceux qui servent les clients en leur donnant les produits demandés. Agents portail ou coordinateur : composé de deux agents coordinateurs. Leur fonction consiste à assurer la coordination entre les acheteurs et les vendeurs afin d'organiser la vente. Pour préparer son offre d'allocation des ressources chaque agent acheteur consulte l'agent coordinateur pour vérifier l'état des vendeurs (libre/occupé) et chaque agent vendeur consulte l'agent coordinateur pour voir la liste de produit de chaque acheteur. Pour accomplir la tâche de coordination, l'agent coordinateur consulte les agents (acheteurs, vendeurs) pour coordonner entre eux par des messages.

Les résultats obtenus et comparer montre l'efficacité du modèle utilisé. En perspective, on pourrait intégrer d'autre algorithmes d'optimisation (des méta-heuristiques) à l'approche multi-agents pour améliorer encore les résultats obtenus.

Bibliographie

- Azaiez S. Approche dirigée par les modèles pour le développement de systèmes multi-agents. Master's thesis, Université de Savoie, 2007.
- Belaqziz S. *Une approche d'aide à la décision pour la gestion d'un système d'irrigation gravitaire : modélisation multi-agents, télédétection et optimisation par algorithme évolutionnaire*. PhD thesis, Université Cadi Ayyad - Marrakech - Maroc., 2014.
- Bender M Chakrabarti, S. Flow and stretch metrics for scheduling continuous job streams, soda '98 : Proceedings of the ninth annual acm-siam symposium on discrete algorithms. Master's thesis, 1998.
- Barceinas A. *AKQML Inter-Agent Communication Framework in a Digital Library*. 1998.
- Benhawala F. L'adoption d'une approche organisationnelle pour la conception et la réalisation d'un système multi agents d'acquisition coopérative d'information. Master's thesis, Université de la Manouba, 2008.
- Blangi P. Etat de l'art sur les plates formes et les langages multi-agents appliqués aux écosystèmes. Master's thesis, recherche :modélisation et simulation des systèmes complexes, 2005.
- Bourdon M Mouaddib A Stinckwich S Lebarbe T Enjalbert, P. Systèmes multi-agents. Master's thesis, université de C A E N, 2000.
- chami D. *Une plate forme orientée agent pour le data mining*. PhD thesis, Université Hadj Lakhdar – Batna, 2010.
- Chaib-Draa B. *Theory, Implementation and practical applications in multi-agents environment. IEEE Transcation On Knowledge and Data Engineering*. PhD thesis, 2001.
- Daily N. *Systèmes multi-agents dans les EIAH*. PhD thesis, 2007.

- Demazeau Y. *From Interactions to Collective Behavior in Agent-Based Systems*. European Conference on Cognitive Science, Saint-Malo, 1995.
- Daknou A. *Architecture distribuée à base d'agents pour optimiser la prise en charge des patients dans les services d'urgence en milieu hospitalier*. Ecole Centrale de Lille, 2011.
- Demazeau Y. *Populations and Organizations in Open MAS*. 1st National Symposium on Parallel and Distributed AI, Hyderabad, July, 1996.
- Ferber J. *Les systèmes Multi-Agents Vers une intelligence collective*. Inter Edition, Paris, 1995.
- Ferrié J Mossière, J. *Escope :gestion des processus et partage des ressources Volume 8 de cahier de l'RIA*. 1997.
- Hennet J. *Concepts et outils pour les systèmes de production*. Cépaduès Editions, Coordination, 1997.
- Jarras i Chaib-Draa, B. *Aperçu sur les systèmes Multi-Agents*. Série Scientifique du centre inter universitaire de recherche en analyse des organisations, CIRANO, 2002.
- Kocsis T. *Study on application possibilities of Case-Based Reasoning on the domain of scheduling problems*. Doctoral dissertation, Toulouse, 2011.
- Mazyad H. *Une approche Multi-agents à Architecture P2P pour l'apprentissage collaboratif*. PhD thesis, Université du Littoral Côte d'Opale, 2013.
- Medernach E. *Workload Analysis of a Cluster in a Grid Environment, in D. Fietelson, E. Frachtenberg, L. Rudolph et U. Schwiegelshohn (eds), Job Scheduling Strategies for Parallel Processing, Vol. 3834 of Lecture Notes in Computer Science*. Springer Berlin, Heidelberg, Berlin, Heidelberg, chapter 2, pp. 36–61., 2005.
- Medernach E. *Allocation de ressources et ordonnancement multi-utilisateurs : une approche basée sur l'équité. Recherche opérationnelle*. Université Blaise Pascal - Clermont- Ferrand II, tel-00657945v2, 2011.
- Phillips S Torng E Motwani, R. *Non-clairvoyant scheduling, Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms, SODA '93, Society for Industrial and Applied Mathematics*. , Philadelphia, pp. 422–431, 1993.

- Russel S Norvig, P. *Artificial Intelligence. A Modern Approach*, Prentice Hall., 1995.
- Sen A. *Artificial Intelligence. Collective Choice and Social Welfare*, North Holland, Amsterdam, 1970.
- Tan Z Gurd, J. *Market-based grid resource allocation using a stable continuous double auction*. Grid Computing, 8th IEEE/ACM International Conference on pp. 283–290, 2007.
- Vivek V Blessing E Srinivasan, R. *Resource Provisioning Methodologies : An Approach of Producer and Consumer Favorable In Cloud Environment*. International Conference on Modern Trends in Science, Engineering and Technology (ICMTSET), 2013.
- Weerdt M Clement, B. *Introduction to planning in multi-agent systems*. Multi-agent and Grid Systems 5, 2009.
- Young H. *Equity in Theory and Practice*. Russell Sage Foundation, 1994.
- Zgaya H. *Conception et optimisation distribuée d'un système d'information d'aide à la mobilité urbaine : Une approche multi-agent pour la recherche et la composition des services liés au transport*. Modélisation et simulation, 2007.