

*République Algérienne Démocratique Et Populaire*  
*Ministère De L'enseignement Supérieur Et De La Recherche Scientifique*  
*Université De Tlemcen*  
*Faculté De Technologie*  
*Département De Génie Electrique Et Electronique*

## **Thèse de Doctorat en Productique**

Présentée pour l'obtention du **grade** de **DOCTORAT 3<sup>ème</sup> Cycle**

**En :** Productique

**Spécialité :** Productique

**Par :** Kouloughli Imèn

### **Sujet**

**Investigation sur les approches de pilotage des AS/RS à convoyeur gravitationnel  
basée sur les Systèmes Multi-Agents(SMA)**

Soutenue publiquement, le / / , devant le jury composé de :

Mr Med El Amine Abderrahim	MCA	Univ. Tlemcen	Président
Mr Zaki Sari	Professeur	Univ. Tlemcen	Directeur de thèse
Melle Latefa Ghomri	MCA	Univ Tlemcen	Co-Directeur de thèse
Mr Khaled Hachemi	MCA	Univ. Oran	Examineur 1
Mme Noria Taghezout	MCA	Univ. Oran	Examineur 2
Mr Pierre Castagna	Professeur	Univ Nantes	Invité

*A mes parents qui se sont toujours sacrifiés pour moi et ont toujours œuvré pour ma réussite*

## Remerciements

Je tiens en tout premier lieu à remercier Zaki Sari, professeur à l'Université de Tlemcen et directeur de cette thèse, pour le temps qu'il a consacré à ces travaux. C'est grâce à ses contacts et à sa disponibilité immédiate et permanente que les travaux de cette thèse ont pu parvenir à ces résultats.

Je remercie aussi vivement le professeur Pierre Castagna pour toutes les fois où il m'a accueilli et consacré de son temps pour la réalisation d'une bonne partie de cette thèse (toute la partie simulation sous netlogo), en stimulant une motivation de recherche et d'approche scientifique.

Je remercie également Melle Ghomri Latefa, maître de conférences à l'Université de Tlemcen et co-encadreur de cette thèse, pour son aide et son soutien.

Je remercie tous les membres du jury d'avoir accepté de porter leur intérêt sur ces travaux et d'être venus, parfois de loin, assister à la soutenance.

Je souhaite adresser mes remerciements aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de cette thèse et à l'aboutissement de ce travail au sein de l'équipe productive.

Je n'oublie pas ceux qui m'ont soutenu durant toutes mes études : mes parents.

Je remercie également mes sœurs, mes collègues et amis pour leur soutien.

## Résumé

Les entreprises évoluent dans un environnement de plus en plus compétitif et exigeant en termes de flexibilité et de réactivité. L'utilisation de modèles issus de l'intelligence artificielle distribuée (IAD) et plus particulièrement des systèmes multi agents (SMA) dans les outils de gestion des entreprises s'avère être efficace pour simuler et reproduire les comportements collaboratifs et adaptatifs tels qu'ils apparaissent actuellement dans les entreprises.

Les AS/RS suscitent actuellement un intérêt de plus en plus grandissant, en raison des avantages importants qu'ils présentent ; à savoir : un faible coût au niveau de la main-d'œuvre, des dépenses revues à la baisse pour le stockage, une meilleure exploitation de l'espace de stockage, une meilleure traçabilité des produits stockés, un débit optimisé du système et plus de sécurité. Ils ont donc une importance capitale dans la réussite des défis de la nouvelle industrie. Malheureusement, mis à part certains types d'AS/RS (par exemple à charge unitaire), ils ne sont pas beaucoup étudiés, bien qu'ils soient très utilisés dans l'industrie et les services, Ce manque d'intérêt pour les AS/RS particuliers tel que l'AS/RS à convoyeur gravitationnel, fait qu'ils ne sont utilisés que pour des applications particulières et très en deçà de leurs possibilités. [Sari 2003]

Dans le cadre de cette thèse ; nous nous sommes focalisés sur l'apport des systèmes multi agents dans les systèmes automatisés de stockage/déstockage (AS/RS).

Dans une première contribution nous nous sommes intéressés aux AS/RS à convoyeur gravitationnel ; plus particulièrement au stockage et déstockage des produits dans un rack. La technique employée est celle des Systèmes Multi-Agents (SMA) réactifs, ou les agents se basent sur des combinaisons de règles de stockage et de déstockage .Les résultats obtenus après plusieurs simulations ont menés à établir une deuxième contribution basée sur des agents intelligents pouvant communiquer, dans le but de minimiser en permanence le temps de déstockage global.

# Sommaire

Introduction générale .....	1
Chapitre 1 Les systèmes Multi-Agents (SMA).....	4
1. Introduction .....	4
2. La notion d'agent .....	4
2.1 Les caractéristiques d'un agent .....	5
2.2 Typologies d'agents .....	6
3. Système multi-agents .....	8
3.1 Définitions.....	8
3.2 Caractéristiques d'un SMA.....	9
3.3 Les communications dans les SMA .....	10
3.4 Types d'interactions.....	11
3.5 Les langages de communication .....	14
3.6 Communication des agents dans un environnement sans fil.....	20
3.7 Les plateformes SMA .....	23
3.8 Domaines d'application des SMA .....	25
4. Conclusion.....	29
Chapitre 2 Systèmes Automatisés de Stockage /Déstockage (AS/RS).....	30
1. Introduction .....	30
2. Définition d'un AS/RS.....	31
3. Fonctions.....	31
4. Composants .....	31
5. Types d'AS/RS .....	33
5.1 AS/RS à charge unitaire .....	34
5.2 AS/RS à mini charge ou a charge réduite.....	35
5.3 AS/RS multi allées .....	35
5.4 AS/RS à racks glissants .....	35
5.5 AS/RS à convoyeur gravitationnel .....	36
5.6 AS/RS à personne embarquée .....	36
5.7 AS/RS à carrousel.....	36
5.8 AS/RS à étagères profondes .....	37
6. Emplacement des produits.....	37
7. Méthodologies de stockage ou de déstockage .....	39
7.1 Stockage aléatoire ouvert.....	39

7.2	Stockage aléatoire par classe.....	40
7.3	Stockage dédié.....	41
8.	Critères de performances d'un AS/RS .....	42
9.	Etat de l'art .....	42
10.	Conclusion.....	50
Chapitre 3 Première contribution : Optimisation du temps de déstockage dans un AS/RS à convoyeur gravitationnel se basant sur un modèle Heuristique et Agent.....		51
1.	Introduction .....	51
2.	La simulation Multi-agents .....	51
3.	Le simulateur Netlogo.....	54
3.1	Définition .....	54
3.2	Fonctionnalités .....	55
4.	Description du modèle SMASD.....	57
4.1	Les principales actions dans SMASD .....	60
4.2	Les règles de stockage .....	61
4.3	Les règles de déstockage .....	62
4.4	Temps de déplacement de la machine de déstockage.....	63
4.5	Résultats obtenus et interprétations.....	64
4.6	Courbes modélisant le temps de déplacement de la machine de déstockage .....	66
4.7	Interprétation des résultats :.....	70
5.	Conclusion.....	77
Chapitre 4 Deuxième Contribution Application avec Jade .....		78
1.	Introduction .....	78
2.	Les outils de développement.....	78
2.1	Le langage de programmation java .....	78
2.2	L'environnement de développement NetBeans.....	79
2.3	La plateforme JADE (Java Agent Development Framework).....	79
3.	Architecture de notre système .....	81
4.	Le système multi agents .....	82
4.1	L'Agent produit .....	83
4.2	L'agent Casier.....	83
5.	UML : Unified Modeling Language .....	83
5.1	Diagrammes de séquences .....	83
5.2	Diagrammes de classes.....	84
6.	ArgoUML.....	84

7. Diagrammes UML de notre système .....	84
8. Les classes de notre programme java .....	87
9. Résultats .....	94
10. Conclusion.....	97
Conclusion générale.....	98
Références .....	101

## Liste des Figures

Figure 1 Un agent avec son environnement.....	6
Figure 2 Représentation schématique d'un SMA [Fabien 2004].....	9
Figure 3 Réseau sans fil.....	20
Figure 4 Types de réseaux sans fil.....	21
Figure 5 Bluetooth.....	22
Figure 6 Communications sans fil dans un SMA.....	23
Figure 7 Représentation des classes ABC.....	40
Figure 8 Interface utilisateur avec le modèle Wolf-Sheep Predation.....	57
Figure 9 L'AS/RS à convoyeur gravitationnel dans le model SMASD.....	58
Figure 10 Principe de fonctionnement du modèle SMAS.....	59
Figure 11 Représentation du SMA.....	59
Figure 12 Interface principale du modèle SMASD.....	61
Figure 13 Principe de fonctionnement des règles de déstockage R3 et R5.....	63
Figure 14 Etat initial du rack.....	65
Figure 15 Temps de déstockage résultant d'un stockage aléatoire combiné aux 5 règles de l'heuristique.....	66
Figure 16 Temps de déstockage résultant d'un stockage par RE2 combiné aux 5 règles de l'heuristique.....	67
Figure 17 Temps de déstockage résultant d'un stockage par RE3 combiné aux 5 règles de l'heuristique.....	69
Figure 18 Architecture de jade.....	79
Figure 19 Comportements des agents dans Jade.....	81
Figure 20 Architecture générale de notre application.....	81
Figure 21 Interface Java du SMA.....	82
Figure 22 Envoi d'une seule requête de déstockage.....	84
Figure 23 Envoi de plusieurs requêtes de déstockage.....	85
Figure 24 Digramme de classes : Produit.....	86
Figure 25 Digramme de classes : Casier.....	86
Figure 26 Enregistrement de services auprès de l'agent DF.....	90
Figure 27 Envoi de message sous Jade.....	90
Figure 28 Messages de type CFP (Call For Proposal).....	91
Figure 29 Messages de type propose.....	92
Figure 30 Choix du meilleur casier et confirmation de la transaction.....	93
Figure 31 Demandes simultanées de déstockage de produits.....	94
Figure 32 Communications intelligentes entre les agents pour l'optimisation du rangement d'un rack.....	95
Figure 33 Résultat de l'application.....	96



## Liste des Tableaux

Tableau 1 Comparaison entre les différents types d'agents.....	7
Tableau 2 Composition d'un message ACL .....	17
Tableau 3 Plateformes de simulation Multi-agents .....	54
Tableau 4 Le temps moyen de déstockage obtenu en combinant deux règles.....	65

# Introduction générale

Actuellement, les entreprises sont dans l'obligation de mettre en œuvre toutes leurs compétences et leurs efforts afin de rester compétitives et en concurrence. Pour cela, il faut s'adapter plus rapidement, garantir leur réactivité, augmenter la productivité, minimiser les stocks et le coût et gagner du temps (délais de livraison).

Afin d'atteindre ces objectifs les entreprises doivent maîtriser la complexité et la souplesse de leurs systèmes de production. Plusieurs approches sont apparues pour la modélisation et l'étude des systèmes de production, L'approche SMA (Systemes Multi-Agents) est l'une de ces approches. Elle est la plus adaptée à la nature des systèmes de production.

Grace aux SMA la conception de systèmes intelligents est possible, elle est caractérisés principalement par :

L'intelligence des agents : les agents du SMA sont autonomes, dotés de capacités cognitives (comme la décision, le raisonnement et l'apprentissage) et d'un comportement réflexif leur permettant d'agir, d'observer leurs actions et d'en changer le cours.

L'interactivité des agents : les agents sont situés dans un environnement partagé et ouverts vers des utilisateurs ou d'autres systèmes.

La complexité du système : en conséquence des deux propriétés précédentes, les agents sont hétérogènes, multiples, voire très nombreux.

Les systèmes de stockage les plus élaborés sont les systèmes automatiques de Stockage/Déstockage (Automated Storage Retrieval Systems AS/RS). Ces systèmes, de plus en plus utilisés, sont constitués de magasins de stockage (racks), de machines S/R de stockage/déstockage et d'un convoyeur reliant les machines S/R et les points d'entrée/sortie de l'aire de stockage [Keserla, 94].

Les systèmes de stockage automatiques sont efficaces, rapides flexibles et fiables. Ils assurent un meilleur stockage et une meilleure distribution des marchandises ainsi que des délais de livraison particulièrement courts. De plus, ils permettent un flux de matériaux optimal et sans rupture. Avec les systèmes de stockage automatiques, les processus de stockage, de déstockage et de réaffectation sont réalisés automatiquement. De cette manière,

les employés n'ont plus à effectuer de tâches monotones et pénibles physiquement, et le nombre d'erreurs est réduit à son minimum. Ces facteurs peuvent s'avérer être un avantage concurrentiel décisif pour les entreprises de l'industrie et du commerce.

L'AS/RS à convoyeur gravitationnel est composé d'un rack profond regroupant un ensemble de casiers pouvant contenir plusieurs produits placés les uns à la suite des autres. Chaque casier est muni d'un convoyeur gravitationnel à base de rouleaux ou roues libres inclinées de façon à permettre le glissement des produits d'un bout à l'autre du casier, donc d'une face à l'autre du rack. Les produits sont stockés d'un côté du rack (face de stockage) par la machine de stockage (MS), ils glissent sur les convoyeurs gravitationnels, et ils sont déstockés de l'autre côté du rack (face de déstockage) par la machine de déstockage (MD). Les machines de stockage et de déstockage peuvent se déplacer simultanément sur deux axes. Ces deux axes forment le plan x-y parallèle aux deux faces du rack. Une station de dépôt est située sur la face de stockage, où la machine de stockage récupère les produits à stocker. Une station de livraison est située sur la face de déstockage, où la machine de déstockage dépose les produits pour la livraison. Un convoyeur de re-stockage incliné en sens inverse relie la machine de déstockage à la machine de stockage. Il permet de transférer les produits à restocker vers la station de dépôt.

Cette thèse est organisée en quatre chapitres en plus d'une introduction générale et d'une conclusion générale.

Dans le premier chapitre nous présentons une description générale des systèmes multi-agents. Nous définirons d'abord les différentes notions et concepts gravitant autour de cette thématique. Ensuite nous présenterons les différentes manières de communications qui puissent exister entre les agents pour passer ensuite aux différents langages de communications. Nous nous sommes intéressés dans les sections qui suivent aux différentes plateformes multi-agents tout en définissant chacune d'entre elles, aux agents communiquant dans un environnement sans fil et pour finir nous établirons les domaines d'application des systèmes multi-agents utilisés jusqu'ici.

Le second chapitre est composé de deux parties, la première partie est réservée à une présentation des systèmes automatisés de stockage/déstockage. Ainsi les définitions, fonctions, compositions sont développées. Nous avons présenté une section exposant les différents types d'AS/RS suivie d'une section décrivant les emplacements de ces systèmes, ensuite une section pour les méthodologies de stockage et enfin nous avons cité les principaux

critères de mesures de performances des AS/RS. Dans la seconde partie, un état de l'art détaillé regroupant les travaux les plus importants dans le domaine est présenté.

Le troisième et quatrième chapitre englobe nos deux contributions, une première proposant une solution robuste, et efficace pour optimiser le stockage/déstockage de produits dans un rack. Ceci dans le but de minimiser le temps de déstockage d'une liste de produits de plusieurs types.

La solution consiste à considérer le système étudié comme étant un SMA, dans un premier temps les agents sont réactifs ; nous stockons et déstockons une liste de produits en appliquant une combinaison bien choisie de règles de stockage et de déstockage (heuristique) ; ceci dans le but de sélectionner la meilleure combinaison.

Notre travail consiste alors à associer les concepts de systèmes Multi-agent et de l'optimisation heuristique, et en proposant un modèle d'optimisation heuristique distribuée baptisé SMASD (Système Multi-Agents de Stockage et Déstockage) une solution applicable et facilement modifiable à tout type de systèmes de production, ceci grâce à l'utilisation de variables globales facilement modifiables.

En établissant notre programme sous Netlogo nous avons ressenti au fur et à mesure la nécessité d'une plateforme mettant en évidence la notion de « Communications entre les agents » ; ce qui a mené à une seconde contribution représentant une application sous la plateforme multi-agents « Jade : Java Agent DEvelopment Framework » ,qui utilise le langage ACL (Agent Communication Language) avec lequel les agents s'envoient des messages dans le but de choisir le meilleur casier d'où le produit en question devra être déstocké c.-à-d. dans les plus brefs délais .

Enfin, dans la partie conclusion, les contributions de cette thèse sont résumées. Plusieurs problèmes et directions de recherche possibles sont également examinés.

# Chapitre 1 Les systèmes Multi-Agents (SMA)

## 1. Introduction

Depuis une dizaine d'années, les systèmes multi-agent ont connu un grand essor et sont appliqués à des domaines très variés comme, par exemple, le domaine de la simulation et de la vie artificielle, la robotique, le traitement d'images, etc.

Cet essor manifeste des SMA émane de la critique des approches séquentielles ou fonctionnelles et de l'intelligence artificielle classique. En résumé, il est intéressant d'observer la place qu'occupent les systèmes multi-agent dans l'évolution des paradigmes de l'informatique selon [Jam, 2005].

La conception, l'implémentation et la validation des systèmes multi-agents soulèvent de nombreuses questions de recherche qui leur sont spécifiques. Elles incluent la mise en œuvre de stratégies de coordination qui permettent effectivement à des groupes d'agents de résoudre des problèmes; de mécanismes de négociation qui permettent d'amener un ensemble d'agents dans un état global acceptable ; de techniques pour détecter les conflits et les résoudre; de protocoles par l'intermédiaire desquels les agents peuvent communiquer et raisonner sur leurs communications ; et enfin de mécanismes par lesquels les agents peuvent maintenir leur autonomie tout en contribuant au fonctionnement général du système.[Taghezout 2012]

Dans la suite, nous présentons les notions d'agents et de systèmes multi-agent, nous aborderons les différentes questions que soulève l'approche SMA telles que : les interactions et la coopération, la coordination, la négociation et la communication.

Dans la section qui suit nous nous intéressons aux langages de communication dans les SMA, nous présentons juste après les différentes plateformes multi-agent qui existent.

Enfin nous établirons une petite partie état de l'art concernant l'utilisation des SMA dans le domaine industriel.

## 2. La notion d'agent

Un agent peut être défini comme une entité capable d'agir sur elle-même et son environnement. Disposant d'une représentation partielle de cet environnement, pouvant communiquer avec d'autres agents et dont le comportement est la conséquence de ses

observations, de sa connaissance et des interactions avec les autres agents. Un agent est une "entité computationnelle", comme un programme informatique ou un robot, qui peut être vu comme percevant et agissant de façon autonome sur son environnement. Nous parlons d'autonomie parce que son comportement dépend au moins partiellement de son expérience.

En d'autres termes l'**agent** est :

Une entité (physique ou abstraite) caractérisée par :

- Son autonomie dans la prise de décision,
- Ses connaissances sur lui même et sur les autres,
- Sa capacité d'agir.

Un agent ne peut exister sans environnement. L'environnement est une structure dans laquelle l'agent évolue. Un agent va agir sur son environnement et vice versa.

## 2.1 Les caractéristiques d'un agent

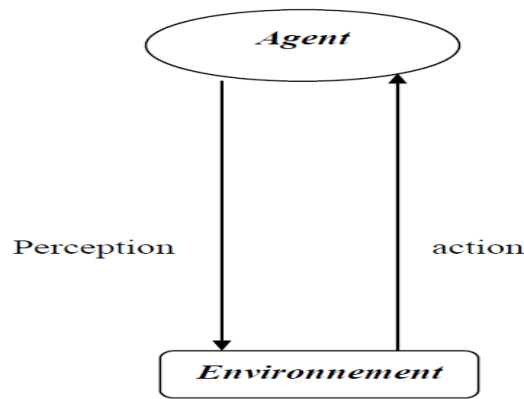
- **Autonomie** : Selon [Russel et Norvig, 2003], « l'agent agit sur son environnement sans l'intervention de l'être humain ou des autres agents », nous pouvons le définir comme une existence indépendante dont il contrôle les actions et ses états internes.

- **Capacités sociales** : l'agent interagit et communique avec d'autres agents dans le but de l'échange de l'information pour un bon fonctionnement interne, ou bien dans le but de la négociation, la coopération ou la coordination.

- **Réactivité** : l'agent maintient un lien constant avec l'environnement dont il appartient et répond aux changements qui y surviennent.(Figure 1)

- **Pro-activité** : l'agent a la capacité de fixer ses propres buts et objectifs par ses propres initiatives.

- **La continuité** : Selon [Florez, 1999], un agent a la capacité de la persistance d'une identité et d'un état sur une longue période et de la différence des objets logiciels, cela permet à l'agent d'être en performance active sans avoir à être provoqué par un stimulus interne.



**Figure 1 Un agent avec son environnement**

## 2.2 Typologies d'agents

Trois types d'agent peuvent être distingués selon l'architecture interne (Tableau 1)

### **Agent réactif :**

Comme son nom l'indique, un agent réactif ne fait que réagir aux changements qui surviennent dans l'environnement. Autrement dit, un tel agent se contente simplement d'acquiescer des perceptions et de réagir à celles-ci en appliquant certaines règles prédéfinies.

Étant donné qu'il n'y a pratiquement pas de raisonnement, ces agents peuvent agir et réagir très rapidement.

### **Agent cognitif :**

Il est intelligent par lui-même c'est-à-dire qu'il effectue un certain raisonnement pour choisir ses actions. Un tel raisonnement peut se faire soit en se basant sur les buts de l'agent, soit sur une certaine fonction d'utilité.

### **Agent hybride :**

Chaque agent hybride est caractérisé par la notion de couches et chaque couche représente soit les agents cognitifs, soit les agents réactifs. Donc l'agent hybride combine entre les deux comportements (comportement réactif et comportement cognitif).

Agents réactifs	Agents cognitifs	Agents hybrides
<p>-Ont des capacités réactives : réagir directement à l'environnement perçu, par pulsion.</p> <p>- Ne possèdent pas de moyen de mémorisation.</p> <p>-N'ont pas de représentation explicite de leur environnement.</p> <p>-Fonctionnent selon un modèle stimuli/ réponse.</p> <p>-Dès qu'ils perçoivent une modification de leur environnement, ils répondent par une action programmée.</p> <p>-L'exemple le plus célèbre est celui de la fourmilière étudié par Alexis Drogoul [Drogoul ,2000].</p> <p>-Les sources des travaux sur ce type d'agents sont les sciences de la nature et de la vie.</p>	<p>-Possèdent des capacités cognitives : anticiper, prévoir le futur, mémoriser des choses... ils réfléchissent.</p> <p>-Proviennent d'une métaphore du modèle humain.</p> <p>-Possèdent une représentation explicite de leur environnement, des autres agents et d'eux-mêmes.</p> <p>-Sont aussi dotés de capacités de raisonnement et de planification ainsi que de communication.</p> <p>-Sont structurés en société où il règne donc une véritable organisation sociale.</p> <p>-Le travail le plus représentatif de cette famille d'agents porte sur le modèle Believe Desir Intention BDI [Rao et al. 1995].</p> <p>-Les sources de ces travaux sont les sciences humaines et sociales.</p>	<p>-Ont des capacités cognitives et réactives.</p> <p>-Conjuguent en effet la rapidité de réponse des agents réactifs ainsi que les capacités de raisonnement des agents cognitifs.</p> <p>-Le modèle est un compromis autonomie/coopération et efficacité/complexité.</p> <p>-L'exemple de l'architecture ASIC [Boissier et al. 1997] utilisée pour le traitement numérique d'images.</p> <p>-L'architecture ASTRO [Demazeau et al. 1998] développée pour être utilisée dans les systèmes multi-agents soumis à des contraintes de type temporelles.</p>

**Tableau 1 Comparaison entre les différents types d'agents**



### 3. Système multi-agents

#### 3.1 Définitions

Un système multi-agent (SMA) est un système distribué :

- Composé d'un ensemble d'agents distribués,
- Ces agents sont situés dans un certain environnement et interagissent selon entre eux.

Un SMA permet de résoudre des problèmes complexes en exploitant l'intelligence collective des agents qui le compose.

Ferber [Ferber, 1995] définit un système multi-agent de la manière suivante :

" Un système multi-agent est un système composé des éléments suivants :

- Un environnement est un espace disposant généralement d'une métrique.
- Un ensemble d'objets situés dans l'espace, ils sont passifs, ils peuvent être perçus, détruits, créés et modifiés par les agents.
- Un ensemble d'agents qui sont les entités actives du système.
- Un ensemble de relations qui unissent les objets entre eux.
- Un ensemble d'opérations permettant aux agents de percevoir, de détruire, de créer, de transformer et de manipuler les objets.
- Un ensemble d'opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification (les lois de l'univers)."

Un système multi-agent est un ensemble homogène ou hétérogène d'agents situés dans un même environnement et qui interagissent. De ces interactions peuvent émerger différents phénomènes que l'on peut classer en trois catégories [Marcenac, 1996]: l'émergence structurelle, l'émergence de comportements et l'émergence de propriétés. L'émergence traite donc de l'apparition soudaine non programmée et irréversible de phénomènes dans un système (éventuellement multi-agent) confirmant ainsi la maxime grecque " le tout est plus que la somme de chaque partie ".

Selon la vision de Demazeau [Demazeau, 1995] :

SMA = Agent + Environnement + Interaction + Organisation.

Cette décomposition d'un SMA est dite décomposition Voyelles qui identifie un axe Agent, un axe Environnement, des Interactions et une structure Organisationnelle explicite ou non. Un SMA est généralement caractérisé par :

- chaque agent a des informations ou des capacités de résolution de problèmes limitées, ainsi il a un point de vue partiel.
- il n'y a aucun contrôle global du système multi-agent.
- les données sont décentralisées.
- le calcul est asynchrone.

Les types courants d'interaction dans un SMA incluent la coopération (travailler ensemble à la résolution d'un but commun) ; la coordination (organiser la résolution d'un problème de telle sorte que les interactions nuisibles soient évitées ou que les interactions bénéfiques soient exploitées) et la négociation (parvenir à un accord acceptable pour toutes les parties concernées).

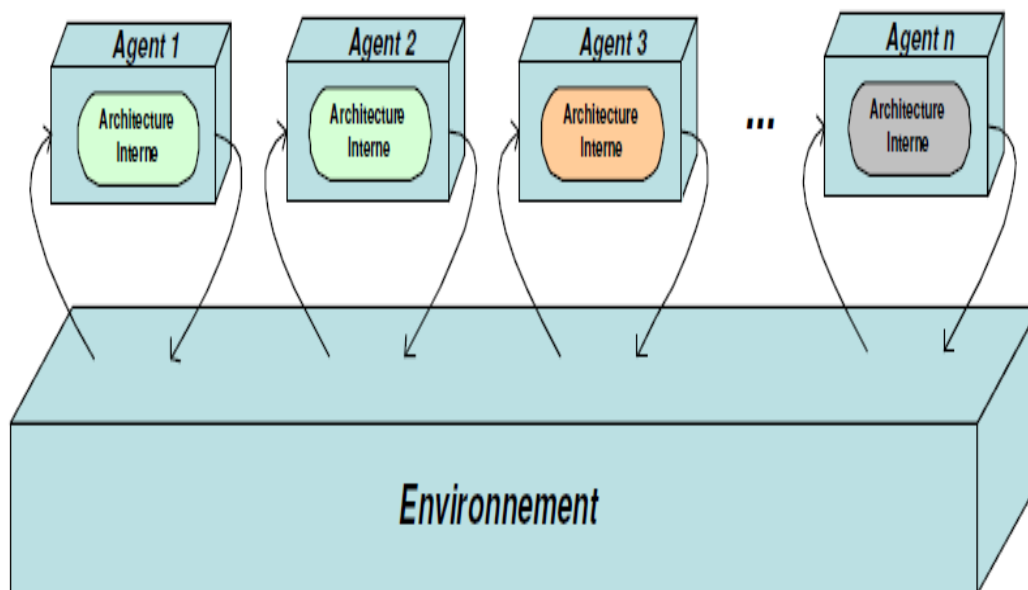


Figure 2 Représentation schématique d'un SMA [Fabien 2004]

### 3.2 Caractéristiques d'un SMA

Les principales caractéristiques des SMA sont :

- Chaque agent ne dispose que d'informations incomplètes et a un champ d'action limité

- Le contrôle du système est réparti
- Les données manipulées sont décentralisées
- Les traitements sont asynchrones

De plus un SMA peut être :

- Ouvert : les agents y rentrent et sortent librement
- Fermé : l'ensemble d'agents reste le même
- Homogène : tous les agents sont construits sur le même modèle
- Hétérogène : les agents sont construits à partir de différents modèles

### **3.3 Les communications dans les SMA**

La notion d'interaction est au centre de la problématique des SMA. De ce fait, les systèmes d'interactions constituent l'objet d'étude essentiel des SMA et servent à formuler le rapport qui existe entre l'agent d'une part et l'organisation dans sa globalité d'autre part.

Une interaction est une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Les interactions s'expriment ainsi à partir d'une série d'actions dont les conséquences exercent en retour une influence sur le comportement futur des agents. Les agents interagissent le long d'une suite d'événements pendant lesquels ils sont d'une certaine manière en contact les uns avec les autres, que ce contact soit direct ou qu'il s'effectue par l'intermédiaire d'un autre agent ou de l'environnement.

Un agent peut agir sur le monde qui l'environne, c'est à dire sur les autres agents présents dans son univers et sur l'environnement lui-même (les objets passifs composant le monde dans lequel évolue l'agent). Cette intervention peut prendre la forme d'une modification de l'état des autres agents qu'il côtoie, que ce soit au niveau de leurs connaissances (ex : envoi d'information) ou au niveau de leur activité (ex : demande d'engagement).

L'interaction est à la base de la constitution d'organisations et simultanément elle suppose la définition d'un espace et généralement d'une organisation préétablie dans lesquels les interactions peuvent se produire. Cette notion est à la fois source et produit de la permanence de toute organisation. Enfin, l'interaction est la conséquence de l'aspect pluriel des sociétés multi-agents, en particulier une dimension supplémentaire à l'individu.

Ce dernier, n'est plus le centre de l'univers, mais il constitue un nœud d'un ensemble d'échanges et d'interdépendances qui le façonnent et lui donnent tout son sens : un agent sans interaction avec d'autres agents n'est plus qu'un corps isolé, qu'un système de traitement d'informations, dépourvu de caractéristiques adaptatives [Ferber 1995].

### **3.4 Types d'interactions**

Les interactions entre agents s'expriment sous diverses formes. Mais au-delà de la diversité superficielle, il est possible de dégager des différences et des points communs à un niveau plus profond. La notion de situation d'interaction permet de décrire les types d'interaction en les reliant aux éléments qui la composent. Elle sert à définir des catégories abstraites indépendantes de leurs réalisations concrètes.

« On appelle situation d'interaction un ensemble de comportements résultant du regroupement qui doivent agir pour satisfaire leurs objectifs en tenant compte des contraintes provenant des ressources plus ou moins limitées dont ils disposent et leurs compétences individuelles » [Ferber 1995].

La notion de situation d'interaction permet la classification des différents types d'interaction à travers trois paramètres caractérisant n'importe quelle situation. Ces trois composantes principales de l'interaction à savoir la nature des buts, l'accès aux ressources et les compétences des agents vont permettre de faire une typologie des situations.

#### **3.4.1 L'interaction par la communication**

La communication est la forme d'interaction la plus élémentaire dans un SMA, elle suppose l'existence de trois éléments [Erwan 2001]:

Médium physique : par exemple le réseau informatique, il est implémenté selon des choix techniques (architecture du réseau, type de connections, protocoles de communication)

Médium linguistique : ou langage d'expression, il nécessite la spécification d'un langage commun à un ensemble d'agents leur permettant d'interpréter un message.

Les recherches en communication inter-agents ont mis en valeur l'apport de la théorie des actes de langages (speech act theory [Searle 1972]). Parmi les langages les plus connus reposant sur cette théorie, nous notons KQML (Knowledge Query Manipulation Language), développé dans le cadre du projet DARPA [Fritzson 1994], largement utilisé dans les SMA, et aussi le langage ACL (Agent Communication Language), proposé par la FIPA (Foundation for Intelligent Physical Agent [FIPA 2002]). Brièvement, chaque acte de discours (i.e un envoi de

message dans un SMA) est considéré comme une action convoyant des informations sur l'intention de son émetteur et le résultat attendu.

**Mode de diffusion** : Il existe deux types de diffusion :

Modes anonymes : peuvent se faire au travers d'un tableau noir (blackboard System), ou par des envois multiples (broadcasting). Dans le premier cas, les agents envoient leurs informations ou requêtes à une zone de connaissance ou mémoire partagée que chaque agent consulte [Durfee 1995]. Les agents lisent ainsi tous les messages inscrits dans le tableau noir mais ne réagissent qu'à ceux qui les concernent, par contre dans le deuxième cas, il s'appuie sur l'envoi d'un message à tous les autres agents sans destinataire spécifique.

Modes nominatifs (Mode par message) : Dans ce type, l'agent envoie un message à chaque agent identifié dans sa base d'acointances, autrement dit, les agents qu'il connaît. Aussi est-il nécessaire de prévoir un mécanisme permettant à chaque agent de savoir comment contacter un agent spécifique i.e. de connaître son adresse.

### **3.4.2 L'interaction par la coordination d'activités**

Dans les environnements où les ressources sont limitées et partagées, chaque agent cherche à satisfaire ses besoins en les utilisant. La coordination peut être vue comme le comportement individuel de chaque agent visant à satisfaire ses besoins en premier lieu et le but global du système en deuxième lieu. Les caractéristiques reflétant les besoins de la coordination dans un SMA [Erwan 2001] se résument en trois points:

- Dépendance entre les actions d'agents ;
- La volonté d'atteindre des objectifs globaux ;
- Aucun des agents ne possède individuellement toutes les compétences, les ressources ou les informations nécessaires pour résoudre un problème donné [Jennings 1993].

### **3.4.3 L'interaction par la coopération d'agents**

Le terme coopération est souvent utilisé dans des sens très différents. Selon Malone et Crowston [Malone 1994], la coopération est une forme de coordination. Par contre, Ferber [Ferber 1999] considère que la coordination est une méthode de coopération.

Il existe une définition intermédiaire entre les deux, proposée par Durfee et al. [Durfee 1989] : la coopération permet d'améliorer la coordination. La coopération se résume dans le fait que les agents travaillent à la satisfaction d'un but commun, ou individuel, dont l'objectif

est d'améliorer le mode de travail des agents. Le recours à une coopération entre agents coopératifs se justifie lorsque le SMA doit Durfee et al. [Durfee 1989] :

- Augmenter la vitesse de complétion d'une tâche par le parallélisme : mais en évitant un recours exclusif à des modes de coordination.
- Elargir l'ensemble ou la portée des tâches réalisables par le partage de ressources (information, expertise,...) : c'est-à-dire améliorer la gestion des ressources. Ferber cite la spécification comme un moyen d'atteindre cet objectif [Ferber 1999].
- Augmenter la fiabilité du système quant à l'achèvement d'une tâche en dupliquant les moyens de l'effectuer : il s'agit donc d'un objectif de robustesse du système face à des défaillances éventuelles de l'un des agents. L'emploi de plusieurs robots d'exploitation illustre cette préoccupation de robustesse : la panne de l'un est compensée par l'intervention d'autres robots [Ferber 1999]. La redondance d'activités est dans ce cas désirée.
- Diminuer les interférences entre les tâches en évitant les interactions préjudiciables ce qui signifie améliorer la coordination d'activités autour de ces tâches en gérant les conflits éventuels.

#### **3.4.4 L'interaction par la négociation entre ressource**

La négociation est le processus par lequel un groupe d'agents parviennent à un accord mutuellement acceptable sur une question. Les tentatives de négociation sous-tendent à coopérer et à coordonner, à la fois lorsque les agents sont autonomes ou lorsqu'ils sont coopératifs. Pour qu'un agent puisse influencer un autre pour agir d'une certaine manière, le premier doit être capable de convaincre le second en utilisant ses propres connaissances.

Le moyen d'atteindre cet objectif est de faire des propositions, négocier les options, offrir des concessions et arriver à un accord mutuellement acceptable.

Ainsi, malgré toute la variété d'approches, la recherche dans la négociation peut être envisagée pour faire face à trois grands thèmes : les protocoles de négociation, les objets de négociation et les modèles de prise de décision. [Halioui 2010]

##### **- Protocoles de négociation**

Un protocole de négociation est l'ensemble des règles qui régissent l'interaction. Ce protocole concerne les types des participants admissibles (négociateur, tierce partie...), les états de négociation (acceptation de l'offre, négociation conclue...), les événements qui changent les états de négociations (offre acceptée, absence d'offres...) et les actions valides

d'un participant dans des situations particulières (quel message peut être envoyé?, par qui?, à qui?, à quel stade ?...)

