

UNIVERSITE KASDI MERBAH OUARGLA

**Faculté des Sciences et de la Technologie et des Sciences de la
Matière**

Département de Mathématiques et d'Informatique



Mémoire

MASTER ACADEMIQUE

Domaine : Mathématiques et d'Informatique

Filière : Informatique

Spécialité: Informatique Industrielle

Présenté par : Ahfouda Djamila

Habbi Yamina

Thème

**Modélisation et Simulation multi-agents
d'un environnement multi-robots**

Soutenu publiquement

Le : 30/06/2013

Devant le jury :

M.	Ben Mir	Abdelkader	MA (A)	Président	UKM Ouargla
Dr.	Said	Bachir	MC (A)	Encadreur	UKM Ouargla
M.	Mezati	Messaoud	MA (A)	Examineur	UKM Ouargla

Année Universitaire : 2012 /2013



REMERCIEMENTS

Avant de présenter ce travail, nous tenons à remercier الله tout puissant, de notre avoir permis d'arriver à ce niveau d'étude, et aussi pour notre avoir donné beaucoup de patience et de courage pour réaliser ce mémoire.


Travers ce modeste travail, nous tenons à présenter nos sincères remerciements et notre profonde reconnaissance à notre encadreur Mr. SAID Bachir.

Nous remercions très sincèrement Monsieur Mohamed Cherif Bellaouer, Pour son aide précieuse, qui nous avons permis de mettre en valeur toute notre connaissance

Mes remerciements les membres du jury Mr Ben Mir Abdelkader et Mr Mezati Messaoud qui ont bien accepté de juger et d'évaluer ce travail.

Nous adressons aussi notre sincère reconnaissance à tous les enseignants au département des Mathématiques Et Informatiques de l'Université Kasdi Merbah Ouargla pour leurs aides et leurs conseils. Sans oublier tout le staff administratif du département.

Grand merci à tous ceux et toutes celles qui ont contribué de près ou de loin dans l'accomplissement de ce travail.






DEDICACES

Je dédie ce mémoire

À toutes nos familles habbi et ahfouda,

À tous nos collègues de promotion,

*ET à tous ceux qui m'ont aidé de près ou de loin à accomplir
ce travail.*



ملخص:

المحاكاة القائمة على أساس وكيل أو متعدد وكيل أصبحت الآن أكثر كفاءة و بساطة. في الواقع، هذا النوع من المحاكاة يوفر على عكس الأنواع الأخرى القدرة على تمثيل مباشر للكائنات المحاكاة و سلوكهم.

يتضمن مشروعنا هذا نمذجة و محاكاة بيئة متعددة الروبوت علي أساس نظام متعدد الوكيل الذي يعتمد علي الانطولوجيا . و عملنا يتمثل في تصميم محاكي لعمال النظافة في مدينة و بالأخص في حي ما و الفكرة هنا هي تمثيل عمال النظافة في المحاكي بروبات ذكية تستند في تفكيرها و اتخاذها للقرارات على قواعد SWRL المبنية على الأنطولوجيا التي تمثل نموذج مصغر للمدينة.

إن بنية و مفهوم الأنطولوجيا يساعد على نمذجة العمل حيث أنها عبارة عن مجموعة أقسام و أجزاء من الأقسام تجمع بينهم علاقات تسمح لنا بتعريف قواعد SWRL، و كل قسم يحتوي على مجموعة من الأفراد.

أما الواجهة تم تصميمها باستعمال جافا أف-أكس لأنها تقلل من التعليمات مقارنة بالجافا، تحتوي واجهتنا على ثلاثة أقسام: شريط القوائم، منطقة الخيارات، منطقة العرض.

نحن مهتمون أساسا بالأنظمة متعددة الوكلاء التعاونية المكونة من وكلاء يشتركون في نفس الأهداف، أما بمصالح مشتركة أو بمهارات متكاملة، و من اجل هذا قمنا بتمثيل كل من روبوت، شاحنة، سلة قمامة في المحاكي بوكيل مما يسمح له بالتواصل مع الوكلاء الآخرين في المحاكي.

الكلمات الدالة: النمذجة، المحاكاة، نظام متعدد الوكيل، نظام متعدد الروبوت، تصرفات الروبوت، الأنطولوجيا.

Résumé :

La simulation à base d'agents ou encore Multi-Agents s'impose de plus en plus pour son efficacité et sa simplicité. En effet, ce type de simulation offre, au contraire des autres types, la possibilité de représenter directement les entités simulées et leurs comportements.

Notre projet comprend la modélisation et la simulation multi agents d'environnement multi robots qui dépend d'une ontologie, donc la tâche qui nous a été affecté afin de concevoir un simulateur des nettoyeurs de la ville, précisément dans le quartier.

L'idée ici présente les nettoyeurs dans le simulateur par les robots intelligent se basant sur sa pensée et prendre les décisions sur les règles de SWRL qui constituent l'ontologie qui représente un petit modèle de la ville.

La structure et le concept de l'ontologie aide à modéliser le travail par conséquent, il s'agit d'un ensemble de classes et sous classes reliées entre eux par des relations permettant de définir les règles SWRL et chaque classe contient une collection d'individus.

L'interface a été conçu à l'aide de javafx parce qu'il possède moins de code comparé au java pour générer une interface et des composants graphiques. L'interface de notre simulateur contient trois parties qui sont : Bar des menus, zone d'options, et zone d'affichage.

Nous nous intéressons aux systèmes multi agents coopératifs composés d'agents qui partagent les mêmes buts, tels que les sociétés d'agents ayant des intérêts communs ou les sociétés d'agents dont les compétences sont complémentaires. Et pour cela, nous devons représenter chacun des robots, camion et poubelle dans notre simulateur par un agent qui permet de communiquer avec d'autres agents dans le simulateur.

MOTS-CLÉS : Modélisation, Simulation, Système Multi-Agents, Système Multi-Robots, Comportement de robot, ontologie.

Summary:

The stimulation that is based on agents or multi-agents becomes now more competent, skillful and simplified. In contrast to other kinds of stimulation, the current stimulation provides us with opportunity to represent directly the stimulated things and their behaviours.

Our project includes a modulation and stimulation of multi-robot environments using a multi- agents system. This latter depends on anthology. We aim for designing a stimulator to cleaning workers. More precisely, the assumption is that, in the stimulator, the cleaning workers represents in form of intelligent robot which depends in its thinking on taking decisions according to SWRL rules. The latter build on anthology that shows small representation of the city.

The structure and the definition of anthology help in modeling the work. Also, it is a group of hierarchical classes that are related to each other via relationships which allow us to define SWRL rules and each class consists of set of members.

The front of the current stimulator was designed using javafx as it needs a little instruction in comparison with java. The front of the project consists of three classes: “Bar des menu “,” zone d’options” and “zone d’affichage”.

We are interested in multi-agent co-operative system which contains a group of members. These members share the same goals and complementary skills. Hence, we present all the robot, lorry, and waste basket in stimulators with agents which allow them to communicate with each other in the stimulators.

KEYWORDS: Modeling, Simulation, Multi-Agent System, System Multi-Robots, robot Behavior, Ontology

Table des Matières

<i>Table des Matières</i>	4
<i>Table des Figures</i>	7
<i>Table des Tableaux</i>	8
<i>Introduction générale</i>	9
Chapitre I : Les systèmes Multi agents	11
I.1 Introduction	12
I.2 La notion d'agent	13
I.2.1 Définition.....	13
I.2.2 Environnement :.....	13
I.2.3 Caractéristiques des agents :	14
I.2.4 Classification des agents	15
I.3 Les systèmes Multi agents	15
I.3.1 Définition.....	15
I.3.2 Caractéristiques des systèmes multi agents	16
I.3.3 Interaction dans un système multi agents.....	17
I.3.4 Communication dans les systèmes multi agents :	17
I.4 Domaines d'application des systèmes multi agents	19
I.4.1 La résolution de problèmes :.....	19
I.4.2 Simulation multi agent :.....	19
I.4.3 La construction de mondes synthétiques :	19
I.4.4 La robotique distribuée :	19
I.5 Simulation multi agent :	20
I.5.1 Définition de simulation multi agent :	20
I.5.2 Méthodologie de simulation Multi-Agents:.....	20
I.5.3 Caractéristiques de la simulation multi-agents.....	21
I.6 Conclusion :	22
Chapitre II : La robotique	23
II.1 Introduction	24
II.2 Les robots réels : définis comme sont :.....	25
II.2.1 La robotique :	25
II.3 Les robots virtuels :	28

II.3.1 Définition de simulateur robot :	28
II.3.2 Les simulateurs des robots existants :	28
II.3.3 Comparaison entre les simulateurs :	32
II.4 Conclusion	33
Chapitre III : Modélisation	34
III.1 Introduction	35
III.2 Problème à résoudre :	36
III.2.1 Description du problème :	36
III.2.2 Résultats attendus :	36
III.3 Modélisation d'un environnement :	36
III.3.1 Les composant d'environnement :	36
III.3.2 Architecture d'environnement :	37
III.4 Définition d'Ontologie et SWRL :	37
III.4.1 Définition d'ontologie :	37
III.4.2 Définition de SWRL(Sémantique Web Règle Langage) :	38
III.4.3 L'Ontologie de notre travail :	38
III.5 Modélisation d'un robot virtuel intelligent :	42
III.5.1 Description de robot virtuel :	42
III.5.2 Architecture des robots virtuels :	42
III.6 Conclusion :	48
Chapitre IV : Implémentation	49
IV.1 Introduction :	50
IV.2 Les outils utilisés :	51
IV.2.1 NetBeans :	51
IV.2.2 JavaFX :	51
IV.2.3 La plate-forme JADE :	52
IV.2.4 PROTÉGÉ-OWL :	53
IV.3 Organisation de code :	54
IV.3.1 Diagramme de classes :	54
IV.3.2 Détail des classes	56
IV.4 Présentation de simulateur :	57
IV.4.1 Présentation de l'interface graphique :	57
IV.4.2 Descriptions détaillées des différentes parties de l'interface :	57
IV.4.3 L'interface en cours d'exécution :	59

IV.4.4 Les communications en cours d'exécution :.....	60
IV.5 Résultats des simulations :.....	61
IV.5.1 Discussion	63
IV.6 Conclusion :.....	64
<i>Conclusion générale</i>	65

Table des Figures

Figure I-1: Architecture d'un environnement	13
Figure I-2 : Architecture d'un SMA.....	16
Figure I-3 : Communication par envoi de messages.....	18
Figure I-4 : Communication via un Tableau Noir.....	18
Figure I-5 : Les éléments dans la simulation multi-agents.....	20
Figure II-1 : Robot humanoïde HRP2.....	26
Figure II-2 : Capteurs	26
Figure II-3 : L'interface de simulateur Webots.....	29
Figure II-4 : L'interface de simulateur Easy-Rob	29
Figure II-5 : L'interface de simulateur simbad	30
Figure II-6 : L'interface de simulateur USARSim.....	31
Figure III-1 : Architecture d'environnement.....	37
Figure III-2 : diagramme d'ontologie	41
Figure III-3 : digramme d'états de robot agent	43
Figure IV-1 : Fenêtre pour ajout un Library	52
Figure IV-2 : Boite de saisie le nom de Library.....	53
Figure IV-3 : Fenêtre de parcourir le fichier jar.....	53
Figure IV-4: Structure du programme.....	55
Figure IV-5 : interface du simulateur.....	57
Figure IV-6 : description de barre des menus	57
Figure IV-7: interface de l'environnement 1	58
Figure IV-8: interface de l'environnement 2.....	58
Figure IV-9: description de barre des menus.....	58
Figure IV-10 : Capture d'écran du simulateur SimNettoyage avant lancement	59
Figure IV-11 : Capture d'écran du simulateur SimNettoyage après lancement.....	59
Figure IV-12 : Capture d'écran du simulateur SimNettoyage fin du lancement.....	60
Figure IV-13 : L'interface de l'agent Sniffer pour la communication entre robotAgent et poubelleAgent	60
Figure IV-14 : L'interface de l'agent Sniffer pour la communication entre robotAgent et camionAgent	61
Figure IV-15 : Les courbes représentent la Temps de nettoyage Inter-Lancement nombre de (robot), pour chaque valeur de nombre de (déchet).....	63

Table des Tableaux

Tableau I.2-1: Comparaison entre agent cognitif et agent réactif	15
Tableau II.3-1 : Comparaison entre les simulateurs robot	32
Tableau III.4-1: Règle déplace sans équipement avec chemin	38
Tableau III.4-2: Règle déplace sans équipement.	38
Tableau III.4-3 : règle mettre ramasser.	39
Tableau III.4-4 : règle déplace avec équipement.	39
Tableau III.4-5 : règle déplace avec équipement avec chemin.	40
Tableau IV.5-1:le tableau de résultat de simulation.....	63

Introduction générale

L'Intelligence Artificielle est parmi les axes de recherche qui s'intéressent à la robotique pour réaliser des systèmes artificiels qui s'adaptent à un environnement changeant ; réaliser des systèmes où l'intelligence est distribuée entre de nombreux agents artificiels qui interagissent ; être capable de raisonner sur le temps et l'espace ; développer des robots autonomes ou capables d'assister l'humain dans ses tâches, seuls ou opérant en collectivité ; et surtout, passer à l'échelle, se confronter aux problèmes réels...