### **- Objet de négociation**

Les objets de négociation sont un ensemble de questions sur lesquelles un accord doit être atteint. A l'extrême, l'objet de négociation peut contenir une seule question (prix), comme il peut contenir plusieurs questions (prix, qualité, délais...). Dans le cas le plus simple, la structure et le contenu de l'accord sont fixés et les participants peuvent accepter ou refuser (c.-à-d. prendre ou laisser l'offre).

A un niveau plus élevé, les participants ont la flexibilité de modifier les valeurs relatives aux questions attachées à l'objet de négociation (c.-à-d. faire des contre propositions pour assurer le meilleur accord en fonction de leurs objectifs de négociation).

### **- Modèle de prise de décision**

Les participants utilisent des modèles de prise de décision pour agir en conformité avec le protocole de négociation en vue d'atteindre leurs objectifs. La complexité du modèle ainsi que l'ensemble des décisions qui doivent être prises, sont influencés par le protocole en place, par la nature de l'objet de négociation ainsi que la gamme d'opérations qui peuvent être effectuées.

L'importance relative de ces trois thèmes varie selon la négociation et le contexte environnemental. Ainsi dans certaines circonstances, le protocole de négociation est la préoccupation dominante [Jen 2000]. Par exemple, le concepteur du système peut montrer que la négociation est mieux organisée au moyen d'une forme particulière de vente aux enchères (ex Anglaise, Néerlandaise, Vickrey...).

## **3.5 Les langages de communication**

Lorsque la notion de groupe d'agents est apparue, les recherches sur des mécanismes permettant aux agents de communiquer entre eux se sont amorcées. Celles-ci ont mené à des langages de communication et de représentation de l'information qui sont devenus des « standards » en SMA : KQML (Knowledge Query Manipulation Language) qui détermine un format pour les messages et un protocole pour la réception et l'envoi de ces derniers. KQML permet aux agents de partager des informations avec les autres agents du système afin de coopérer pour résoudre un problème. Un autre langage est apparu : le langage (FIPA ACL), qui est semblable à KQML ; des protocoles d'interaction comme le contractnet et d'autres protocoles populaires lui ont été ajoutés.

Tout langage multi agent est représenté par une structure de donnée comprenant les champs :

- Emetteur.
- Récepteur.
- Langage utilisé : langage dans lequel le vrai message est rédigé.
- Contenu du message : le vrai message qui fait l'objet de la communication.
- L'ontologie : le vocabulaire dans un domaine donné pour que les agents puissent se comprendre.
- Ensemble de définitions concernant le message.

### Exemple de KQML :

L'agent A veut connaître toutes les personnes définies comme étant des hommes.

```
(ask-all
:sender A
:receiver B
:receiver B
:language PROLOG
:ontology phylosophie
:content
homme (=x)
:reply_with question1
)
```

### 3.5.1 ACL : Agent Communication Language

L'organisation Foundation for Intelligent Physical Agents (FIPA) a été créée en 1996. Parmi ses préoccupations, une place importante concerne l'élaboration des spécifications du langage de communication entre agents FIPA-ACL.

Ayant une syntaxe similaire à KQML le langage s'appuie sur la définition de deux ensembles :

- Un ensemble d'actes de communication primitifs, auquel s'ajoutent les autres actes de communication pouvant être obtenus par la composition des ces actes de base
- Un ensemble de messages prédéfinis que tous les agents peuvent comprendre
- FIPA-ACL possède 21 actes communicatifs, exprimés par des performative, qui peuvent être groupés selon leur fonctionnalité de la façon suivante :



Passage d'information: inform\*, inform-if (macro act), inform-ref (macro act), confirm\*, disconfirm\*

- Réquisition d'information : query-if, query-ref, subscribe
- Négociation : accept-proposal, cfp, propose, reject-proposal
- Distribution de tâches (ou exécution d'une action) : request\*, request-when, request-whensoever, agree, cancel, refuse
- Manipulation des erreurs : failure, not-understood

En FIPA-ACL il n'existe pas de primitives de gestion ni de facilitation.

Les actes communicatifs peuvent être primitifs ou composés. Les actes communicatifs primitifs sont définis de façon atomique, c'est-à-dire qu'ils ne sont pas définis à partir d'autres actes (dans la classification ci-dessus ils sont suivis d'une étoile - "\*"). En revanche, les actes communicatifs composés sont définis à partir d'autres actes par l'une des opérations suivantes

Un acte fait partie du contenu d'un autre acte ; à travers l'opérateur de composition " ; " pour indiquer une séquence d'actions

À travers l'opérateur de composition " | " pour indiquer un choix non déterministe de l'action.

#### - **Structure d'un message**

Un message comprend plusieurs éléments qui sont présentés dans le tableau suivant (Tableau 2) :

<b>Elément</b>	<b>Signification</b>
Performative	Type de l'acte communicatif
Sender	L'émetteur du message
Receiver	Le destinataire du message
reply-to	Participant à l'acte de communication
Content	Le contenu du message (l'information transportée par la performative)
Language	Le langage dans lequel le contenu est représenté
Encoding	Décrit le mode d'encodage du contenu du message

Ontology	Le nom de l'ontologie utilisée pour donner un sens aux termes utilisés dans le content
Protocol	Contrôle la conversation
conversation-id	Identificateur de la conversation
reply-with	Identificateur unique du message, en vue d'une référence ultérieure
in-reply-to	Référence à un message auquel l'agent est entrain de répondre (précisé par l'attribut reply-with de l'émetteur)
reply-by	Impose u délais pour la réponse

**Tableau 2 Composition d'un message ACL**

Dans ce qui suit un exemple de message ACL

```
(inform
  :sender A
  :receiver B
  :reply-with laptop
  :language KIF
  :ontology ordinateurs
  :content (= (prix HP-Jet) (scalar 1500
USD))
  :reply-by 10
  :conversation-id conv01
)
```

#### - **Sémantique du modèle**

SL (Semantic Language) est le langage formel employé pour définir la sémantique des performatives de FIPA-ACL. SL utilise une logique multi-modale avec les opérateurs modaux pour les croyances (B), les désirs (D), les croyances incertaines (U) et les intentions (buts persistants, PG). SL peut représenter des propositions, des objets, et des actions. L'origine de SL remonte au travail de [COH90b], mais sa forme courante est principalement basée sur le travail de [SAD91]. Une description détaillée de SL, y compris sa propre sémantique peut être trouvée dans les spécifications de FIPA-ACL [FIPA97].

Le modèle mental d'un agent est basé sur la représentation de trois attitudes primitives : croyance, incertitude et choix (ou, dans une certaine mesure, but). Elles sont respectivement formalisées par les opérateurs modaux B, U, et C.

B<sub>i</sub>(p) : "i croit (implicitement) que p est vrai"  
U<sub>i</sub>(p) : "i pense que p est plutôt vrai que faux "  
C<sub>i</sub>(p) : "i souhaite que p soit vrai. Pour ce faire, l'agent établit un plan d'actions lui permettant d'atteindre le but désiré p"

Les composantes des actes de communication planifiés au sein d'un agent caractérisent à la fois le but pour lequel l'acte est choisi et les conditions qui doivent être satisfaites pour que l'acte soit exécuté. Pour un acte donné, le premier est désigné sous le nom de l'effet rationnel (rational effect), et le dernier comme pré-conditions de faisabilité (feasibility preconditions) qui sont les qualifications de l'acte. Ainsi, chaque acte de communication est défini de cette manière, ce qui permet aux agents de raisonner et d'avoir un comportement rationnel cohérent.

De plus, un ensemble de propriétés ont été définies et doivent être respectées.

#### - **Propriétés**

L'intention d'un agent de réaliser un but donné se traduit par l'intention que l'un des actes connus par l'agent soit exécuté de telle sorte que son effet rationnel corresponde exactement au but de l'agent.

L'ensemble des pré-conditions de faisabilité pour un acte de communication peut être décomposé en deux sous-ensembles : les pré-conditions de capacité et les pré-conditions du contexte dans lequel se trouve l'agent.

Les pré-conditions de capacité caractérisent les capacités intrinsèques d'un agent dans l'exécution d'un acte donné. Par exemple, pour affirmer sincèrement une certaine proposition P, un agent doit croire que P est vrai.

Les pré-conditions associées au contexte caractérisent la pertinence de l'acte par rapport au contexte dans lequel il est exécuté. Par exemple, un agent peut faire une promesse tout en croyant que l'action promise est nécessaire pour le destinataire.

Un agent ayant observé (reçu) un acte de communication doit croire que les conditions préalables de faisabilité sont vraies et que l'agent qui a exécuté l'acte (envoi de message) a l'intention de provoquer l'effet rationnel (effet intentionnel).

### 3.5.2 Similitudes et différences entre KQML et FIPA-ACL

KQML et FIPA-ACL sont presque identiques en ce qui concerne leurs concepts de base et les principes qu'ils observent. Les deux langages diffèrent principalement dans les détails de leurs cadres sémantiques. Dans un sens, cette différence est substantielle :

Il n'est pas possible de proposer une traduction systématique entre les performatives de KQML et celles complètement équivalentes de FIPA, ou vice-versa.

Les différences inéluctables pourraient avoir peu d'importance pour les programmeurs d'agents intelligents, si leurs agents ne sont pas de véritables agents BDI.

Les deux langages ont la même syntaxe. C'est-à-dire, un message de KQML et un message de FIPA-ACL sont syntaxiquement identiques excepté, naturellement, dans leurs différences sur les noms des primitives de communication.

Les deux langages n'impliquent aucun engagement de base vis-à-vis d'un langage pour le contenu. Cependant, dans le cas de FIPA-ACL, comme nous l'avons mentionné, il est nécessaire de doter les agents d'une capacité de compréhension de  $\$SL\$$  pour traiter correctement un message reçu.

Sémantiquement, les deux langages diffèrent au niveau de ce qui constitue la description de la sémantique :

pré-conditions, post-conditions, et conditions d'accomplissement pour KQML

pré-conditions de faisabilité et effets rationnels pour FIPA-ACL

Ils diffèrent également au niveau du choix et des définitions des modalités qu'ils utilisent (le langage employé pour décrire les états des agents).

Une autre différence entre les deux ACL est dans la gestion des agents (enregistrement dans le système, localisation, appartenance à un groupe,...). Dans KQML, ces tâches sont associées aux performatives que le langage traite en tant que propositions de premier ordre. FIPA-ACL, prévu pour être un ACL pur, ne considère pas ces tâches individuellement comme des actes de communication. Au lieu de cela, il les traite en tant qu'actions rattachées à la primitive "request" et définit ainsi une gamme d'actions réservées qui couvrent les tâches de gestion dans le cycle de vie des agents telles que la tâche d'enregistrement. Dans cette approche, les actions réservées n'ont pas été formellement définies et le sont en termes de descriptions en langage naturel.

Enfin, tout système qui doit utiliser KQML ou FIPA-ACL ou tout autre ACL doit fournir les éléments suivants :

- Une suite d'APIs qui facilitent la composition, l'envoi et la réception des messages de ACL
- Une infrastructure des services qui aident la gestion des agents comme l'enregistrement, l'identification, la recherche d'agents, déclaration des services offerts, etc.
- Un code pour chaque type de message réservé (acte communicatif ou performative) qui prend en charge la sémantique des actions relatives à un domaine d'application

Idéalement, un programmeur devrait seulement fournir le point trois. Les points un et deux devraient être des composants réutilisables que nous pouvons intégrer dans le code de l'application.

Les spécifications des langages FIPA-ACL et de KQML n'imposent aucune condition sur le langage de programmation ni sur la plate-forme utilisée pour l'implantation, pour autant que l'implantation soit conforme aux spécifications du langage.

Sur Internet nous remarquons remarquer plusieurs efforts d'implantation de ces langages, dont les plus connus sont JADE, JATLite, KAPI et COBALT proposé par l'équipe IRIT/SIERA de Toulouse.

### 3.6 Communication des agents dans un environnement sans fil

#### 3.6.1 Définition d'un réseau sans fil

Un réseau sans fil (wireless network) est, comme son nom l'indique, un réseau (Figure 3) dans lequel au moins deux terminaux peuvent communiquer sans liaison filaire. [Benmammar 2009]

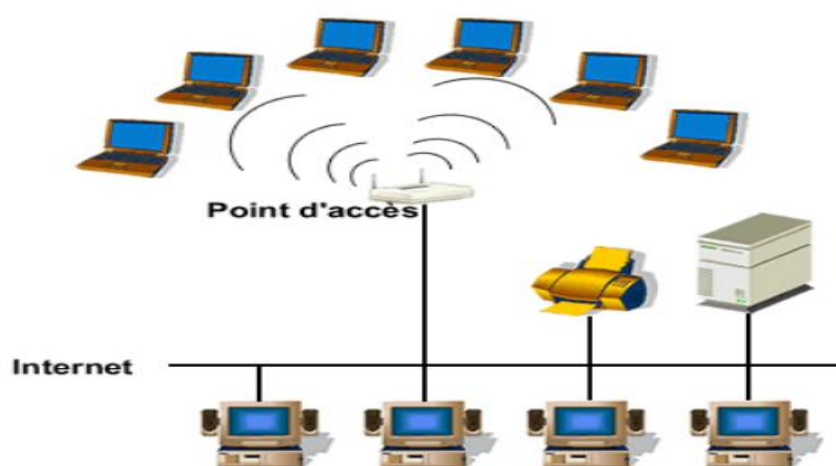
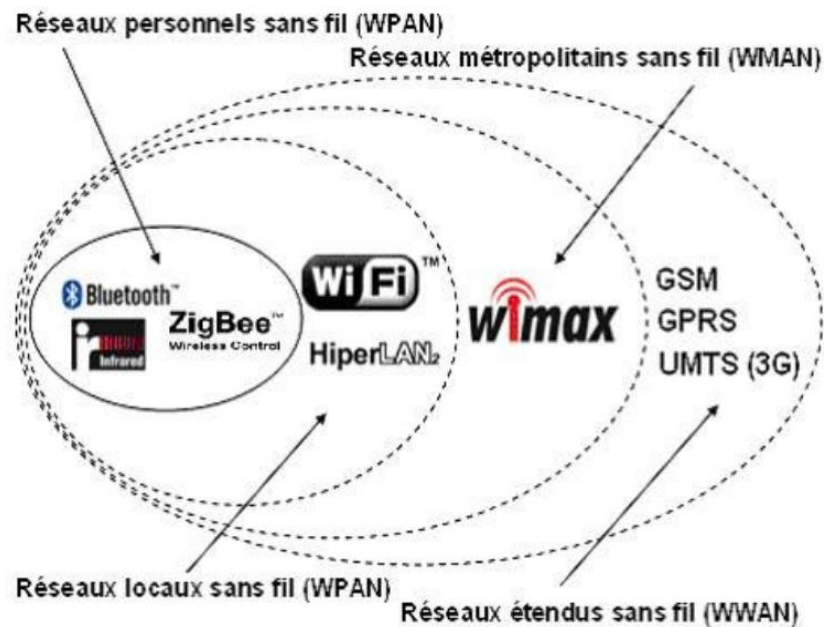


Figure 3 Réseau sans fil

### 3.6.2 Les catégories de réseaux sans fil

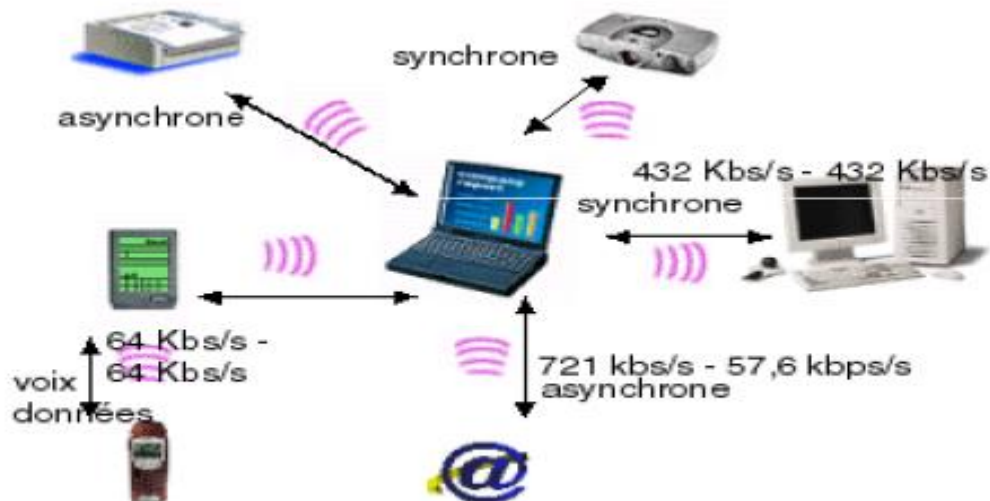
On distingue habituellement plusieurs catégories de réseaux sans fil, selon le périmètre géographique offrant une connectivité (appelé zone de couverture) :



**Figure 4** Types de réseaux sans fil

D'après la figure précédente (Figure 4), nous parvenons aux types de réseaux sans fil suivants :

- WPAN: Bluetooth,**  
-Wireless Personal Area Network
- WLAN: Le Wifi,**  
-Wireless Local Area Network
- WMAN: WiMAX,**  
-Wireless Metropolitan Area Network
- WWAN: GSM, GPRS, UMTS**  
-Wireless Wide Area Network



**Figure 5 Bluetooth**

**- Le WiFi**

Le WiFi permet la mise en place des transmissions dans les endroits où la pose de câble est difficile, voir impossible.

**- WiMAX**

La norme de réseau métropolitain sans fil la plus connue est le WiMAX, permettant d'obtenir des débits de l'ordre de 70 Mbit/s sur un rayon de plusieurs kilomètres.

**- Réseaux étendus sans fil WWAN**

Les principales technologies sont les suivantes :

- GSM (Global System for Mobile Communication ou en français Groupe Spécial Mobile)
- GPRS (General Packet Radio Service)
- UMTS (Universal Mobile Telecommunication System)

**- UMTS**

L'UMTS signifie système universel de télécommunications mobiles, c'est-à-dire un réseau mobile capable d'offrir des services multimédias, partout et à tout moment.

**- Les couches de communication**

La communication entre agents dans un environnement sans fil peut être divisée en quatre couches :

- La couche d'Interaction

- La couche de Langage de Communication
- La couche de Message
- La couche de Transport

### 3.6.3 Communication des agents dans un environnement sans fil

La communication entre agents dans un environnement sans fil peut s'effectuer à différents niveaux et de différentes façons. (Figure 6)

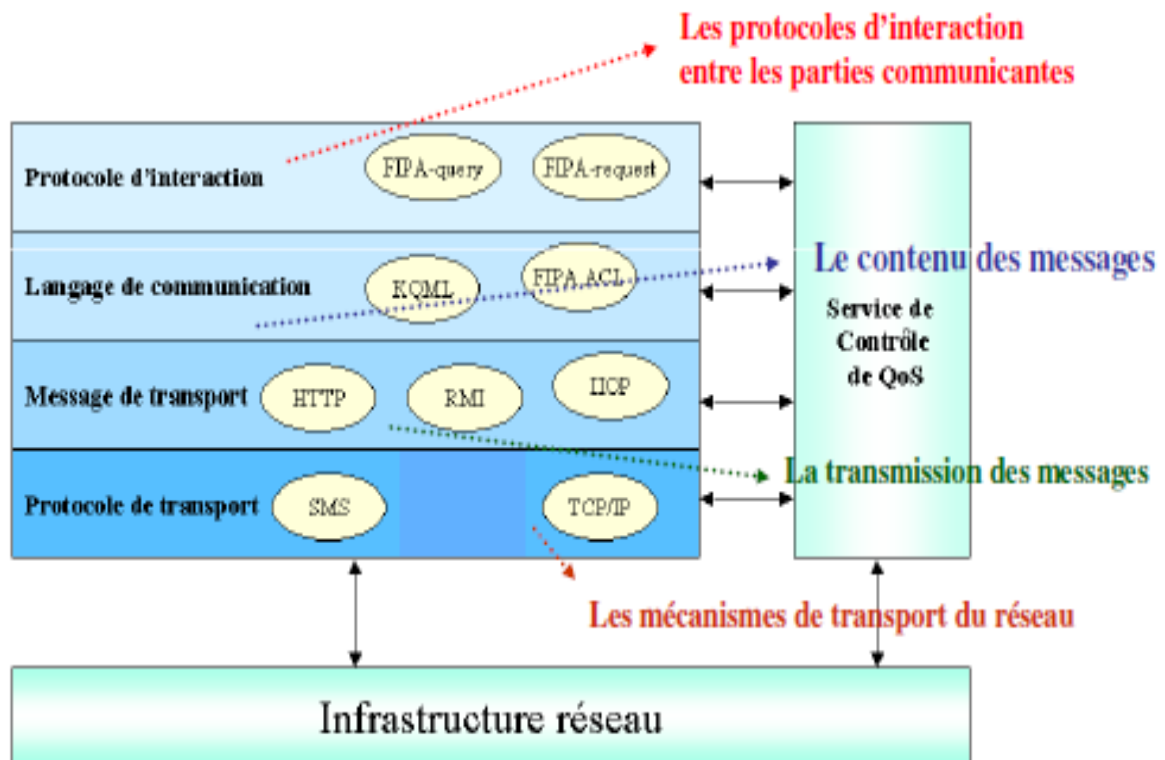


Figure 6 Communications sans fil dans un SMA

### 3.7 Les plateformes SMA

Il existe plusieurs plateformes multi-agents :

- **JACK** : est un langage de programmation et un environnement de développement pour agents cognitifs, développé par la société Agent Oriented Software comme une extension orientée agent du langage Java.
- **JADE** : (Java Agent DEvelopment) est un framework de développement de systèmes multi-agents, open-source et basé sur le langage Java. Il offre en particulier un support avancé de la norme FIPA-ACL, ainsi que des outils de validation syntaxique des messages entre agents basé sur les ontologies.
- **Jadex** : est une plate-forme agent développée en JAVA par l'université de Hambourg



qui se veut modulaire, compatible avec de nombreux standards et capable de développer des agents.

- **Jagent** : est un framework open source réalisé en Java dont l'objectif est de faciliter le développement et le test de systèmes multi-agents.
- **Janus** : est une plateforme multi-agents modulaire écrite en Java. Elle permet de créer des systèmes multi-agents avec ou sans une approche organisationnelle basée sur le modèle Capacité-Rôle-Interaction-Organisation(CRIO).
- **Jason** : est un environnement open source de développement d'agents dans le formalisme AgentSpeak, et développé en Java
- **MadKit** : est une plate-forme multi-agents modulaire écrite en Java et construite autour du modèle organisationnel Agent/Groupe/Rôle. C'est une plateforme libre basée sur la licence GPL/LGPL développée au sein du LIRMM.
- **MAGIQUE** : est une plate-forme pour agents physiquement distribués écrite en Java et fournissant un modèle de communication original d'appel à la cantonade. Dans MAGIQUE, les compétences sont dissociées des agents. L'architecture des agents et les différentes compétences sont développées séparément. Les compétences sont ensuite greffées comme plugin dans les agents au gré du concepteur. Cette plate-forme est développée au sein du LIFL.
- **OMAS** : Open Multi-Agent Asynchronous Systems est une plateforme de recherche développée par l'équipe d'intelligence artificielle de l'Université de technologie de Compiègne, sous la direction de Jean-Paul Barthès.
- **SemanticAgent** : est basé sur JADE et permet le développement d'agents dont le comportement est représenté en SWRL.
- **SemanticAgent** est développé au sein du LIRIS, il est open-source et sous licence GPL V3.
- **SPADE**: est un environnement de développement d'organisations multi-agents basé sur le protocole XMPP et est écrit en Python.
- **MASSIVE** : est un logiciel pour la simulation de foule, basé multiagents, qui a permis la création d'effets spéciaux dans un grand nombre de films, ayant été développé à l'origine pour les scènes de combat dans Le Seigneur des anneaux.
- **Golaem Crowd** est un plug-in pour Maya (logiciel) basé multi-agent et permettant d'effectuer des simulations de foule pour les effets spéciaux directement dans Maya.

### **3.8 Domaines d'application des SMA**

Nous distinguons généralement trois types d'utilisation : la simulation de phénomènes complexes, la résolution de problèmes, et la conception de programmes.

#### **3.8.1 Simulation de phénomènes complexes**

Nous utilisons les systèmes multi-agents pour simuler des interactions existant entre agents autonomes. Nous cherchons à déterminer l'évolution de ce système afin de prévoir l'organisation qui en résulte. Par exemple, en sociologie, nous pouvons paramétrer les différents agents composant une communauté. En ajoutant des contraintes, nous pouvons essayer de comprendre quelle sera la composante la plus efficace pour parvenir à un résultat attendu (construction d'un pont). Ils permettent même d'expérimenter des scénarios qui ne seraient pas réalisables sur des populations réelles, que ce soit pour des raisons techniques ou éthiques. [Gilbert 2007], ce qui importe c'est le comportement d'ensemble et non pas le comportement individuel. Des applications existent en physique des particules (agent = particule élémentaire), en chimie (agent = molécule), en robotique (agent = robot, dans le cas d'une implémentation sur robot réel, nous parlerons de système multi-robots), en biologie cellulaire (agent = cellule), en éthologie (agent = animal), en sociologie et en ethnologie (agent = être humain). L'autonomie permet ici de simuler le comportement exact d'une entité.

Comme exemples concrets, nous pouvons citer par exemple EpiSIM, logiciel qui permet de simuler la propagation de maladies dans un espace géographique paramétrable, ou encore DS, qui simule l'évolution de la population de l'île de La Réunion et permet d'anticiper son urbanisation. Nous trouvons également des simulations de trafic routier [Doniec 2008] ou d'évacuation de foule paniquée dans un bâtiment, qui permettent d'étudier l'origine des problèmes (embouteillages, bousculades...) afin de réfléchir à des solutions pour les restreindre.

#### **3.8.2 Résolution de problèmes**

L'intelligence artificielle distribuée est née pour résoudre les problèmes de complexité des gros programmes monolithiques de l'intelligence artificielle : l'exécution est alors distribuée, mais le contrôle reste centralisé. Au contraire, dans les SMA, chaque agent possède un contrôle total sur son comportement. Pour résoudre un problème complexe, il est en effet parfois plus simple de concevoir des programmes relativement petits (les agents) en interaction qu'un seul gros programme monolithique. L'autonomie permet au système de s'adapter dynamiquement aux changements imprévus qui interviennent dans l'environnement.

### **3.8.3 Conception de programmes**

Dans le même temps, le génie logiciel a évolué vers des composants de plus en plus autonomes. Les SMA peuvent être vus comme la rencontre du génie logiciel et de l'intelligence artificielle distribuée, avec un apport très important des systèmes distribués. Par rapport à un objet, un agent peut prendre des initiatives, peut refuser d'obéir à une requête, peut se déplacer... L'autonomie permet au concepteur de se concentrer sur une partie humainement appréhendable du logiciel.

### **3.8.4 Dans la finance**

Les systèmes multi-agents sont également utilisés dans le monde de la finance. Par exemple, la plateforme MetaTrader 4 permet d'utiliser des agents experts en trading automatique qui suivent les cours du Forex.

### **3.8.5 Dans le domaine industriel**

L'industrie a vu une exploitation régulière des SMA grâce à leurs capacités de modélisation des systèmes complexes, à tel point qu'un nombre important de travaux de recherches se sont consacrés à leur étude comme : Chaib-draa [**Chaib-draa 1995**], Kouiss et Pierreval [**Kouiss 1995**], Müller et Parunak [**Müller 1998**], Parunak [**Parunak 1996**] et [**Parunak 1998**], Shen et al., [**Shen 2001**], Blecker et Graf [**Blecker 2003**] et Mahesh et al., [**Mahesh 2006**]. Le parallélisme des traitements, l'existence d'outils de validation et de méthodologies de conceptions conduisent à faciliter la maintenance des systèmes ainsi développés et donc à améliorer leur réutilisabilité.

Des travaux ont été réalisés dans le domaine des systèmes multi-agents appliqués à l'industrie. Mansour et al. [**Mansour 2014**] présentent une solution logicielle flexible pour contrôler les systèmes multi-agents intelligents de production "FMCC". Cette dernière permet aux utilisateurs d'éditer différentes séquences de fabrication pour chaque produit avec une multitude de combinaisons possibles (flexibilité dans l'ordre des opérations de fabrication, le temps d'attente maximum pour un produit donné, le temps d'attente minimum entre deux opérations de production ...). Les principaux éléments de cette application sont l'utilisation du protocole OPC, la technologie RFID et le multi-threading. Anane et al., [**Anane 2009**] modélisent la coordination des différentes parties collaboratives aussi bien internes qu'externes d'une chaîne logistique en utilisant les modèles de coordination par formation de coalitions proposés dans les SMA. Ils ont proposé une approche agent, un algorithme de formation de coalitions ainsi qu'un protocole d'interaction entre agents indispensable pour

mettre en œuvre cette coordination distribuée. Ils illustrent leur démarche par un exemple pris dans le domaine de l'industrie avionique.

L'article de Atalla [**Atalla 2011**] traite les différents aspects de la conception d'un système multi-agents pour gérer et contrôler les installations industrielles complexes de la conceptualisation jusqu'au déploiement. Une étude approfondie de la littérature est faite pour donner une grande image de ce qui a été accompli dans le milieu universitaire et industriel pour ce sujet.

Dans [**Pannequin 2006**], les auteurs proposent une plateforme basée sur l'émulation du système opérant ; ainsi il est possible de représenter des systèmes complexes, d'échelle réaliste. Ils introduisent des primitives de modélisation permettant de représenter les aspects spécifiques au contrôle par le produit. Puis ils ont spécifié la manière dont un système de contrôle (éventuellement multi-agents) peut être intégré à leur plateforme. Enfin ils appliquent leur démarche au développement du modèle d'émulation d'un site industriel.

Dans [**Clair 2008**], une plate-forme de simulation de gestion de production a été développée afin de comparer différentes approches. Le problème consiste en l'affectation de tâches à des opérateurs sur des machines d'usinage de pièces, en fonction d'un cahier de commandes. L'auteur souhaite modéliser le problème sous forme d'un problème de satisfaction de contraintes distribuées et le résoudre en utilisant des techniques d'auto-organisation, afin de proposer des solutions capables de répondre à la dynamique de l'environnement. Cette approche va mettre en action des agents qui vont coopérer pour parvenir à trouver la meilleure solution possible au scénario proposé. Après avoir analysé le problème, plusieurs approches classiques développées ont été présentées et ont été confrontées à l'approche auto-organisatrice imaginée. Pour comparer les avantages et les limites de cette approche l'étude des résultats de quelques tests suivant divers critères et métriques a été établie.

L'auteur dans [**Reguieg 2010**] propose une approche basée sur l'élaboration d'un protocole de coordination pour une architecture multi-agents réalisée dans la plateforme Jade. Le modèle proposé consiste principalement en cinq composantes : Agent Ressource (AR), Agent Interface(AI) ,Agent Superviseur (AS), Data Manager Agent (DMA) ,et l'Agent Coordinateur (AC).Le protocole de coordination assure les échanges de messages entre les agents ainsi que qu'il leur offre des comportements prédéfinis .Le contexte de notre approche est la gestion de production dynamique ce qui nécessite des interactions entre les agents pour résoudre des problèmes d'allocation de ressource ,de gestion de conflits ,et une meilleure

réaction lors d'une panne de ressource .L'agent coordinateur (AC) est doté d'un mécanisme d'apprentissage .L'apprentissage de l'AC facilite la prise de décision pour le système de production et requiert moins de temps d'exécution ce qui libère l'agent AC pour des éventuels traitements et rend le système rapide ,flexible et plus réactif.

Dans [Taghezout 2011], l'intégration des agents aux Systèmes Interactifs d'Aide à la Décision (SIAD) a été proposée ; afin de résoudre certains problèmes d'incertitude dans le système d'ordonnement de la production dynamique. Le système développé donne aux centres de décision la possibilité pour prendre des décisions dans un contexte dynamique. Plus précisément, les Stations Intégrées de Production (SIP) seront équipées d'un comportement suffisant pour exécuter des opérations concrètes et réagir simultanément à la complexité des problèmes causés par l'ordonnement dynamique dans des situations réelles. Ces agents expriment leurs préférences en utilisant la méthode ELECTRE III, afin de résoudre les différences. Le mécanisme de négociation est basé sur le Protocole Contract Net (CNP). Le protocole développé sur JADE fournit des échanges de messages entre les agents et leur propose des comportements prédéfinis. L'approche est testée à travers des scénarii simples.

Moyaux [Moyaux 2001] traite la manière dont les SMA pourraient aider à la gestion de la production, de la distribution et des stocks ; il se base sur l'approche DDP(Demand Driven Production and Distribution) pour étudier comment les SMA pourraient résoudre le principal problème de cette approche :la centralisation des décisions .A cet effet les SMA et la planification distribuée ont été présentés .Une réflexion a été menée sur les architectures multi-agents et les types d'interactions inter-agents aptes à permettre la planification de la production ,du stockage et de la distribution dans une chaine logistique .

Les auteurs dans [Wenrong 2014] examinent l'utilisation éventuelle des approches DI dans les systèmes de gestion d'entrepôt (WMS). En outre, le document examine les défis dans la gestion d'entrepôt et compare ces défis avec les caractéristiques des problèmes de production avec lesquels l'approche DI s'adapte.

Une nouvelle approche a été proposée dans [Gazdar 2008]; baptisée COSAH (COntainer Stacking via multi-Agent approach and Heuristic method) qui permet de simuler, résoudre et optimiser l'espace de stockage disponible pour manier les départs et les arrivées des conteneurs dans un port fluvial ou maritime. Les résultats obtenus, présentés et illustrés, montrent l'efficacité de COSAH en particulier, et d'une méthode d'optimisation heuristique distribuée alliant les deux concepts : Agent et Heuristique, en général.

#### **4. Conclusion**

Tout au long de ce chapitre nous avons décrit les SMA de manière générale ; leurs concepts et fonctionnements. L'interaction des agents y joue un rôle primordial ; afin de pouvoir coordonner les différentes tâches qu'ils accomplissent ; sans qu'il y ait d'erreurs, de retard ou bien d'échec dans l'accomplissement du but global.

Nous avons également abordé les différents langages de communication entre les agents, la communication des agents dans un environnement sans fil.

Dans la section qui suit nous avons défini les différentes plateformes multi-agents développées jusqu'à maintenant et pour finir nous avons étudié les différents domaines d'application des SMA en mettant l'accent sur le domaine que nous allons étudier par la suite dans notre application c.-à-d. le domaine industriel.

# Chapitre 2 Systèmes Automatisés de Stockage /Déstockage (AS/RS)

## 1. Introduction

Le marché des systèmes automatisés de stockage et de déstockage (ASRS) est actuellement confronté à une demande croissante des industries de l'automobile, des produits alimentaires, de boissons...

Ils sont fortement recommandés dans ces industries pour stocker et récupérer des produits en ayant un inventaire de la base de stockage en temps réel.

La notion de stockage concerne les biens transformés ou pas, se situant en amont, à l'intérieur ou en aval du flux de production.

Ils ont pour vocation principale de réguler les consommations tout en maîtrisant les flux d'entrée. Ils offrent une disponibilité immédiate d'un composant pour servir un besoin. Ils sont utilisés dans le cycle d'exploitation à travers, les matières, les marchandises, les produits intermédiaires, les produits finis, mais aussi les diverses fournitures utilisées directement ou indirectement dans les principaux processus de production.

Les AS/RS engendrent actuellement un intérêt de plus en plus grandissant, en raison de l'importance du stockage dans l'industrie. Le stockage permet le lissage des irrégularités d'approvisionnements et/ou de la production en réduisant les risques de ruptures et en favorisant le maintien d'une activité continue. Les stocks permettent aussi de maintenir les articles à proximité de leur lieu de consommation. Ils limitent considérablement les délais d'attente.

Ces systèmes contribuent à apporter des produits et des matériaux au bon endroit comme par exemple un lieu de travail d'un employé et au bon moment avec précision, ils permettent de construire des tampons de charge de stockage appropriés pour la préparation des commandes et la production, ils offrent aussi : une meilleure exploitation de la surface de stockage, une gestion des stocks plus organisée, plus de rendement dans la production, une réduction de la main d'œuvre, une flexibilité dans les types de produits, l'assurance de garder les produits en bon état, plus de sécurité, une réduction des risques dus aux opérations dangereuses, ils sont interfaçables avec d'autres systèmes comme les FMS, les AGV, ...

Des facteurs tels que la croissance du secteur e-commerce, l'accroissement de l'industrie automobile à l'échelle mondiale, et la tendance des magasins d'automobiles en sont quelques-uns des principaux moteurs de croissance pour le marché de ces systèmes.