La modélisation informatique permet de construire un prototype caractérisé par un programme informatique d'une réalité. Ce prototype traduit les comportements de cette réalité ainsi que son interaction avec l'environnement. Plusieurs sortes de motivations poussent au recours à la modélisation et à la simulation. Il est ainsi courant d'utiliser ces techniques afin de savoir comment fonctionne un système (service de nettoyage) puis formuler et mettre au point ses lois d'évolution.

La simulation Multi-Agents nous apparaît être la démarche la plus adéquate à la modélisation de ce type de systèmes. La simulation Multi Agents est un domaine de recherche relativement récent, qui se base sur la notion d'agent. Un système Multi-Agents est naturellement adapté pour décrire et simuler un système composé d'entités, il permet au modèle de paraître plus proche de la réalité. De plus, un système Multi-Agents est particulièrement bien adapté à la description d'un système du point de vue de l'activité de ses composantes.

Pour implémenter notre simulateur de nettoyage par les systèmes Multi-Agents comme base d'environnement multi robot, nous avons adopté la plate-forme de développement jade et le protégé owl pour implémenter la base de connaissance de robot.

Notre mémoire se compose de quatre chapitres organisés de la façon suivante:

- Le premier chapitre, nous introduisons la notion d'agent, de système Multi agents et les applications concernées à la robotique.

- Dans le deuxième chapitre, nous présentons la robotique en détaille et quelque simulateur robot existant.
- Dans le troisième chapitre sera consacré à l'ontologie de notre travail et la modélisation de notre environnement et robot avec l'architecture du dernier.
- Dans le dernier chapitre, une implémentation de notre application avec la plate-forme Multi-agents Jade, Protege OWL, ainsi que la présentation et critique des résultats obtenus.
- Finalement, nous terminons par une conclusion générale .

Chapitre I :

Les systèmes Multi agents

I.1 Introduction

Historiquement, les systèmes multi agents ont été appliqués pour la résolution de problèmes d'intelligence artificielle dans les années 70. En effet, à cette époque, l'évolution des domaines d'application de l'intelligence artificielle a investi des univers complexes et hétérogènes tels que l'aide à la décision, la reconnaissance et l'interprétation des formes, la conduite et le contrôle de processus industriels,...etc.

Ce chapitre est consacré à la présentation de notions fondamentales de Systems multi-agent qu'est considéré comme plate forme à la robotique. Ce système se positionne au carrefour de plusieurs domaines qui sont : la programmation, l'intelligence artificielle et les systèmes répartis (distribués). Ce domaine représente, en quelque sorte ; le résultat de la rencontre de l'intelligence artificielle des Systèmes Distribués.

I.2 La notion d'agent

I.2.1 Définition

Etymologiquement, le mot agent est inspiré du latin "agere" qui signifie agir, donc un agent est une entité qui agit.

Jacques Ferber définit l'agent comme étant : "*une entité autonome physique ou abstraite qui est capable d'agir sur elle même et sur son environnement, qui dans un univers Multi Agents peut communiquer avec d'autres agents et dont le comportement est la conséquence de ses observations, de ses connaissances, et des interactions avec les autres agents*" [11].

I.2.2 Environnement :

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 l'environnement va agir sur l'agent [8].

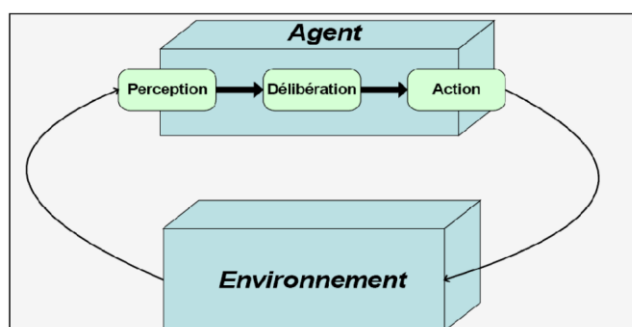


Figure I-1: Architecture d'un environnement.[15]

L'interaction agent/environnement étant essentielle dans le problème que pose l'intelligence artificielle, nous commencerons ici par présenter les principales propriétés qu'il est proposé d'observer sont [4]:

Accessible/inaccessible: L'environnement auquel l'agent peut accéder est dit accessible sinon il lui est inaccessible.

Déterministe/non déterministe (Stochastique): dans un environnement déterministe, l'état de l'environnement à l'instant suivant est complètement déterminé par l'état courant et par les actions effectuées par les agents. Dans un environnement stochastique, l'état suivant est aléatoire.

Épisodique/non épisodique (séquentiel): on parle d'environnement épisodique (ou de temps discrétisé) si l'expérience de l'agent peut être divisée en "épisodes". Chaque épisode

consiste en une phase de perception puis en une phase d'action, le résultat de l'action ne dépendant que de la période courante.

Statique/dynamique: si l'état de l'environnement ne peut pas être changé sans l'intervention de l'agent on dit qu'il est statique. L'environnement est dynamique si son état peut changer sans intervention d'agent.

Discret /continu: si tout passage d'un état de l'environnement à un autre nécessite le passage par une séquence d'états intermédiaires, on dit que l'environnement est continu, sinon il est discret.

I.2.3 Caractéristiques des agents :

Les différentes caractéristiques de l'entité agent peuvent être résumées dans la définition suivante : « un agent est un système informatique situé dans un environnement, et qui agit de façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu » [3].

L'autonomie : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne.

La situation : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement. Exemples: systèmes de contrôle de processus, systèmes embarqués.

La flexibilité : L'agent est capable de réaliser des actions d'une façon autonome et réflexible afin d'atteindre les objectifs qui lui ont été fixé. La réflexibilité signifie dans ce cas:

- ❖ la réactivité : l'agent est capable de percevoir les changements dans son environnements, et doit élaborer une réponse dans les temps requis (changer son état interne, exécuter une tâche particulière...etc.).
- ❖ la pro-activité : l'agent doit exhiber un comportement proactif, opportuniste et orienté objectifs, c'est-à-dire il n'agit pas uniquement en réponse à son environnement mais, il est également capable de prendre l'initiative au "bon" moment.
- ❖ la sociabilité : l'agent doit être capable d'interagir avec les autres agents (logiciels et humains) quand la situation l'exige afin de compléter ses tâches ou aider ces agents à accomplir les leurs.

I.2.4 Classification des agents

Les agents sont généralement classés selon deux courants de penser : l'école cognitive et l'école réactive.

I.2.4.1 Les agents réactifs:

Ce sont des agents qualifiés de non intelligents, ils répondent d'une façon opportune aux modifications de leurs environnements résultants des stimuli externes, les agents réactifs agissent en fonction de ces dernières sans nécessiter de compréhension de leurs univers ni de leurs buts [1].

I.2.4.2 Les agents cognitifs:

L'agent dispose d'une capacité de raisonnement, il possède des informations et des savoir-faire nécessaires à la réalisation de sa tâche et à la gestion des interactions avec son environnement et les autres agents. L'agent possède un but à atteindre à l'aide des plans explicites [22].

Différence entre agent cognitif et agent réactif.

Système d'agents cognitifs	Système d'agents réactifs
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire de son historique
Agent complexe	Fonctionnement stimulus/réaction
Petit nombre d'agent	Grand nombre d'agent

Tableau I.2-1: Comparaison entre agent cognitif et agent réactif

I.2.4.3 Les agents hybrides:

Un agent hybride est une combinaison entre l'agent cognitif et l'agent réactif.

I.3 Les systèmes Multi agents

I.3.1 Définition

Les systèmes Multi Agents (SMA) sont des systèmes distribués conçus et implantés idéalement comme un ensemble d'agents interagissant, le plus souvent, selon des modes de coopération, de concurrence et de coexistence [3].

Jacques Ferber définit les systèmes Multi-Agents comme des systèmes composés des éléments suivants [11]:

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

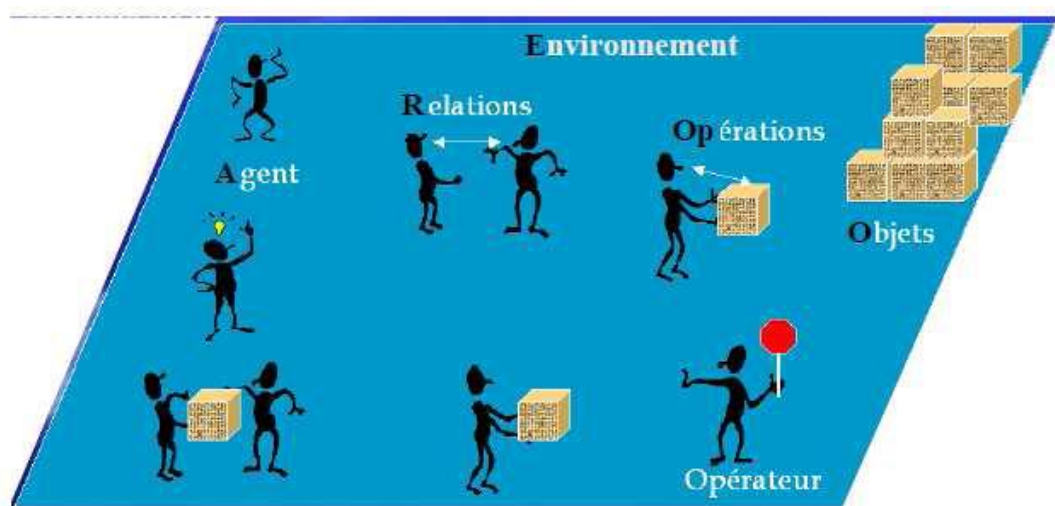


Figure I-2 : Architecture d'un SMA. [15]

I.3.2 Caractéristiques des systèmes multi agents

Un système Multi Agents possède généralement les caractéristiques suivantes:

- ✓ Les agents agissent et travaillent indépendamment les uns des autres.
- ✓ Chaque agent est une partie du système.
- ✓ Chaque agent travaille dans le but d'accomplir ses tâches.
- ✓ Chaque agent est capable de communiquer et d'interagir avec d'autres agents.
- ✓ Un agent coopère avec les autres agents lorsque nécessaire.

- ✓ Un agent est capable de coordonner ses activités avec les autres agents pour accéder à des ressources et à des services partagés dont il a besoin (pour réaliser ses buts).
- ✓ Les agents ont un but commun.
- ✓ Chaque agent a une vue partielle du SMA. [15]

I.3.3 Interaction dans un système multi agents

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 dans les conséquences exercent en retour une influence sur le comportement futur des agents » [11].

Nous allons donner par la suite quelques définitions de la coopération, la coordination et la négociation, ces types d'interactions étant parfois difficiles à différencier.

◆ *La coopération :*

Jacques Ferber définit la coopération entre plusieurs agents comme "une situation dans laquelle, soit l'ajout d'un nouvel agent permet d'accroître les performances du groupe, soit l'action des agents sert à éviter ou à résoudre des conflits potentiels ou actuels".

◆ *La coordination :*

Jacques Ferber donne la définition suivante: "la coordination d'actions dans un système Multi-Agents est définie comme l'ensemble des tâches effectuées par les agents pour réaliser les autres actions (actions effectives) dans les meilleures conditions".

◆ *La négociation :*

La négociation définie comme le processus d'améliorer les accords (en réduisent les inconsistances et l'incertitude) sur des points communs [21].

I.3.4 Communication dans les systèmes multi agents :

La communication représente la base de réalisation de tous les modes d'interaction qu'on a vue précédemment, soit la négociation ou bien la coordination...

Une communication peut être définie comme une forme d'action locale d'un agent vers d'autres agents.