Dans ce chapitre, nous définirons d'abord dans la section 1.2 ce qu'est un système automatisé de stockage/déstockage. Une présentation de ses composants est faite dans la section 1.3, ainsi que ses fonctions dans la section 1.4. Dans la section 1.5 une description des différents types d'AS/RS est réalisée. Dans la section 1.6 nous expliquons comment sont placés les produits dans les AS/RS. Par la suite dans la section 1.7 les différentes méthodologies de stockage/déstockage sont mises en évidence. La section 1.8 traite les critères de performances. Dans la section 1.9 un état de l'art englobant les principaux travaux sur les systèmes automatisés de stockage/déstockage est détaillé.

## **2. Définition d'un AS/RS**

Material Handling Institute (1977) définit un système de stockage/déstockage automatisé comme suit : « C'est une combinaison d'équipements et de systèmes de contrôle qui prend en charge, stocke et déstocke des produits avec précision, exactitude et célérité sous un certain degré d'automatisation ».

## **3. Fonctions**

Un système automatisé de stockage/déstockage peut réaliser automatiquement les opérations suivantes:

- Déstockage d'un article d'un emplacement de stockage,
- Transport de cet article vers un point de livraison,
- Stockage d'un article, dans un emplacement prédéterminé, à partir d'un point de dépôt.

En plus des opérations précédentes un AS/RS peut réaliser d'autres fonctions comme : recevoir, identifier, classer, emballer..., des produits finis ou en cours de production, de la matière première, des outils,...

## **4. Composants**

Un AS/RS est composé : de racks de stockage formé chacun d'un ensemble de casiers, d'un ensemble d'allées chacune placée entre deux racks, de stations dépôt/livraison ou stations d'entrées/sorties, de machines de stockage/déstockage, et d'un système de contrôle supervisant le tout.



### **- Rack de stockage**

C'est un ensemble de cellules ou casiers mient l'un à coté de l'autre ce qui forme une ligne et l'un au dessus de l'autre formant une colonne. Dans chaque cellule on peut stocker un ou plusieurs produits en multi couches.

### **- Station dépôt/livraison**

La station de dépôt/livraison appelée aussi station d'entrée/sortie est le point par lequel passent tous les produits entrants ou sortants du système. C'est l'interface entre la machine S/R et le système de transfert externe à l'AS/RS. Le transfert des produits de l'AS/RS aux autres systèmes de production se fait soit manuellement soit en utilisant des systèmes de transfert tel que : les AGV, les convoyeurs.

### **- Machine de stockage/déstockage (S/R)**

La machine de stockage/déstockage (S/R: Storage/Retrieval machine) appelée aussi transstockeur est une structure mobile servant à stocker ou déstocker les produits dans un rack, souvent à grande hauteur. C'est un portique en forme de rectangle creux, les cotés horizontaux glissent sur des rails fixées le long des allées (l'un au sol et l'autre au plafond). Tandis que sur les cotés verticaux coulisse un plateau glissant à deux degrés de liberté (linéaires, vertical et horizontal perpendiculaire à l'allée) permettant de charger et décharger les palettes [Sari 2003]

La nouvelle génération de ces machines a spécialement été conçue pour assurer des cadences très élevées et donc faire gagner en efficacité et en productivité. Elles sont utilisées pour réaliser des opérations de stockage et de déstockage qui peuvent être longues ou même dangereuses pour l'intervention humaine. Ces machines peuvent se déplacer simultanément dans le sens horizontal et vertical.

La machine S/R peut opérer en simple ou en double cycle. Dans un simple cycle elle réalise une opération à la fois soit un stockage soit un déstockage : elle se déplace alors de la station dépôt /livraison vers le casier de stockage ou de déstockage dépose ou récupère le produit et revient à cette même station. Dans un double cycle, elle réalise les deux opérations à la fois un stockage suivi d'un déstockage : cette dernière se déplace du point d'entrée/sortie vers le casier de stockage dépose le produit à stocker ensuite elle se déplace du casier de stockage vers le casier de déstockage récupère le produit à déstocker et revient vers la station de livraison.

## **- Système de contrôle**

Il permet un contrôle en temps réel dans le système de production. Il est composé de trois entités principales :

- Les calculateurs ou ordinateurs puissants en calcul.
- Les logiciels développés pour chaque opération.
- Les bases de données
- Les calculateurs : ce sont des ordinateurs à base de microprocesseurs dotés de mémoire et de puissances de calculs plus ou moins grandes.
- Les logiciels : développés pour chaque type d'opération, ils sont le cœur du système de contrôle.
- Les bases de données, les systèmes de transmission d'information, les capteurs..., ils représentent la troisième entité. Ils permettent la transmission de l'information comme par exemple les emplacements de stockage des différents produits.

La forme de base d'un AS/RS est composée de deux racks séparés par une allée de desserte, chaque rack comportant un certain nombre de casiers. L'allée est desservie par une machine S/R permettant de stocker et déstocker les produits de et vers les casiers. Cette forme d'AS/RS est dite AS/RS à charge unitaire. C'est la forme la plus étudiée. Les AS/RS peuvent être classés suivant la disposition des casiers de stockage et le nombre de machines S/R servant ces casiers.

## **5. Types d'AS/RS**

La classification des AS/RS dépend de plusieurs paramètres, nous citons quelques uns :

- Taille ou dimensions des produits à stocker: largeur, hauteur, profondeur, poids. Les produits peuvent être de petites pièces, des palettes, des containers, ...
- Les produits doivent être stockés un par casier ou bien en multi couches
- Nombre de machines S/R utilisé
- Nombre et positions des points d'entrée/sortie
- Dimensions des casiers : largeur, hauteur, profondeur,
- Nombre total de casiers,

- Rendement : le nombre de stockage/déstockage par unité de temps
- Nombre d'allées, d'étagères,...
- Type de cycle effectué par la machine S/R : cycle simple, cycle double, cycle multi-adresses,...
- Taux d'utilisation des machines.

Il existe différents types d'AS/RS, pour savoir lequel convient le mieux pour être installé dans un environnement de production précis, nous basons notre étude sur les paramètres cités auparavant.

Les principaux types d'AS/RS étudiés sont :

- Les AS/RS à charge unitaire.
- Les AS/RS à mini charge ou à charge réduite.
- Les AS/RS multi allées.
- Les AS/RS à racks glissants.
- Les AS/RS à convoyeur gravitationnel.
- Les AS/RS à personne embarquée.
- Les AS/RS à carrousel.
- Les AS/RS à étagères profondes.

### **5.1 AS/RS à charge unitaire**

L'AS/RS à charge unitaire est composé de plusieurs allées, chacune placée entre deux racks mitoyens et servie par une machine S/R. Chaque casier a une capacité unitaire. Ce type d'AS/RS est utilisé lorsque les charges à stocker sont palettisées ou en container. Le poids de la charge dépasse 250kg. Les machines S/R sont conçues de façon à pouvoir supporter des poids et des volumes importants.

Le contrôle du système à charge unitaire est géré par ordinateur avec des logiciels appropriés.

L'AS/RS à charge unitaire représente la base, les autres types d'AS/RS ne sont que des variations de celui ci.

## **5.2 AS/RS à mini charge ou a charge réduite**

Un système à charge réduite est utilisé pour stocker de petites charges identiques ou différentes tel que : des outils, pièces détachées,... qui sont contenues dans des casiers du système de stockage (un casier peut contenir un ou plusieurs produits en multi couches).

En cas de stockage ou de déstockage, La machine S/R est conçue pour retirer le container en entier qui est déplacé vers un opérateur qui va procéder à la sélection des produits à déstocker ou à stocker. Une fois l'opération effectuée, le container ou casier est remis à sa place dans le système.

## **5.3 AS/RS multi allées**

Un AS/RS multi allées est constitué d'un ensemble de racks, disposés deux à deux en parallèle et séparés par des allées. Chacune de ces allées, appelée allée de desserte donne accès à deux racks. Une allée commune placée perpendiculairement aux racks relie toutes les allées de desserte. Ce type de système est composé d'une seule machine de stockage/déstockage qui dessert l'ensemble des racks, cette machine S/R se déplace, suivant trois axes : verticalement, le long des colonnes formants les racks, horizontalement, le long des allées de desserte et transversalement, le long de l'allée commune dont l'une des extrémités est équipée d'une station de dépôt/ livraison (P/D). La topologie de l'AS/RS multi allées est similaire à celle de l'AS/RS à charge unitaire avec en plus une allée commune reliant toutes les allées de service permettant à la machine S/R de circuler.

## **5.4 AS/RS à racks glissants**

Les AS/RS à racks glissants appelés aussi AS/RS à racks mobiles sont une variation des AS/RS multi allées. Ils sont composés d'un ensemble de racks disposés en parallèles et d'une seule machine de stockage/déstockage (S/R). La particularité de ces systèmes est que les allées de service n'apparaissent que lorsqu'une opération de stockage ou déstockage est prévue dans les racks correspondants. Les racks formant le système glissent latéralement sur des rails de telle sorte que l'on puisse ouvrir une allée entre n'importe quels deux racks mitoyens. Au repos, il existe une seule allée de service, celle-ci pourra être placée entre n'importe quels deux racks mitoyens en faisant glisser les racks. Au repos, il existe une seule allée de service, celle-ci pouvant être ouverte entre n'importe quels deux racks mitoyens.

## **5.5 AS/RS à convoyeur gravitationnel**

L'AS/RS à convoyeur gravitationnel est composé :

- D'un rack qui lui-même est composé de casiers, chaque casier est formé de plusieurs couches de stockage, il est muni d'un convoyeur gravitationnel à base de rouleaux ou roues libres inclinées de façon à permettre le glissement des produits d'un bout à l'autre du casier, donc d'une face à l'autre du rack.
- D'une station de dépôt située sur la face de stockage, où la machine de stockage récupère les produits à stocker.
- D'une station de livraison située sur la face de déstockage, où la machine de déstockage dépose les produits pour la livraison. La station de dépôt et la station de livraison sont situées respectivement au coin bas de la face de stockage et de la face de déstockage.
- D'une machine de stockage sur la face avant du rack et une machine de déstockage sur la face arrière. Les deux machines peuvent se déplacer simultanément sur deux axes. Ces deux axes forment le plan x-y parallèle aux deux faces du rack. Ces deux machines sont reliées entre elles par un convoyeur de restockage permettant à la machine S/R d'accéder aux produits pour les restocker.
- D'un convoyeur de restockage qui est un convoyeur gravitationnel, incliné en sens inverse, reliant les deux faces du rack. Il permet de transférer les produits à restocker vers la station de dépôt.

## **5.6 AS/RS à personne embarquée**

Ce type de système permet de stocker plusieurs petites charges dans chaque casier. Une personne se trouvant sur une plate-forme mobile, choisit les produits à déstocker des casiers, et stocke les produits à emmagasiner dans le casier correspondant. Ce système permet de réduire les temps de stockage/déstockage grâce à la possibilité de stocker/déstocker plusieurs produits en une seule opération. L'opérateur, après les avoir déstockés, charge les produits sur la machine S/R qui les transporte vers la station de dépôt/livraison.

## **5.7 AS/RS à carrousel**

L'AS/RS à carrousel est composé d'un ensemble de casiers montés en carrousel et se déplaçant horizontalement vers une extrémité du rack où l'opération de stockage/déstockage sera réalisée.

Dans ces systèmes, ce sont les casiers qui se déplacent vers l'extrémité du rack ou un opérateur stocke ou déstocke les produits.

Les carrousels sont une bonne alternative au système à mini charge [Tompkins 1984].

### **5.8 AS/RS à étagères profondes**

Dans les AS/RS à étagère profonde, plusieurs charges unitaires peuvent être stockées dans le même casier l'une après l'autre et possédant la même adresse. Au niveau de chaque casier, les produits peuvent se déplacer horizontalement. Ils sont stockés d'un côté du casier et déstockés de l'autre côté. Ce déplacement est possible grâce à un plateau mobile qui se déplace de la machine S/R à l'entrée du casier. Donc ce type d'AS/RS n'est qu'une variation de l'AS/RS à charge unitaire offrant en plus la possibilité de stocker plusieurs charges unitaires en multi couches dans un même emplacement.

## **6. Emplacement des produits**

### **6.1 L'adressage des produits**

Après le contrôle et la réception, les produits destinés à l'AS/RS sont rangés suivant deux méthodes d'affectation :

- **Emplacement fixe et invariable** : c'est une méthode de rangement claire, les produits affectés à un emplacement régulier sont plus faciles à trouver physiquement. Cependant, cette forme de rangement utilise un système de stockage lourd et un espace plus grand.
- **Emplacement variable en fonction des espaces disponibles** : après chaque réception, le produit est rangé à n'importe quel emplacement libre. Il y a risque de trouver la même pièce à différents endroits. Bien que cette forme de rangement présente l'avantage d'une meilleure exploitation de l'espace de stockage, elle est mieux indiquée pour des stocks unitaires. Elle demande beaucoup d'écritures car à chaque fois, il faut mettre à jour le nouvel emplacement sur les documents et le logiciel de gestion des stocks.

Dans la pratique, l'utilisation des emplacements ou des aires de stockage fixes est prédominante. L'emploi d'emplacements variable est admis seulement lorsqu'il y a insuffisance d'espace (capacité de stockage limitée).

### **6.2 Critères de choix des emplacements**

Le choix de l'emplacement dans un AS/RS est fonction des caractéristiques du produit comme :

- **La nature du produit** : pièce solide, vrac solide, vrac liquide, vrac gaz ou encore marchandise palettisée...
- **Les formes et dimensions du produit** : poids, volume, encombrement, possibilité d'empilage ...
- **L'influence de l'environnement sur le produit** : humidité, chaleur, poussières, lumière,...

Il est bien entendu que la capacité de stockage d'un AS/RS est une contrainte supplémentaire au rangement. Les quantités d'un produit ne seront affectées qu'à un emplacement qui peut les contenir en totalité.

### **6.3 Stockage selon la nature du produit**

Le système de stockage est choisit et construit en fonction de la nature des marchandises stockées. On présente les principales formes trouvées dans l'industrie :

#### **- Stockage de petites pièces**

Les pièces de petites tailles sont rangées dans des magasins fermés. L'emploi des systèmes de rayonnage à casiers ouverts est bien adapté pour cette catégorie de produits. Grace à leur petite taille et un encombrement peu contraignant, il faut peu d'espace pour un grand nombre de pièces stockées. Généralement chaque casier a une adresse fixe, unique invariable et peu contenir plusieurs produits disposés en multi couches. L'utilisation d'un AS/RS à mini charge peut être intéressante dans ce cas.

#### **- Stockage de charges unitaires**

Les charges unitaires dont nous faisons allusion sont généralement des marchandises palettisées ou en container de poids important déjà conditionnées dans un dispositif visant à faciliter la manutention et le transport.

Ce sont des catégories de marchandises qui selon les cas sont stockées dans des magasins ouverts ou fermés. Elles peuvent faire l'objet d'un stockage de masse (caisses, ou toutes autres unités superposables). Les unités non superposables requièrent quant à elles l'utilisation de rayonnages pour une exploitation optimale des aires de stockage. Pour un suivi ordonné, il est impératif que l'adressage de ces marchandises obéisse à un système d'emplacements fixes et invariables.

#### **- Stockage du vrac solide**

Le vrac désigne des marchandises qui ne sont pas emballées. Le vrac solide se décompose en matières sensibles aux intempéries (produit alimentaires pour

la plupart des cas) et les matières inertes (sables, gravier, minerais divers). L'utilisation d'emplacements variables en fonction des espaces disponibles est moins contraignante pour cette catégorie d'articles.

Les matières non sensibles aux intempéries sont généralement stockées dans des yards (formes de dépôts ouverts qui ne nécessitent en réalité aucune construction d'un bâtiment. Seul le balisage de la zone par une barrière s'avère cependant nécessaire pour la sécurisation des lieux).

Les matières sensibles aux intempéries nécessitent outre la construction d'au moins un hangar couvert, mais quelques fois l'utilisation de dispositifs de contenance divers (caisses, silos ...).

### **- Stockage du vrac liquide**

Le vrac liquide rassemble l'ensemble des produits liquides de grande masse. Les petites quantités déjà conditionnées en fûts ou bidons obéissent aux règles de rangement des matières unitaires, tel que nous les avons énoncées plus haut. Lorsque nous parlons de vrac, il s'agit donc de quantités très importantes et non emballées.

### **- Stockage du vrac gazeux**

Les produits gazeux sont stockés dans des réservoirs particulièrement adaptés pour supporter des pressions élevées. Leur manipulation demandant le respect de règles strictes de sécurité.

## **7. Méthodologies de stockage ou de déstockage**

Dans les systèmes de stockage, il existe plusieurs façons d'attribuer les emplacements aux produits, on définira les méthodes classiques telles que le stockage aléatoire et le stockage dédié et on parlera des méthodes heuristiques.

### **7.1 Stockage aléatoire ouvert**

Mode de stockage qui consiste à placer les produits, les matières, les pièces ou les articles dans n'importe quel espace ou casier libre au moment de leur réception, plutôt qu'à leur attribuer un emplacement spécifique selon leur identification.

Le stockage aléatoire nécessite moins d'espace que le stockage à emplacement fixe, mais il requiert l'utilisation d'un système informatisé afin de permettre la localisation des matières des pièces ou des produits.



## 7.2 Stockage aléatoire par classe

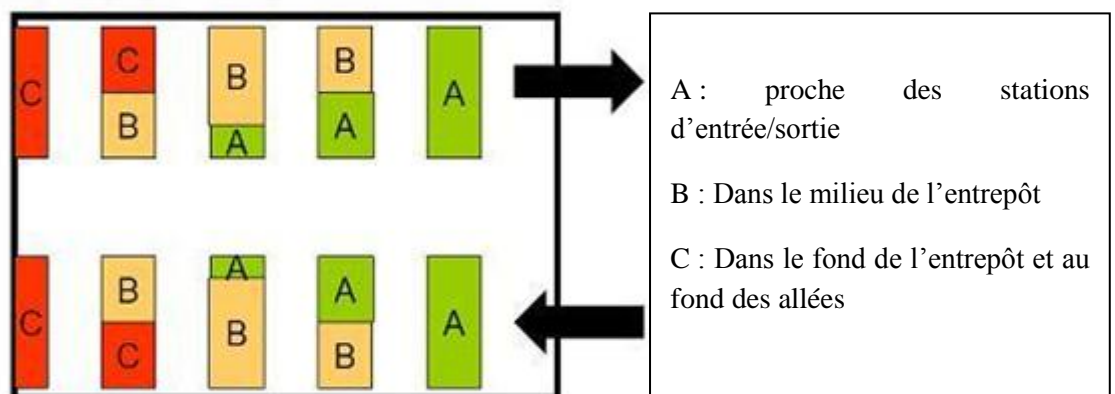
La méthode de stockage par classe organise les positions de stockage en se basant sur les fréquences de demande des différents produits.

Dans les AS/RS les produits sont généralement différents, il y a des produits très utilisés et donc stockés et déstockés fréquemment, d'autres moins utilisés ayant une durée de stockage plus importante. Si les produits dont la durée de stockage est petite sont mis dans des emplacements proches de la station dépôt/livraison et les produits dont la durée de stockage est plus grande sont attribués à des emplacements plus éloignés, le temps de cycle de la machine S/R sera réduit et le rendement du système sera amélioré.

Le stockage par classe subdivise l'espace de stockage en classes, ces dernières sont définies par rapport à la station dépôt/livraison et par rapport à la durée pendant laquelle le produit reste dans le système, les casiers les plus proches de la station dépôt/livraison forment la première classe, dans ces casiers on stocke les produits ayant la plus petite durée de stockage.

Les casiers les plus loin par rapport à la station de dépôt/livraison forment la dernière classe, on y stocke les produits ayant la plus grande durée de stockage.

Généralement le nombre de classes est limité à trois, 'A' représente la classe des produits qui ont une forte rotation, 'B' comprend la classe des articles avec une rotation moyenne et 'C' couvre le reste.



**Figure 7 Représentation des classes ABC**

### **7.3 Stockage dédié**

Dans le stockage dédié, l'espace de stockage est partagé en blocs ou parties, chaque partie est utilisée pour stocker un type de produit déterminé. Les principaux avantages de cette méthode sont :

- La gestion des stocks est simplifiée car la quantité de chaque produit peut être contrôlée.
- Chaque partie ou aire de stockage peut être différente suivant le type de produit stocké, les casiers peuvent avoir des tailles et des profondeurs variables, les allées peuvent être plus ou moins larges,...
- Chaque produit ayant une partie réservée prédéterminée à l'avance, le contrôle du système est facile est simplifié.

Les limites de cette méthode sont :

- L'introduction d'un nouveau type de produit dans le système est difficile car ceci entraîne des modifications dans la conception du système.
- Pour chaque type de produits nous devons réserver suffisamment d'espace et un stock maximum ceci entraîne une augmentation importante dans l'espace total de stockage.

La somme des stocks maximum de chaque produit étant supérieure au stock maximum de tous les produits. Puisque les stocks maxima pour chaque produit sont de l'ordre du double des stocks moyens, le taux de remplissage du système de stockage sera relativement bas [Sari 2003].

### **7.4 Heuristiques**

Pour améliorer la manière de stocker les produits dans et dans le but de minimiser le temps de cycle et l'espace de stockage dans les AS/RS un certain nombre d'heuristique ont été proposé et utilisé à la place des méthodologies classiques de stockage. Lorsque les méthodes exactes ne permettent pas de trouver des solutions optimales, l'utilisation des heuristiques permet de trouver des solutions satisfaisantes aux différents problèmes d'optimisation. Une heuristique est un algorithme qui fournit rapidement une solution réalisable, pas nécessairement optimale, pour un problème d'optimisation NP-difficile. Une heuristique, ou méthode approximative, est donc le contraire d'un algorithme exact qui trouve une solution optimale pour un problème donné. Les algorithmes de résolution exacts étant de complexité exponentielle, il est généralement plus judicieux de faire appel à des méthodes heuristiques pour des problèmes difficiles. Généralement une heuristique est conçue pour un

problème particulier, en s'appuyant sur sa structure propre, mais les approches peuvent contenir des principes plus généraux.

Gaouar et al., [Gaouar 2005, 2006] ont développé une heuristique de stockage/déstockage pour réduire ce temps moyen de déstockage. L'heuristique a été développée de manière à stocker les produits de mêmes références dans le même casier au plus près de la station de livraison. Cette heuristique a permis une grande amélioration du temps moyen de déstockage.

## 8. Critères de performances d'un AS/RS

Plusieurs mesures de performance peuvent être appliquées pour évaluer un AS/RS. Les plus importantes sont : le taux d'utilisation de la machine S/R, le temps de cycle figure parmi les paramètres les plus importants d'un AS/RS étant donné qu'il influe directement sur les performances du système global et l'espace de stockage utilisé.

- **Le taux d'utilisation de la machine S/R** : il correspond au nombre de demandes de stockage ou de requêtes de déstockage traitées sur une période de temps.

- **Le temps de cycle de la machine S/R** : Temps moyen nécessaire pour servir une requête de stockage ou de déstockage ou bien temps qui s'écoule depuis l'apparition de la requête, jusqu'à la fin de son exécution.

Le temps de cycle comprend le temps d'attente si la requête n'est pas satisfaite au moment de son apparition, le temps de déplacement de la machine S/R de la station dépôt/livraison vers le casier de stockage/déstockage, et le temps de déplacement du casier de stockage/déstockage vers la station de dépôt/livraison.

- **Taux de requêtes satisfaites par unité de temps** : il dépend du temps de cycle moyen mais aussi des méthodologies de stockage et des techniques de pilotage du système.

- **Espace de stockage utilisé** : l'espace de stockage utilisé dans les AS/RS ou la quantité de produits qu'on peut stocker (par unité de volume) dans un AS/RS dépend du type de ce dernier.

## 9. Etat de l'art

Dans la littérature, les principales propriétés relatives aux AS/RS étudiées sont : la modélisation mathématique du temps de cycle de la machine S/R, l'optimisation des dimensions des AS/RS et du temps de cycle, les différentes méthodologies de stockage

appliquées dans ces systèmes, conception, le positionnement du point de repos de la machine S/R, l'ordonnancement des opérations de stockage/déstockage.

### 9.1 Temps de cycle de la machine S/R

Dans plusieurs travaux, les auteurs se sont intéressés aux mesures de performance permettant d'évaluer les AS/RS et principalement au temps de cycle de la machine de stockage/déstockage S/R. On retrouve généralement la modélisation mathématique du temps de cycle et l'optimisation de ce critère.

Les AS/RS à charge unitaire sont les plus étudiés. Hausman et al. [**Hausman 1976**] ont été parmi les premiers à développer un modèle analytique pour l'étude des AS/RS à charge unitaire. Bozer et white [**Bozer 1984**] ont utilisé une approximation continue du rack de stockage pour développer une expression mathématique permettant de calculer le temps de simple et de double cycle. Les AS/RS considérés sont à charge unitaire, le rack est rectangulaire et la stratégie de stockage est aléatoire. Par ailleurs, ils ont étudié plusieurs positions du point d'entrée/sortie. Dans [**Hwang 1988**] les auteurs ont proposé une expression mathématique pour les AS/RS multi allées, dans ce travail, chaque rack est considéré comme une face continue et pour chaque rack, ils ont développé une expression analytique du temps de cycle. Hwang et Lee, [**Hwang 1990**] ont reproché aux travaux cités précédemment le fait qu'ils considèrent une vitesse uniforme pour la machine S/R et ont affirmé que toute conception se basant sur ces modèles est loin d'être optimale. Ils ont donc proposé un modèle tenant compte des caractéristiques opérationnelles de la machine S/R, à savoir l'accélération, la décélération et la vitesse maximale. Sari [**Sari 1998**] a établie les expressions analytiques discrètes pour les AS/RS multi allées et pour les AS/RS à racks glissants. Lehrer et al. [**Lehrer 2010**] ont proposé des modèles analytiques pour le calcul des temps de simple et double cycle pour un AS/RS à charge unitaire à double profondeur. Egbelu [**Egbelu 1991**] a développé des formulations pour minimiser le temps de cycle et les temps de réponse maximum. Dans deux travaux différents, Guezzan et al. [**Guezzan 2011a, 2011b**], ont développé deux modèles analytiques continus permettant l'estimation du temps moyen de simple cycle pour les AS/RS à racks glissants. Dans le premier modèle, les auteurs considèrent le cas où le temps de glissement des racks est inférieur au temps de déplacement sur les allées et dans le deuxième modèle ils ont considéré le cas où le temps de glissement des racks est supérieur au temps de déplacement sur les allées. Ghomri et al. [**Ghomri 2008**] ont proposé une expression analytique du temps moyen de simple cycle d'un AS/RS multi allées en considérant que l'ensemble des racks représentent un espace continu. Les auteurs

dans [Le-Duc 2004], [Le-Duc 2005], [De Koster 2007], ainsi que [Roodbergen 2001] ont estimé le temps de cycle d'un déstockage par lots, c'est-à-dire regroupant plusieurs demandes de déstockage en une livraison afin de réduire le temps de réponse moyen. Lerher et al. [Lerher 2005] ont proposé un modèle analytique du temps de cycle d'un AS/RS multi allées dans lequel, les auteurs ont pris en considération l'accélération et la décélération de la machine S/R. Sari [Sari 2003] et Sari et al. [Sari 2005] ont développé l'expression du temps de cycle moyen d'un AS/RS à convoyeur gravitationnel pour quatre positions différentes du convoyeur de restockage et des stations de dépôt/livraison. Une étude comparative entre l'AS/RS à convoyeur gravitationnel et l'AS/RS à charge unitaire a ensuite été établie. Dans [Sari 2010] les auteurs ont développé une expression analytique pour le temps de cycle d'un nouveau type d'AS/RS composé d'une seule machine de stockage/déstockage (S/R) et de convoyeurs gravitationnels. Les auteurs dans [Gaouar 2004] se sont intéressés à l'AS/R à convoyeur gravitationnel. Dans le but de réduire le temps moyen de déstockage, et d'améliorer les performances du système de stockage, ils ont développé une heuristique de stockage/déstockage permettant de réduire le temps de déstockage. L'heuristique a été développée de manière à stocker les produits de mêmes références dans le même casier au plus près de la station de livraison.

## 9.2 Optimisation du temps de cycle et des dimensions de l'AS/RS

L'optimisation du temps de cycle et des dimensions des AS/RS a été traité dans quelques travaux, nous citons quelque uns : Le-Duc et al. [Le-Duc 2006] ont considéré un système automatisé de stockage/déstockage tridimensionnel compact nouvellement conçu. Le système se compose d'un pont automatisé prenant en considération les mouvements dans le sens horizontal et dans le sens vertical. Un mécanisme de transport prend soin du mouvement en profondeur. Les auteurs ont optimisé les dimensions de l'AS/RS en minimisant le temps de cycle. Ils ont estimé le temps de déplacement de la machine S/R en simple cycle. A partir du temps estimé, ils ont calculé le rapport optimal entre les trois dimensions qui minimise le temps de parcours. La méthodologie de stockage utilisée est aléatoire. Ensuite ils ont déduis une expression mathématique pour le temps de double cycle. Rosenblatt et al. [Rosenblatt 1993] ont considéré deux aspects des AS/RS jusque là étudiés séparément ; l'optimisation du coût et le comportement dynamique du système. A l'aide d'une heuristique récursive comportant une optimisation et une simulation, ils ont déterminé les caractéristiques physiques de l'AS/RS. La relation entre les dimension du rack et la capacité de la machine S/R qui pourrait affecter sa performance n'a pas été prise en compte dans leur modèle. Kouloughli et al. [Kouloughli 2010] ont optimisé les dimensions d'un AS/RS multi allées en

minimisant le temps de simple cycle de la machine S/R. Les expressions analytiques optimisées ont été développées par Ghomri et al. [**Ghomri 2008**]. Dans [**Kouloughli 2015**], les auteurs ont considéré une contrainte supplémentaire au problème d'optimisation initial du système multi allées. Ils ont supposé que l'une des trois dimensions du système était fixe, et ils ont déterminé les deux dimensions optimales restantes. Dans un autre travail [**Kouloughli 2013**], l'optimisation des dimensions de l'AS/RS à racks glissants pour un temps de simple cycle minimal a été réalisée. Dans [**Azzouz 2001a**], les auteurs ont appliqué la méthode Branch and Bound à l'optimisation des dimensions d'un AS/RS à convoyeur gravitationnel, ils ont défini l'optimalité des dimensions du rack en vue de minimiser le temps moyen de déstockage. Dans un autre travail de Azzouz [**Azzouz 2001b**] une synthèse sur l'optimisation des dimensions d'un AS/RS à convoyeur gravitationnel a été faite.

### 9.3 Méthodologie de stockage

Le choix de la stratégie de stockage des produits dans un AS/RS est très important car il influe sur la minimisation de plusieurs paramètres comme : le temps de déstockage, le temps de cycle de la machine S/R, l'espace de stockage,...

Hausman et al. [**Hausman 1976**] ont traité le problème de l'assignement optimal de stockage en utilisant les trois méthodologies : le stockage aléatoire, le stockage dédié et le stockage par classe. Une réduction importante du temps de cycle est obtenue grâce à l'utilisation du stockage dédié par rapport au stockage aléatoire. Graves et al. [**Graves 1977**] ont montré que la forme en L des différentes régions, pour un stockage par classes, n'est pas nécessairement optimale pour un rack carré. Dans [**Rosenblatt 1989**] et [**Rosenblatt 1994**] les auteurs ont cherché à déterminer le nombre de classes optimal pour un AS/RS en utilisant la méthode de Hausman [**Hausman 1976**] et concluent qu'il est préférable d'utiliser des racks avec un facteur de forme  $b \geq 0.6$  et moins de 7 classes. Eynan [**Eynan 1994**] a choisi la méthode de Hausman et al. [**Hausman 1976**] pour déterminer le nombre de casiers à dédier par classe, Cette méthode a été développée pour un rack à charge unitaire. La position des classes est assez particulière elle est limitée par des arcs de cercles autour de la station de livraison. Grace à la profondeur du rack, la disposition des classes est adaptée de façon à prendre en considération la position du couloir de restockage. Caron et al. [**Caron 2000**] proposent un modèle de temps de cycle pour un entrepôt à 2-block avec un stockage COI (Cube Order Index) par classes. Dallari et al. [**Dallari 2000**] ont appliqué les différentes méthodologies de stockage, ils ont fait varier le nombre d'opérateurs et l'algorithme d'ordonnancement utilisé pour estimer le temps de cycle dans un AS/RS à personne à bord.

Les auteurs dans [Ashayeri 2002] ont proposé une approche géométrique pour modéliser le temps de cycle dans un AS/RS utilisant le stockage par classe. Dans [Kouvelis 1995], les auteurs considèrent un AS/RS avec deux classes. Ils ont développé des relations explicites pour les limites optimales des régions ainsi que pour le temps de transport du simple cycle pour des racks optimisés. Mansuri [Mansuri 1991] a développé un modèle par simulation capable d'analyser et d'évaluer un type de stockage. Le stockage aléatoire, dédié et une combinaison des deux ont été étudiés. Lee [Lee 1992] a étudié le stockage dédié dans un AS/RS à personne embarquée, il a développé une heuristique pour la disposition des espaces de stockage en se basant sur la technologie de groupe. Guenov et al. [Guenov 1992] comparent trois configurations de la forme des zones dans un AS/RS. Ils concluent que la configuration des zones pour le stockage par classes dépend de la position du point d'entrée/sortie du rack pour un AS/RS à charge unitaire. Les auteurs dans [Eynan 1994] ont présenté une procédure d'utilisation du stockage par classes. Trois types de classes sont considérés, et leurs temps de cycle correspondants sont modélisés. Ils ont montré que les meilleures performances sont obtenues avec un petit nombre de classes. Van Den Berg et al. [Van Den Berg 2000] ont considéré une grande variété de règles de contrôle et de règles de stockage pour estimer le temps de cycle. Pour le stockage à classes, un nouvel algorithme, permettant de jouer entre l'espace de stockage nécessaire et le temps de transport, a été utilisé. Les auteurs dans [Le-Duc 2005], [Le-Duc 2007] ainsi que [Roodbergen 2001] ont estimé le temps de cycle d'un déstockage par lots c'est-à-dire regroupant plusieurs demandes de déstockage en une seule livraison afin de réduire le temps de réponse moyen. Dans [De Koster 2008] et [Yu 2008] les auteurs ont étudié un système à 3D intégré avec convoyeurs circulaires : le système est entièrement automatisé, et chaque palette stockée dans un rack multi profondeur est individuellement accessible sans l'aide d'un convoyeur tampon. Ils ont étudié le stockage aléatoire, le stockage à base de rotation, pour la conception d'un support optimal en minimisant le temps de déplacement de la machine S/R. Ils ont conclu que le rapport optimal entre les trois dimensions dans les directions verticale, horizontale et convoyeur est respectivement de 0,72: 0,72: 1 pour les commandes simple cycle et pour un stockage aléatoire.