Dans les S.M.A deux stratégies principales ont été utilisées pour supporter la communication entre agents:

➤ **Communication par envoi de messages :**

La communication directe correspond à l'envoi volontaire d'un message à un ou plusieurs agents. Il s'agit d'une conversation entre agents cognitifs, capables d'interpréter des actes de langage et de raisonner sur des messages ambigus. Un acte de langage est une action intentionnelle effectuée au cours d'une communication.

Les langages de communication entre agents, ou ACL (Agent Communication Languages), définissent en particulier des types de messages indiquant l'intention de l'agent émetteur (requête, ordre, information spontanée, proposition de service, affirmation, croyance, etc.). C'est le cas du langage KQML (Knowledge Query and Manipulation Language), du langage de la FIPA (Foundation of Intelligent Physical Agents) [1].

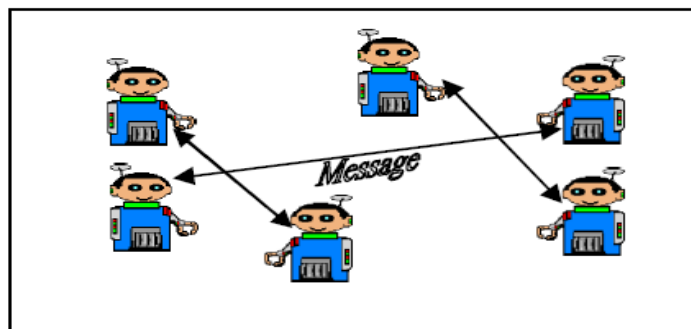


Figure I-3 : Communication par envoi de messages

➤ **Communication par tableau noir :**

Dans les systèmes fonctionnant par partage de ressources, les différents composants ne sont pas directement liés entre eux. Ils communiquent au travers d'une zone de données commune appelée tableau noir (Blackboard), dans laquelle sont stockées les connaissances du système [1].

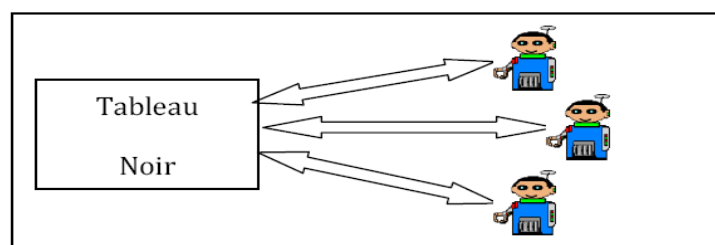


Figure I-4 : Communication via un Tableau Noir

I.4 Domaines d'application des systèmes multi agents

Les domaines d'application des systèmes multi agents qui consiste à cinq grands domaines sont :

I.4.1 La résolution de problèmes :

Telle qu'elle avait été définie en Intelligence Artificielle, étendue à un contexte distribué. Dans ce cadre, l'objectif est de mettre en œuvre un ensemble de techniques pour que des agents, pertinents pour la résolution d'une partie ou l'ensemble du problème, participent de manière efficace et cohérente à la résolution du problème global [7].

I.4.2 Simulation multi agent :

Les simulations dont l'objectif est la modélisation de phénomènes du monde réel, afin d'observer, de comprendre et d'expliquer leur comportement et leur évolution. Les systèmes multi-agents ont trouvé rapidement un champ extrêmement propice à leur développement dans le domaine de la modélisation de systèmes complexes ne trouvant pas de formalisation mathématique adaptée [11].

I.4.3 La construction de mondes synthétiques :

Dans la construction de monde synthétique ne permet pas la résolution de problème, n'utilise pas d'agent physique et ne simulent aucun monde réel mais est une large part dans les recherches portant sur les systèmes multi-agents parce qu'elle permet d'analyser certains mécanismes d'interaction de manière plus fine que ne pourrait le faire une véritable application [11].

I.4.4 La robotique distribuée :

La robotique distribuée porte sur la réalisation non pas d'un seul robot, mais d'un ensemble de robots qui coopèrent pour réaliser une tâche. La robotique distribuée utilise des agents concrets qui se déplacent dans un environnement réel, il y a deux types de robotique est [11]:

- La robotique cellulaire : s'intéresse à la constitution modulaire de robots. Dans ce cadre, un robot qui est représenté comme un système multi-agent, chacun de ses composants sera regardé comme un agent. La réalisation d'un travail est le résultat de la coordination d'un ensemble d'agents.
- La robotique mobile : contient au moins deux robots qui doivent coordonner leurs déplacements et coopérer à la réalisation de tâches telles que nettoyer le plancher,

surveiller des bâtiments, intervenir pour secourir des personnes, réparer des canalisations ou explorer des espaces lointains ou dangereux.

I.5 Simulation multi agent :

I.5.1 Définition de simulation multi agent :

La simulation Multi-Agents est bien adaptée aux systèmes complexes constitués de plusieurs entités, elle consiste à séparer ces entités qui seront représentées par des agents et définir leurs différents comportements et interactions, puis faire évoluer les agents en fonction du temps, et étudier et analyser la progression du système global suite aux actions produites par les agents et leurs interactions [24].

Un système de simulation Multi-Agents est un ensemble d'agents et de règles d'interaction entre eux. Chaque agent est capable d'évaluer individuellement sa situation et de prendre des décisions à partir d'un ensemble de règles. Il peut exécuter divers comportements relatifs au système représenté, Les agents peuvent ensuite être capables d'évoluer, permettant à des comportements non anticipés d'apparaître [24].

- Simulations de Systèmes Multi-Agents : Il a trois éléments: agents, interactions et environnement.

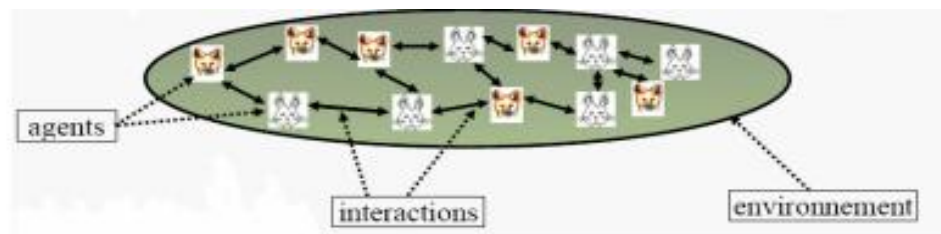


Figure I-5 : Les éléments dans la simulation multi-agents

I.5.2 Méthodologie de simulation Multi-Agents:

La Méthodologie de simulation se base sur la définition d'une théorie de modélisation. La modélisation d'un phénomène dans un perspectif multi-agent se traduit par:

I.5.2.1 Une décomposition du phénomène :

En un ensemble d'éléments discrets autonomes dont les interactions reproduisent le phénomène. Cette phase est semblable au processus de réification intervenant dans les méthodes de conception orientées objet. Il est à noter que ce préliminaire nécessite une vision déjà distribuée du phénomène à modéliser (à moins que, comme nous le faisons remarquer plus loin, l'un des objets de la simulation soit justement de déterminer les éléments ou les niveaux pertinents à étudier) [10].

I.5.2.2 La modélisation de chacun de ces éléments par un agent :

Dans cette phase intervient un nécessaire choix quant à la théorie à employer pour définir les connaissances de l'agent, ses capacités fonctionnelles, ses comportements et les modes d'interaction qu'il adoptera à l'encontre des autres agents. On se reportera utilement à l'intelligence artificielle et aux modèles d'agents qui y sont présentés pour comprendre que ce choix peut être délicat. C'est au cours de cette étape que le modèle théorique de comportement individuel proposé par le domaine de recherche dans lequel s'effectue la modélisation est traduit en un modèle informatique [10].

I.5.2.3 La définition de l'espace :

Dans lequel évoluent ces agents et des lois qui le gouvernent. On appellera cet espace l'environnement des agents. Sa définition permettant d'affiner la description des actions possibles des agents, ainsi que celle de leurs moyens de communication, on définira souvent agents et environnement de manière concomitante [10].

I.5.2.4 La définition des objets inertes. :

Seront considérés comme objets inertes les agents du système qui ne sont dotés d'aucune capacité d'action ni de communication.

I.5.3 **Caractéristiques de la simulation multi-agents**

Dans des approches de modélisation basées sur la définition d'équations mathématique, la dynamique du système est définie dès le début par des relations fonctionnelles entre entités. Au contraire, la simulation multi-agents se propose de modéliser explicitement les comportements des entités et considère que la dynamique globale d'un système résulte directement de l'interaction des comportements. En effet, l'approche multi-agents permet de modéliser chaque individu, son comportement et les interactions qui découlent de la mise en commun de ces individus. La dynamique globale du système est alors issue de l'ensemble de ces individus [25].

I.6 Conclusion :

La vision centralisée des problèmes est difficile à mettre en œuvre dans le cas de processus complexes, réunir la décentralisation et l'autonomie grâce aux systèmes Multi-Agents ouvre les portes vers une nouvelle démarche de résolution des problèmes et de simulation des phénomènes complexes, cette dernière a bien exploité les avantages qu'offrent les systèmes Multi-Agents qui lui permettent de maîtriser la complexité et la difficulté des systèmes à simuler

Dans ce chapitre, nous avons présenté un état de l'art sur les notions théoriques en rapport direct avec le thème de notre mémoire. Cet état présente en façon globale les définitions et explications sur les concepts d'agents et de Système Multi Agent.

Et puisque les SMA servent aujourd'hui à toutes sortes d'applications dans divers domaines, nous citons quelque domaine d'application comme résolution de problème, simulation multi agent et robotique.

Chapitre II :

La robotique



II.1 Introduction

La robotique est la science qui s'intéresse aux robots. En fait, il s'agit d'un domaine multidisciplinaire : on y trouve des aspects concernant la mécanique, l'informatique, l'électronique. Puisque la simulation de robotique avant son déploiement réel représente une alternative moins coûteuse et plus rapide que son expérimentation en environnement réel, On va présenter dans ce chapitre a présenté et décrive la robotique et on va citer quelques simulateurs robots qui sont actuellement existants.

II.2 **Les robots réels** : définis comme sont :

II.2.1 **La robotique** :

La *robotique* peut être définie comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles [6].

II.2.1.1 Définitions de robot :

On définit le robot comme étant un appareil automatique capable de manipuler des objets, ou d'exécuter des opérations selon un programme fixe ou modifiable.

Un robot est, a priori, un système matériel mécanique et informatique, qui possède des capacités de perception, d'action, de décision et de communication. Il est parfois capable d'améliorer ses propres performances par apprentissage automatique ou supervisé par des humains. Il peut agir dans un environnement ouvert ou confiné, dynamique et imparfaitement modélisé ou même inconnu [9].

Les robotiques se répartissent selon cinq grandes thématiques:

- la conception mécanique, électronique et informatique
- l'action et le mouvement, et notamment la commande et les contrôles sensorimoteurs
- la perception (multi-sensorielle) et la représentation de l'environnement
- la prise de décision, le raisonnement et l'apprentissage
- l'interaction et la coopération avec d'autres robots ou avec l'homme.

II.2.1.2 Composantes d'un robot :

Les robots sont équipés des effecteurs et des capteurs.

- effecteur leur permet d'agir dans l'environnement (roue, bras, jambes et pinces) [19]



Figure II-1 : Robot humanoide HRP2

- capteur leur permet de percevoir l'environnement dans lequel ils évoluent [19]



Figure II-2 : Capteurs

- Il existe autre composantes comme périphériques de stockage, interfaces de communication et unité d'alimentation [19].

II.2.1.3 Classification des robots :

II.2.1.3.1 Robots manipulateurs :

Robots manipulateur existe dans les applications industrielles .Ce robot est un série de segment relativement rigides dont les articulations sont commandées par des actionneurs et chaque segment équipé d'un effecteur (pince, main articulée, outil, ...) servant à la réalisation de la tâche à accomplir, il y a différent type de ce robot : robots industriels et les robots pour l'assistance médicale [12].

➤ Classification des robots manipulateurs [12]: La classification des robots manipulateurs est basée sur des critères suivant :

- ✓ La technologie des actionneurs.
- ✓ La géométrie de l'enveloppe de travail.
- ✓ Le type de déplacement.

II.2.1.3.2 Robots mobiles :

Un robot mobile est une machine automatique capable de se déplacer dans un environnement donné. Ils sont équipés ou non de manipulateurs suivant leur utilisation [18].

Il y a différents types de ce robot : les robots explorateurs et les robots de service et les robots ludiques.