#### 9.4 Conception des AS/RS

La conception des AS/RS a intéressé beaucoup de chercheurs :

Bozer et al. [Bozer 1980] ont développé une méthodologie de conception basée sur des techniques analytiques. Rosenblatt et al. [Rosenblatt 1993] ont considéré deux aspects

des AS/RS l'optimisation du coût et le comportement dynamique du système. L'heuristique d'optimisation/simulation développée a permis de déterminer les paramètres de conception nécessaire à un certain niveau de performance. L'auteur dans [Derks 1993] a considéré la conception et réalisation d'un AS/RS en suivant les étapes essentielles dans la construction des systèmes et la notion de hiérarchie dans les le but de montrer que l'approche holistique dans la conception des systèmes est beaucoup plus performante que l'approche réductionniste conventionnelle. Trevino et al. [Trevino 1995] ont développé une procédure de conception des AS/RS à carrousels. Cette procédure est basée sur la satisfaction de trois aspects : la capacité de stockage, les contraintes d'espace et le rendement tout en minimisant le coût. Lee et al. [Lee 1996] ont simulé un AS/RS à allées étroites dans le but d'examiner la logique de fonctionnement du système. Ce qui permet de déterminer le nombre optimal de machines de stockage/déstockage pour un rendement maximal. Dans [Van den Berg 1996], l'auteur propose un programme dynamique qui assigne à chaque référence de produits sa classe. Chew et al. [Chew 1999] ont estimé le cycle moyen dans un entrepôt utilisant un stockage par classes. Dans [Malmberg 2000], les auteurs ont considéré la conception et modélisation d'AS/RS à double navette. Ils ont déterminé la distribution de la probabilité de l'état de la queue de transactions de stockage et déstockage en utilisant des valeurs estimées des temps de cycles. Dans un autre travail [Malmberg 2001] ils ont présenté une nouvelle règle d'évaluation de la configuration des racks pour AS/RS. Cette nouvelle règle permet d'éviter de faire deux hypothèses nécessaires pour les règles précédentes : la proportion de simple et double cycle ainsi que la capacité de stockage lorsque le stockage aléatoire est considéré. Eldemi et al. [Eldemi 2004] ont établie un modèle mathématique permettant d'estimer le temps de cycle et l'espace de stockage pour les différentes méthodologies de stockage : aléatoire, dédié et par classe. Grâce à ce modèle ils ont pu conclure que le stockage par classe utilise moins d'espace et moins de temps de stockage et de déstockage en comparaison avec les deux autres méthodologies. Lee [Lee 2005] propose d'abord un AS/RS avec rack à cellules modulaires en premier, ensuite, ils mettent en valeur la flexibilité et l'utilisation de l'espace de stockage ; ils comparent un nouveau modèle réalisé au modèle existant à travers des exemples numériques. Park [Park 2006] a déterminé la moyenne et la variance des temps de déplacement des commandes simple et double cycle pour les systèmes AS/RS pour les cas non carré dans le temps avec deux classes de stockage, et a analysé l'influence des facteurs de rack et des paramètres d'asymétrie sur le rendement du système. Gue et al. [Gue 2007] ont étudié les systèmes de stockage de très haute densité où les unités de charges sont stockées en multi-profondeur. Le système étudié permet un stockage en unités de charges qui ne sont pas



accessibles individuellement. Muppani et al. [Muppani 2008] ont développé un algorithme de type Branch & Bound permettant de comparer le stockage par classes au stockage dédié, en considérant l'espace de stockage et son coût, le coût de manutention et de préparation de commande pour un stockage par classes. L'approche présentée a permis de démontrer qu'il existe des économies importantes en utilisant le stockage par classes, bien que le stockage dédié et le stockage aléatoire soient couramment utilisés dans la planification de stockage en entrepôt. De Koster et al. [De Koster 2008] ont considéré une nouvelle configuration optimale pour un AS/RS 3-D en appliquant un stockage aléatoire. Le but était d'analyser les performances et la dimension optimale du système afin d'améliorer le temps de cycle. Sari et al. [Sari 2010] étudient un système 3D, où les palettes sont stockées et récupérées à différents côtés du rack par des machines S/R respectivement utilisées pour le stockage et la récupération. Les palettes sont stockées à profondeurs multiples. Afin de récupérer une palette particulière à l'intérieur du rack, la machine S/R de déstockage doit déplacer l'ensemble des palettes en face de celui-ci et les stocker sur un convoyeur tampon spéciale utilisant la règle FIFO (premier entré, premier sorti).

### 9.5 Point de repos de la machine S/R

Le problème de la position du point de repos de la machine de stockage déstockage (machine S/R) a été largement étudié par beaucoup de chercheurs, nous présentons les principaux travaux dans le domaine :

Bozer et al. [Bozer 1984] ont suggéré plusieurs règles statiques de positionnement du point de repos, mais ils n'ont fourni aucune comparaison quantitative de leurs performances. Egbelu [Egbelu 1991] a utilisé la programmation linéaire pour étudier le problème de positionnement du point de repos de la machine S/R pour minimiser le temps de cycle. Hwang et al. [Hwang 1993] ont utilisé les règles développées par Egbelu [Egbelu 1991] et les ont transformés en un problème de positionnement unique en utilisant les distances de Tchebyshev pour la première règle et le minimax de Tchebyshev pour la seconde. Dans [Egbelu 1993], les auteurs ont utilisé la simulation pour comparer les règles de positionnement du point de repos. Ha et al. [Ha 1994] ont montré que le stockage par classes à 2 classes réduit le temps de stockage et le temps de déstockage par rapport à un stockage aléatoire pour une simple et double commande pour système de stockage à carrousel. Peters et al. [Peters 1996] ont développé une expression analytique pour la position du point de repos sous une variété de configurations d'AS/RS concluant que le meilleur point de repos de la machine S/R se trouve au milieu du rack. Chang et al. [Chang 1997] ont proposé un

algorithme permettant de déterminer le point de repos optimal dans le but de minimiser le temps de cycle sur l'ensemble des opérations de stockage et de déstockage. Park [**Park 2001**] a déterminé la position du point de repos optimale pour des racks à distribution non uniforme. L'auteur a déterminé un modèle analytique basé sur la probabilité que la prochaine opération soit un stockage ou un déstockage. Pour revenir au point de repos après une opération, différentes alternatives ont été étudiées pour un fonctionnement optimal de la machine S/R. Van den Berg [**Van den Berg 2002**] a développé des modèles analytiques pour déterminer les points de repos de la machine S/R sous deux stratégies de stockage : aléatoire et par classe.

## **9.6 Ordonnancement des opérations de stockage et de déstockage**

Les problèmes d'ordonnancement des opérations de stockage ou de déstockage, ont trouvé un écho très favorable de la part des chercheurs. Les travaux dans ce domaine sont très nombreux nous citons quelque uns :

Hwang et al. [**Hwang 1993**] ont développé une heuristique appliquée dans un AS/RS à personne embarquée pour résoudre le problème d'ordonnancement dans requêtes de stockage et de déstockage. Randhawa et al. [**Randhawa 1995**] ont simulé, en utilisant SIMAN, six différentes configurations d'AS/RS à charge unitaire combinées à trois règles d'ordonnancement. Les résultats ont été comparés selon différents critères, le taux de charge de l'AS/RS étant le principal. Lee et al. [**Lee 1995**] ont considéré un AS/RS à charge unitaire pour lequel ils ont étudié l'ordonnancement des opérations de stockage et de déstockage en respectant la contrainte du juste à temps. Quatre heuristiques ont été développées basées sur deux étapes, la constitution des doubles cycles puis leur ordonnancement optimisé. Une solution optimale du problème fut proposée en utilisant la programmation en nombres entiers. Elsayed et al. [**Elsayed 1996**] ont étudié les ordres de déstockage ayant des dates spécifiées dans un AS/RS. Ils ont développé des règles permettant la combinaison des ordres de stockage et déstockage en double cycle ainsi que l'ordonnancement de ces ordres de tel sorte que le retard global des ordres de déstockage soit minimisé. Lee et al. [**Lee 1996**] ont considéré le problème d'ordonnancement des demandes de déstockage dans un AS/RS à charge unitaire. Ils ont développé un algorithme permettant de séquencer les demandes en double cycle afin de minimiser les temps de transport. Bozer et al. [**Bozer 1996**] ont déterminé le nombre minimum d'opérateurs en bout d'allée dans un AS/RS à mini charge. L'algorithme calculant ce nombre est basé sur un modèle analytique estimant le taux d'utilisation de l'opérateur. Les auteurs dans [**Mahajan 1998**] ont considéré un AS/RS à mini

charge pour lequel ils traitent le problème d'ordonnement suivant : les demandes de déstockage actuelles deviennent demandes futures de stockage, puisque les charges sont retournées dans le système après leur utilisation. Par conséquent, le problème peut être reformulé de sorte que seule une file d'attente de demandes de déstockage existe et se traduit par un problème moins complexe. Les demandes de déstockage sont réarrangées de telle sorte que les demandes successives sont situées les unes à côté des autres. Le stockage et le déstockage qui sont proches l'un de l'autre peuvent être couplé par une heuristique du plus proche voisin. Van Den Berg et al. [Van Den Berg 1999] ont déterminé le routage optimal en appliquant une stratégie de stockage dédié dans un AS/RS. Le problème d'ordonnement d'une liste de demande de déstockage, est similaire à celui du voyageur de commerce et donc NP-complet. Ce problème a été réduit à un problème d'ordonnement en stockage dédié pour être facilement résolu. Dans [Hwang 1999], les auteurs ont étudié le problème de d'ordonnement des opérations de stockage et de déstockage pour les AS/RS à carrousel, en considérant les demandes simple cycle et les demandes réalisées en double cycle (Une heuristique du plus proche voisin).

## **10. Conclusion**

Dans ce chapitre nous nous sommes intéressés aux systèmes automatisés de stockage/déstockage, nous avons traité les différents aspects de ces systèmes comme leurs Fonctions, leurs composants, les différents types d'AS/RS, l'adressage des produits stockés, les principaux critères de choix des emplacements, le stockage selon la nature du produit, les différentes méthodologies de stockage ou de déstockage ainsi que les plus importants critères de performances servant à évaluer ces systèmes.

Enfin un état de l'art détaillant les principaux travaux concernant les AS/RS à été étalé. L'intérêt que portent les chercheurs aux systèmes de stockage a été illustré. Cet intérêt est justifié par le fait que les AS/RS ont une importance capitale dans les systèmes de production.

# **Chapitre 3 Première contribution : Optimisation du temps de déstockage dans un AS/RS à convoyeur gravitationnel se basant sur un modèle Heuristique et Agent**

## **1. Introduction**

La mise en place d'une solution de stockage automatisé permet d'actionner plusieurs facteurs d'amélioration : diminution du nombre de références, souplesse d'approvisionnement et rapidité de transmission des informations. Mais leur mise en œuvre prend du temps et grand nombre d'entreprises n'ont pas forcément le pouvoir de négociation pour y parvenir aisément. Il existe des solutions agiles qui réduisent les coûts d'exploitation, qui conduisent à la souplesse et la modularité de l'unité de stockage mais aussi qui entraînent une forte réactivité des opérations de stockage et déstockage.

Dans ce qui précède, nous avons dressé une étude bibliographique exhaustive concernant les systèmes automatisés de stockage et déstockage, l'importance d'optimiser l'espace de stockage dans un magasin et les méthodes de résolution qui leur ont été dédiées. Nous avons conclu par un travail de synthèse et critique de cet état de l'art qui nous a permis de situer notre thèse au sein des travaux existants, de montrer certaines limites et insuffisances quant aux approches de résolution du problème étudié, et en conséquence de justifier nos choix et directives quant à la modélisation, la simulation et la résolution de ce problème. Dans le reste de ce manuscrit nous allons alors nous intéresser aux différentes méthodes et concepts adoptés dans notre modèle proposé : SMASD : Système Multi-Agents de Stockage et Déstockage.

## **2. La simulation Multi-agents**

La simulation multi-agents, en anglais Multi-Agent Based Simulation (MABS). Le terme de « simulation », dans le cadre informatique, recouvre en fait au moins deux pratiques complémentaires. La première correspond à l'acceptation scientifique de ce terme, et consiste à « pratiquer des tests sur un substitut de la situation réelle (le substitut étant sup- posé reproduire fidèlement les caractéristiques pertinentes essentielles à la situation à caractériser)

» [Picault 2013] et ce afin d'évaluer la plausibilité des hypothèses qui sous-tendent un modèle. Comme dans une expérience, il s'agit donc de contrôler les conditions dans lesquelles se déroulent les événements à étudier, afin d'aider à l'évaluation d'un modèle issu d'une autre discipline scientifique.

La seconde correspond à l'usage courant du terme « simulation » : l'animation d'un monde artificiel, proche ou non du monde réel, et dont l'usage peut être ludique (jeu vidéo, cinéma), pédagogique (simulateur de conduite, Serious Game), artistique. [Hutzler 2004]

En pratique, les objectifs visés dans chacune de ces deux grandes catégories peuvent être assez diversifiés : prédiction de l'évolution d'un système, explication de ses mécanismes, découverte de phénomènes inattendus, preuve, entraînement, divertissement, remplacement d'acteurs humains, etc. [Axelrod 1997]

Cela nous amène à proposer la définition suivante pour les simulations multi agents :

La simulation multi-agents est le domaine qui étudie les méthodes et algorithmes, dans le cadre conceptuel et opérationnel des SMA, qui permettent de construire des techniques de représentation et de production des connaissances ainsi que de calcul d'événements, à la fois :

- Fiables, c'est-à-dire sans biais introduits dans les résultats par des choix implicites de modélisation ou d'implémentation ;
- Efficaces en termes de temps d'exécution ou de volume d'information pouvant être traité ;
- Manipulables autant que possible par des thématiciens (experts non informaticiens du domaine auquel s'applique la simulation visée) qui doivent pouvoir paramétrer voire programmer eux-mêmes les outils de simulation ;
- Intelligibles en ce qui concerne la représentation du modèle et l'interprétation des résultats de simulation par ces mêmes thématiciens ;
- Modulaires pour permettre une révision aisée de parties du modèle ;
- Polyvalentes, autrement dit permettant de traiter de la même façon non seulement des simulations scientifiques appliquées à des domaines variés, mais également des simulations destinées à produire des mondes artificiels.

L'une des difficultés de la conception d'une simulation réside dans le fait qu'elle suit un processus complexe au cours duquel le modèle du phénomène établi par le thématicien, dit « modèle thématique » (parfois appelé « modèle non formel »), est d'abord reformulé dans le cadre d'un formalisme donné, pour constituer un « modèle conceptuel » (ou « modèle formel ») [Robinson 2009]. Celui-ci est à son tour transformé en un « modèle computationnel » (ou « modèle exécutable », ou encore « modèle opérationnel ») [Edmonds 2003] qui comporte les spécifications nécessaires à l'implémentation. Enfin ce dernier est traduit en pratique en un

programme informatique au moyen d'un langage de programmation ou au sein d'une plateforme de simulation. Chacune de ces étapes est susceptible de déroger aux exigences que nous avons formulées ci-dessus [Edmonds 2005].

<b>Plateforme</b>	<b>Domaine</b>	<b>License</b>	<b>Langage de programmation</b>	<b>Système d'opération</b>	<b>FIPA compilant</b>	<b>Capacité 3D</b>
Cormas : (Common-pool Resources and Multi-Agent Systems)	Gestion des ressources naturelles, du développement rural et de l'écologie	Libre de modifier, mais de ne pas distribuer la version modifiée	Smalltalk (besoin VisualWorks pour exécuter)	Linux; Macintosh Windows	Inconnu	Inconnue
JADE (Java Agent Development Framework)	Application distribuées composée d'entités autonomes	LGPL version 2	Java	Toute Java Platform	Oui	Inconnue
Madkit (Multi Agent Development Kit)	Plate-forme multiagent avec une couche de simulation agent	LGPL; GPL	Java	JVM (Java 2)	Inconnu	Inconnue
MASON (Multi-Agent Simulator Of Neighborhoods)	Complexité sociale, la modélisation physique, modélisation abstraite	License académique gratuite (open source)	Java	Toute java platform	Inconnu	Oui
MASS (Multi-Agent Simulation Suite)	simulations distribuées, simulations participatives	Propriétaire, version gratuite disponible	Java; Exécuter des simulations avec Repast	Tout système d'exploitation avec	Non	A partir de Java, toute visualisation est peut-être ajoutée

			et NetLogo	Java 1.5		
RepastS (REcursive PorousAge nt Simulation Toolkit Symphony	Sciences sociales	BSD	Java	Java version 1.4	Inconnu	Oui
SeSAM (Shell for Simulated Agent Systems)	Recherche, l'enseigne ment, les ressources, la théorie de graphe	LGPL	Simulation compilées à partir de la spécification visuelle	Java 5.0 ou supérieur	Plugin disponib le	Plugin disponible
Netlogo	l'ingénierie informatiqu e, physique ,chimie, mathématiq ue ...	Gratuit Pas open source	Netlogo	Tout Java Virtual Machine	Inconnu	Oui
GAMA	sciences sociales et naturelle	Logiciel gratuit LGPL	Java	Tout Java Virtual Machine	Oui	Oui

**Tableau 3 Plateformes de simulation Multi-agents**

### 3. Le simulateur Netlogo

#### 3.1 Définition

NetLogo est un environnement de modélisation programmable permettant de simuler des phénomènes naturels et sociaux. Il a été créé par Uri Wilenski en 1999 et son développement est poursuivi de manière continue par le Center for Connected Learning and Computer-Based Modeling.

NetLogo convient tout particulièrement à la modélisation de systèmes complexes évoluant au cours du temps. Les « modélisateurs » peuvent donner des instructions à des centaines ou des milliers d'« agents » opérant indépendamment les uns des autres. Ce qui

permet d'explorer les liens entre les comportements des individus à leur niveau et les schémas généraux (comportements de groupe ou de masse) qui émergent des interactions entre de nombreux individus.

Ces simulations couvrent de nombreux domaines des sciences naturelles et sociales, y compris la biologie et la médecine, la physique et la chimie, les mathématiques et l'ingénierie informatique ainsi que l'économie et la psychologie sociale. De nombreuses sessions d'apprentissage à base de modèles utilisant NetLogo sont actuellement en cours de développement.

NetLogo offre aussi un outil puissant de simulations participatives au niveau de la classe appelé HubNet.

Il représente la nouvelle génération de toute une série de langages de modélisations multi-agents qui a débuté avec StarLogo. Il a été construit à partir des bases fournies par le logiciel StarLogoT auquel il apporte toute une série de nouvelles fonctionnalités significatives ainsi qu'un langage et une interface utilisateur entièrement remaniés. NetLogo étant écrit en Java, il peut tourner sur tous les systèmes majeurs (Mac, Windows, Linux et autres). Il fonctionne en tant qu'application indépendante. Les modèles peuvent même être sauvegardés sous forme d'applets Java et tourner dans tous les navigateurs internet modernes.

### **3.2 Fonctionnalités**

Netlogo offre les fonctionnalités suivantes :

#### **Système :**

Multi-plateforme: tourne sur Mac, Windows, Linux, et autres

#### **Langage :**

Entièrement programmable

Structure du langage simple

Dialecte Logo étendu pour supporter les agents

Agents mobiles (les tortues) se déplaçant sur une grille formée d'agents stationnaires (les patches)

Création de liens entre les tortues pour former des agrégats, des réseaux et des graphes

Important vocabulaire de primitives intégrées au langage

Calculs en virgule flottante double précision (IEEE 754)

Fonctionnement absolument identique sur toutes les plateformes



**Environnement :**

Observation des modèles en 2D ou en 3D (Figure 8)

Formes vectorielles redimensionnables et pivotantes

Étiquetage des tortues et des patches

Centre de commande pour un pilotage interactif (en direct) de la simulation

Interface pouvant recevoir des boutons, des curseurs, des commutateurs, des traceurs de courbes, des sélecteurs, des moniteurs, des boîtes de texte, des notes, des zones de sortie

Variateur de vitesse pour accélérer ou ralentir la simulation

Système de sortie graphique puissant et souple

Panneau d'informations pour annoter les modèles

HubNet: simulation participative utilisant des machines en réseau

Moniteurs d'agent pour inspecter et contrôler les agents

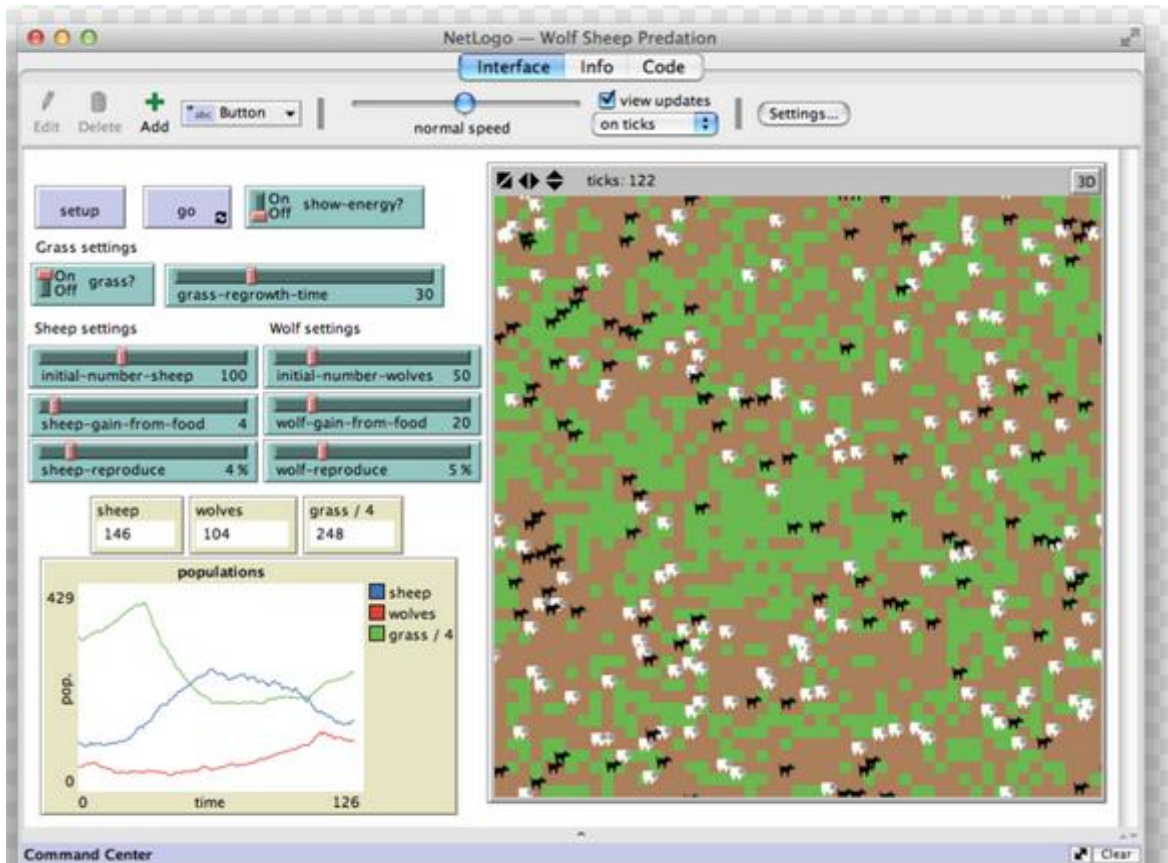
Fonctions d'exportation et d'importation (exportation des données, sauvegarde et restauration de l'état d'un modèle, création et enregistrement d'une animation)

Outils BehaviourSpace (espace de compostements) utilisés pour collecter des données provenant de plusieurs sessions de simulations

Modélisation de systèmes dynamiques

**Web:**

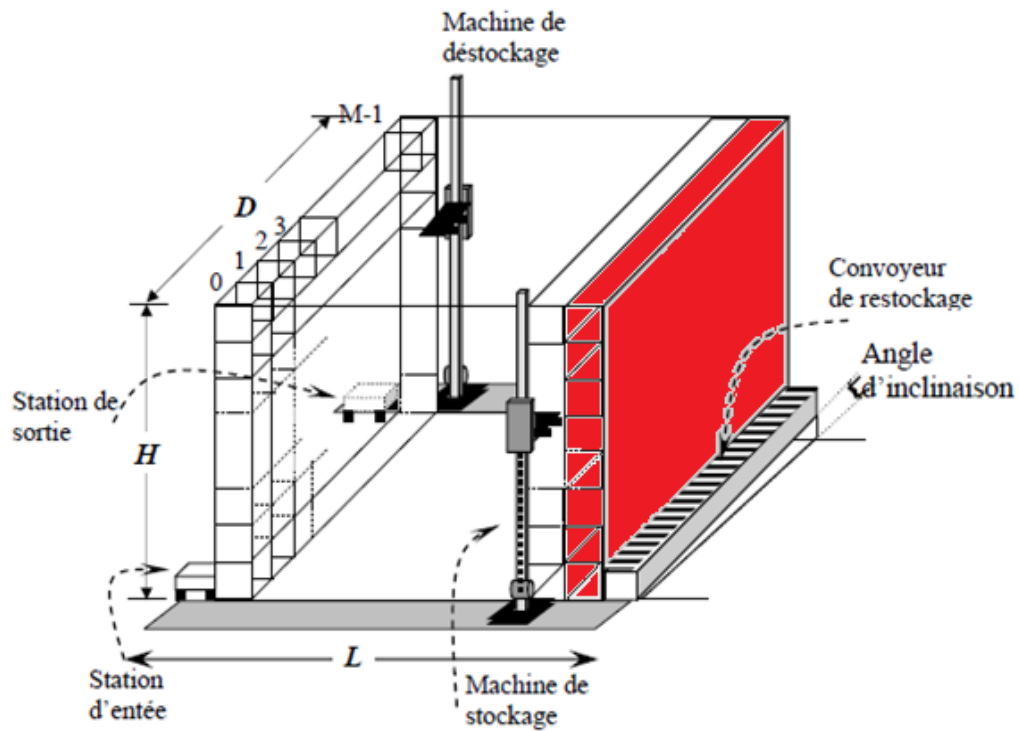
Modèles enregistrables sous forme d'applets pouvant ensuite être intégrés dans des pages web (certaines fonctionnalités ne sont pas disponibles dans les applets, telles que certaines extensions et la visualisation 3D)



**Figure 8 Interface utilisateur avec le modèle Wolf-Sheep Predation.**

#### **4. Description du modèle SMASD**

Dans notre système nous nous contentons d'étudier le stockage et le déstockage des produits dans un rack à une seule rangée de casiers ,ce qui signifie que la longueur du rack  $L$  sera égale à 1(Figure9), nous avons fait ce choix car le même scénario se répète pour les autres rangées. Pour la hauteur du rack  $H$  et la profondeur  $D$  nous choisissons les valeurs arbitraires respectives: 35 et 46. Nous précisons que notre programme permet de modifier facilement les valeurs de ces deux dernières dimensions suivant la configuration étudiée du rack.

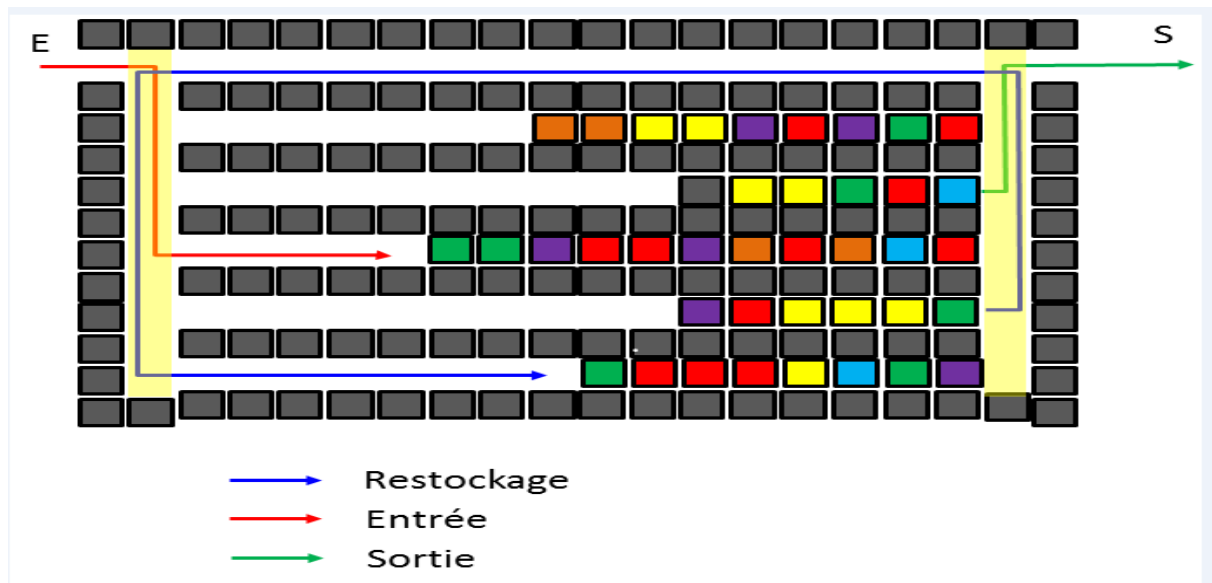


**Figure 9 L'AS/RS à convoyeur gravitationnel dans le model SMASD**

Le modèle SMASD est composé d'un rack formé de 19 casiers ; 18 pour stocker les produits et 1 casier consacré au restockage.

A chaque type de produit a été attribuée une couleur ; Dans notre modèle nous avons utilisé 140 couleurs.

Notre programme offre la possibilité de pouvoir modifier à tout moment les valeurs prises précédemment.

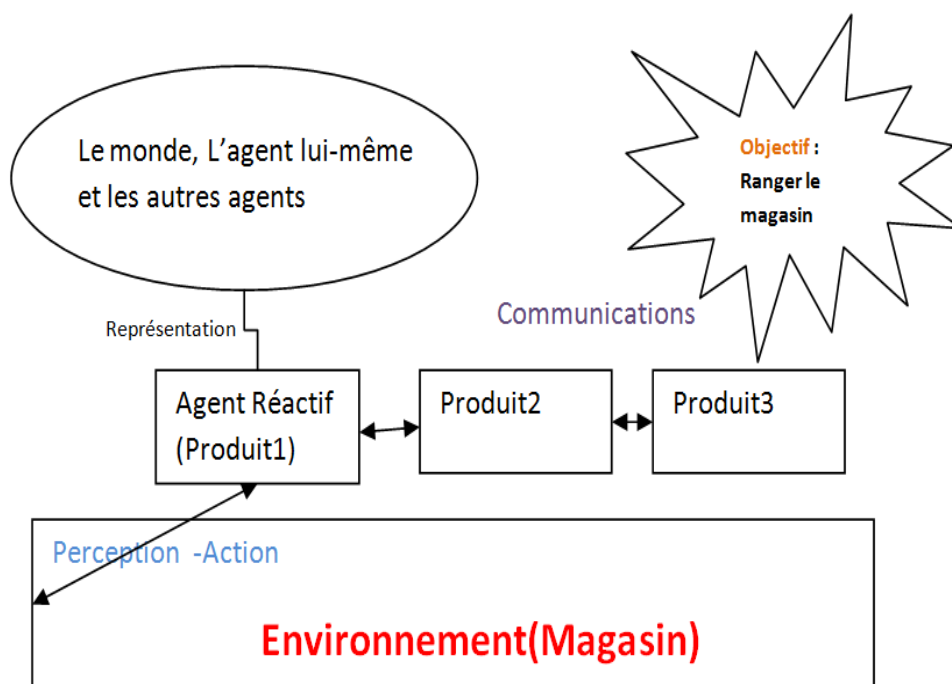


**Figure 10 Principe de fonctionnement du modèle SMAS**

Notre système a été perçu comme étant un système multi agents réactifs ou il existe des agents produits (agents tortues sous Netlogo) qui se déplacent suivant la règle « Si action alors Réaction » ; en communiquant leur distances par rapport à la station de sortie (qui est considérée comme un agent patch sous Netlogo). Ceci dans le but de minimiser au maximum le temps de déstockage des produits (Figure 11).

Les agents produits fonctionnent suivant la règle :

**Si < état interne > et < état de perception > alors < action >**



**Figure 11 Représentation du SMA**

## 4.1 Les principales actions dans SMASD

La procédure Preparer dans la figure qui suit permet d'effacer tous les agents turtles qui existent ;Init permet de mettre le rack dans l'état initial adéquat .

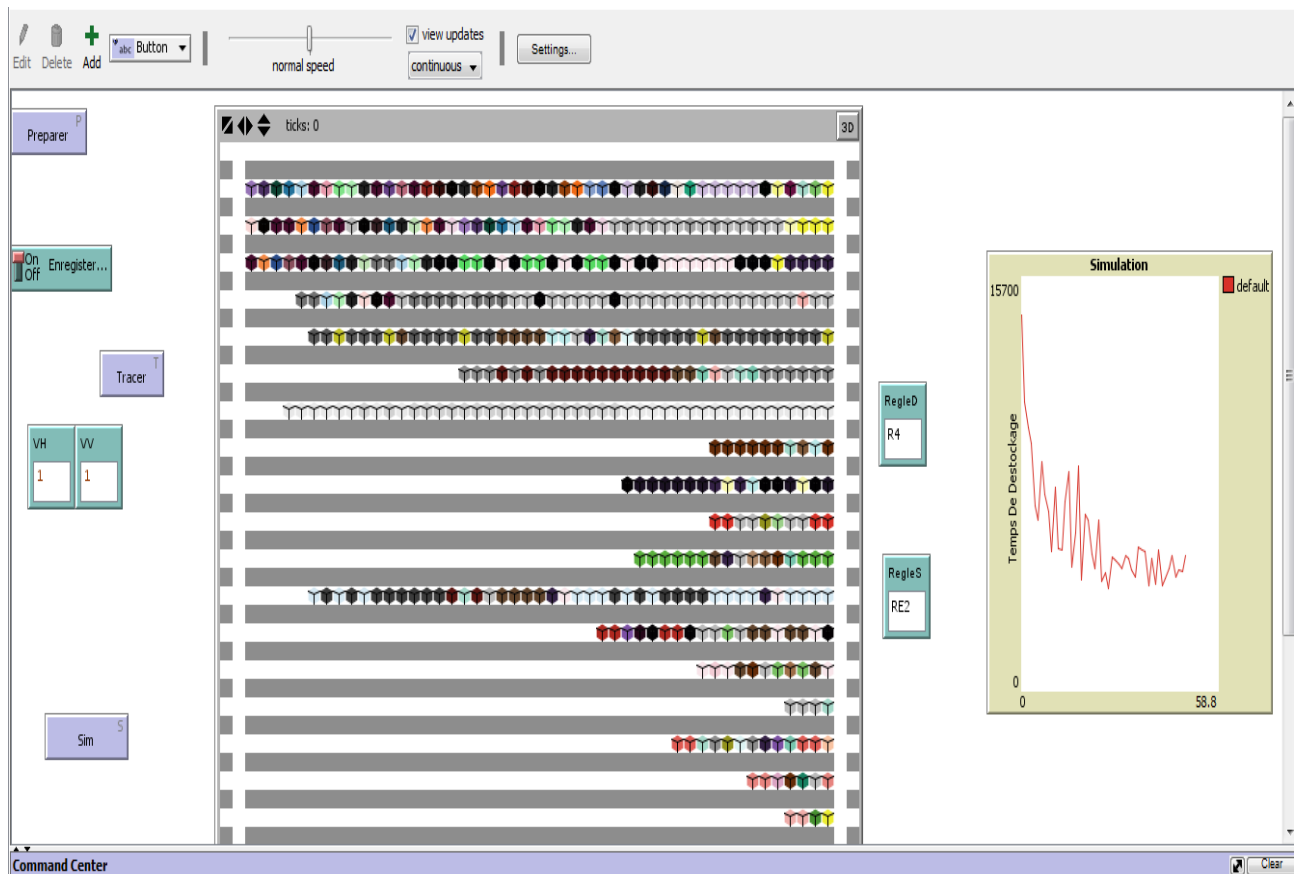
Si par exemple nous mettons EnregisterEtatMagasin à True et que nous faisons des actions de stockage et/ou de déstockage ; si nous appuyons sur Init c'est le dernier état du magasin qui sera mis en place comme état initial.

L'input box « RegleS » nous permettra de choisir parmi les règles de stockage possibles celle que l'on souhaite appliquer ; de la même manière l'input box « RegleD » nous permettra de choisir parmi les règles de déstockage possibles celle que l'on souhaite appliquer.

En combinant deux de ces règles la ; nous lancerons la simulation en faisant appel à la procédure Sim. Cela signifie qu'on aura comme entrée un fichier texte que nous avons appelé « D.txt » qui contient la liste des couleurs des produits que l'on souhaite déstocker.

Suivant la règle « RegleD », nous déstockerons cette liste de produits et par la suite nous restockerons des produits de même type (de même couleur) suivant la règle de stockage « RegleS » ,afin d'avoir une plus importante variation dans les types de produits .

Nous calculerons le temps nécessaire pour déstocker cette liste de produits et puis nous arrangerons l'état du magasin en restockant des produits de même type à chaque fois (figure12) ; nous vérifierons si au bout d'un certain temps (que nous fixons) le temps de déstockage diminue au fur et à mesure qu'on range le magasin selon la règle de stockage RegleS.



**Figure 12 Interface principale du modèle SMASD**

## 4.2 Les règles de stockage

Dans notre programme nous avons défini trois règles de stockage.

### 4.2.1 La règle de stockage RE1 :

RE1 correspond au stockage aléatoire ; d'où l'utilisation de la fonction prédéfinie de Netlogo (random), elle retourne un nombre aléatoire entier compris entre 0 et 17.

### 4.2.2 La règle de stockage RE2 :

Cette règle permet de stocker un produit dans le casier qui contient au minimum un produit du même type, sinon on stocke dans un casier vide.

### 4.2.3 La règle de stockage RE3:

Cette règle cherche le casier qui contient le minimum de produits du même type que celui que l'on souhaite stocker ; le but étant d'avoir la plus grande variation de produits dans chaque casier.

### **4.3 Les règles de déstockage**

#### **4.3.1 La règle de déstockage R1 :**

Cette règle teste s'il existe un produit de la même couleur que celui que l'on souhaite déstocker dans la première couche ; s'il en existe plusieurs elle choisit le plus proche de la station de déstockage ; s'il n'en existe aucun elle passe à la seconde couche ...

#### **4.3.2 La règle de déstockage R2 :**

Dans cette règle nous avons choisi de découper notre rack en quarts carrés ; nous parcourons le premier, si on ne trouve pas le bon produit à déstocker dans le premier carré ; on passe au second...

#### **4.3.3 La règle de déstockage R3 :**

Nous recherchons tout d'abord tous les produits du même type que celui que l'on souhaite déstocker ; nous calculons juste après la distance qu'il y a entre le produit et la station de sortie (ceci en traçant une ligne droite entre les deux points). On choisit le produit ayant la distance minimale (Figure 7).