II.2.1.4 Caractéristiques des robots :

II.2.1.4.1 Volume de travail :

Le volume accessible par l'outil du robot est :

- de la géométrie du robot
- de la longueur des segments
- du débattement des articulations (limité par des butées) [16].

II.2.1.4.2 Répétabilité ou Précision :

On dit que le positionnement absolu est imprécis lorsqu'il dépasse 1 mm, alors on a l'un des erreurs suivantes [16]:

- Erreurs de modèle géométrique.
- Erreurs de quantification de la mesure de position.
- Flexibilité.

Répétabilité : la répétabilité d'un robot est l'erreur maximale de positionnement répété de l'outil en tout point de son espace de travail [16].

- en général répétabilité < 0.1 mm

II.2.1.4.3 Performances :

- ◆ Vitesse maximale : C'est la vitesse de translation ou de rotation de chaque axe du robot. Souvent les constructeurs donnent une vitesse maximale de l'outil ou l'organe terminal [16].
- ◆ Accélération maximale : Elle dépend de l'inertie, donc de la position du robot. Pour chaque axe, cette accélération est donnée dans la configuration la plus défavorable (inertie maximale, charge maximale) [16].

II.2.1.4.4 Charge utile :

La charge maximale qui peut porter le robot sans perturber ni la respectabilité ni les performances dynamiques du robot.

Cette charge utile est nettement inférieure à la charge maximale que peut porter le robot et directement dépendante des actionneurs [16].

II.3 Les robots virtuels :

Les simulateurs des robots permettent de s'attaquer à un large éventail de problèmes de conception qui se posent tout en développant des systèmes robotisés.

II.3.1 Définition de simulateur robot :

Un logiciel de simulation de robot ou un simulateur est un environnement virtuel complet utilisée pour créer des applications des robots simples peuvent être construits, programmé et testé. Dans quelque cas, ces applications peuvent être transférées sur le robot réel sans modification ce qui réduit les coûts et le temps[34].

Les principales caractéristiques de simulateur robot est très simple à utiliser et intuitive.

II.3.2 Les simulateurs des robots existants :

il existe plusieurs simulateurs des robots comme Simbad, SimRobot ,Webots , Easy-Rob, CARMEN, USARSim et openCitye. nous présentons dans cette partie quelques simulateurs.

II.3.2.1 Les simulateurs commerciaux:

II.3.2.1.1 Webots :

Webots est un logiciel commercial de simulation de robots mobiles. Il a été initialement développé au LAMI (Laboratoire de micro-informatique, Prof. Jean-Daniel Nicoud), puis transféré en 1998 à la société Cyberbotics [28].

Webots est un environnement de développement utilisé pour modéliser, programmer et simuler des robots mobiles. Avec Webots l'utilisateur peut concevoir des configurations complexes robotiques, avec un ou plusieurs robots, semblables ou différents, dans un environnement partagé. [28]

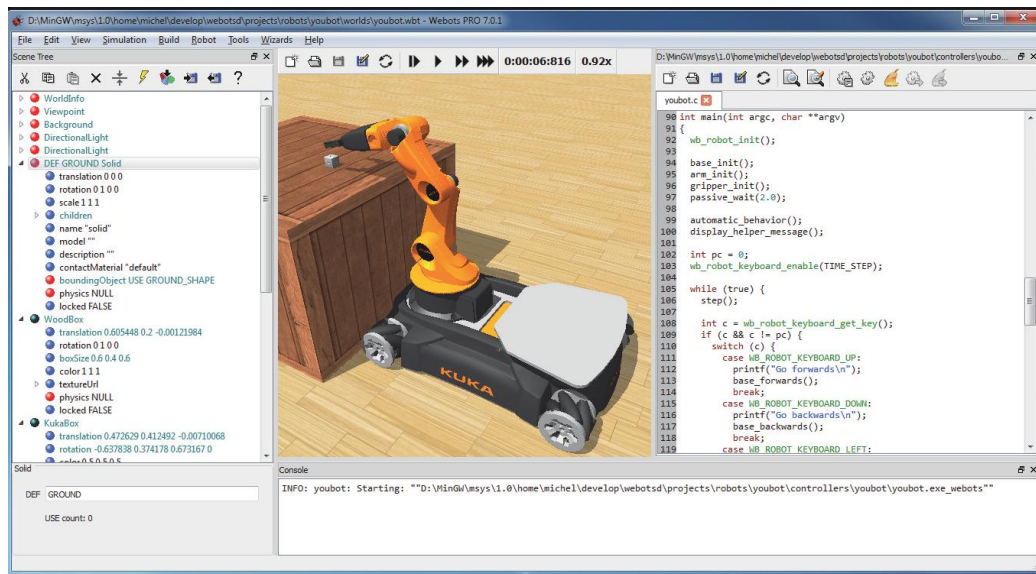


Figure II-3 : L'interface de simulateur Webots

II.3.2.1.2 Easy-Rob:

Easy-Rob est logiciel de planification et de simulation pour les plates-formes robotiques dans les usines de fabrication de cellules de travail, pour créer des images de haute qualité et à haute vitesse fournis au format 3D. Une autre fonctionnalité de Easy-Rob, c'est la capacité de créer des films AVI des simulations. Les utilisateurs peuvent utiliser l'API fournie par Easy-Rob ou externes bibliothèques robotiques, des outils intégrés permettent à l'utilisateur d'importer ou d'exporter des dessins AutoCAD [29].

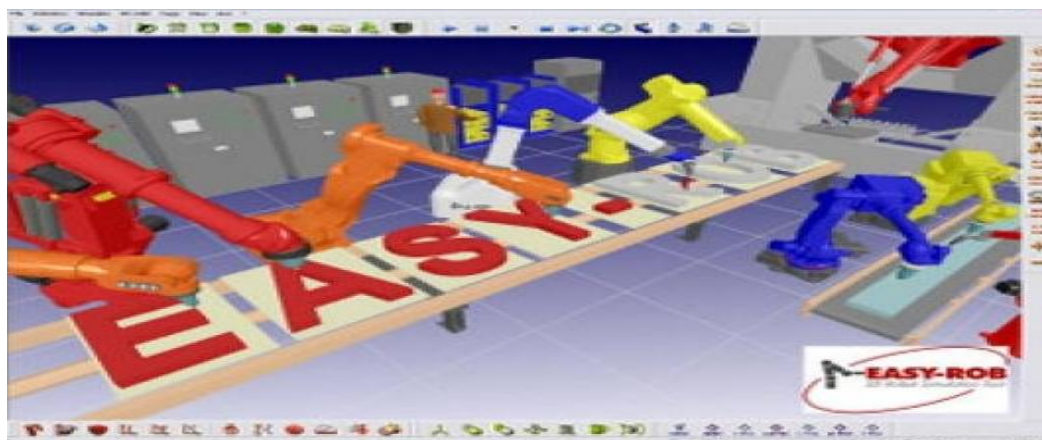


Figure II-4 : L'interface de simulateur Easy-Rob

II.3.2.2 Les simulateurs Open-source: ➤

II.3.2.2.1 Simbad :

Simbad est un simulateur 3D de robot open source en Java à usage scientifique et pédagogique permettant de programmer facilement un ou plusieurs robots. Il est dédié aux chercheurs et programmeurs qui souhaitent avoir une base simple pour étudier " l'intelligence artificielle, l'apprentissage de machine, et les algorithmes de l'IA dans le cadre des robots autonomes " [35].

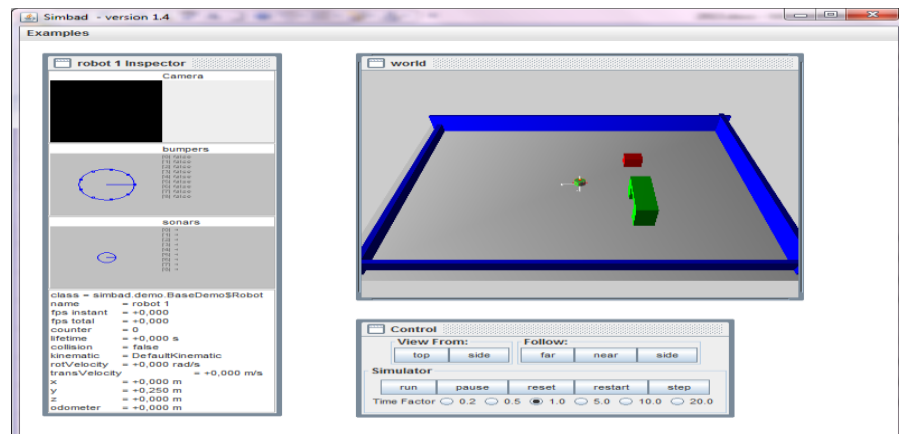


Figure II-5 : L'interface de simulateur simbad

II.3.2.2.2 CARMEN :

CARMEN (Carnegie Mellon Robot Navigation Toolkit) est une collection open source des logiciels pour le contrôle de robot mobile. CARMEN est un logiciel modulaire conçu pour fournir primitives de navigation de base, y compris: la base et le contrôle du capteur, l'exploitation forestière, l'évitement d'obstacles, localisation, planification de chemin, et la cartographie. Fonctionnalités principales de CARMEN a écrit en C, mais fournit le support de Java et exécute sous Linux et est disponible sous licence GPL. CARMEN est un simulateur robot 2D [27].

II.3.2.2.3 USARSim:

USARSim (Unified System for Automation and Robot Simulation) est un simulateur 3D basé sur le moteur Unreal Tournament jeu. USARSim a été créé pour simuler la recherche urbaine et robots de sauvetage et des environnements destinés à l'étude de l'interaction homme-robot et la coordination multi-robots [5].



Figure II-6 : L'interface de simulateur USARSim

II.3.2.2.4 MRDS:

MRDS (Microsoft Robotics Developer Studio) est Windows basé sur un simulateur 3-D pour créer des applications robotiques à travers une variété de plates-formes robotiques et des capteurs. L'objectif de MRDS est facilité le développement d'applications robotiques. Elle prend en charge diverses configurations matérielles de robots. Elle comporte un runtime, un environnement de développement et des outils de simulation.

MRDS n'est pas techniquement un simulateur open source, mais il est accessible au public sans frais et comprend les fonctionnalités de toutes les éditions et les boîtes à outils [31].

II.3.3 Comparaison entre les simulateurs :

	Webots	Easy-Rob	Simbad	CARMEN	USARSim	MRDS
SE	Linux, Mac, Win.	Win.	Linux, Mac, Win.	Linux	Linux Win.	Win.
Type de Simulateur	3-D	3-D	3-D	2-D	3-D	3-D
Langage de programmation	C, C++, Java, Matlab, Python	C++	Java	C, Java	C, C++, Java	VPL, C#, Visual Basic, JScript, IronPython
Documentation	haut niveau	haut niveau	haut niveau	bas niveau	haut niveau	haut niveau
Tutoriel	Oui	Oui	Limité	Limité	Limité	Oui
Portabilité	Oui	Non	Limité	Oui	Oui	Oui
Capteurs	odomètre gamme, caméra, GPS	odomètre	vision, gamme, contact	odomètre gamme, GPS	odomètre gamme, caméra, GPS	odomètre gamme, GPS
Débogage	Oui	Oui	Oui	Oui	Oui	Oui
Interface utilisateur graphique	Oui	Oui	Non	Non	Oui	Oui

Tableau II.3-1 : Comparaison entre les simulateurs robot

II.4 Conclusion

La robotique est un domaine récent, comprend nombreux des applications dans diversité des disciplines et domaine important de l'informatique.

Dans ce chapitre, nous avons présenté la robotique en générale et plus particulièrement le simulateur de robot. En début, les robots réel en mettant l'accent sur la notion de robot, ses composants et classification des robots. En suit, les robots virtuels en mettant l'accent sur la notion de simulateur robot, présent quelque simulateur commerciaux et open-source. En fin, comparaison entre les simulateurs.

Chapitre III :

Modélisation

III.1 Introduction

Dans ce chapitre, nous présentons et décrivons la modélisation de notre application sommes les Nettoyage dans un quartier par des robots autonomes et plus particulièrement des robots intelligents.

Tout d'abord, une présentation de problème à résoudre, en suite modélisation de notre environnement en détaille le plant de quartier et le group de nettoyage et définition de son ontologie.

En fin, la modélisation d'un robot virtuel intelligent dans ce point nous parlons à l'architecture de robot et les différents comportements.

III.2 Problème à résoudre :

III.2.1 Description du problème :

Aujourd'hui, le champ de modélisation et simulation du système complexe est de plus en plus développer. Il aide des chercheurs dans plusieurs domaines à étudier et à faire des recherches des systèmes complexes qu'ils ne peuvent pas faire en réalité.