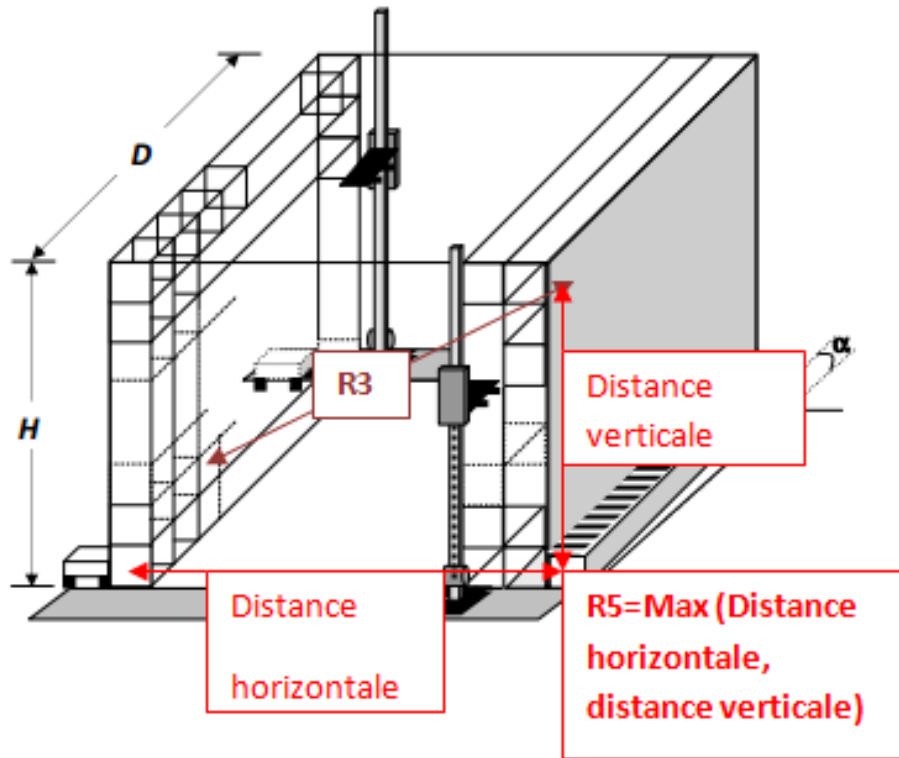
#### **4.3.4 La règle de déstockage R4 :**

Cette règle tient compte du nombre de produits qu'il y a avant celui que nous souhaitons déstocker mais aussi de la distance qu'il y a entre ce dernier et la station de sortie.

#### **4.3.5 La règle de déstockage R5:**

Cette règle tient compte du nombre de produits qu'il y a avant celui que nous souhaitons déstocker mais aussi de du maximum entre la distance horizontale et la distance verticale de la machine de déstockage (Figure 13).

Nous considérons que le déplacement horizontal vaut une unité de distance car notre système se résume à une seule rangée.



**Figure 13 Principe de fonctionnement des règles de déstockage R3 et R5**

#### **4.4 Temps de déplacement de la machine de déstockage**

Il nous a semblé intéressant et adéquat de schématiser et mesurer le temps de déplacement de la machine de déstockage.

Il faut donc calculer le temps des déplacements horizontaux de la machine de déstockage mais aussi celui des déplacements verticaux.

L'emplacement de repos de la machine de déstockage a été choisi juste à côté de la station de sortie.



```

/*A : Pour le déplacement de tous les produits qui précèdent celui
que l'on va déstocker du casier courant jusqu'au convoyeur de
restockage*/

/* i : représente le numéro de casier */

/* pour l'aller-retour --pour la distance unitaire*/
A= (i - 1) * 2*2
/* Pour le déplacement du produit choisi pour le déstockage
jusqu'à la station de sortie*/

/*j: représente l'abscisse du produit à déstocker c.à.d. le numéro
de couche */

/*B : Pour l'aller-retour --pour la distance élémentaire
B= i * 2*2 * (j-1)

TempsDestoc=A + B + TempsDestoc

```

#### 4.5 Résultats obtenus et interprétations

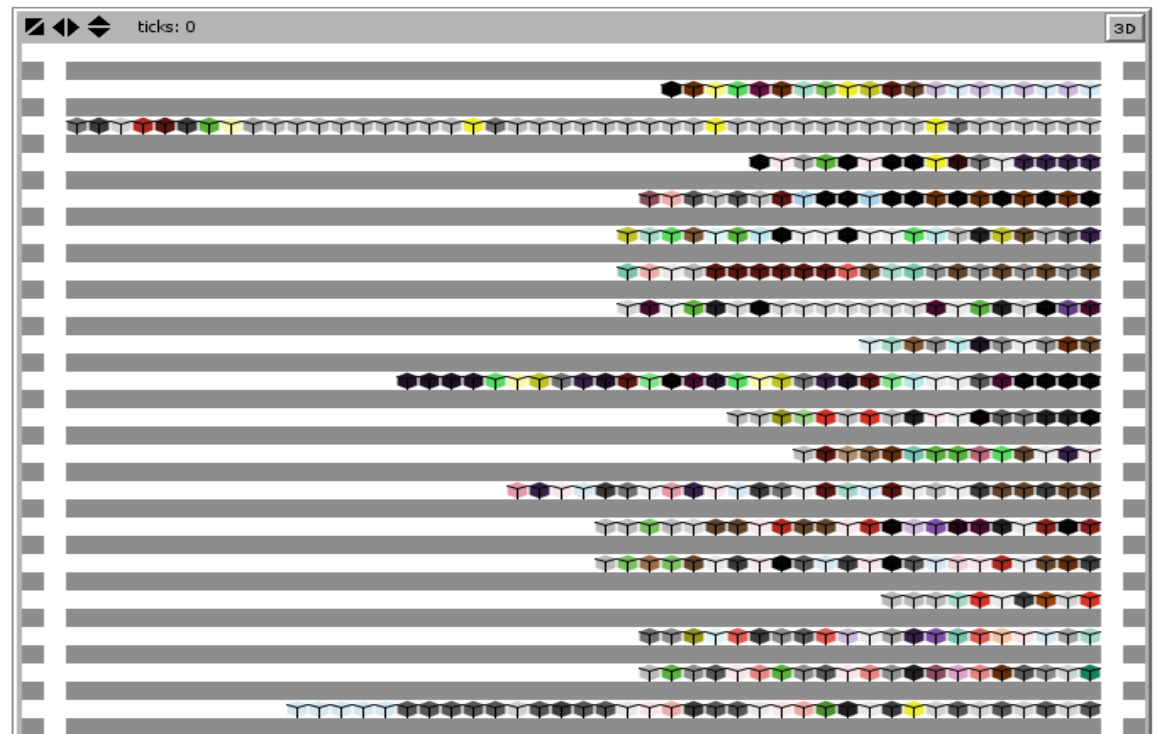
Nous mesurons le temps de déplacement moyen de la machine de déstockage ; en partant à chaque fois d'un état initial du magasin.

Chaque courbe dans ce qui suit modélise cinquante temps de déplacement de la machine de déstockage (puisque'il y a cinquante itérations dans la boucle Repeat de la procédure Sim).

Nous nous sommes contentés de calculer cinquante temps de déplacement, car cela a suffi pour avoir des résultats intéressants et interprétables. Cependant ce nombre peut être augmenté.

Les vitesses horizontale et verticale de la machine S/R sont représentées par des variables globales dans l'interface principale de notre programme, on peut changer leurs valeurs à n'importe quel moment. Nous leur attribuons par exemple la valeur 1.

#### 4.5.1 Etat initial du magasin E0



**Figure 14 Etat initial du rack**

Nous avons choisi l'état initial du magasin E0 (Figure 14) avec un taux de remplissage presque égal à 48%. Nous appliquons les règles de stockage et de déstockage en partant de cet état là. Nous procédons au déstockage d'une liste de produits que nous définissons.

Le tableau qui suit résume les résultats obtenus :

<b>Règle de stockage</b>	RE1	RE1	RE1	RE1	RE1	RE2	RE2	RE2	RE2	RE2	RE3	RE3	RE3	RE3	
<b>Règle de destockage</b>	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
<b>Temps moyen de déplacement de la machine de déstockage</b>	3238 2.88	396 36.5	42178 .24	37618 .24	439 58	899 8.4	13989 .36	15813 .12	5906. 72	6089. 52	11479 .76	8810. 72	888 2	1027 5.92	721 6.8

**Tableau 4 Le temps moyen de déstockage obtenu en combinant deux règles**

## 4.6 Courbes modélisant le temps de déplacement de la machine de déstockage

### 4.6.1 Règle de stockage « RE1 » combinée aux règles de déstockage « R1, R2, R3, R4 »

Le stockage aléatoire a été étudié en premier comme référence de comparaison aux stockages heuristiques présentés dans les sections 6.2.2 et 6.2.3.

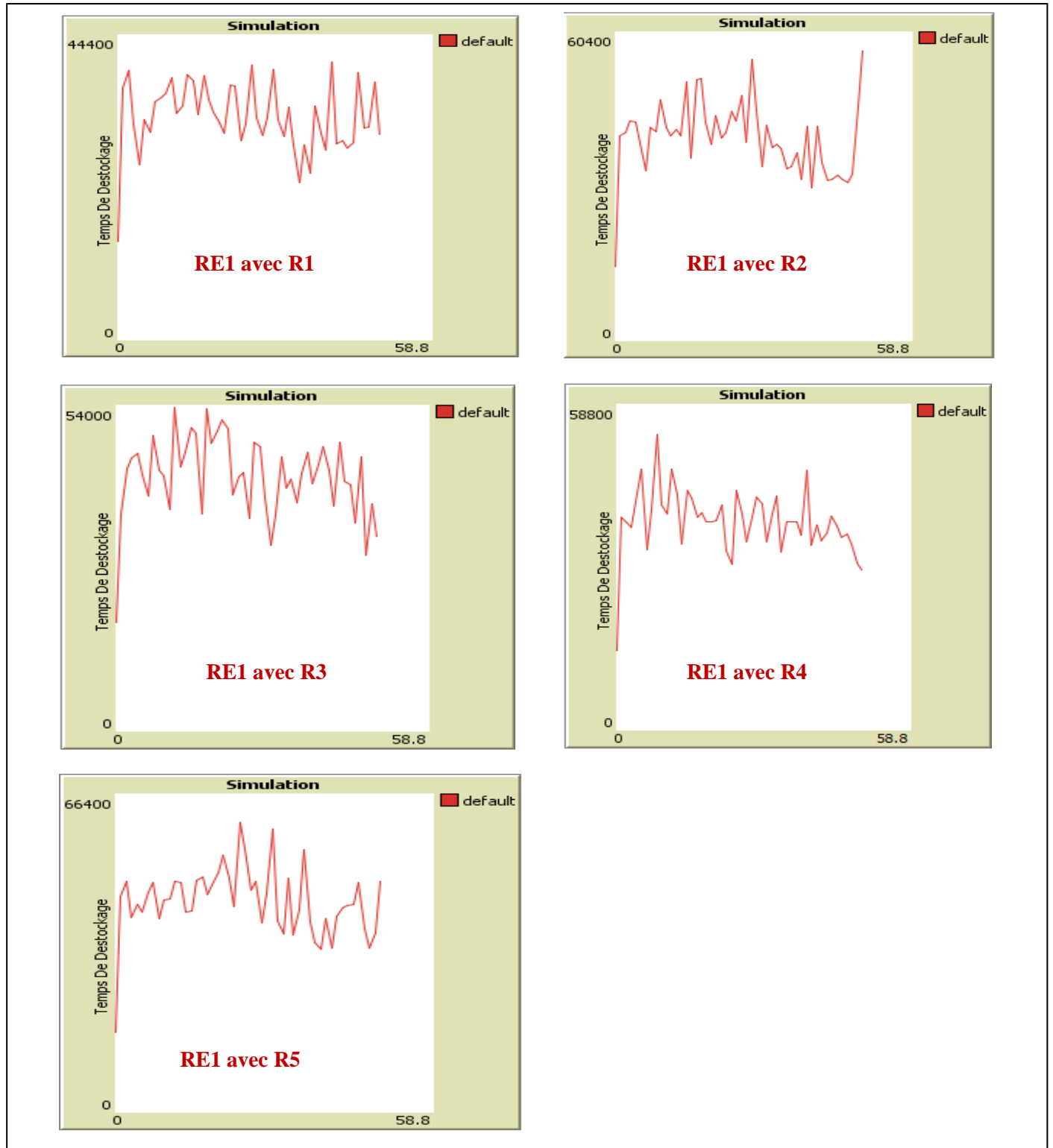


Figure 15 Temps de déstockage résultant d'un stockage aléatoire combiné aux 5 règles de l'heuristique

- Nous constatons que pour le cas du stockage aléatoire l'évolution du temps de déstockage au fil du temps est instable (figure 15).

#### 4.6.2 Règle de stockage « RE2 » combinée aux règles de déstockage « R1, R2, R3, R4 »

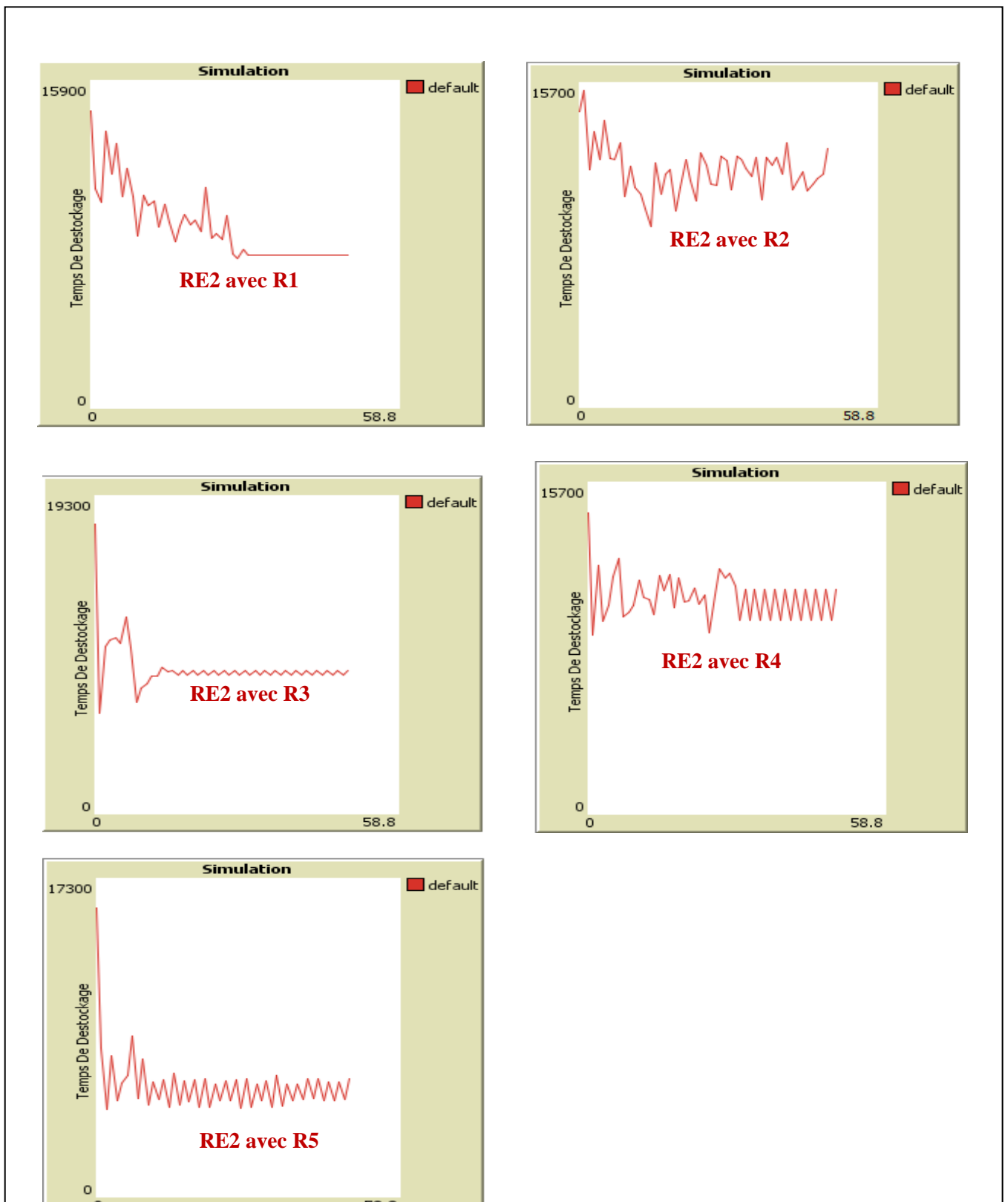


Figure 16 Temps de déstockage résultant d'un stockage par RE2 combiné aux 5 règles de l'heuristique

- Nous remarquons que les 5 temps de déstockage moyens obtenus en appliquant la règle RE2 sont de loin plus petits que ceux obtenus en appliquant un stockage aléatoire.

- Le temps moyen de déstockage qui résulte de la combinaison RE2, R4 s'avère être le meilleur, par rapport à celui qui résulte de la combinaison RE2, R2 et RE2, R3, R5 (Figure 16).

- La règle de déstockage R3 peut donner de bons résultats ; dans le cas où la moyenne du nombre de produits stockés avant celui que l'on souhaite déstocker est petite. Le cas contraire n'est pas intéressant car nous devrions restocker à chaque fois beaucoup de produits.

#### 4.6.3 Règle de stockage « RE3 » combinée aux règles de déstockage « R1, R2, R3, R4 »

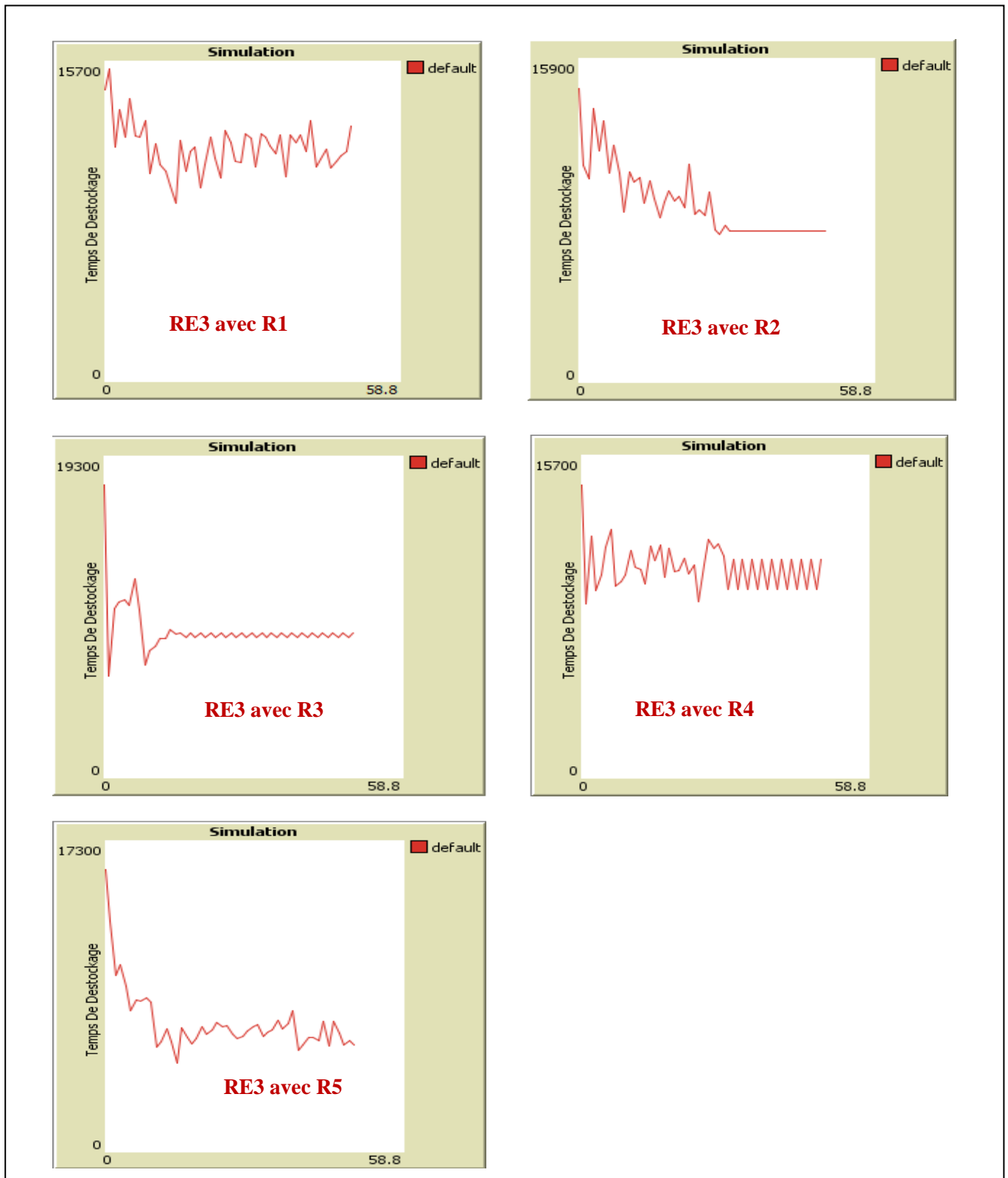


Figure 17 Temps de déstockage résultant d'un stockage par RE3 combiné aux 5 règles de l'heuristique

- Les meilleurs temps moyens de déstockage obtenus résultent de la combinaison de la règle de stockage RE3 avec les règles de déstockage R2 et R3.
- En appliquant RE3, nous avons un nombre très petit de produits de même type dans un même casier donc lors d'un déstockage le nombre de produits à restocker sera petit.

#### **4.7 Interprétation des résultats :**

##### **4.7.1 Règle de stockage RE1**

Il est clair que le temps de déplacement de la machine de déstockage se basant sur un stockage aléatoire (RE1) a donné des résultats médiocres comparés à ceux obtenus en appliquant des règles de stockage.

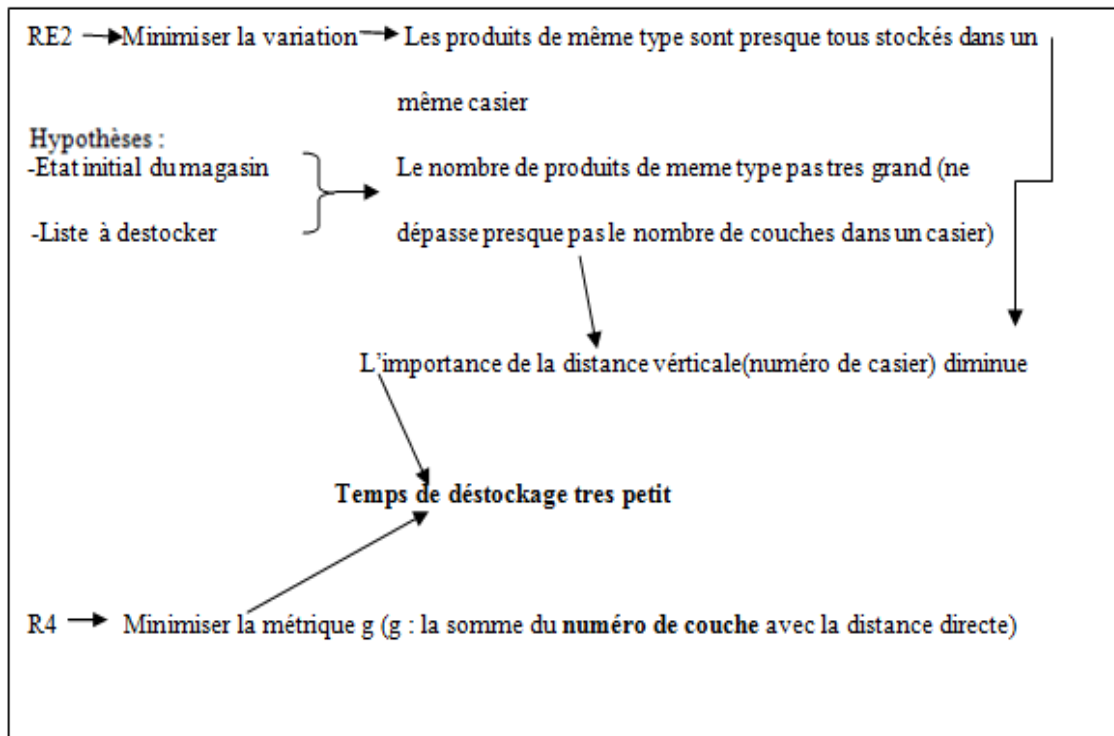
##### **4.7.2 Règle de stockage RE2**

Après combinaison de la règle de stockage RE2 avec les différentes règles de déstockage : R1, R2, R3, R4 et R5, nous avons constaté que le temps minimal de déstockage obtenu correspond à la combinaison (RE2, R4) ceci s'explique par :

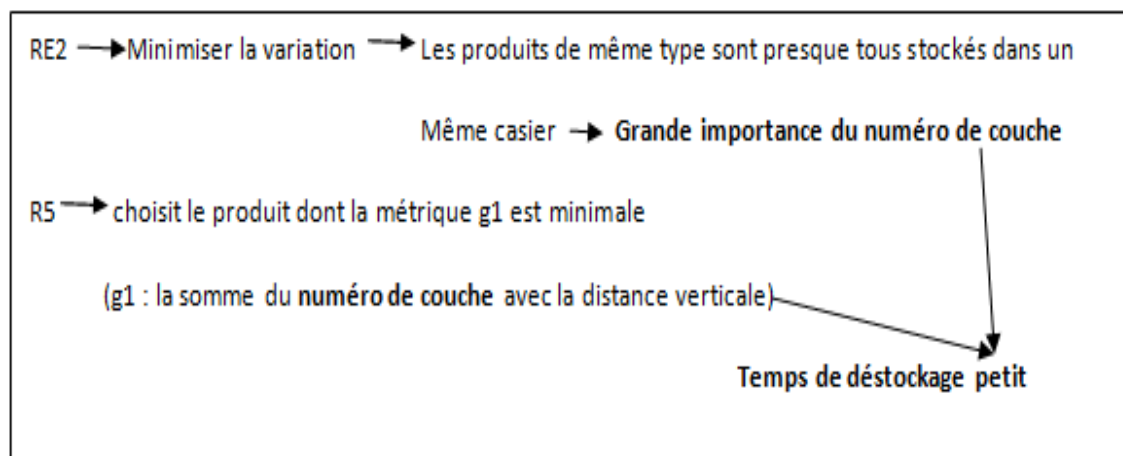
La règle RE2 minimise la variation de types de produits dans un même casier ceci permet de trouver le produit à déstocker dans les premières couches.

La règle R4 calcule la moyenne entre le nombre de produits à restocker et la distance directe entre le produit à déstocker et la station de sortie.

Plus le nombre de couches est réduit, plus la distance directe entre le produit à déstocker et la station de sortie est petite et plus le nombre de produits se trouvant avant le produit à déstocker diminue ce qui explique l'obtention du meilleur temps de déstockage pour la combinaison (RE2, R4).

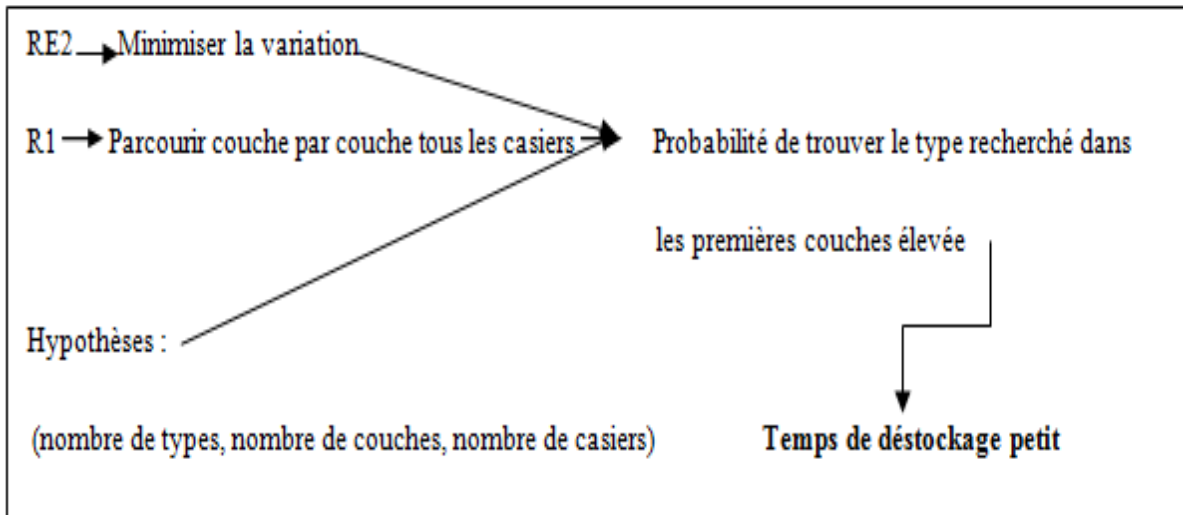


- (RE2, R5) est la combinaison qui donne un bon résultat, car la règle de déstockage R5 combinée à n'importe quelle règle de stockage donne un bon résultat c.à.d. que si nous devons changer les procédures de stockage à tout moment ; autant appliquer la règle de déstockage R5.

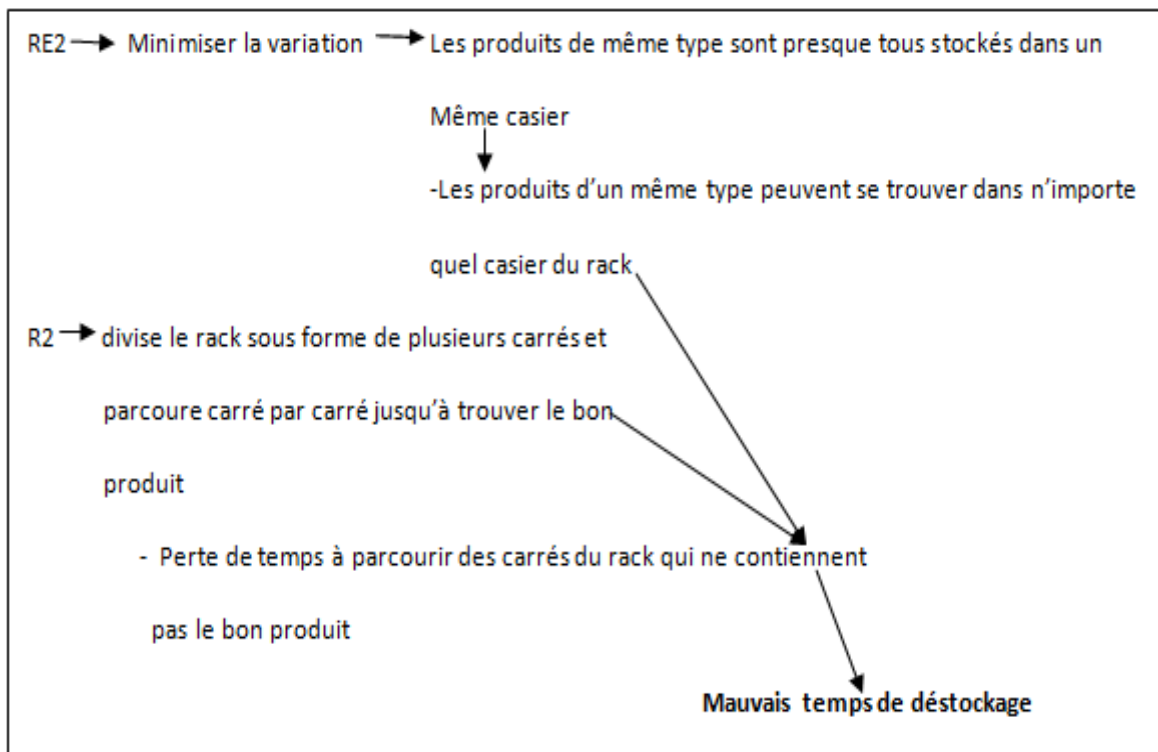


- (RE2, R1) a donné un bon résultat car R1 a tendance à chercher le produit souhaité en parcourant couche par couche dans une rangée du rack (c.à.d. nous cherchons le produit dans la 1ere couche du 1 er casier ensuite dans la 1ere couche du 2eme casier ...) ; et comme RE2 minimise la variation de produits dans un même casier c.à.d. il y a de fortes chances que nous trouvions le produit dans les premières couches ,cela devrait donné un bon résultat.

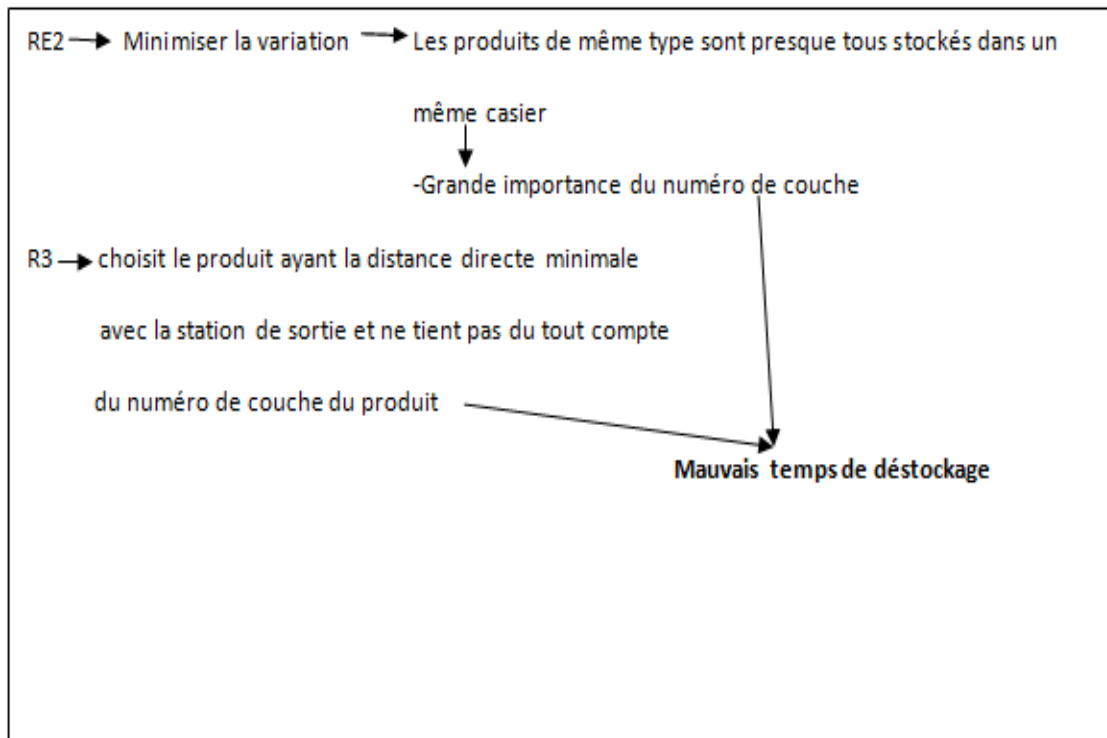




- Il n'est pas conseillé de combiner RE2 avec R2 car il se peut que le produit recherché soit situé dans les derniers casiers du rack (si nous cherchons un produit jaune par exemple et que le casier 17 est celui qui contient la plupart des produits jaunes) nous devrions alors parcourir plusieurs carrés du rack avant d'arriver au bon.