Un des systèmes intéressants, c'est le nettoyage dans une ville et plus particulièrement dans un quartier. Il est un facteur important dans le quartier parce que du manque d'attention au nettoyage, cette étude pose des solutions pour éviter la pollution environnementale.

III.2.2 Résultats attendus :

Le résultat attendu de l'étape travaille pratique est une simulation plus proche de la réalité qui rend possible de nettoyage d'un quartier ce résultat est la conséquence de l'analyse expérimentations de plusieurs effectué sur des environnements différents.

III.3 Modélisation d'un environnement :

III.3.1 Les composent d'environnement :

III.3.1.1 Le plan du quartier :

Le quartier de notre modélisation, c'est un ensemble des objets, on a explique en détail à le suivant :

- Les résidentiels : qui comprendre les maisons, le hôpital, le poste police, boutique, l'école... Etc.
- Les poubelles: il y a un nombre limite des poubelles, le caractérisé de chaque poubelle est le poids et la capacité.
- Les déchets : est l'ensemble de tout déchet comme déchet de mange, Bouteille plastique, sachet et pomme ; chaque déchet est caractérisé par un poids spécifique.
- L'autre : arbre, véhicule, zone vert, route.

Les objets sont classés à deux types obstacle ou non.

III.3.1.2 Le group de Nettoyage :

Le group de Nettoyage est :

- Les robots : est joue le rôle de personne qui faire le nettoyage et caractérisé par une capacité.

- matériel de nettoyage :
- Un camion : rassemble tous les déchets qui nettoyé par les robots, et caractérisé par un poids et une capacité.
- Les poubelles : on a utilise par les robots pour mettre les déchets qui ramasse, le nombre de panier est inferieur ou égale le nombre des robots, et caractérisé par un poids et une capacité.

III.3.2 Architecture d'environnement :

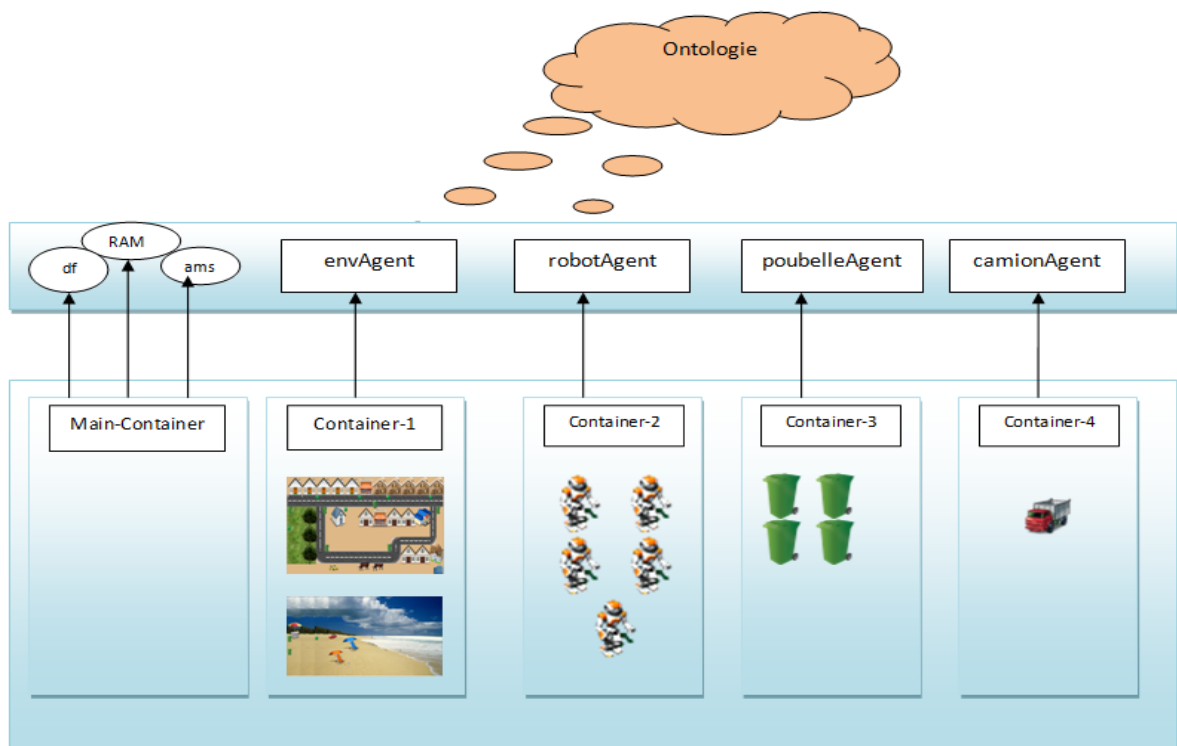


Figure III-1 : Architecture d'environnement

III.4 Définition d'Ontologie et SWRL :

III.4.1 Définition d'ontologie :

La définition du mot Ontologie par :

- La métaphysique d'Aristote, c'est une science fait partie de la philosophie, et traite de la description des entités du monde réel. Son but est de définir les catégories générales employées pour classifier toutes entités du monde (les êtres humains, les animaux, les objets, etc.)[13].

- L'intelligence artificielle, c'est les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire.[chapitreI]

Les domaines d'application d'Ontologie c'est Web sémantique et system d'information.

III.4.2 Définition de SWRL(Sémantique Web Règle Langage) :

destine à être le langage de règles du Web sémantique. Il est basé sur OWL: toutes les règles sont exprimées en termes de concepts OWL (classes, propriétés, des individus, des littéraux ...) [20].

III.4.3 L'Ontologie de notre travail :

On a défini l'ontologie pour présenter la construction de quartier avec les équipes de nettoyage pour ce la utilise le protégé OWL qui noté à le chapitre suivant.

Dans notre ontologie, on a utilisons les règles suivant qui permettent le robot pour réalité une décision :

- *depalace_sans equip_avec_chemin*:

Règle 1 :

$$\begin{aligned} &robot(?x) \wedge Environnement_multi_robot(?y) \wedge capte(?x,?y) \wedge poid_ch(?y,?z) \\ &\quad \wedge swrlb:greaterThan(?z,10) \wedge action(?a) \\ &\quad \wedge has_name(?a,"deplace_avec_chemin") \rightarrow faire(?x,?a) \end{aligned}$$

si le robot capte un déchet dans l'environnement, le poids de déchet supérieur dix et nom de l'action est «deplace_avec_chemin » alors applique cette règle.

Tableau III.4-1: Règle déplace sans équipement avec chemin

- *deplace_sans equip*

Règle 2 :

$$robot(?x) \wedge action(?z) \wedge has_name(?z,"deplacer") \rightarrow faire(?x,?z)$$

si un robot sans équipement et nom de l'action est «deplacer» alors applique cette règle.

Tableau III.4-2: Règle déplace sans équipement.

➤ mettre_ramasser_avec equip:

Règle 3:

$$\begin{aligned} & robot(?x) \wedge Environnement_multi_robot(?y) \wedge capte(?x,?y) \\ & \wedge poid_ch(?y,?pch) \wedge action(?b) \\ & \wedge has_name(?b,"mettre_ramasser") \wedge panier(?eq) \\ & \wedge equip(?x,?eq) \wedge equip_nes(?b,?eq) \\ & \wedge swrlb:lessThanOrEqual(?pch,10) \rightarrow faire(?x,?b) \end{aligned}$$

si un robot avec équipement, capte un déchet dans l'environnement, le poids de déchet inférieur ou égale dix et nom de l'action est «mettre_ramasser» alors applique cette règle.

Tableau III.4-3 : règle mettre ramasser.

➤ deplace_avec equip:

Règle 4:

$$\begin{aligned} & robot(?x) \wedge Environnement_multi_robot(?y) \wedge action(?b) \wedge \\ & has_name(?b,"deplacer") \wedge panier(?eq) \wedge poid_panier(?eq,?z) \wedge \\ & cap_panier(?eq,?cap) \wedge equip(?x,?eq) \wedge equip_nes(?b,?eq) \wedge \\ & swrlb:divide(?a,?cap,2) \wedge swrlb:lessThanOrEqual(?z,?a) \rightarrow faire(?x,?b) \end{aligned}$$

si un robot avec équipement, le poids de panier inférieur ou égale le demi de capacité de panier et nom de l'action est «deplacer» avec un équipement nécessaire panier alors applique cette règle.

Tableau III.4-4 : règle déplace avec équipement.

➤ *deplace_avec equip_avec_chemi*

Règle 5:

$$\begin{aligned}
 & robot(?x) \wedge Environnement_multi_robot(?y) \wedge action(?b) \\
 & \quad \wedge has_name(?b, "deplace_avec_chemin") \wedge panier(?eq) \\
 & \quad \wedge poid_panier(?eq, ?z) \wedge cap_panier(?eq, ?cap) \\
 & \quad \wedge equip(?x, ?eq) \wedge equip_nes(?b, ?eq) \\
 & \quad \wedge swrlb:divide(?a, ?cap, 2) \\
 & \quad \wedge swrlb:greaterThanOrEqual(?z, ?a) \rightarrow faire(?x, ?b)
 \end{aligned}$$

si un robot avec équipement, le poids de panier supérieur ou égal le demi de capacité de panier et nom de l'action est «deplacer_avec_chemin» avec un équipement nécessaire panier alors applique cette règle.

Tableau III.4-5 : règle déplace avec équipement avec chemin.

Notre ontologie qui contient des classes, des sous-classes et les relations entre les deux en expliquent on détaille au diagramme suivant:

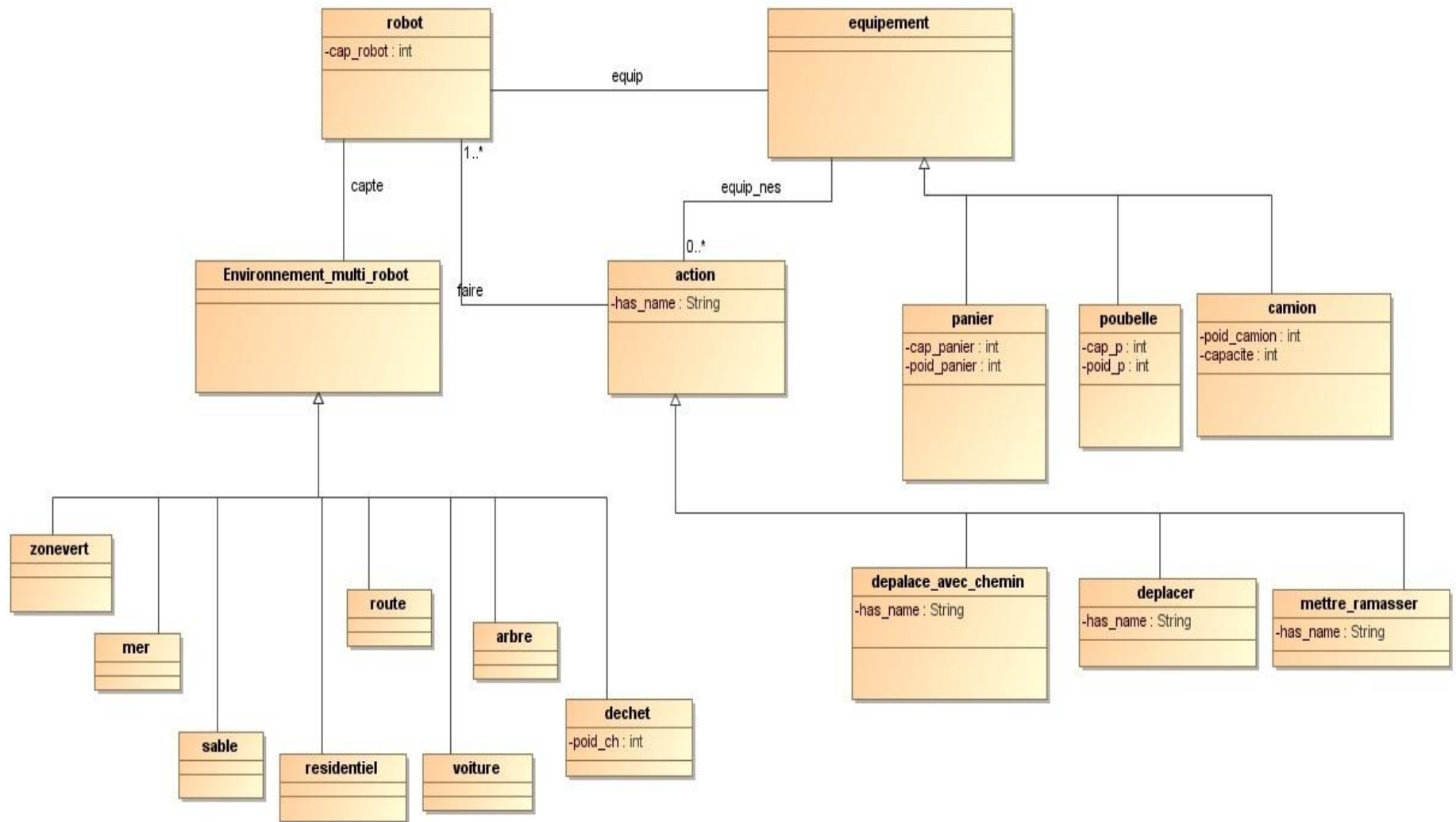


Figure III-2 : diagramme d'ontologie

III.5 Modélisation d'un robot virtuel intelligent :

III.5.1 Description de robot virtuel :

Le robot virtuel de notre travail est composée à deux partie est :

- ◆ Partie visuel : permet d'ajoute le composent robot dans l'environnement.
- ◆ Partie robot agent: est un partie le plus important dans robot virtuel, qui permet de gestion de robot visuel.

Ce robot est capable d'extraire l'information de son environnement et d'utiliser la connaissance de son environnement afin de se déplacer de manière réfléchie pour ramasser ou mettre des déchets.

III.5.2 Architecture des robots virtuels :

Nous avons mettre en évidence un système multi-agent sur lequel repose notre étude, chaque robot (agent) est autonomie, de coopèrent et de s'interagir dans le groupe. Pour ces raisons, nous avons choisi d'adopter pour un agent-robot une architecture, cette architecture est composée de quatre principaux comportements (états de robot) déplacement, mettre ramasser, déplace avec chemin et vidage poubelle.

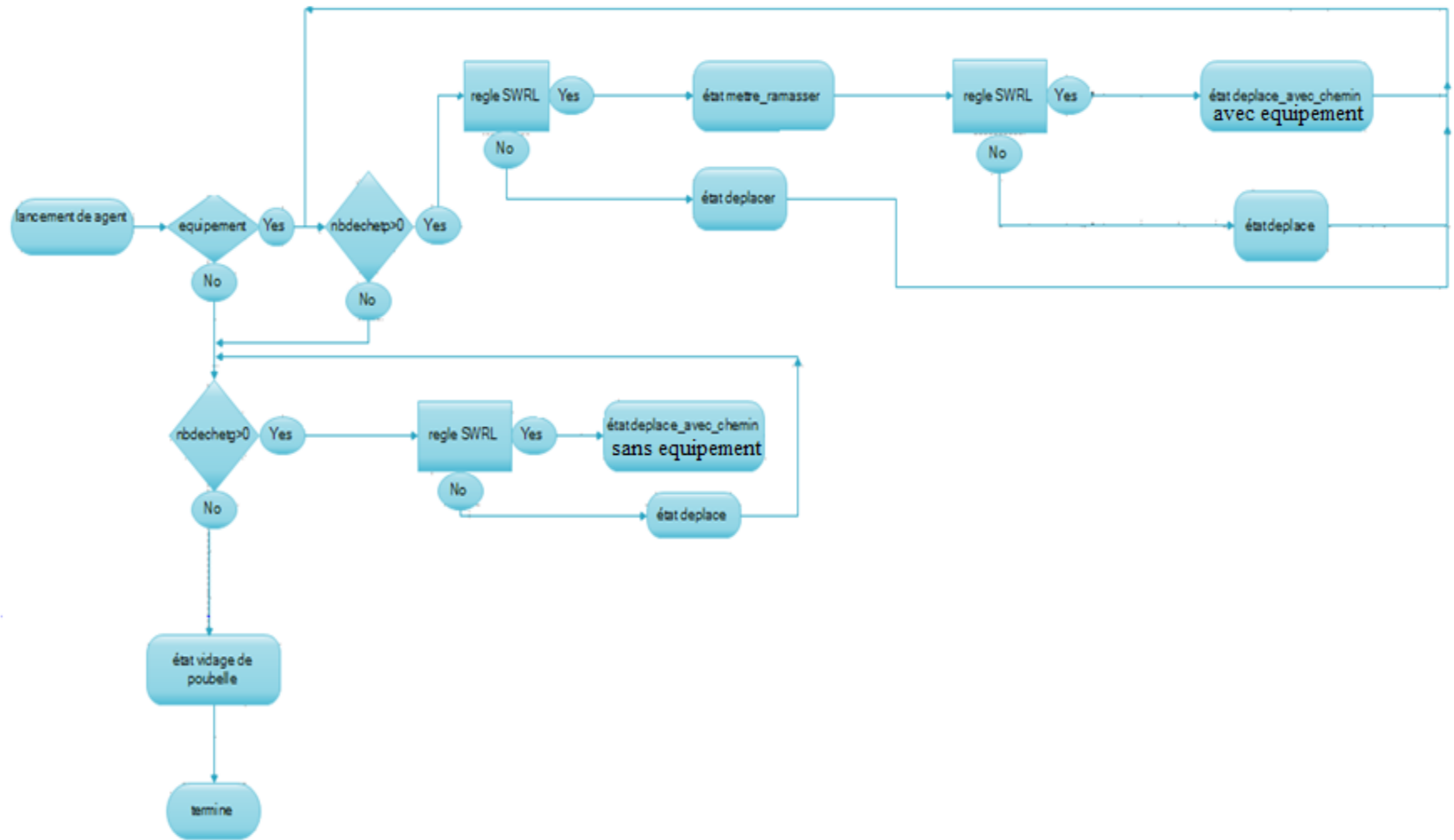


Figure III-3 : digramme d'états de robot agent

III.5.2.1 État déplacement :

Chaque robot du groupe se déplace avec un choix de direction « Droite, Haut, Gauche, Bas ». C'est-à-dire, si le robot peut, il cherchera toujours à aller à droite sans revenir sur ses pas, sinon il cherchera à aller en haut, sinon à gauche et sinon en bas. Le but de cet état est détecté les différents objets «Camion, Poubelle, déchets », évite les obstacles. Pour réaliser ce but nous implémente l'algorithme de déplacement.

Le principe d'algorithme :

- pour n robots le coordonnée de robot_i est (x_i,y_i).
- Chaque robot récupère votre coordonnée de case située robot_i(x_i,y_i), et enregistrer cette coordonnée dans un table des cases qui a été visitée.
- Créer les quatre coordonnées correspondant à la case de robot :
 $\{ \text{Haut}(x_i, y_{i+1}), \text{Gauche}(x_{i+1}, y_i), \text{Bas}(x_i, y_{i-1}), \text{Droite}(x_{i-1}, y_i) \}$.
- Calcule et trie la somme des distances entre chaque direction et les coordonnées des autres robots.

$$X = \{ x_i, x_{i+1}, x_i, x_{i-1} \}, Y = \{ y_{i+1}, y_i, y_{i-1}, y_i \}$$

$$\text{Somme} = \sum_{i=1}^n ((X_j - x_i)^2 + (Y_j - y_i)^2) \quad i/i \text{ de robot, } j=1..4$$

- Choisie la somme la plus grande (max des sommes) et récupère coordonnée de cette somme.
- Si cette case non nettoie déplace à se case. Sinon choisie la case ensuite à cette case dans le tri de somme des distances.
- Si toutes les cases directions déjà nettoie, sélectionne la case du la somme plus grande et se case n'est pas la case précédente de robot.

III.5.2.2 État mettre ramasser :

Cet état est la plus simple de robot. Principe est si le robot capte un ou plusieurs déchets dans votre position ramasse les déchets et mettre dans le panier ou poubelle.

III.5.2.3 État déplace avec chemin :

- ✚ Première partie : dans cette partie chaque robotagent envoie à tous les poubelleagents une demande de coordonnée poubelle dans l'environnement.

- Deuxième partie : les poubelleagents réceptionne toutes les demandes et envoie un message contient voter position au robotagent envoyé.
- Troisième partie : les robotagents réceptionnent toutes les réponses. La figure (IV.5-8) qui explique la communication de cet état.
- Quatrième partie : chaque robotagent applique l’algorithme A* avec tous les positions des poubelles pour choisie la poubelle le plus proche et sélectionne le chemin de déplacement. En fin déplace à cette poubelle.

Algorithme du A étoile :

A*(A-étoile) est un algorithme qui permet de rechercher le plus court chemin dans un graphe entre un "sommet source" (appelé début) et un "sommet destination" (appelé fin). Il est notamment utilisé dans les jeux vidéo [17].

Le principe de cet algorithme est :

- Recherche arborescente avec heuristique: algorithme meilleur d'abord
- Chaque nœud n est évalué selon une fonction :

$$f(n) = g(n) + h(n) \text{ où } \begin{cases} g(n) \text{ est le cout de l'état initial au noeud } n \\ h(n) \text{ est un estimé du cout du noed } n \text{ à l'état final (heuristique)} \end{cases}$$

Cette algorithme est utilisé une méthode Examiner* et le dernier appelé à la méthode Ajuster Arborescence pour confirme la solution.

En suite la méthode :

```

Procédure A*(G : Graphe, s : sommet)
  Init(G, s)
  f(s) ← g(s)
  x ← s
  TantQue x ≠ p faire
    Examiner *(x)
  Choisir un sommet ouvert x d’approximation f(x) minimale
  FinTantQue
    
```


Méthode Examiner* :

Procédure Examiner*(x : Sommet)
Pour tout successeur y de x **faire**
Si $h(x) + c(x, y) < h(y)$ **alors**
 Ajuster Arborescence(x, y)
 Ouvrir(y)
 $f(y) = h(y) + g(y)$
Finsi
Finpour
Fermer(x)

Méthode Ajuster Arborescence

Procédure Ajuster-Arborescence(x, y : Sommet)
 $h(y) = h(x) + c(x, y)$
 $p(y) \leftarrow x$

III.5.2.4 État vidage poubelle :

Cette comportement lance dans le cas l'environnement est nettoie.

- ✚ Première partie : dans cette partie chaque robotagent envoie à tous les poubelleagent et camionagent une demande de coordonnée poubelle et camion dans l'environnement.
- ✚ Deuxième partie : les poubelleagent et camionagent réceptionnent toutes les demandes et envoie un message contient voter position au robotagent expéditeur.
- ✚ Troisième partie : les robotagent réceptionnent toutes les réponses. Et applique l'algorithme A* entre la position de camion et l'autre position des poubelles trie en ascendant les distances.
- ✚ Quatrième partie : cet robotagent sélectionne la poubelle la plus proche au camion, applique l'algorithme A* entre la position de robot et la position de poubelle selecte, calcule la distance entre aux et envoyé cette distance à tout autre robotagent.
- ✚ Cinquième partie : chaque robotagent calcule les nombre des robots (nbr) qui ramasse cette poubelle (la plus proche au camion), trie les distance qui réceptionnent et sélectionnent les premiers nbr des robotagents.

- ✓ si cette robotagent dans les robotagents selecte donc déplacent à la position de poubelle en suit déplacent à la position de camion et vidage la poubelle.
- ✓ si non déplacent à la position de deuxième poubelle « de deuxième poubelle plus proche au camion ». Aller à Quatrième partie et répéter jusque vidage toute poubelle.

III.6 Conclusion :

Dans ce chapitre, nous avons commencé par proposer notre problème à résoudre et le résultat attendu. Dans la deuxième partie, nous avons présenté un modèle de Service Nettoyage Environnement adapté à la simulation de son fonctionnement. Ce modèle est décomposé à deux modèles le premier est modèle d'environnement le deuxième est le modèle de robot, en ce partie, on a présente spécialement le modèle d'environnement et l'architecture de dernière. Dans la troisième partie, nous avons exposé l'ontologie de notre travail et les règles SWRL utilise. Dans le quatrième partie, nous avons présente le modèle et l'architecture de robot et les algorithmes utilise.

Chapitre IV :

Implémentation

IV.1 Introduction :

Nous présentons, dans ce chapitre les principales étapes de notre réalisation. Cette implémentation traduit le passage du modèle consiste qu'il construise à partir d'une représentation mentale du système liée à une certaine connaissance qu'en a acquis le modélisateur vers un produit informatique.