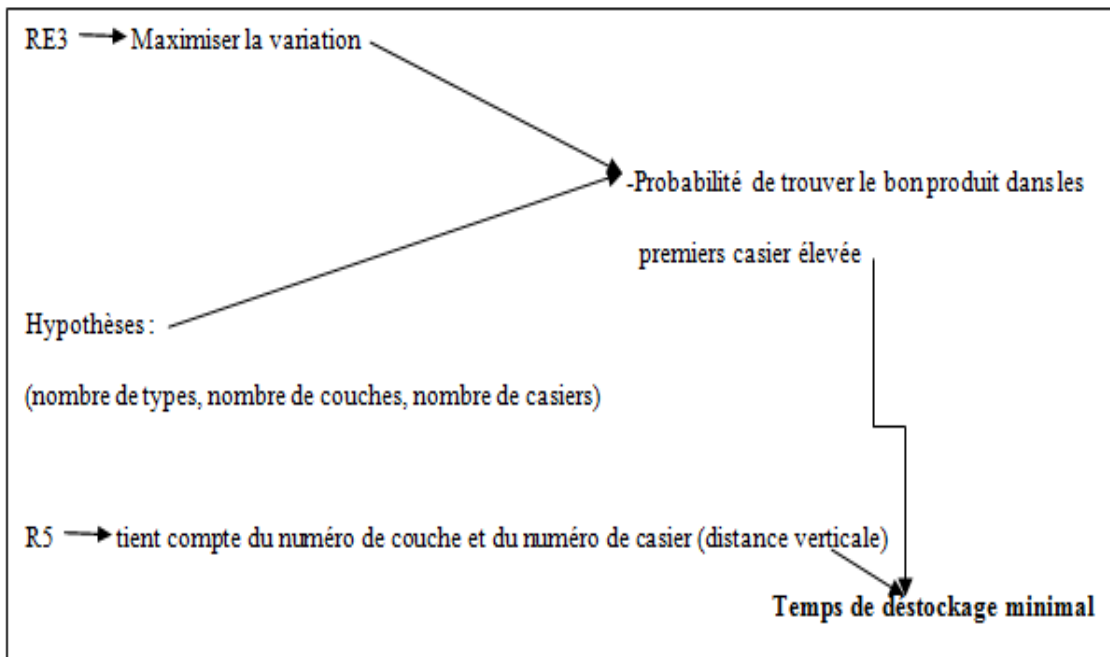
La combinaison (RE2, R3) est aussi à éviter car R3 ne tient pas du tout compte de la couche où est située le produit à déstocker alors que dans le cas où nous appliquons RE2 comme règle de stockage nous avons plus de chance de trouver les produits du type souhaité tous stockés dans un même casier d'où la grande importance du numéro de couche du produit.



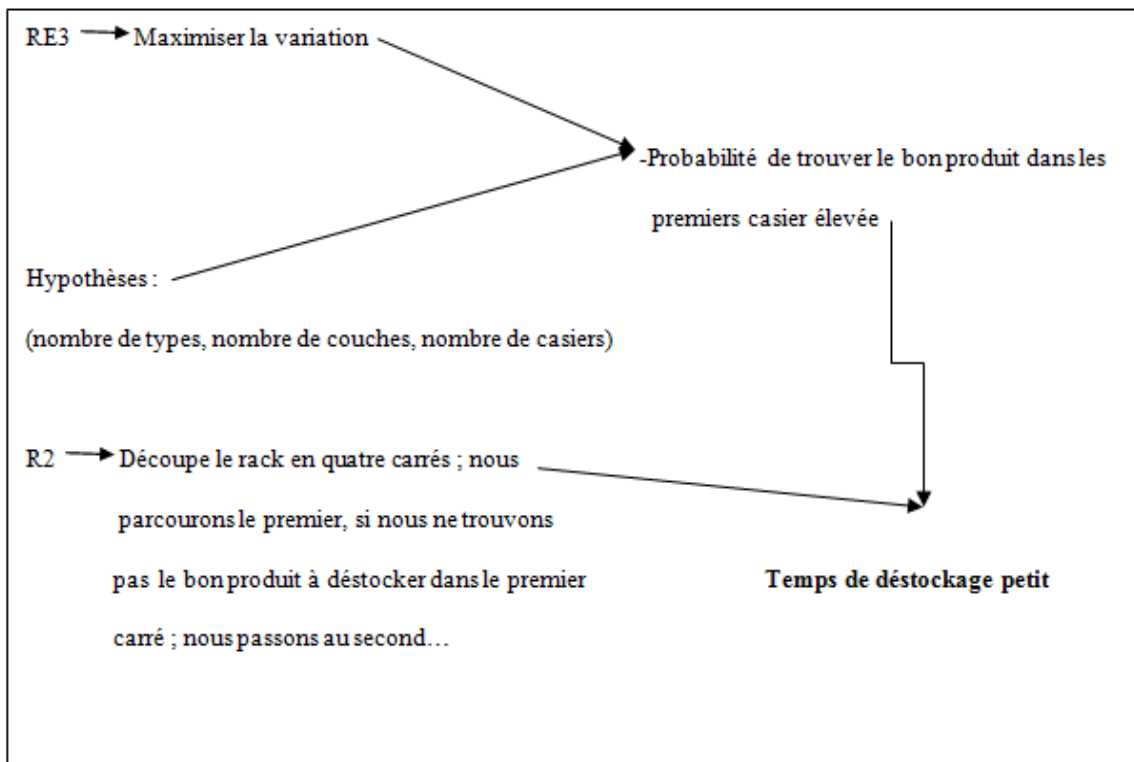
#### 4.7.3 Règle de stockage RE3

- Après combinaison de la règle de stockage RE3 avec les différentes règles de déstockage : R1, R2, R3, R4 et R5, nous avons constaté que le temps minimal de déstockage obtenu correspond à la combinaison (RE3, R5) ceci s'explique par :

- Suivant la règle de stockage RE3; il est probable de trouver le bon produit à déstocker dans n'importe quelle couche d'où la grande importance du numéro de couche.
- La règle RE3 maximise la variation de produits dans chaque casier et vu les hypothèses que nous avons ;il y a de grandes chances que nous trouvions le bon produit dans les premiers casiers ;ceci dit le numéro de casier a tout de même son importance .
- Comme la règle de déstockage R5 est celle qui combine ces deux facteurs importants (numéro de couche et numéro de casiers) ; elle a conduit au temps de déstockage minimal.



- La combinaison (RE3, R2) a donné un bon résultat car comme nous l'avons constaté dans ce qui précède nous avons beaucoup de chance de trouver le bon produit dans les premiers casiers et justement R2 découpe le rack en quatre carrés et commence par parcourir le premier (le carré qui englobe les premiers casiers); ensuite le second... ce qui mène à obtenir un bon résultat.

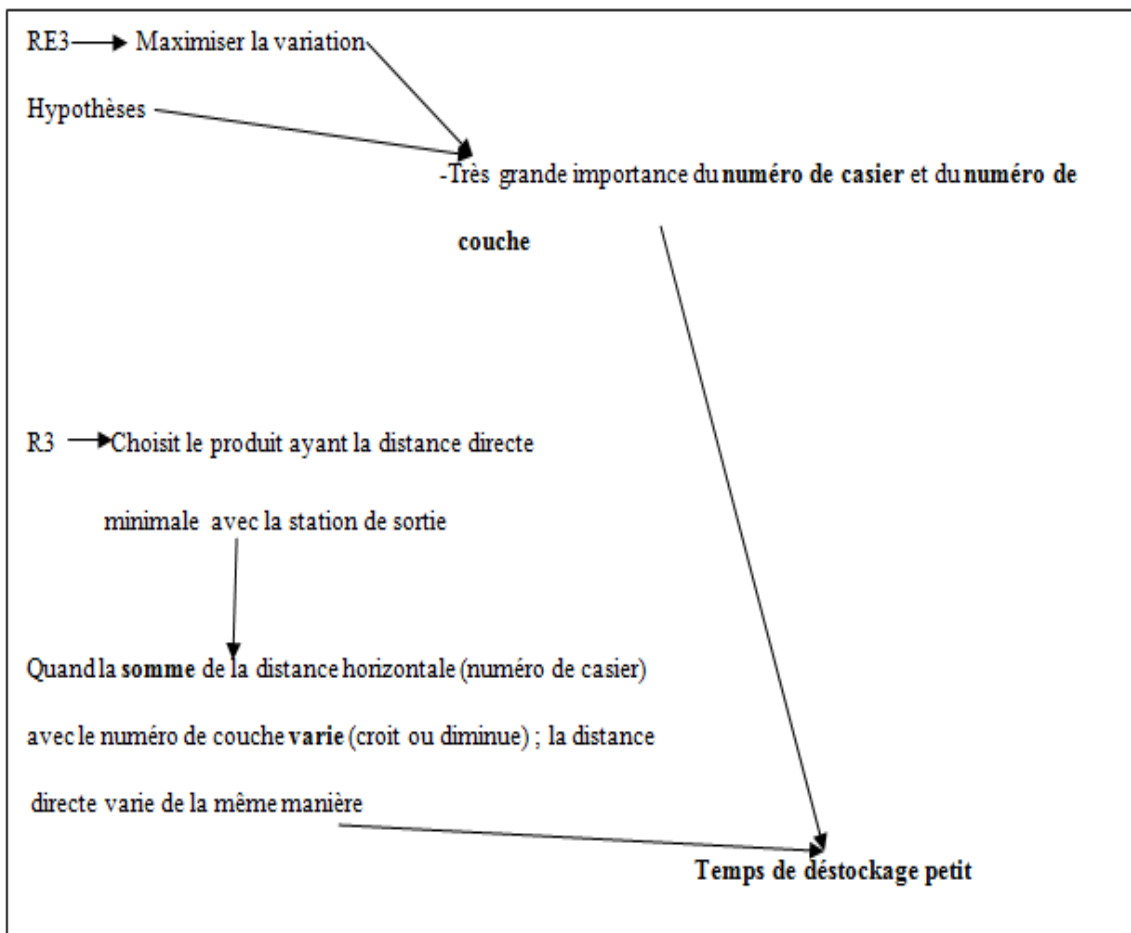


- (RE3,R3) donne elle aussi un bon résultat ;ceci s'explique par :

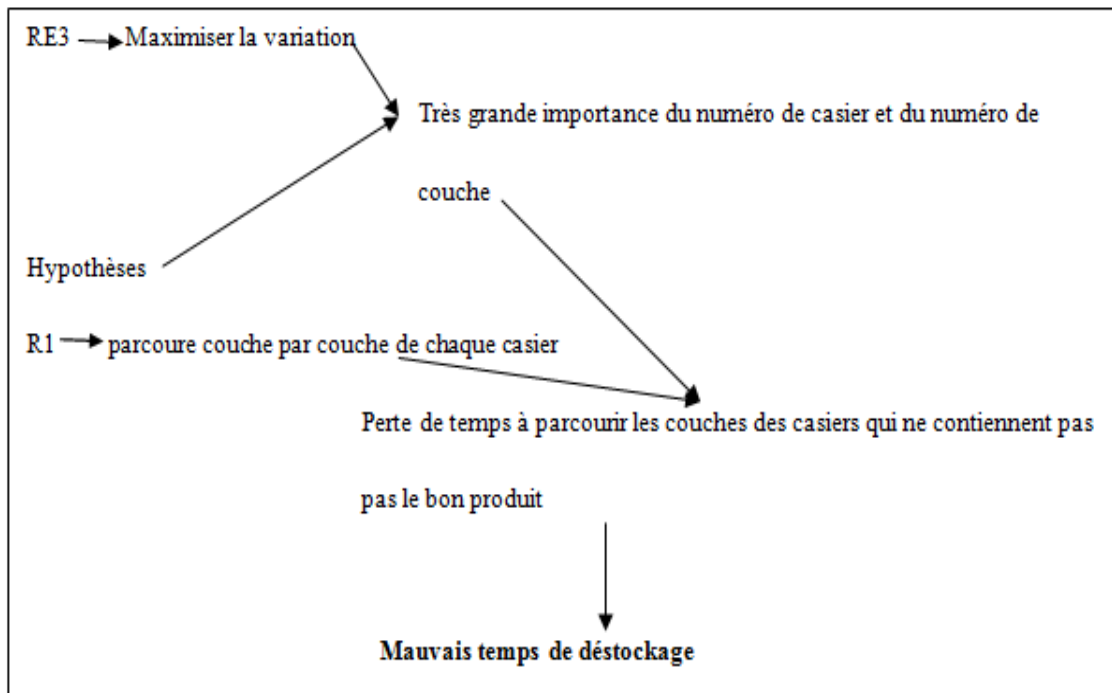
- En appliquant la règle de stockage RE3 ;nous donnons une grande importance au numéro de casier et au numéro de couche ;et d'après le théorème de Pythagore nous concluons que la distance directe entre un produit et la station de sortie est égale à la racine carrée de la somme du carré du numéro de casier(distance verticale) avec le carré du numéro de couche :

$$\text{Distance directe} = \sqrt{(\text{numéro de couche})^2 + (\text{numéro de casier})^2}$$

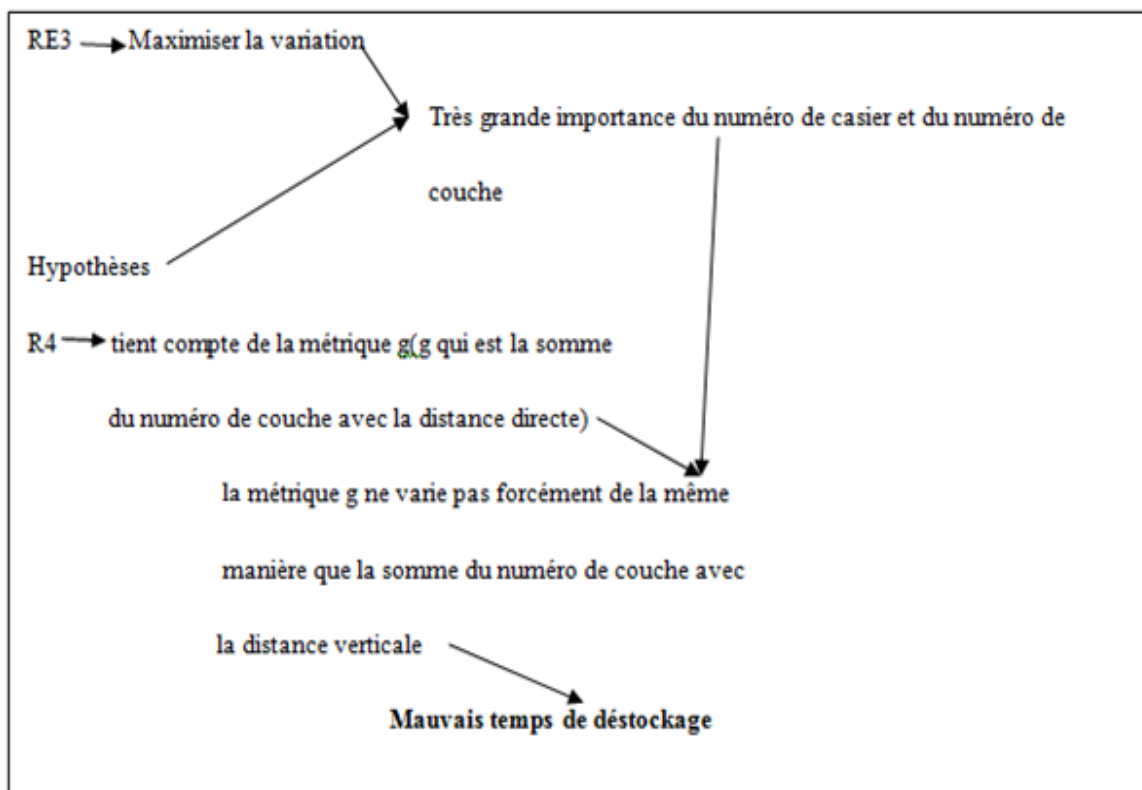
- De plus nous savons que la somme (numéro de couche+numéro de casier) évolue(croît ou diminue) de la même manière que la distance directe c'est pourquoi combiner RE3 avec R3 a donné un bon résultat.



- Contrairement à RE2, avec RE3 nous ne sommes pas sur que le produit du bon type est situé dans les premières couches, et puisque le principe de R1 est de parcourir couche par couche chaque casier ; nous risquons de perdre beaucoup de temps à parcourir des couches inutilement.



Nous savons qu'avec RE3 il est important de tenir compte de deux choses : le numéro de casier et le numéro de couche .Même si la règle de déstockage R4 tient compte du numéro de couche ; elle ne le combine pas avec la bonne distance (c.à.d. la distance verticale) d'où l'obtention d'un mauvais temps de déstockage.



- Le choix de la combinaison de règles de stockage et de déstockage dépend de deux facteurs très importants qui sont : le taux de remplissage du rack ; mais aussi de la manière dont il est rempli au départ c.à.d. que si nous changions l'état initial du magasin de E0 à E1 c'est sur que le choix de la meilleure combinaison de règles changera avec ; c'est pourquoi il serait très intéressant d'ajouter dans l'interface principale ces deux paramètres.
- La perception du système comme un SMA est très importante ; car lorsque les agents produits appliquent les meilleures combinaisons de règles de stockage et de déstockage dans l'environnement (le magasin) ; cela améliorerait au fur à mesure le temps moyen de déstockage (le but à atteindre).
- Les agents produits communiquent leur métrique (suivant la règle de déstockage appliquée) à l'agent station de sortie et à partir de cette communication nous pouvons conclure le choix du meilleur agent produit à déstocker. Cette intelligence est distribuée entre beaucoup d'agents réactifs (produits) ; ce comportement intelligent émerge de l'interaction entre ces agents réactifs et l'environnement.

## 5. Conclusion

Dans ce chapitre, nous avons conçu notre propre modèle, que nous avons baptisé « SMASD ».

La nouveauté et l'originalité de ce modèle réside dans la fusion de deux techniques minutieusement choisies et adaptées aux AS/RS à convoyeur gravitationnel.

Nous avons tout d'abord mis en place une stratégie de stockage (règles de stockage) et de déstockage (règles de déstockage). nous choisissons la meilleure combinaison de ces règles ; ceci dans le but de ranger le mieux possible notre magasin.

Ensuite nous avons perçu notre AS/RS à convoyeur gravitationnel comme étant un SMA (système multi agents) ; ou ces derniers sont des agents autonomes en interaction ; ils se communiquent des informations les concernant telles que leur type, leur distance par rapport à la station de sortie ...

La fusion des deux a permis une complémentarité visant un meilleur rangement du rack.

Nos perspectives sont de reconsidérer le même système ; mais cette fois ci en mettant plus en évidence la communication entre les agents c'est-à-dire en suivant un langage dédié aux SMA « ACL : Langage de Communication entre Agents ». Pour cela nous utiliserons dans une deuxième contribution une plateforme spécifique nommée **Jade**.

# Chapitre 4 Deuxième Contribution

## Application avec Jade

### 1. Introduction

Dans ce travail nous nous sommes intéressés au déstockage de produits dans un AS/RS (Automated Storage Retrieval System) ; le but étant de minimiser en permanence le temps de déstockage d'une liste de produits. Le système a été perçu comme un SMA : Système multi-agents ; où les agents communiquent entre eux afin de prendre la meilleure décision de déstockage.

Le modèle proposé dans cette partie ; sera implémenté sous la plateforme multi-agents Jade ; il repose principalement sur trois types d'agents : l'agent DF, l'agent Produit qui devra être déstocké et puis sur l'agent Casier.

Le déstockage d'un produit se fait généralement suivant une règle de déstockage bien spécifique ; chacune de ces règles est fondée sur une certaine métrique mathématique.

L'objectif est à présent de mettre en évidence les communications entre les différents agents de notre système afin de choisir le meilleur casier d'où le produit en question devra être déstocké (la règle de déstockage offrant la métrique minimale).

### 2. Les outils de développement

Notre application est implémentée en utilisant le langage JAVA, sous l'environnement de développement NetBeans. Pour le système multi agents nous avons utilisé la plateforme JADE.

#### 2.1 Le langage de programmation java

Le choix du langage JAVA a été fait pour les raisons suivantes :

Les agents développés sous la plate-forme JADE, sont entièrement écrits en Java. Ce langage nous est donc imposé par le choix de la plate-forme.

Java assure une totale indépendance des applications vis-à-vis de l'environnement d'exécution : c'est à dire que toute machine supportant Java est en mesure d'exécuter un programme sans aucune adaptation (ni recompilation, ni paramétrage de variables d'environnement) ;

JAVA possède une bibliothèque immense d'objets prêts à l'emploi, ce qui facilite pleinement la procédure d'implémentation.

## 2.2 L'environnement de développement NetBeans

NetBeans est un EDI (Environnement de Développement Intégré), un environnement libre et facile à utiliser. Il supporte plusieurs langages de programmation à savoir JAVA, C++, PHP, et bien d'autres langages. Il fournit plusieurs outils tels qu'un éditeur de texte doté d'un pré-compilateur avancé, un gestionnaire de projets, ainsi que des outils de débogage et de test de programmes. C'est un outil qui facilite énormément la phase de développement et de tests.

## 2.3 La plateforme JADE (Java Agent Development Framework)

La plateforme JADE (Java Agent DEvelopment Framework) est un environnement de développement d'agents implanté totalement dans le langage JAVA (Figure 18). Il facilite la mise en place d'un système multi agent répondant aux spécifications de FIPA (Fondation for Intelligent Physical Agent) à travers un ensemble d'outils.

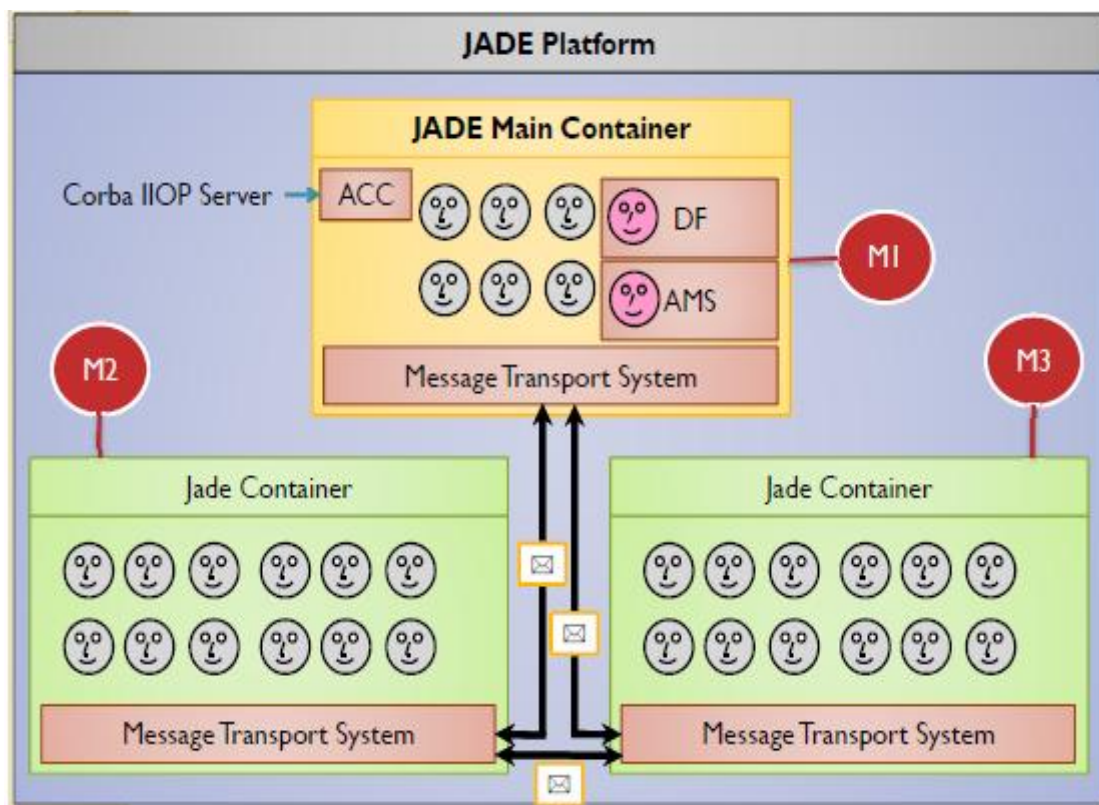


Figure 18 Architecture de jade

La plateforme JADE inclut tous les composants obligatoires qui contrôlent un système multi-agent. Ces composants sont l'ACC (Agent Communication Channel), l'AMS (Agent



Management System) et le DF (Directory Facilitator). Toute la communication entre agents est exécutée par messages FIPA/ACL.

L'Agent Platform (AP) constitue l'infrastructure physique sur laquelle se déploient les agents (Container). Il contient un ou plusieurs Containers ; ceci dit un et un seul main-container. Chaque main-container comporte: **AMS** « Agent Management System » qui supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système (pages blanches).

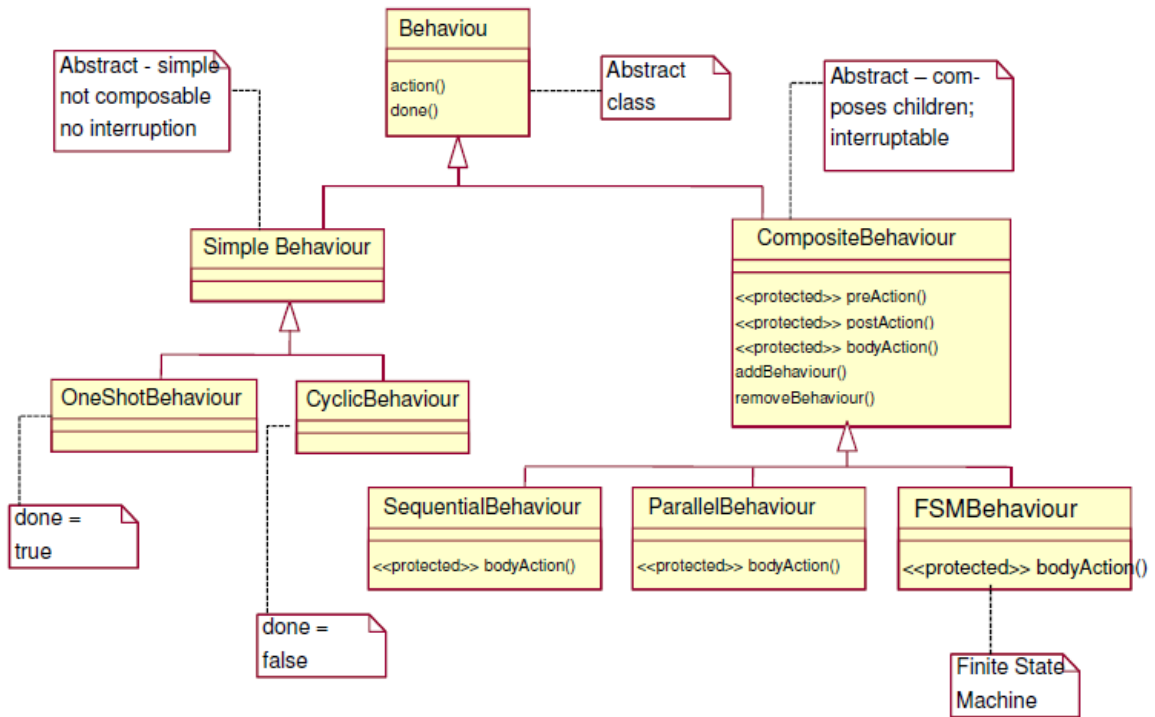
**DF** « Directory Facilitator » qui fournit un service de « pages jaunes » à la plate-forme (enregistrement, dé- enregistrement des services, recherche,...).

**ACC** «Agent Communication Channel » qui gère la communication entre les agents ;

**MTS** est une méthode par défaut de communication entre agents de différentes plateformes. Cela permet l'interconnexion entre systèmes hétérogènes

La plateforme d'agent peut être répartie sur plusieurs serveurs. Une seule application Java, et donc une seule machine virtuelle Java Java Virtual Machine-JVM, est exécutée sur chaque serveur. Chaque JVM est un conteneur d'agents qui fournit un environnement complet pour l'exécution d'agent et permet à plusieurs agents de s'exécuter en parallèle sur le même serveur. L'architecture de communication offre la transmission de messages flexibles et efficaces. JADE crée et contrôle une file d'attente des messages entrants pour chaque agent. Le modèle global de communication entre ces agents FIPA a été mis en application. Ses composants ont été distingués clairement et ont été entièrement intégrés : protocoles d'interaction, ACL, langues, schémas de codage, protocoles de transport, etc. A chaque agent on associe un processus léger thread mais les agents ont souvent besoin d'exécuter des tâches en parallèle. L'environnement de développement pourvoit aussi des comportements tout " prêt à exécution " pour les tâches les plus souvent sollicitées dans la programmation orientée agent.

Il existe plusieurs types de comportements des agents (Figure 19):



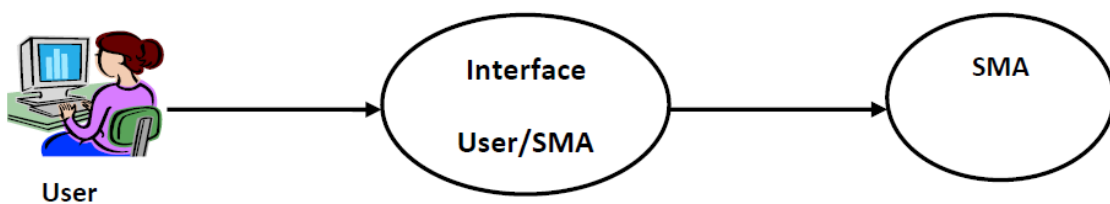
**Figure 19 Comportements des agents dans Jade**

Dans notre programme nous avons utilisé les comportements d’agents suivants :

- Cyclic Behaviour qui est une instance de la classe jade.core.behaviours.CyclicBehaviour. Un cyclic Behaviour exécute sa tâche d'une manière répétitive. La classe CyclicBehaviour implémente la méthode done() qui retourne toujours false.
- ParallelBehaviour pour qu’un agent puisse exécuter plusieurs comportements à la fois.

### 3. Architecture de notre système

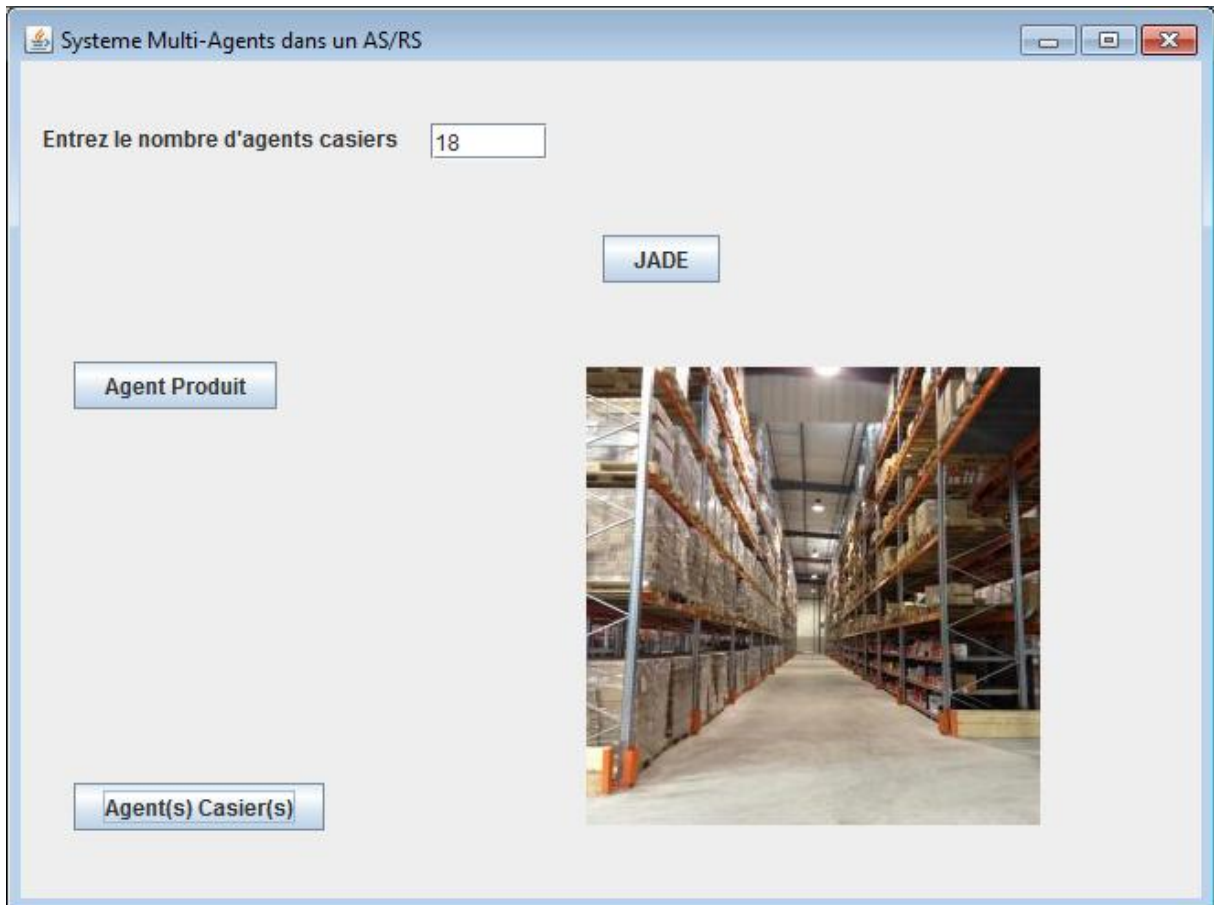
L’application implémentée se compose d’un système multi-agents et d’une interface de communication entre le SMA et l’utilisateur (Figure 20).



**Figure 20 Architecture générale de notre application**

## Interface Java

Nous donnons un aperçu de l'interface graphique de notre application (Figure 21). A travers cette interface, l'utilisateur lance la plateforme multi agents, contrôle le nombre d'agents Casiers qui composent le rack et crée dès qu'il le souhaite un agent Produit souhaitant être déstocké du rack.



**Figure 21 Interface Java du SMA**

### 4. Le système multi agents

Notre SMA se compose de deux types d'agents : un agent produit et plusieurs agents casiers.

- Des agents de type « Casier » qui ont pour objectif de proposer des offres aux agents produits qui souhaitent être déstockés.
- Un agent de type « Produit » qui a pour objectif d'être déstocké du meilleur casier (le casier offrant la meilleure métrique)

Le SMA fonctionne de la manière suivante :

#### **4.1 L'Agent produit**

Devra envoyer une requête contenant la couleur (le type) du produit que nous souhaitons déstocker.

#### **4.2 L'agent Casier**

En supposant que le choix de la règle de déstockage a déjà été effectué auparavant ; chaque casier propose une valeur de la métrique choisie (suivant la règle de déstockage).

L'agent produit devra par la suite consulter toutes les propositions et choisir celle qui offre la valeur minimale de la métrique de la règle de déstockage.

### **5. UML : Unified Modeling Language**

Le langage de modélisation unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

#### **5.1 Diagrammes de séquences**

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language.

L'utilité du diagramme de séquences :

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

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

## 5.2 Diagrammes de classes

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

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

Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations sont possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

Elles sont finalement instanciées pour créer des objets (une classe est un moule à objet : elle décrit les caractéristiques des objets, les objets contiennent leurs valeurs propres pour chacune de ces caractéristiques lorsqu'ils sont instanciés).

## 6. ArgoUML

ArgoUML est un logiciel libre de création de diagrammes UML. Programmé en Java, il est édité sous licence EPL 1.0. Il est multilingue, supporte la génération de code et l'ingénierie inverse.

## 7. Diagrammes UML de notre système

### - Diagrammes de séquences

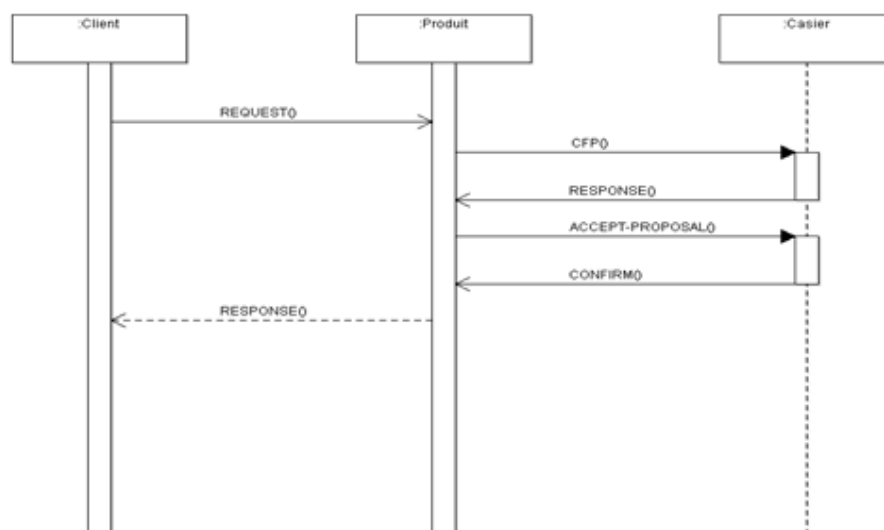
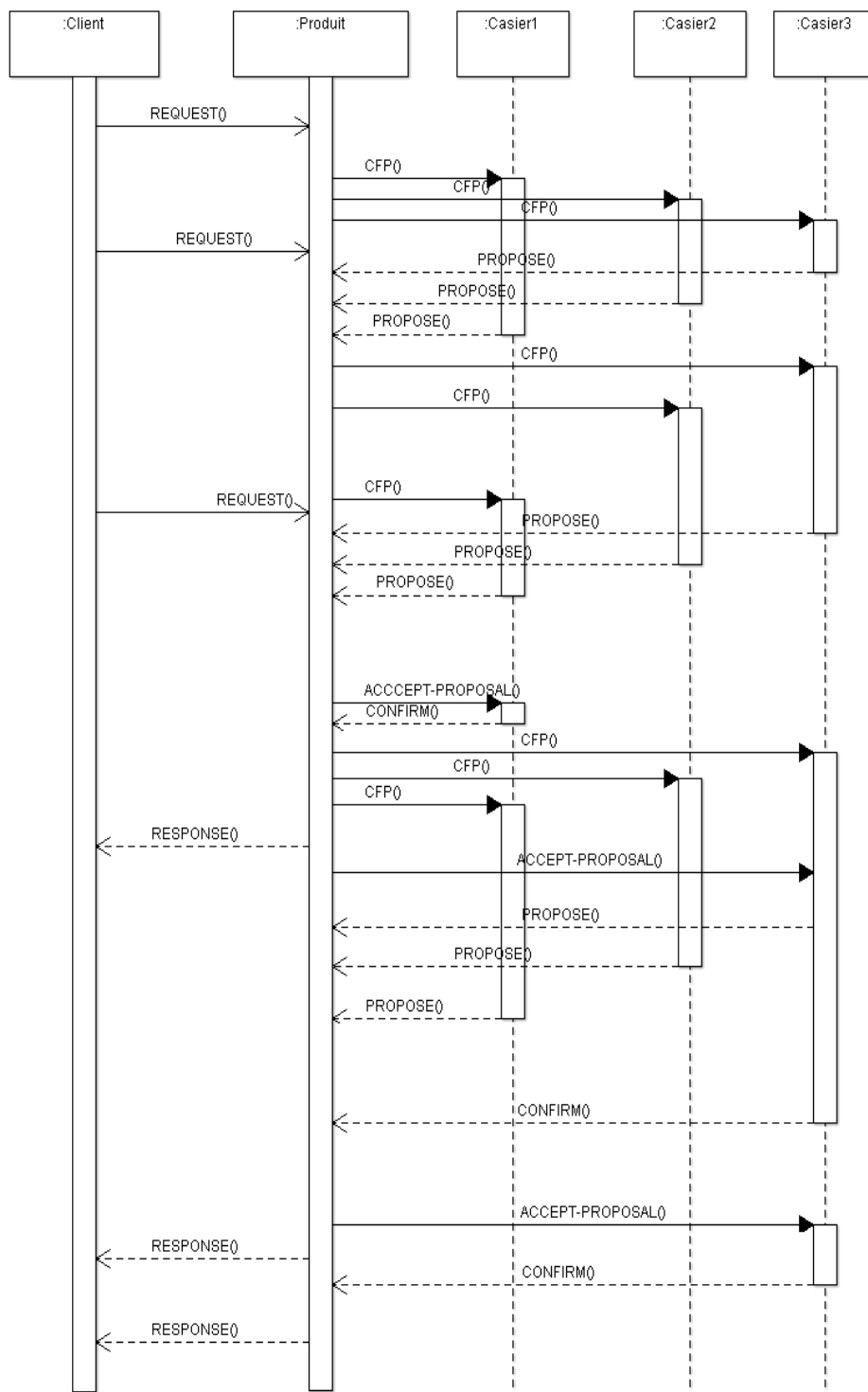


Figure 22 Envoi d'une seule requête de déstockage



**Figure 23** Envoi de plusieurs requêtes de déstockage

- Diagrammes de classes

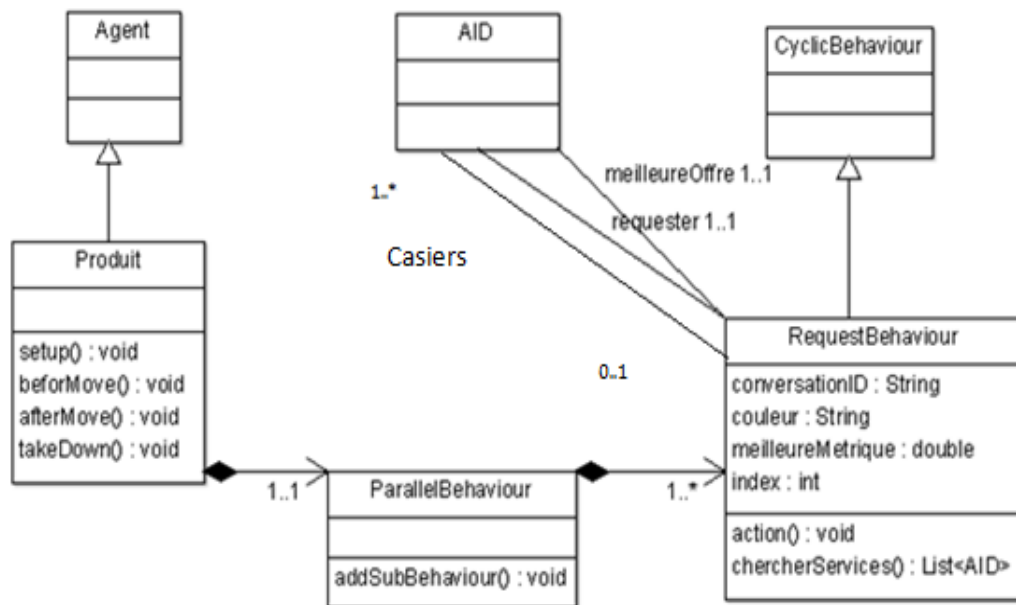


Figure 24 Diagramme de classes : Produit

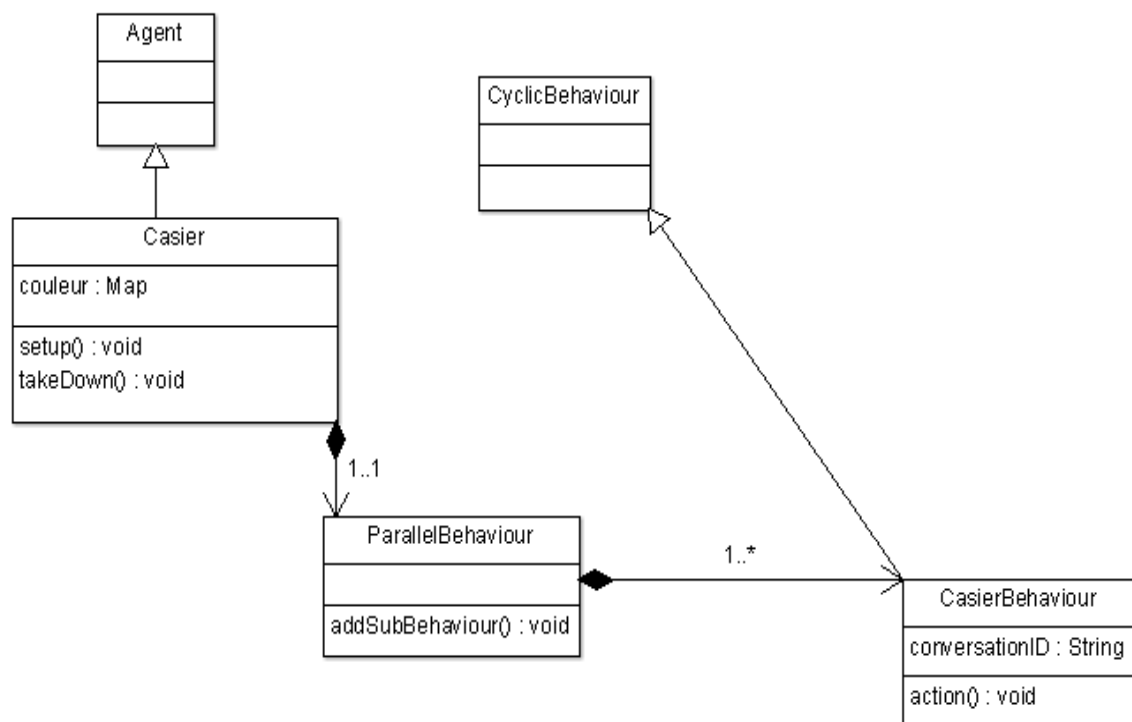


Figure 25 Diagramme de classes : Casier

## 8. Les classes de notre programme java

Notre programme java est composé de sept classes :

Casier.java, CasierBehaviour.java, Contcasier.java, Produit.java, RequestBehaviour.java, Contpro.java et SMA.java.

**Casier.java** : qui représente les agents Casiers ; celle-ci hérite de la classe Agent.

Chaque agent Casier possède une collection de type HashMap dont la clé représente le type de produits que nous souhaitons déstocker (sa couleur) et dont la valeur représente la métrique proposée par ce casier suivant une règle de déstockage choisie auparavant. Les métriques seront choisies aléatoirement car l'objectif principal de cette partie est de mettre en évidence les communications entre les différents agents du système et non pas la programmation d'heuristiques comme dans la partie précédente (avec netlogo).

Dans une seconde étape nous avons enregistré les services que les agents Casiers offrent auprès de l'agent DF.

Enfin nous sommes passés à la programmation des comportements des casiers ;

Les agents Casiers peuvent avoir deux comportements de type CyclicBehaviour qui peuvent se passer en même temps .Pendant qu'un agent Casier publie les services qu'il offre ; une autre requête de déstockage peut survenir, d'où le « parallelBehaviour ».

Le premier comportement concernera la réception d'un message de type **CFP** et l'envoi d'un message de type **propose**.

Le second comportement englobe l'acceptation de la proposition et l'envoi de la confirmation ; la classe **CasierBehaviour.java** lui est dédiée.

**Contcasier.java** : pour déployer un agent Casier dans un container.

La classe **Produit.java** dans la quelle nous avons instancié un objet de la classe **RequestBehaviour.java**. Les deux classes permettent d'établir les comportements des produits.

**RequestBehaviour.java** qui hérite de la classe **CyclicBehaviour** de jade.

**Contpro.java** : pour déployer un agent Produit dans un container.

**SMA.java** : pour l'interface principale de notre SMA.



```

public class Casier extends Agent {

private Map<String, Double> data=new HashMap();

private ParallelBehaviour parallelBehaviour;

protected void setup() {

//Charger les types de produits et les métriques proposées par les casiers suivant la règle de
déstockage appliquée

data.put("rouge", new Double(230+Math.random()*200));

data.put("jaune", new Double(460+Math.random()*200));

data.put("bleu", new Double(540+Math.random()*200));

data.put("blanc", new Double(230+Math.random()*200));

.....

//Enregistrement des services qu'offrent les casiers dans le DF

//Description des comportements que peuvent avoir les agents Casiers

parallelBehaviour=new ParallelBehaviour();

addBehaviour(parallelBehaviour);

parallelBehaviour.addSubBehaviour(new CyclicBehaviour() {

public void action() {

MessageTemplate messageTemplate=

MessageTemplate.MatchPerformative(ACLMessage.CFP);

ACLMessage aclMessage=receive(messageTemplate);

System.out.println("Conversation ID:"+aclMessage.getConversationId());

String couleur=aclMessage.getContent();

Double g=data.get(couleur);

ACLMessage reply=aclMessage.createReply();

reply.setPerformative(ACLMessage.PROPOSE);

reply.setContent(g.toString());

Thread.sleep(5000);send(reply);

}

});

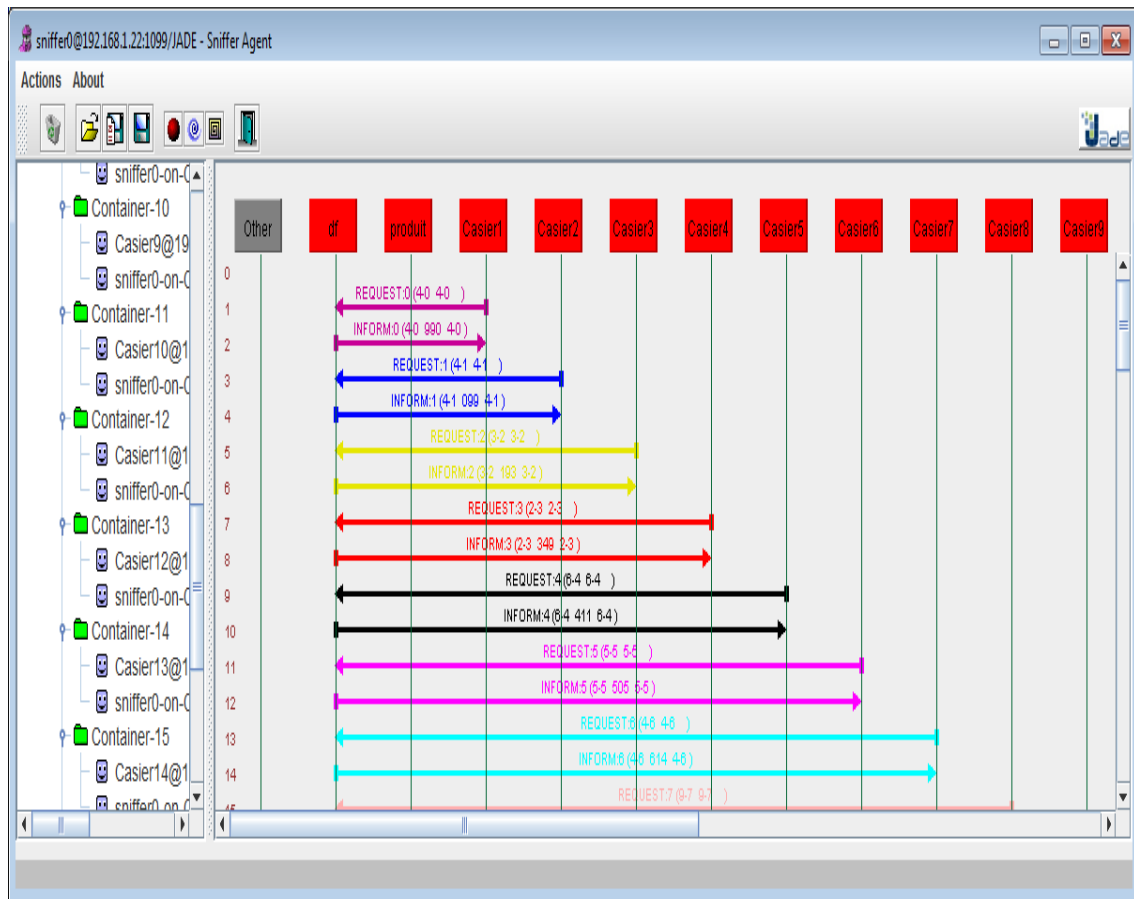
parallelBehaviour.addSubBehaviour(newCasierBehaviour(myAgent,aclMessage.getConversa
tionId()));

```

## Déroulement des communications

- Dans un premier temps tous les agents casiers devront enregistrer les services qu'ils offrent auprès de l'agent DF qui existe par défaut dans Jade. La figure suivante (Figure 26) est extraite à partir de l'outil « agent sniffer » de la plateforme JADE.
- Ensuite, chaque minute, l'agent Produit tente de chercher dans le service des pages jaunes DF, les agents qui offrent le service de déstockage.

```
Recherche des services...  
Liste des casiers trouvés :  
  
====Casier1@192.168.1.44:1099/JADE  
  
====Casier2@192.168.1.44:1099/JADE  
  
====Casier10@192.168.1.44:1099/JADE  
  
====Casier7@192.168.1.44:1099/JADE  
  
====Casier3@192.168.1.44:1099/JADE  
  
====Casier9@192.168.1.44:1099/JADE  
  
.....
```



**Figure 26 Enregistrement de services auprès de l'agent DF**

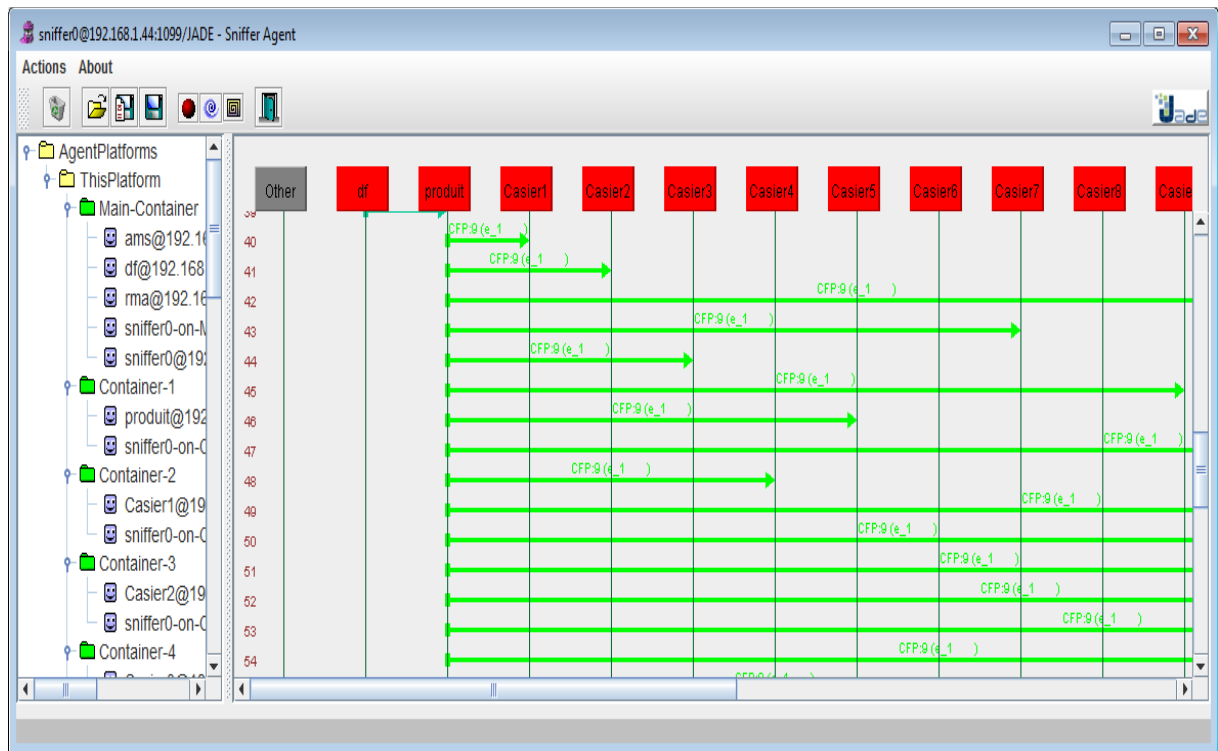
- L'agent produit envoie un message contenant la couleur du produit à déstocker. La requête envoyée par l'agent produit peut contenir n'importe quel type d'informations ; elle peut par exemple contenir en plus de la couleur du produit que nous souhaitons déstocker : la quantité souhaitée ... (Figure 27)

The 'ACL Message' dialog box is shown with the 'Envelope' tab selected. The fields are as follows:

- Sender: Set
- Receivers: produit@192.168.1.44:1099/JADE
- Reply-to: (empty)
- Communicative act: request
- Content: orange
- Language: (empty)
- Encoding: (empty)
- Ontology: (empty)
- Protocol: Null
- Conversation-id: (empty)
- In-reply-to: (empty)
- Reply-with: (empty)
- Reply-by: Set
- User Properties: (empty)

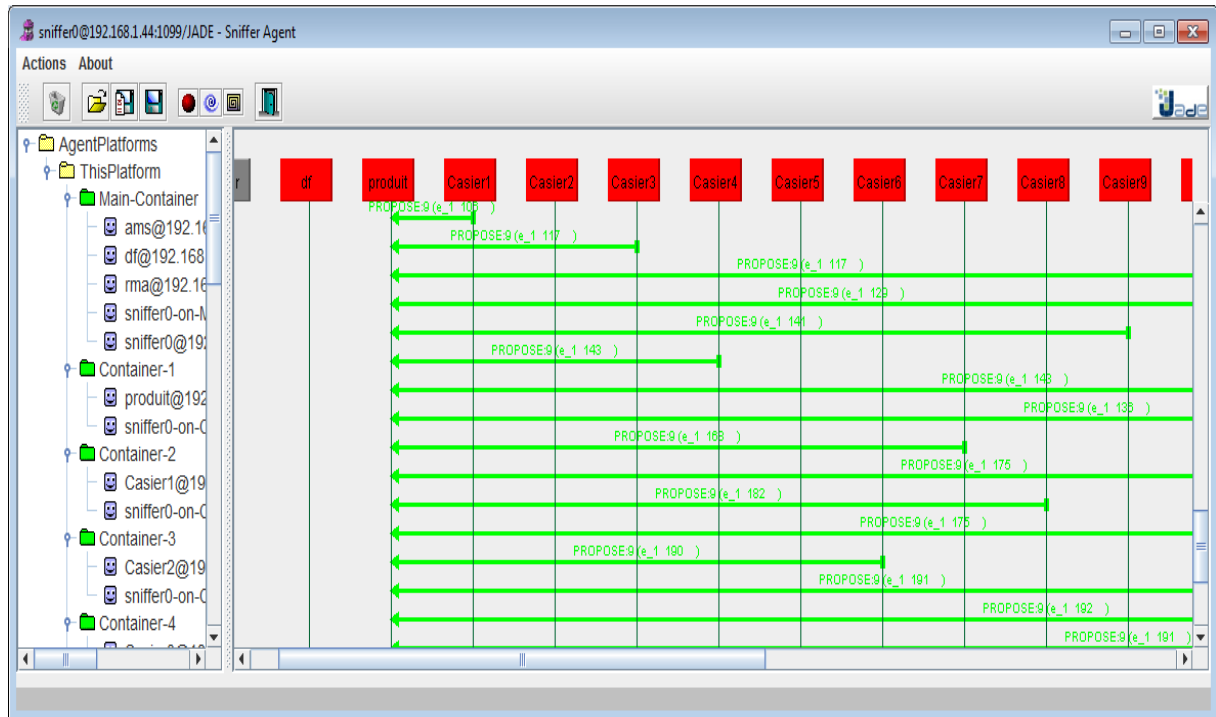
**Figure 27 Envoi de message sous Jade**

- La requête est envoyée à l'agent DF ; Ceci déclenchera un appel de propositions pour tous les casiers (des messages de type CFP : Call For Proposal). (Figure 28)
- Par la suite nous pourrions modifier le contenu du message ; il pourra contenir d'autres informations sur le produit souhaité ; tout dépendra du système étudié ; si par exemple il s'agit d'un rack contenant des médicaments la date de péremption aurait également son importance dans le contenu du message...



**Figure 28 Messages de type CFP (Call For Proposal)**

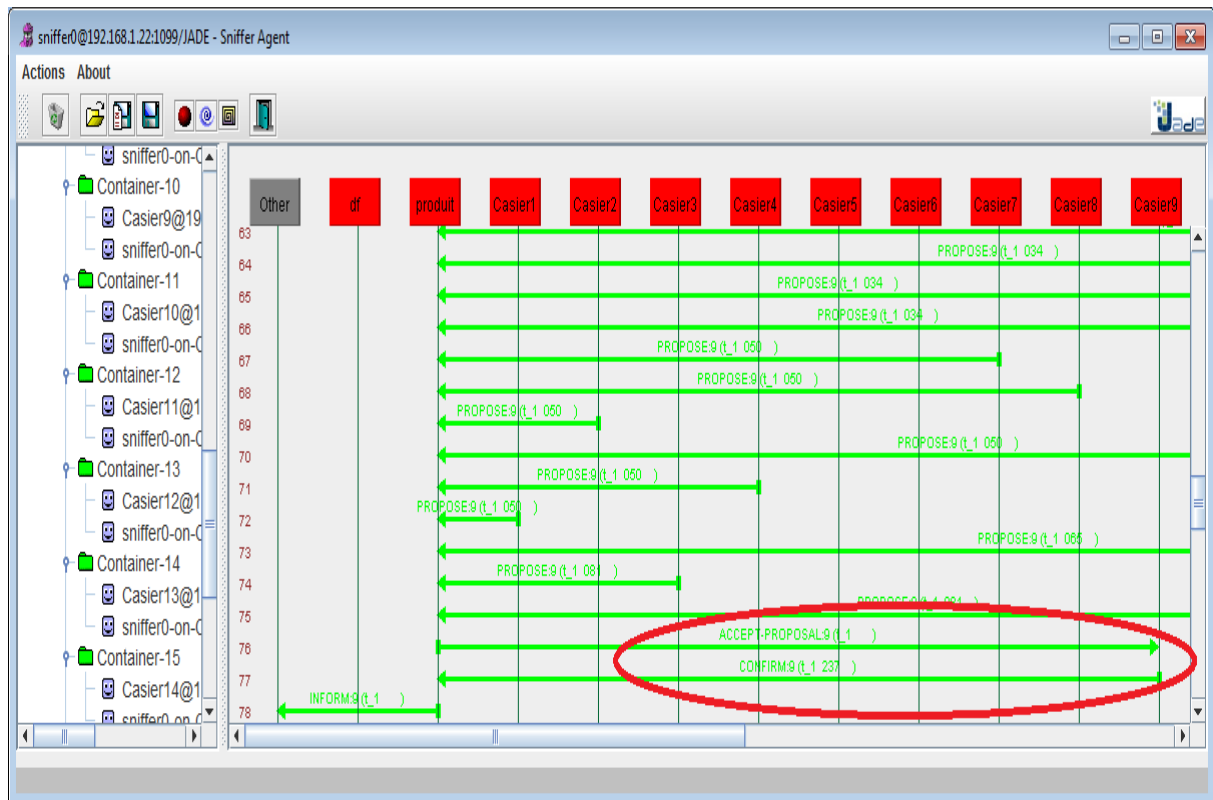
- Chacun des agents Casiers postera sa proposition (la métrique qu'il propose) ; et à partir de ces propositions l'agent produit choisira la meilleure (le meilleur casier d'où il devra être déstocké). (Figure 29)



**Figure 29 Messages de type propose**

- A partir de plusieurs messages reçus par l'agent produit ; ce dernier choisira le meilleur agent avec qui il procédera à une nouvelle communication (message de confirmation).

- Dans le cas réel plusieurs produits peuvent être déstockés simultanément ; dans notre application ceci revient à envoyer plusieurs requêtes de déstockage.



**Figure 30 Choix du meilleur casier et confirmation de la transaction**

Demandes simultanées de déstockage de produits

Dans le cas réel ; il arrive que nous souhaitons déstocker plusieurs produits à la fois.

Dans l'exemple qui suit, il y a eu envoi de trois requêtes simultanées.

La première demandant un produit bleu ; la seconde un produit orange et la troisième un produit rouge. En supposant qu'à chaque couleur est associée un certain type de produit. (Figure 31)

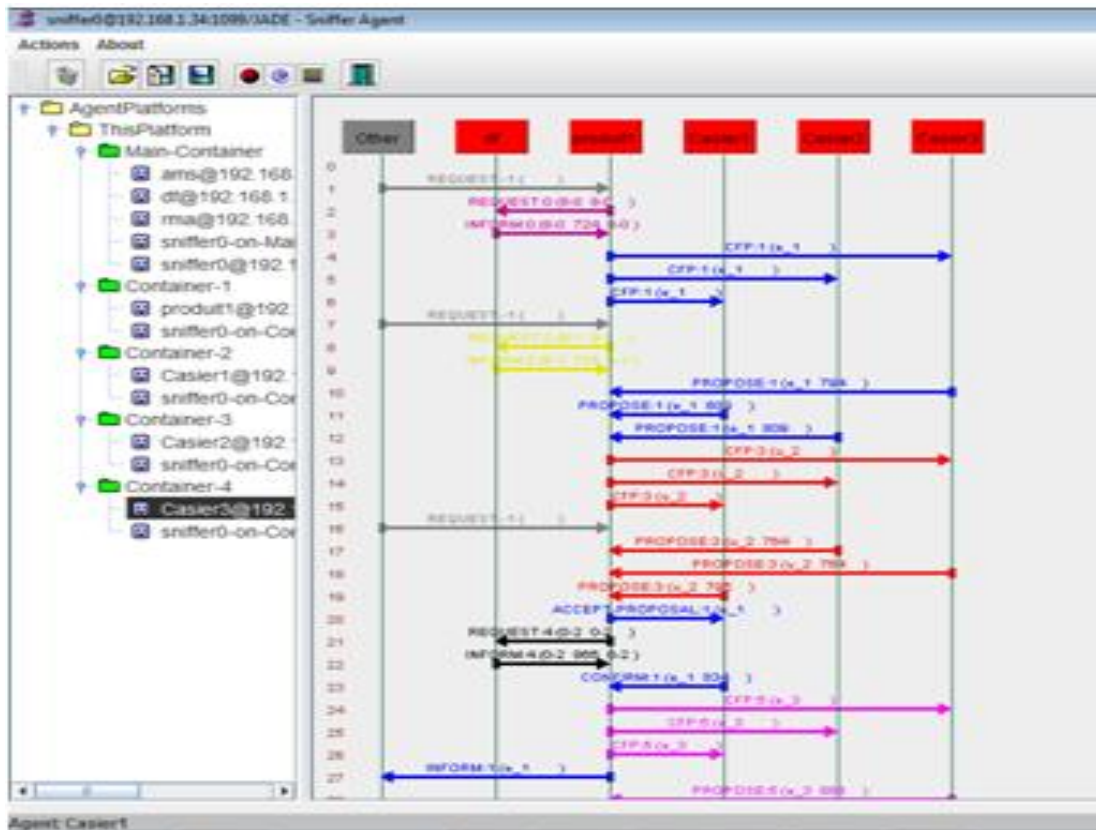


Figure 31 Demandes simultanées de déstockage de produits

## 9. Résultats

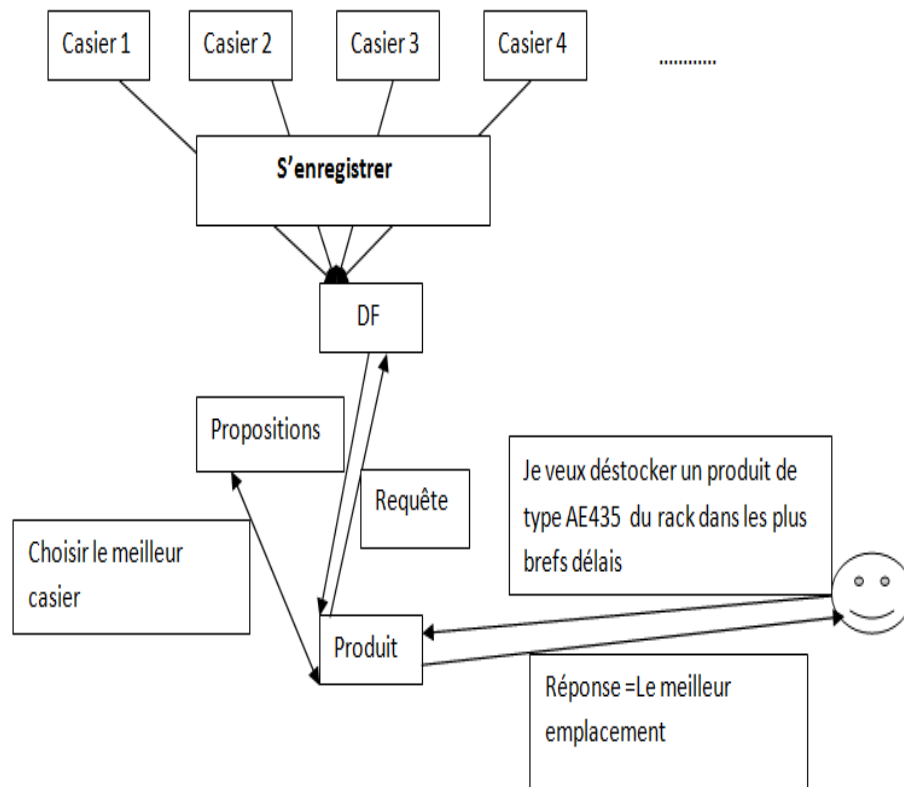
À sa création un agent Casier doit publier un service dans les pages jaunes. Ce service concerne Le déstockage de produits.

Chaque agent « Casier » possède une collection de type HashMap dont la clé représente la couleur (le type) du produit et dont la valeur représente la métrique de la règle de déstockage appliquée ; par exemple si nous souhaitons déstocker un produit rouge et que la métrique appliquée dans ce cas la représente le temps de déstockage minimal de chaque produit rouge dans chaque casier ; les métriques seront choisies aléatoirement dans un premier temps.

À la demande d'un agent « Organisateur », l'agent de type « Produit » commence par récupérer le type de produit à déstocker, passé comment argument au moment du déploiement ; ensuite, chaque minute, il tente de chercher dans le service des pages jaunes DF, les agents qui offrent le service de déstockage de produits.

Ensuite, il envoie un message de type CFP (Call For Proposal) à tous ces agents « Casiers », pour leur demander de lui fournir la proposition de la métrique du produit souhaité.

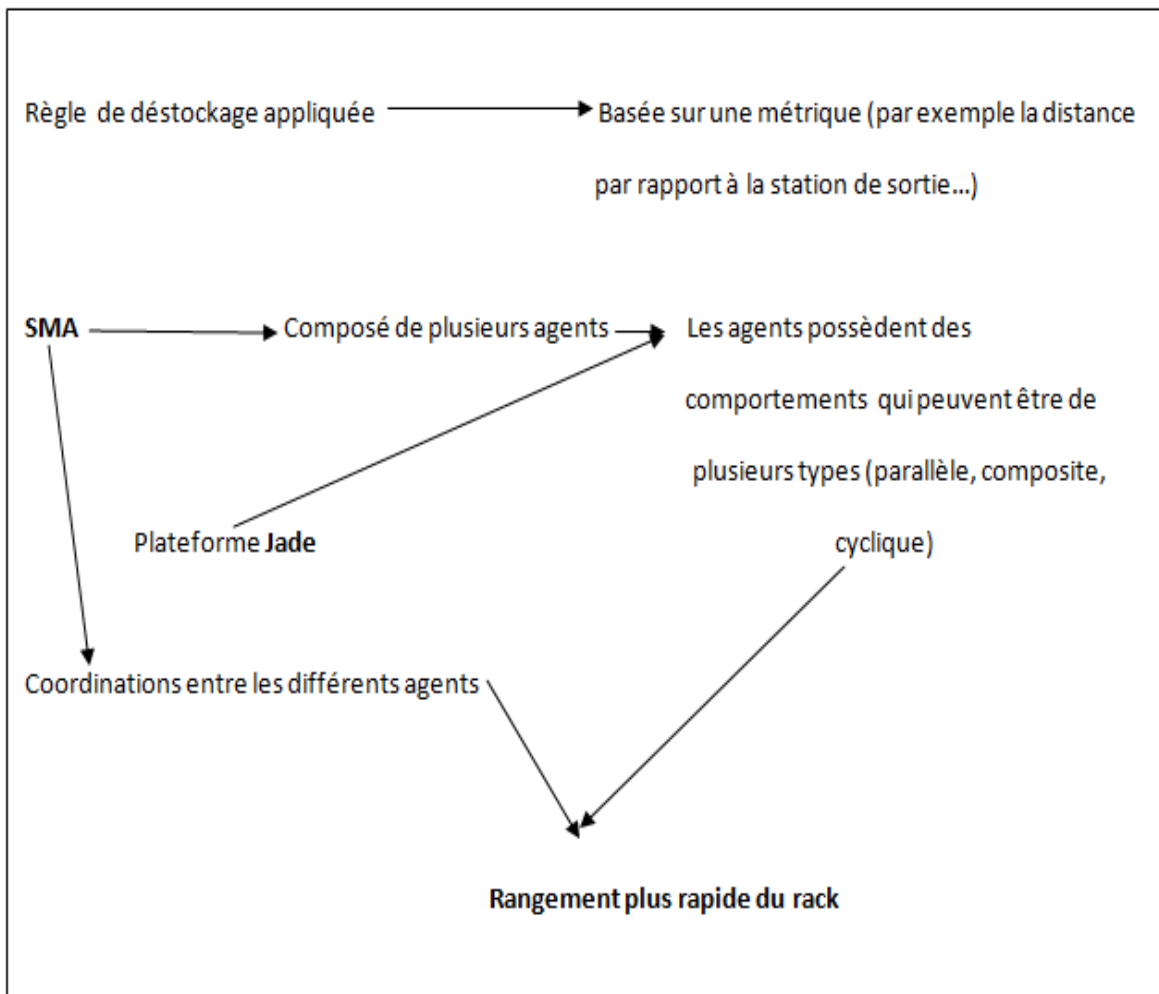
Si les propositions arrivent, il doit déterminer l'agent qui offre la meilleure métrique. Ensuite il envoie un message à cet agent pour finaliser le déstockage (Figure 32).



**Figure 32 Communications intelligentes entre les agents pour l'optimisation du rangement d'un rack**

- En utilisant la plateforme Multi-agents Jade ; nous sommes parvenus à programmer des comportements d'agents (Behaviours) qui peuvent se passer parallèlement. (Figure 31); par parallèle nous comprenons qu'après avoir invoqué la méthode action() d'un sous-Behaviour, nous pointons sur le suivant sans attendre que le premier termine son exécution. L'ajout de sous-Behaviour a été fait par la méthode addSubBehaviour() .
- Plusieurs requêtes de déstockage de produits peuvent être envoyées et traitées simultanément ce qui permet d'optimiser le temps de déstockage d'une liste de produits.
- Ce travail a montré une parfaite collaboration entre les différents agents pour atteindre le but final (rangement du magasin).
- Une certaine intelligence dans notre système résulte à partir des communications entre nos différents agents.