Pour l'implémentation de notre logiciel, utilise la plate-forme jade. La mise en œuvre des différentes entités se fait en utilisant les outils correspondants dans la plate-forme jade dans la partie SMA et dans la partie graphique en utilisant javafx. Les signaux de commande sont représentés par des messages et les comportements des agents sont implémentés en utilisant du code jade et les règles de SWRL de notre ontologie. Le simulateur ainsi formé constitue le système Multi-Agents dont le service de nettoyage dans un quartier.

IV.2 Les outils utilisés :

IV.2.1 NetBeans :

NetBeans est un projet open source ayant un succès et une base d'utilisateur très large. Sun Microsystems a fondé le projet open source NetBeans en Juin 2000 et continue d'être le sponsor principal du projet.

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X [32].

Aujourd'hui, deux projets existent: L'EDI NetBeans et la Plateforme NetBeans.

- ❖ L'EDI NetBeans (Environnement de Développement Intégré) : est un environnement de développement - un outil pour les programmeurs pour écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java - mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre l'EDI NetBeans [30].
- ❖ La Plateforme NetBeans : c'est également une plateforme. Il vous est possible de créer votre propre application Awt ou Swing, basée sur la plateforme NetBeans.

IV.2.2 JavaFX :

IV.2.2.1 Définition :

JavaFX est une famille de produits et de technologies de Sun Microsystems qui appartient à Oracle, qui base sur la machine virtuel Java pour fonctionner donc la communication avec des applications java standard est très simple. Une application JavaFX a accès à toutes les classes fournies par la machine virtuelle java [2].

Les produits JavaFX ont pour but de créer des applications internet riches (RIA) et Facilite le développement avec images, graphiques, audio et vidéo. Actuellement JavaFX est constitué de JavaFX Script et de JavaFX Mobile, bien que d'autres produits soient prévus [23].

IV.2.2.2 Les avantages :

- ◆ Basé sur Java (Java SE et ME).
- ◆ Utilisable sur tous les écrans : navigateurs, mobile, TV, etc.
- ◆ Open Source.

- ◆ Déploiement sur navigateur et ordinateur de bureau "Desktop" sans modification.
- ◆ Collaboration designers et développeurs.
- ◆ Possibilité d'intégrer des codes en Java et JavaFX
- ◆ Moins de code pour générer une interface et des composants graphiques(NSY)

IV.2.3 La plate-forme JADE :

IV.2.3.1 Définition :

JADE (Java Agent DEvelopment framework) est une plate-forme multi-agent créé par le laboratoire TILAB et décrite par Bellifemine et al. JADE permet le développement de systèmes multi-agents et d'applications conformes aux normes FIPA. Elle est implémentée en JAVA et fourni des classes qui implémentent « JESS » pour la définition du comportement des agents. JADE possède trois modules principaux (nécessaire aux normes FIPA).

DF « Director Facilitator » fournit un service de « pages jaunes» à la plate-forme ;

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

AMS « Agent Management System » supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système.

Ces trois modules sont activés à chaque démarrage de la plate-forme [26].

IV.2.3.2 Configuration Jade avec NetBeans :

Pour configuration jade avec NetBeans, procédons de la manière suite :

Démarré NetBeans et à partir du menu Tools, sélectionné Ant Library comme suite :

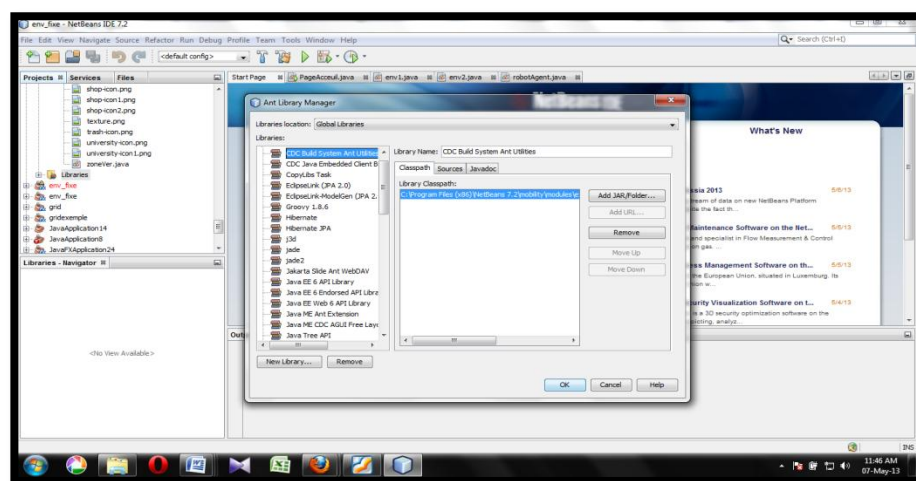


Figure IV-1 : Fenêtre pour ajout un Library

1. Cliquez sur le bouton **New Library** dans la boîte de Ant Library et saisissez le nom de la bibliothèque par exemple **JADE** et cliquez sur **Ok**.

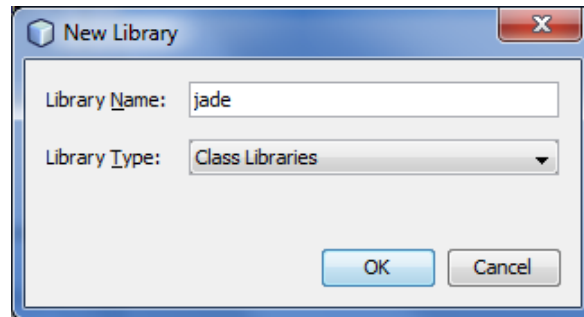


Figure IV-2 : Boîte de saisie le nom de Library

2. Cliquez sur les boutons **Add JAR/Folder** pour parcourir les fichiers jar de la plateforme jade dans le fichier lib est `http.jar`, `iio.jar`, `jade.jar`, `jadeTools.jar` et cliquez sur les boutons **Add JAR/Folder**, puis cliquez sur **OK**.

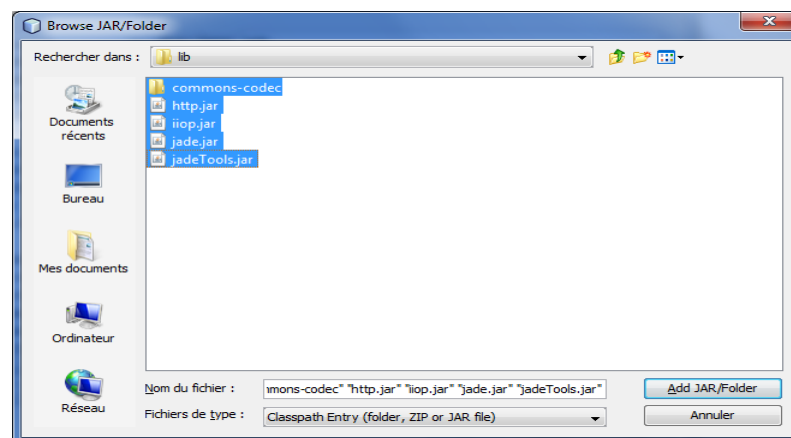


Figure IV-3 : Fenêtre de parcourir le fichier jar

3. Si on a besoin des bibliothèques de la plateforme jade on clique avec le bouton droit de la souris sur **Libraries** et sélectionne le libraire et cliquez sur **Ok**.

IV.2.4 PROTÉGÉ-OWL :

L'éditeur Protégé-OWL est une extension de Protégé qui prend en charge le langage d'ontologie Web (OWL). Protégé est développé par l'Université de Stanford en open-source et en Java, et avec une vision Plug-and-Play, une ontologie OWL peut inclure des descriptions des classes, propriétés et leurs instances [33].

OWL qui permet de faire la comparaison entre des propriétés et des classes: identité, équivalence, contraire, symétrie, cardinalité, transitivité, disjonction, etc [Holger Knublauch].

L'éditeur Protégé-OWL permet aux utilisateurs de:

- Charger et sauvegarder des ontologies OWL et RDF.
- Éditer et visualiser des classes, les propriétés et les règles SWRL.
- Définir les caractéristiques de classes logiques comme des expressions OWL.
- Exécuter raisonneurs comme la description classificateurs logiques.
- Modifier individus OWL pour balisage sémantique Web.

IV.3 Organisation de code :

IV.3.1 Diagramme de classes :

La construction du simulateur a été effectuée après modélisation. Grâce à la modélisation objet du langage Java, on commence par modéliser les environnements et les robots en cherchant leurs attributs et leurs méthodes en fonction de leurs nécessités, Dans un second temps, il faut réfléchir à la partie visuelle : réfléchir aux fonctionnalités qui peuvent être utiles à l'utilisateur ou encore réfléchir à la disposition des éléments d'une fenêtre pour que ce soit le plus pratique et le plus joli possible.

C'est ainsi que le programme de simulateur a aboutit au diagramme de classe ci-dessous.

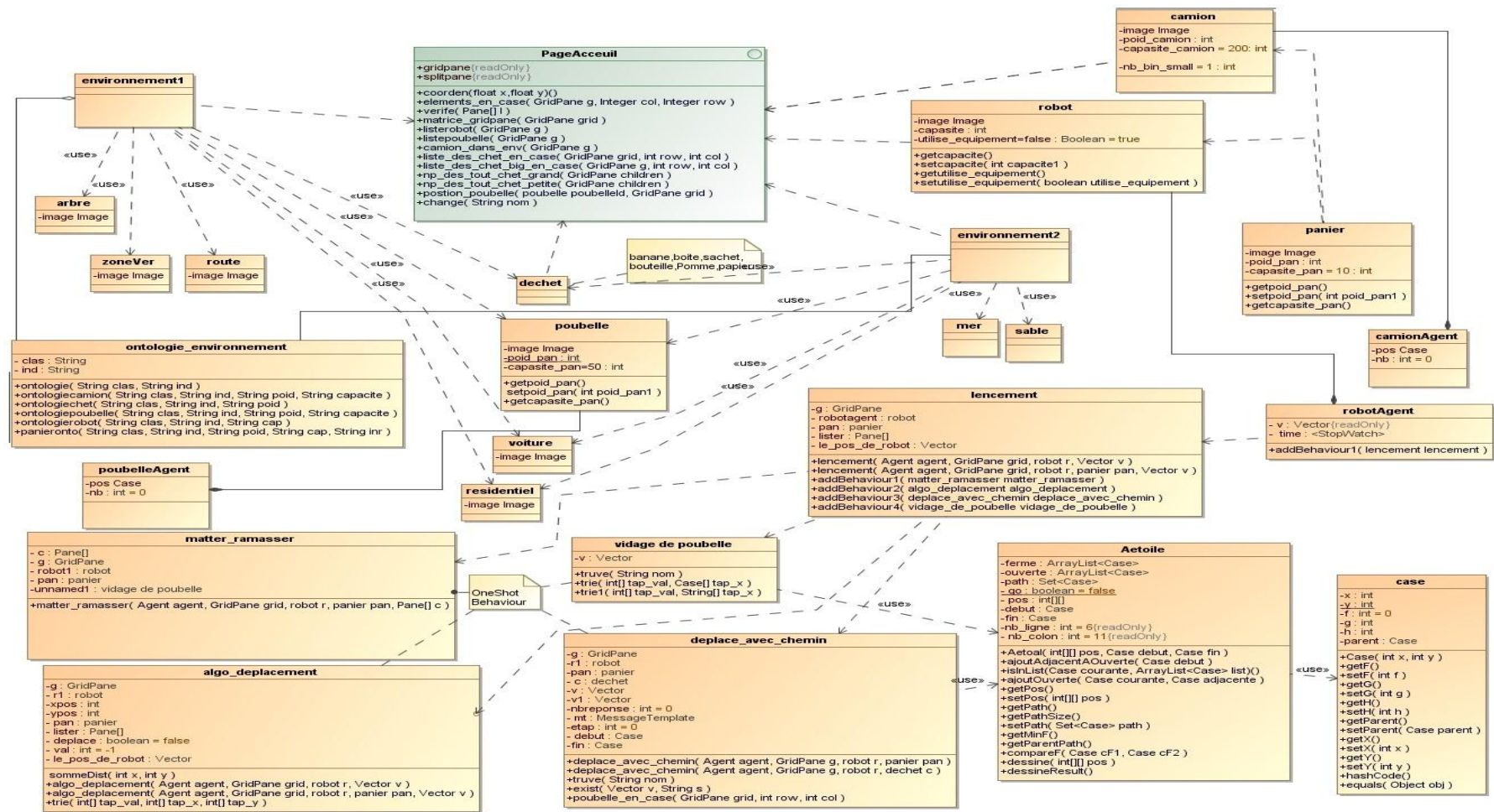


Figure IV-4: Structure du programme

IV.3.2 **Détail des classes**

Nous allons présenter plus en détail le cœur du programme, et chaque élément dans le diagramme précédant sera expliqué.