**Figure 33 Résultat de l'application**

- Nous avons établi une application de base dans l'objectif de choisir à chaque fois le meilleur produit à déstocker après demande du client ; ceci a été réalisé en tirant profit des avantages que peuvent offrir les SMA.

En utilisant la plateforme Jade les coordinations entre les différents agents du système peuvent être programmées ; à chaque type d'agent ont été affectés des comportements bien spécifiques et bien étudiés.

Cette application peut être considérée comme une référence de base que nous pourrions améliorer et adapter au système étudié.

## 10. Conclusion

Dans cette partie nous avons conçu une application multi-agents appliquée aux systèmes automatisés de stockage et déstockage ; celle-ci peut être considérée comme l'application de référence dont d'autres peuvent s'inspirer et la modifier selon les besoins du système étudié.

L'optimisation du temps de déstockage dans ce genre de système a déjà été faite par d'autres auteurs ; ceci dit ces derniers ont implémenté une heuristique pour résoudre le problème (des règles de stockage et de déstockage se basant sur des métriques mathématiques).

La nouveauté et l'originalité de ce travail réside dans la considération du système comme étant un SMA .ce qui a engendré des communications entre les différents agents de notre système ; ces derniers partagent des informations les concernant en se basant sur un langage bien spécifique : ACL.L'envoi de plusieurs messages de manière simultanée a été réalisé en utilisant la plateforme Jade ; à partir des comportements apportés aux agents.

Ces collaborations entre les agents permettraient sans aucun doute un gain de temps de déstockage très important.

Nos perspectives seront de :

- 1) Attribuer à chaque type de produits un code spécifique ; puisque le nombre de produits dans un magasin peut être très grand.
- 2) Définir notre propre ontologie pour la validation syntaxique des messages entre les agents casiers et les agents produits.
- 3) Programmer un protocole de coordination qui répond aux besoins de notre système.
- 4) Programmer plus de cas particuliers qui peuvent survenir dans notre application.

## Conclusion générale

Dans ce travail, nous nous sommes intéressés aux systèmes automatisés de stockage/déstockage (AS/RS). Nous avons étudié en particulier l'AS/RS à convoyeur gravitationnel.

Dans ce type d'AS/RS les casiers sont profonds et chacun est muni d'un convoyeur gravitationnel et de deux machines S/R. La première sur la face avant du rack pour le stockage et la seconde sur la face arrière pour le déstockage. Ces deux machines sont reliées entre elles par un convoyeur de restockage permettant à la machine S/R d'accéder aux produits pour les restocker.

Dans les AS/RS le temps de cycle est utilisé comme principale mesure de performance, étant égale à la somme de plusieurs temps parmi lesquels nous retrouvons bien entendu le temps de déstockage. Dans ce travail nous nous sommes justement intéressés à la minimisation de ce dernier en proposant une nouvelle technique ; celles des systèmes multi-agents.

Dans le premier chapitre, nous avons introduit le concept des SMA en donnant un aperçu général sur ce type de systèmes. Nous avons défini les notions d'agents et de système multi-agents, nous avons également établi les caractéristiques liés à ces systèmes.

Nous avons mis l'accent sur les différents moyens de communication entre les agents, plus particulièrement sur la communication entre les agents dans un environnement sans fil et qui semble d'ailleurs avoir un avenir très prometteur.

Les domaines d'application de ce type de systèmes (SMA) ont été étudiés, nous nous sommes particulièrement attardés sur le domaine industriel sur lequel nous travaillons justement, en établissant une partie état de l'art sur les différentes contributions qui ont été faites dans ce domaine en se basant sur les SMA.

Dans le second chapitre une présentation des systèmes automatisés de stockage/déstockage a été faite. Ainsi les définitions, fonctions, compositions ont été développées. Nous avons présenté une section exposant les différents types d'AS/RS suivie d'une section décrivant les emplacements de ces systèmes, ensuite une section pour les méthodologies de stockage, nous avons cité les principaux critères de mesures de performances des AS/RS et enfin un état de l'art détaillé regroupant les travaux les plus importants dans le domaine a été présenté.

Le troisième chapitre présente notre première contribution qui consiste à considérer le système étudié (l'AS/RS à convoyeur gravitationnel) en tant que système multi-agents réactifs.

Plusieurs simulations sous Netlogo ont été faites afin de pouvoir trouver la meilleure combinaison de règles de stockage et de déstockage.

Nous mesurons le temps de déplacement moyen de la machine de déstockage qui dans notre cas s'avère être égale au temps de déstockage ; en partant à chaque fois d'un état initial du magasin.

Chaque courbe obtenue modélise cinquante temps de déplacement de la machine de déstockage ; nous nous sommes contentés de calculer cinquante temps de déplacement, car cela a suffi pour avoir des résultats intéressants et interprétables. Cependant ce nombre peut être augmenté.

Les vitesses horizontale et verticale de la machine S/R sont représentées par des variables globales dans l'interface principale de notre programme, on peut changer leurs valeurs à n'importe quel moment. Nous leur attribuons par exemple la valeur 1.

La deuxième contribution de cette thèse, consiste en une nouvelle solution permettant d'optimiser le temps de déstockage.

Dans une première étape nous avons commencé par la modélisation de notre système ,en nous basant sur le langage UML « Unified Modeling Language » ;pour cela nous avons eu besoin de deux types de diagrammes :des diagrammes de séquences pour définir les communications qui vont être déclenchées entre les différents agents et des diagrammes de classes .

Après l'étape « modélisation » nous sommes passés à la programmation des différentes classes Agents sous java. Sous la plateforme Jade les différents agents communiquent ; après demande de déstockage d'un produit il y a déclenchement de plusieurs messages, la requête est envoyée à tous les agents casiers leur demandant des propositions d'offres, pour trancher ensuite et choisir la meilleure proposition c.-à-d. le casier offrant la valeur minimale de la métrique.

Plusieurs requêtes de déstockage peuvent être envoyées et traitées à la fois, cette partie avec jade montre l'intérêt que peuvent avoir les interactions entre les agents.

Nos perspectives seront de :

- 1- Programmer plus de règles de stockage et de déstockage dans la première contribution.
- 2- Reprogrammer la même chose pour les autres rangées du rack (vision 3D).
- 3- Attribuer à chaque type de produits un code spécifique ; puisque le nombre de produits dans un magasin peut être très grand.
- 4- Définir notre propre ontologie pour la validation syntaxique des messages entre les agents casiers et les agents produits.
- 5- Programmer un protocole de coordination qui répond aux besoins de notre système.
- 6- Programmer plus de cas particuliers qui peuvent survenir dans notre application.
- 7- Reprendre la même chose de ce qui a été réalisé sous Netlogo avec jade et établir une comparaison entre les deux.

## Références

- [Anane 2009] D. Anane, S. Pinson, S. Aknine, « Les approches agents pour la coordination d'activités dans les chaînes logistiques ». Cahier du Lamsade 291, Université Paris-Dauphine, Décembre 2009.
- [Ashayeri 2002] J. Ashayeri, RM. Heuts, MWT. Valkenburg, HC. Veraarts, MR. International Journal of Production Research, 40(17), p. 4467- 4483, 2002.
- [Atalla 2011] F. Sayda Atalla, « Multi-agent Systems for Industrial Applications: Design, Development, and Challenges, Multi-Agent Systems - Modeling, Control, Programming, Simulations Wilhelm, « A geometrical approach to computing expected cycle times for zone-based storage layouts in AS/RS ». and Applications ». Dr. Faisal Alkhateeb (Ed.), ISBN: 978-953-307-174-9, InTech, Available from: <http://www.intechopen.com/books/multi-agentsystems-modeling-control-programming-simulations-and-applications/multi-agent-systems-for-industrialapplications-design-development-and-challenges>, 2011.
- [Axelrod 1997] R. Axelrod, « Advancing the art of simulation in the social sciences ». Simulating Social Phenomena, Lecture Notes in Economics and Mathematical Systems, vol. 456, Edité par R. Conte, R. Hegselmann et P. Terna, Springer, p. 21- 40, 1997.
- [Azzouz 2001a] A. Azzouz, Z. Sari, N. Ghouali, « La méthode Branch and Bound appliquée à l'optimisation des dimensions d'un AS/RS à convoyeur gravitationnel ». Conférence Internationale sur la Productique, Alger, 9-11 juin 2001.
- [Azzouz 2001b] A. Azzouz, Z. Sari, N. Ghouali, « Une synthèse sur l'optimisation des dimensions d'un AS/RS à convoyeur gravitationnel ». Conférence Internationale sur la Conception et la Production Intégrée, Fès, 24-26 octobre 2001.
- [Benmammar 2009] B. Benmammar, « Systèmes Multi-Agents ». Ecole d'ingénieur, 2009.
- [Blecker 2003] T. Blecker et G. Graf, « Multi agent systems in internet based production environments-an enabling infrastructure for mass customization ». Paper presented at the Second Interdisciplinary World Congress on Mass Customization and Personalization, p.1–27). Munich, Germany, In proceedings, 2003.
- [Boissier 1997] O. Boissier and Y. Demazeau. « Une architecture multi-agents pour des systèmes de vision ouverts et décentralisés ». Technique et Science Informatiques. Numéro

spécial sur l'Intelligence Artificielle Distribuée et des Systèmes Multi-Agents, 16(18), p. 1039 - 1062, 1997.

**[Bozer 1980]** Y.A. Bozer, J.A. White, « Optimum designs of automated storage/retrieval systems ». TIMS/ORSA Joint National Meeting, Washington, DC, 1980.

**[Bozer 1984]** Y.A. Bozer, J.A. White, « Travel time for automated storage/ retrieval systems ». IIE transactions, 16(4), p. 329 - 338, 1984.

**[Bozer 1996]** Y.A. Bozer, J.A. White, « A generalized design and performance analysis model for end-of aisle order picking systems ». IIE Transactions, 28(4), p. 271-280, 1996.

**[Caron 2000]** F. Caron, G. Marchet and A. Perego, « Optimal layout in low-level picker-to-part systems ». International Journal of Production Research, 38(1), p. 101-117, 2000.

**[Chaib-draa 1995]** B. Chaib-draa, « Industrial Applications of DAI ». Communication of ACM, 38(11), p. 49-53, 1995.

**[Chang 1997]** D.T. Chang, U.P. Wen, « The impact on rack configuration on the speed profile of the storage and retrieval machine ». IIE Transactions, 29, p. 525-531, 1997.

**[Chew 1999]** E.P. Chew and L.C. Tang, « Travel time analysis for general item location assignment in a rectangular warehouse ». European Journal of Operation Research, 112. p. 582 - 597, 1999.

**[Clair 2008]** G. Clair, « Gestion de production par système multi-agent auto-organisateur ». Rapport de Master 2 Recherche Web Intelligence (WI), Université Jean Monnet, 2008.

**[Dallari 2000]** F. Dallari, G. Marchet, R. Ruggeri, « Optimization of man-on-board automated storage/ retrieval systems ». Integrated Manufacturing Systems, 11(2), p. 87-93, 2000.

**[De Koster 2007]** R. De Koster, T. Le-Duc, K.J. Roodbergen, « Design and control of warehouse order picking: a literature review ». European Journal of Operational Research, 182(2), p. 481-501, 2007.

**[De Koster 2008]** M.B.M. De Koster, T. Le-ANH, and Y. YU, « Optimal storage rack design for a 3-dimensional compact AS/RS ». International Journal of Production Research, 46(6), p. 1495-1514, 2008.

- [**Demazeau 1995**] Y. Demazeau. « From interactions to collective behaviour in agent-based systems ». Proceedings of the First European conference on cognitive science, pages 117–132, Saint Malo, France, 1995.
- [**Demazeau 1998**] Y. Demazeau and M. Ocelllo. « Software systems development as societies of agents ». 15th IFIP Conference (IT&KNOWS'98), p. 303-304, Vienna, September 1998.
- [**Derks 1993**] R.P. Derks, « Purpose-driven product and process design ». Industrial Engineering, 25 (1), p. 38- 42, 1993.
- [**Doniec 2008**] A. Doniec, R. Mandiau, S. Piechowiak et S. Espié, « A behavioral multi-agent model for road traffic simulation ». Engineering Applications of Artificial Intelligence, 21(8), p. 1443-1454, décembre 2008.
- [**Drogoul 2000**] A. Drogoul. « Systèmes multi-agents situés », Université Pierre et Marie Curie, 2000.
- [**Durfee 1989**] E.H. Durfree, V.R. Lesser et D.D. Corkill, « Trends in cooperative Problem Solving ». IEEE Transaction on knowledge and Data Engineering, KDE-1(1), p. 63
- [**Durfee 1995**] E.H. Durfee, « Distributed Artificial Intelligent ». in The handbook of brain Theory and Neural Networks, M. Arbib (eds), MIT Press, p. 232-239, 1995.
- [**Edmonds 2003**] B. Edmonds, D. Hales, « Replication, replication and replication : Some hard lessons from model alignment ». Journal of Artificial Societies and Social Simulations, 6(4), 2003.
- [**Edmonds, 2005**] B. Edmonds, « Simulation and complexity – how they can relate ». Virtual World of Precision. Computer-based Simulations in the Sciences and Social Sciences, édité par V. Feldmann et K. Mühlfeld, Lit Verlag, p. 5-32, 2005.
- [**Egbelu 1991**] P.J. Egbelu, « Framework for dynamic positioning of storage/retrieval machines in an automated storage/ retrieval system ». International Journal of Production Research, 29(1), p. 17-37, 1991.
- [**Egbelu 1993**] P.J. Egbelu, C.T. Wu, « A comparison of dwell point rules in an automated storage/retrieval system ». International Journal of Production Research, 31(11), p. 2515-2530, 1993.



[**Eldemi 2004**] F. Eldemi, R.J. Graves, C.J. Malmborg, « New cycle time and space estimation models for automated storage and retrieval system conceptualization ». International Journal of Production Research, 42(22), p. 4767- 4783, 2004.

[**Elsayed 1996**] E.A. Elsayed, M.K. Lee, « Order processing in automated storage/retrieval systems with due dates ». IIE Transactions, 28, p. 567-577, 1996.

[**Erwan 2001**] T. Erwan, « IAD et ordonnancement : une approche coopérative du réordonnancement par systèmes multi-agents ». Thèse de Doctorat, Université de Droit, d'Economie et des Sciences d'Aix-Marseille III, Mai 2001.

[**Eynan 1994**] A. Eynan, M.J. Rosenblatt, « Establishing zones in single-command class-based rectangular AS/RS ». IIE Transactions, 26(1), p. 38- 46, 1994.

[**Fabien 2004**] M. Fabien, « Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents ». Thèse de Doctorat Université des Sciences et Techniques du Languedoc, Décembre 2004

[**Ferber 1995**] J. Ferber, « Les systèmes multi-agents : vers une intelligence collective ». 1995.

[**Ferber 1999**] J. Ferber, « Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence ». Addison-Wesley Longman Publishing Co., Inc., 1999.

[**FIPA 2002**] The FIPA ACL Message Structure Specifications, 2002.

[**Florez 1999**] R. Florez, « Towards a standardization of multi-agent system framework ». ACM Crosswords Student Magazine, 1999.

[**Fritzson 1994**] R. Fritzson, T. Finin, D. McKay, R. McEntire, « KQML as an Agent Communication Language ». Proceedings of the third International Conference on Informatique Management (CIKM'94), ACM Press, Nov. 1994.

[**Gaouar 2004**] N. Gaouar, « Amélioration des performances d'un AS/RS à convoyeur gravitationnel ». Thèse de Magister, Université de Tlemcen Algérie, 2004.

[**Gazdar, 2008**] M. Gazdar, « Optimisation Heuristique Distribuée du Problème de Stockage de Conteneurs dans un Port ». Thèse de Doctorat, Ecole Centrale de Lille, 2008.

- [**Ghomri 2008**] L. Ghomri, Z. Sari, A. Guezzan, T. Sari, « Modèle analytique du temps de simple cycle d'un AS/RS multi allées ». Conférence Internationale de Modélisation et Simulation, Paris, 31 mars-02 avril 2008.
- [**Ghomri 2008**] L. Ghomri, Z. Sari, A. Guezzan, T. Sari, « Modèle analytique du temps de simple cycle d'un AS/RS multi allées ». Conférence Internationale de Modélisation et Simulation, Paris, 31 mars - 02 avril 2008.
- [**Gilbert 2007**] N. Gilbert, « Computational Social Science ». Agent-Based Modelling and Simulation in the Social and Human Sciences, Bardwell Press, 2007.
- [**Graves 1977**] S.C. Graves, W.H. Hausman, L.B. Schwarz, « Storage-retrieval interleaving in automatic warehousing systems ». Management Science, 23(9), p. 935-945, 1977.
- [**Gue 2007**] K.R. Gue, B.S. Kim, « Puzzle-based storage systems ». Naval Research Logistics, 54 (5), p. 556-567, 2007.
- [**Guenov 1992**] M. Guenov, R. Raeside, « Zone shapes in class based storage and multi-command order picking when storage/retrieval machines are used ». European Journal of Operational Research, 58, p. 37-47, 1992.
- [**Guezzan 2011a**] A. Guezzan (a), Z. Sari, L. Ghomri, « A Study on Mobil Racks Automated Storage and Retrieval System (M-AS/RS) ». Actes Conférence CCCA' 2011, Hammamet, 2011.
- [**Guezzan 2011b**] A. Guezzan (b), Z. Sari, L. Ghomri, « Continuous Model for Single Cycle Times of a Mobile Racks Automated Storage and Retrieval System ». Actes Conférence MISC' 2011.
- [**Ha 1994**] J.W. Ha, H. Hwang, « Class-based storage assignment policy in carousel system ». Computers and Industrial Engineering, 26(3), p. 489 – 499, 1994.
- [**Halioui 2010**] H. Halioui, « Spécification formelle de la négociation avec les SMA ». <hal-00701892>, 2010.
- [**Hausman 1976**] W.H. Hausman, L.B Schwarz, and S.C Graves, « Optimal storage assignment in automatic warehousing systems ». Management Science, 22(6), p. 629-638, 1976.

- [**Hutzler 2004**] G. Hutzler, B. Gortais, « From computer art to ambient displays ». *Machine Graphics and Vision*, 13, p. 181-191, 2004.
- [**Hwang 1988**] H. Hwang, K.H. Ko, « A study on multi-aisle system served by a single storage/retrieval machine ». *International Journal of Production Research*, 26(11), p. 1727-1737, 1988.
- [**Hwang 1990**] H. Hwang, S.B. Lee, « Travel-time models considering the operating characteristics of the storage/retrieval system ». *International Journal of Production Research*, 28(10), p. 1779-1789, 1990.
- [**Hwang 1993**] H. Hwang, J.M. Lim, « Deriving an optimal dwell point of the storage/retrieval machine in an automated storage/retrieval system ». *International Journal of Production Research*, 31(11), p. 2591-2602, 1993.
- [**Hwang 1999**] H. Hwang, C.S. Kim, and K.H. Ko, « Performance analysis of carousel systems with double shuttle ». *Computers & Industrial Engineering*, 36(2), p. 473-485, 1999.
- [**Jamont 2005**] J.P. Jamont Diamond, « Une Approche pour la conception de Systèmes Multi-agents Embarqués ». PhD Thesis, INPG, Septembre 2005.
- [**Jennings 1993**] N.R. Jennings, « Specification and implementation of a belief-desire-joint-intention architecture for collaboration problem solving ». *International Journal of Intelligent Cooperative Information Systems*, 2(3), p. 289 - 318, 1993.
- [**Jennings 2000**] N.R. Jennings, « A classification scheme for negotiation in electronic commerce », EPSRC, 2000.
- [**Kazuhiro 1995**] K. Kazuhiro, I. Toru, O. Nobuyasu, « AgenTalk: Coordination Protocol Description for Multi-Agent Systems ». *First International Conference on Multi-Agent Systems (ICMAS '95)*, 1995.
- [**Keserla 1994**] A. Keserla, B.A. Peters, « An Analysis of Dual Shuttle Automated Storage/Retrieval Systems ». Working paper, Dept. Of Industrial Engineering, Texas A&M University, 1994.
- [**Kouiss 1995**] K. Kouiss, H. Pierreval, « Systèmes multi-agents: direction actuelles pour les systèmes de productions ». *Congrès International de Génie Industriel, La productivité dans un monde sans frontière*, Montréal, Canada, 18 (2), p. 2029-2039, Octobre 1995.

**[Kouloughli 2010]** S. Kouloughli, Z. Sari, T. Sari, « Optimisation des dimensions d'un AS/RS multi allées basée sur un modèle analytique du temps de simple cycle ». *Journal Européen des Systèmes Automatisés*, 44(2), p. 135-160, 2010.

**[Kouloughli 2013]** S. Kouloughli, « Optimisation de systèmes automatisés de stockage/déstockage multi-allées et à racks glissants ». Thèse de Doctorat, Université de Tlemcen Algérie, 2013.

**[Kouloughli 2015]** S. Kouloughli, Z. Sari, « Multi-aisle AS/RS dimensions optimization for cycle time minimization ». *International Journal of Advanced Manufacturing Technology IJAMT*, (76), DOI 10.1007/s00170-014-6709-3, ISSN 0268-3768, 2015.

**[Kouvelis 1995]** P. Kouvelis, V. Papanicolaou, « Expected travel time and optimal boundary formulas for a two-class-based automated storage/retrieval system ». *International Journal of Production Research*, 33(10), p. 2889-2905, 1995.

**[Le-Duc 2004]** T. Le-Duc, R. De Koster, « Optimal Storage Rack Design for a 3 dimensional Compact AS/RS ». *ERIM Research Program: Business Processes, Logistics and Information Systems*, 2004.

**[Le-Duc 2005]** T. Le-Duc, R. De Koster, « Travel distance estimation and storage zone optimisation in a 2-block ABC-storage strategy warehouse ». *International Journal of Production research*, 43(17), p. 3561-3581, 2005.

**[Le-Duc 2006]** T. Le-Duc, and R. De Koster, « Travel time estimation and order batching in a 2- block warehouse ». Report, Erasmus Research Institute of Management (ERIM), RSM Erasmus University, the Netherlands (in revision for the *European Journal of Operational Research*), 2006.

**[Lee 1992]** M.K. Lee, « A storage assignment policy in a man-on-board automated storage/retrieval system ». *International Journal Production Research*, 30 (10), p. 2281-2292, 1992.

**[Lee 1995]** M.K. Lee, S.Y. Kim, « Scheduling of storage/retrieval orders under a just-in-time environment ». *International Journal Production Research*, 33 (12), p.3331 - 3348, 1995.

**[Lee 1996]** H.F. Lee, S.K. Schaefer, « Retrieval sequencing for unit-load automated storage and retrieval systems with multiple openings ». *International Journal of Production Research*, 34(10), p. 2943-2962, 1996.

[**Lee 2005**] H.L. Lee, M.H. Lee, L.S. Hur, « Optimal design of rack structure with modular cell in AS/RS ». *International Journal of Production Economics*, 2005.

[**Lehrer 2010**] T. Lerher, M. Sraml, I. Potrc, T. Tollazzi, « Travel time models for double-deep automated storage and retrieval systems ». *International Journal of Production Research* 48, 11, p. 3151-3172, DOI : 10.1080/00207540902796008, 2010.

[**Lerher 2005**] T. Lerher, M. Sraml, J. Kramberger, I. Potrc, M. Borovinsek, B. Zmazek, « Analytical travel time models for multi aisle automated storage and retrieval systems ». *International Journal Advanced Manufacturing Technology*, 30(3), p. 340-356, 2005.

[**Mahajan 1998**] S. Mahajan, B.V. Rao, B.A. Peters, « A retrieval sequencing heuristic for miniload end-of-aisle automated storage/retrieval systems ». *International Journal of Production Research*, 36(6), p. 1715-1731, 1998.

[**Mahesh 2006**] M. Mahesh, S.K. Ong, A.Y.C. Nee, « Communication and coordination in Multi-Agent Manufacturing System ». *International Journal of Manufacturing Research*, 1(1), p. 59-82, 2006.

[**Malmborg 2000**] C.J. Malmborg, « Interleaving models for the analysis of twin shuttle automated storage and retrieval systems ». *International Journal Production Research*, 38 (18), p. 4599 - 4614, 2000.

[**Malmborg 2001**] C.J. Malmborg, « Rule of thumb heuristics for configuring storage racks in automated storage and retrieval systems design ». *International Journal Production Research*, 39 (3), p. 511- 527, 2001.

[**Malone 1994**] T.W. Malone, K. Crowston, « The Indisciplinary Study of Coordination ». *ACM Computing Surveys*, 26(1), p. 87-119, 1994.

[**Mansour 2014**] W. Mansour, K. Jelassi, « Flexible Software solution for intelligent multi-agent manufacturing systems ». *International Conference Electrical Sciences and Technologies in Maghreb (CISTEM)*, 2014.

[**Mansuri 1991**] M. Mansouri, « Simulation model development to analyze, evaluate and design storage allocation and AS/RS operation ». *Simulation*, 57 (4), p. 231-239, 1991.

[**Marcenac 1996**] P. Marcenac, « Emergence of behaviors in natural phenomena agent-simulation ». *Complexity International Review*, 3, 1996.

**[Material Handling Institute]** Material Handling Institute « Consideration for planning and installing an automated storage/retrieval systems ». Inc AS/RS document, 100 7M, 1977.

**[Moyaux 2001]** T. Moyaux, « Comment les SMA pourraient aider à la gestion de la production, de la distribution et des stocks ». Rapport d'un cours, Université Laval, 2001.

**[Muppani 2008]** V.R. Muppani, G.K. Adil, « A branch and bound algorithm for class based storage location assignment ». European Journal of Operational Research, 189 (2), p. 492-507, 2008.

**[Pannequin 2006]** R. Pannequin, A. Thomas, « Proposition d'une plateforme d'expérimentation sur le contrôle par le produit des flux de production ». Conférence francophone de modélisation et de simulation MOSIM'06, Rabat Maroc, 2006.

**[Park 2001]** B.C. Park, « An optimal dwell point for automated storage/ retrieval systems with uniformly distributed, rectangular racks ». International Journal of Production Research, 39(7), p. 1469-1480, 2001.

**[Park 2006]** B.C. Park, « Performance of automated storage/retrieval systems with non square- in-time racks and two-class storage ». International Journal of Production Research, 44 (6), p. 1107-1123, 2006.

**[Parunak 1996]** H.V.D. Parunak, « Applications of Distributed Artificial Intelligence in Industry ». Foundations of Distributed Artificial Intelligence, Chap. 4, G.M.P. O'Hare et N.R. Jennings eds., J. Wiley and sons Inc, p.139-164, 1996.

**[Parunak 1998]** H.V.D. Parunak, « What can Agents do in Industry, and Why ? An Overview of Industrial-Oriented RetD at CEC ». CIA'1998.

**[Peters 1996]** B.A. Peters, J.S. Smith, T.S. Hale, « Closed form models for determining the optimal dwell point location in automated storage and retrieval systems». International Journal of Production Research, 34(6), p. 1757-1771, 1996.

**[Picault 2013]** S. Picault, « De la simulation multi-agents à la simulation multi-niveaux Pour une réification des interactions ». Mémoire d'Habilitation, Université de Lille 1,2013.

**[Randhawa 1995]** S.U. Randhawa, R. Shroff, « Simulation-based design evaluation of unit load automated storage/retrieval systems ». Computers & Industrial Engineering, 28(1), p. 71-79, 1995.

[**Rao 1995**] A. Rao, M. Georgeff, « Bdi-agents : from theory to practice ». Proceedings of the First Intl Conference on Multiagent Systems, San Francisco, 1995.

[**Reguieg 2010**] S. Reguieg, « Elaboration d'un protocole de coordination dans un SMA pour la gestion de production dynamique : vers une approche décentralisée ». Thèse de Magister, Université d'Oran, 2010.

[**Robinson 2009**] S. Robinson, « Conceptual modelling for simulation part I : Definition and requirements ». Journal of the Operational Research Society, 59(3), p. 278-290, 2009.

[**Roodbergen 2001**] K.J. Roodbergen, R. De Koster, « Routing methods for warehouses with multiple cross aisles ». International Journal of Production Research, 39(9), p. 1865-1883, 2001.

[**Rosenblatt 1989**] M.J. Rosenblatt, A. Eynan, « Deriving the optimal boundaries for class-based automatic storage/retrieval systems ». Management Science, 35(12), p. 1519-1524, 1989.

[**Rosenblatt 1993**] M.J. Rosenblatt, Y. Roll, V. Zyser, « A combined optimization and simulation approach for designing automated storage/Retrieval systems ». IIE Trans, 25 (1), p. 40-50, 1993.

[**Russell 2003**] S.J. Russell, P. Norvig, « Artificial Intelligence: A Modern ...».

[**Sari 1998**] Z. Sari, « Expressions analytiques discrètes de simple et double cycle d'un AS/RS multi-allées ». Rapport interne, LAT, Université Abou-Bekr Belkaïd, Tlemcen, Algérie, 1998.

[**Sari 2003**] Z. Sari, « Modélisation, analyse et évaluation des performances d'un AS/RS à convoyeur gravitationnel ». Thèse de Doctorat d'Etat, Université de Tlemcen Algérie, 2003.

[**Sari 2005**] Z. Sari, C. Saygin, N. Ghouali, « Travel-time models for flow-rack automated storage and retrieval systems ». International journal Advanced Manufacturing Technology, 25, p. 979-987, 2005.

[**Sari 2010**] Z. Sari, S. Kouloughli, « Modélisation des temps de cycle d'un AS/RS à convoyeur gravitationnel à une seule machine S/R ». Conférence Internationale de Modélisation et Simulation, Hammamet Tunisie, 10-12 mai 2010.

[**Searle 1972**] J.R. Searle, « Les actes de langage ». Paris, Herman, 1972.

[**Shen 2001**] W. Shen, H.N. Douglas, J.P. Barthes, « Multiagent systems for concurrent intelligent design and manufacturing ». Taylor and Francis Pbs, 2001.

[**Taghezout 2012**] N. Taghezout, « Conception et développement d'un système multi-agent d'aide à la décision pour la gestion de production dynamique ». Thèse de Doctorat, Université Paul Sébastien de Toulouse, 2012.

[**Trevino 1995**] J. Trevino, F. Liao, S. Chopra, « Cardes – carousel design ». International Journal Production Research, 32 (5), p. 1013-1026, 1995.

[**Van den Berg 1996**] J.P. Van den Berg, « Class-based storage allocation in a single command warehouse with space requirement constraints ». International Journal of Industrial Engineering, 3(1), p. 21- 28, 1996.

[**Van Den Berg 1999**] J.P. Van Den Berg, A.J.R.M. Gademann, « Optimal routing in an automated storage/retrieval system with dedicated storage ». IIE Transaction, 31 (5), p. 407 - 415, 1999.

[**Van Den Berg 2000**] J.P. Van Den Berg, AJRM. Gademann, « Simulation study of an automated storage/retrieval system ». International Journal Production Research, 38 (6), p. 1339 -1356, 2000.

[**Van den Berg 2002**] J.P. Van den Berg, « Analytic expressions for the optimal dwell point in an automated storage/retrieval system ». International Journal of Production Economics, 76, p. 13-25, 2002.

[**Wenrong 2014**] L. Wenrong, G. Vaggelis, M. Duncan, H.O. James, « The Role of Distributed Intelligence in Warehouse Management Systems ». p. 63-77, 2014.

[**Yu 2008**] Y. Yu, M.B.M. De Koster, « Designing an optimal turnover-based storage rack for a 3D compact AS/RS ». International Journal of Production Research, forthcoming, 2008.



**Résumé** - Les entreprises évoluent dans un environnement de plus en plus compétitif et exigeant en termes de flexibilité et de réactivité. L'utilisation de modèles issus de l'intelligence artificielle distribuée (IAD) et plus particulièrement des systèmes multi agents (SMA) dans les outils de gestion des entreprises s'avère être efficace pour simuler et reproduire les comportements collaboratifs et adaptatifs tels qu'ils apparaissent actuellement dans les entreprises.

Les AS/RS suscitent actuellement un intérêt de plus en plus grandissant, en raison des avantages importants qu'ils présentent ; à savoir : un faible coût au niveau de la main-d'œuvre, des dépenses revues à la baisse pour le stockage, une meilleure exploitation de l'espace de stockage, une meilleure traçabilité des produits stockés, un débit optimisé du système et plus de sécurité. Ils ont donc une importance capitale dans la réussite des défis de la nouvelle industrie. Malheureusement, mis à part certains types d'AS/RS (par exemple à charge unitaire), ils ne sont pas beaucoup étudiés, bien qu'ils soient très utilisés dans l'industrie et les services, Ce manque d'intérêt pour les AS/RS particuliers tel que l'AS/RS à convoyeur gravitationnel, fait qu'ils ne sont utilisés que pour des applications particulières et très en deçà de leurs possibilités. [Sari 2003]

Dans le cadre de cette thèse ; nous nous sommes focalisés sur l'apport des systèmes multi agents dans les systèmes automatisés de stockage/déstockage (AS/RS).

Dans une première contribution nous nous sommes intéressés aux AS/RS à convoyeur gravitationnel ; plus particulièrement au stockage et déstockage des produits dans un rack. La technique employée est celle des Systèmes Multi-Agents (SMA) réactifs, ou les agents se basent sur des combinaisons de règles de stockage et de déstockage. Les résultats obtenus après plusieurs simulations ont menés à établir une deuxième contribution basée sur des agents intelligents pouvant communiquer, dans le but de minimiser en permanence le temps de déstockage global.

**Abstract-** Companies evolve in an increasingly competitive and demanding environment in terms of flexibility and reactivity. The use of models resulting from distributed artificial intelligence (DAI) and more particularly multi-agent systems (MAS) in companies' management tools proves to be efficient for simulating and reproducing the collaborative and adaptive behavior as they now appear in companies.

AS/RS are currently rising an increasingly growing interest, because of the significant advantages they present; such as: labors low cost, downgraded expenses for storage, a better use of the storage space, a better tracking of the stored products, an optimized flow of the system and also offered better security. Therefore, they have a critical importance in the success and overcome of the new industry challenges.

Unfortunately, except some types of AS/RS (for example those with unit load), they have not been studied enough, although they are widely used in the industry and services. This lack of interest for particular AS/RS such as AS/RS with gravitational conveyor, explain the fact that they are used only for particular applications and well below their potential.

In this thesis; we focus on the multi-agent systems contribution in the storage/retrieval automated systems (AS/RS).As a first contribution, we focus on the AS/RS with gravitational conveyor; more particularly on the storage and retrieval of the products in a rack. The used technique is the technique of the multi-agents reactive systems, where the agents are based on the combination of the storage/retrieval rules. The results obtained after many simulations allowed to establish a second contribution based on intelligent agent which have the ability of communication, in order to minimize permanently the global retrieval time.

**ملخص**- حاليًا الشركات تعمل في بيئة متزايدة التنافس و المتطلبات من حيث القدرة على الاستجابة. استخدام نماذج من الذكاء الاصطناعي الموزع وعلى الأخص المتعدد الوكلاء في أدوات إدارة الشركات ذو فعالية لمحاكاة وتكرار السلوك التعاوني والتكيف كما يظهر حاليًا في الشركات. النظم الآلية للتخزين والدستوكينغ تجذب حاليًا اهتمامًا متزايدًا بسبب المزايا الهامة التي تقدمها. المتمثلة في: انخفاض تكلفة اليد العاملة، انخفاض مصاريف التخزين وتحسين الاستفادة من مساحة التخزين، وتتبع أفضل المنتجات، نظام تدفق أمثل وأكثر سلامة. وبالتالي لديهم أهمية قصوى في نجاح تحديات صناعة جديدة. للأسف، وبصرف النظر عن أنواع معينة من AS / RS، لا يتم دراستها بكثير، على الرغم من أنها تستخدم على نطاق واسع في الصناعة والخدمات.

وفي هذه الأطلوحة. ركزنا على مساهمة نظام متعدد الوكلاء في آلية التخزين/الاسترجاع. في أول مساهمة نحن مهتمون في النظم الآلية للتخزين والدستوكينغ ذو ناقل. لا سيما في تخزين واسترجاع المنتجات في رفوف. التقنية المستخدمة هي النظام المتعدد الوكلاء التفاعلي حيث تستند الوكلاء على مزيج من قواعد التخزين والاسترجاع. النتائج المتحصّل عليها بعد تطبيق عدّة عمليّات محاكاة أدت إلى إنشاء مساهمة ثانية معتمدة على أساس ذكاء بعض الوكلاء بإمكانهم الاتّصال، من أجل تقليل وقت تخليص المنتجات بشكل عام.