- Classe « PageAccueil » : sélectionne l'environnement pour nettoyer, avec la possibilité d'ajoute l'équipe de nettoyage et les déchets dans l'environnement spécifié. Elle contient aussi plusieurs méthodes public et static parce que cette classe représente le lien entre tout les classe.
- Classe « env1 » ou Classe « env2 »: Cette classe représente l'environnement à nettoyer et un agent pour lancement les autre agents (robotAgent, poubelleAgent, camionAgent) , elle contient une GridPane ce dernier est un ensemble de tout les objets dans l'environnement.
- Classe « robotAgent» : C'est l'objet les très important dans notre simulateur, cette classe est aspect virtuel de la classe visuel « classe robot » qui permet l'exécution de tous les traitements.
- Classe « poubelleAgent» , Classe « camionAgent» : ce deux classes sont avoir le même implémentation du la classe robotAgent mais la fonctionnalité de cette classe est réponde les message qui envoyé par robotAgent .
- Classe « algo_deplacement » , « matter_ramasser » , « deplace_avec_chemin » et « vidage_de_puobelle »: ces classe présente les Beauvoir qui utilise par robot pour réalise un action .
- Classe « dechet » : Un déchet peut être soit un banane, soit un pomme, soit une bouteille , soit une sachet , soit un papier, soit une boite.
- Classe « résidentiel » , Classe « voiture » , Classe « arbre » , Classe « mer » : ces classes permettent de dessiner les objet résidentiel, voiture, arbre, mer en respectivement de type obstacle.
- Classe « route » , Classe « robot » , Classe « zonevert » , Classe « poubelle » , Classe « panier » » , Classe « sable » : ces classes permettent de dessiner les objet route ,robot, zonevert ,poubelle, panier ,sable en respectivement de type non obstacle.
- Classe « Aetoile »: Cette classe définit le chemin de déplacement de robot.
- Classe « Case »: Cette classe utilise par les autres classes pour défini le case.
- Classe « lencement » : Cette classe permet de lancer la simulation.

IV.4 Présentation de simulateur :

IV.4.1 Présentation de l'interface graphique :

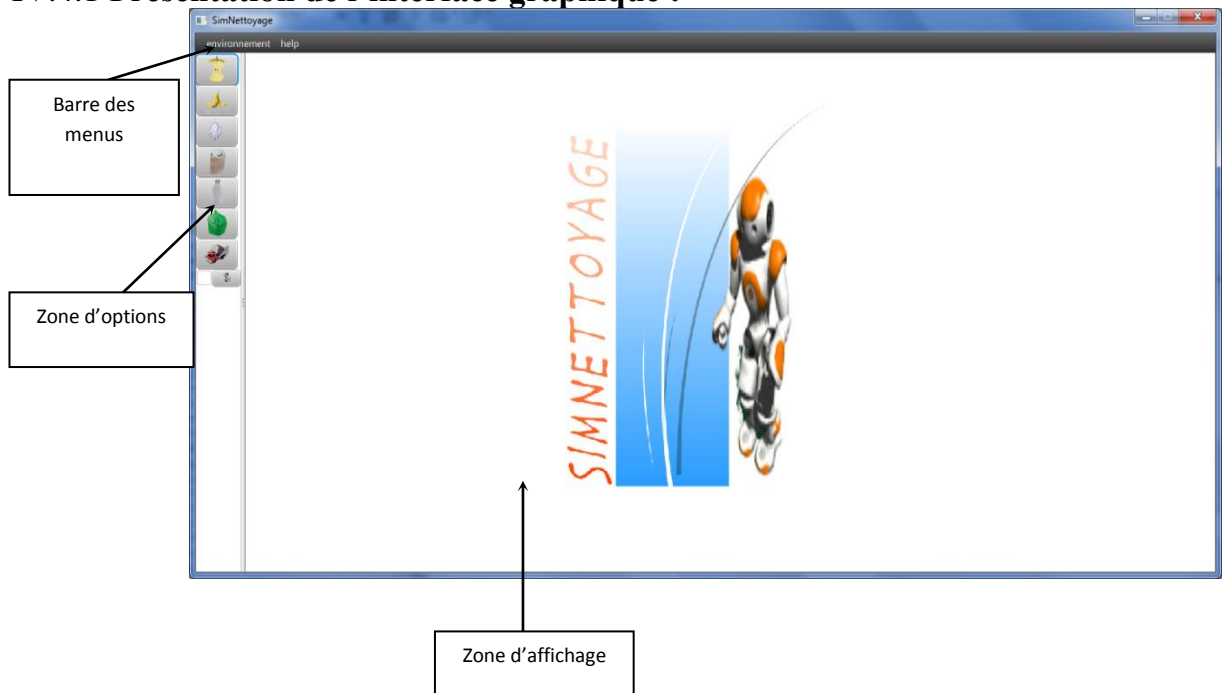


Figure IV-5 : interface du simulateur

IV.4.2 Descriptions détaillées des différentes parties de l'interface :

IV.4.2.1 Barre des menus :

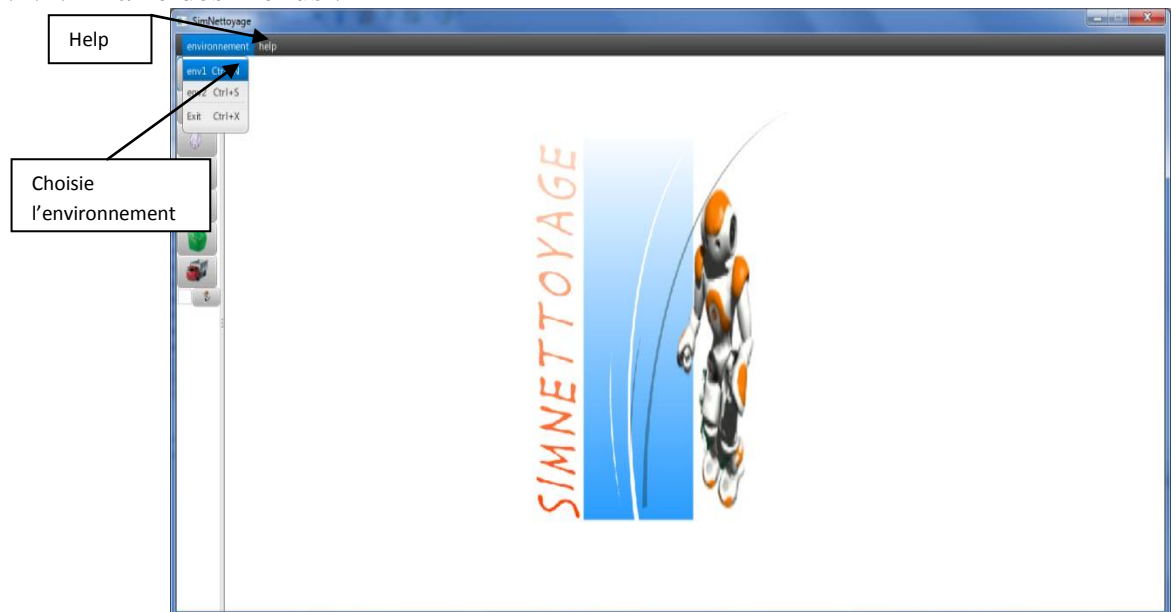


Figure IV-6 : description de barre des menus

Si choisie env1



Lancement de nettoyage

Figure IV-7: interface de l'environnement 1

Si choisie env2



Figure IV-8: interface de l'environnement 2

IV.4.2.2 Zone d'options :

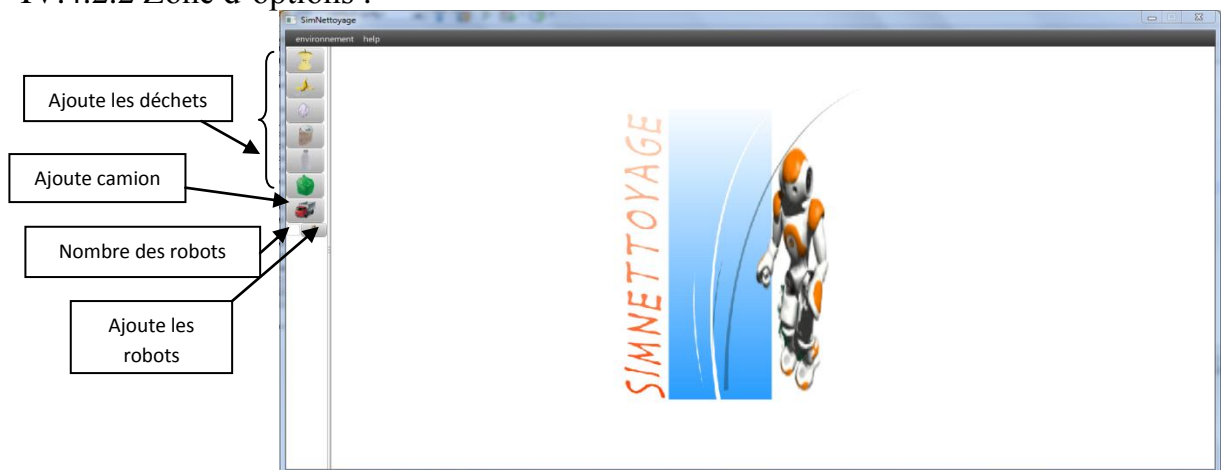


Figure IV-9: description de barre des menus

IV.4.3 L'interface en cours d'exécution :

IV.4.3.1 Avant lancement :

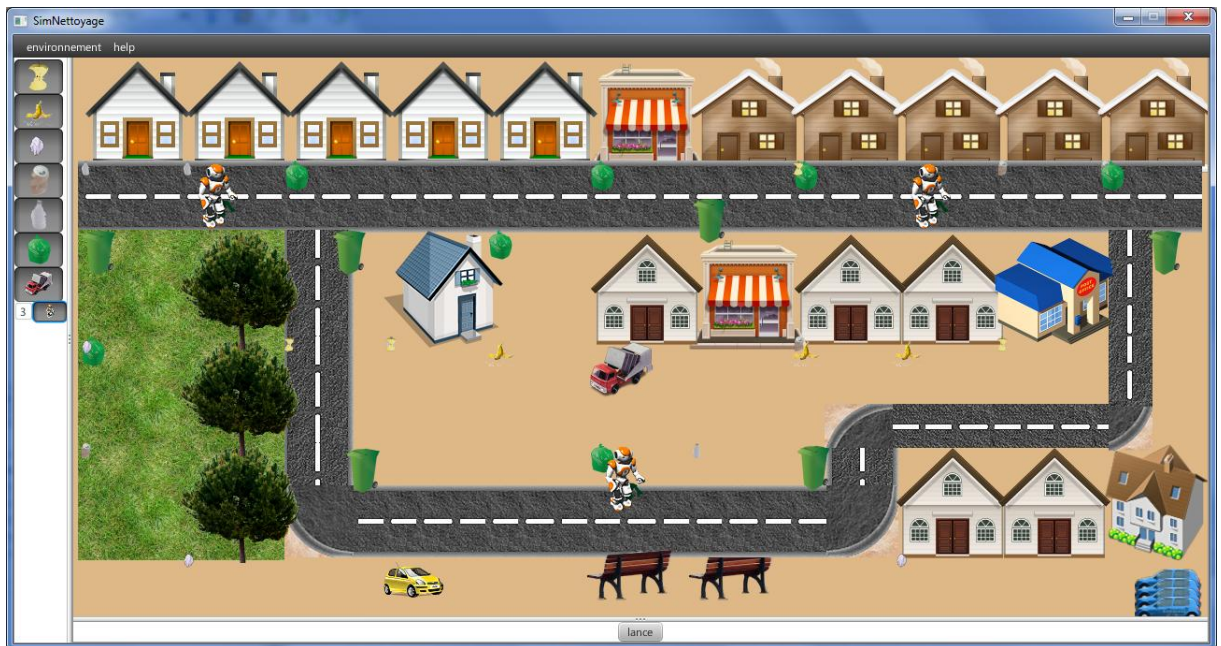


Figure IV-10 : Capture d'écran du simulateur SimNettoyage avant lancement

IV.4.3.2 Après lancement :

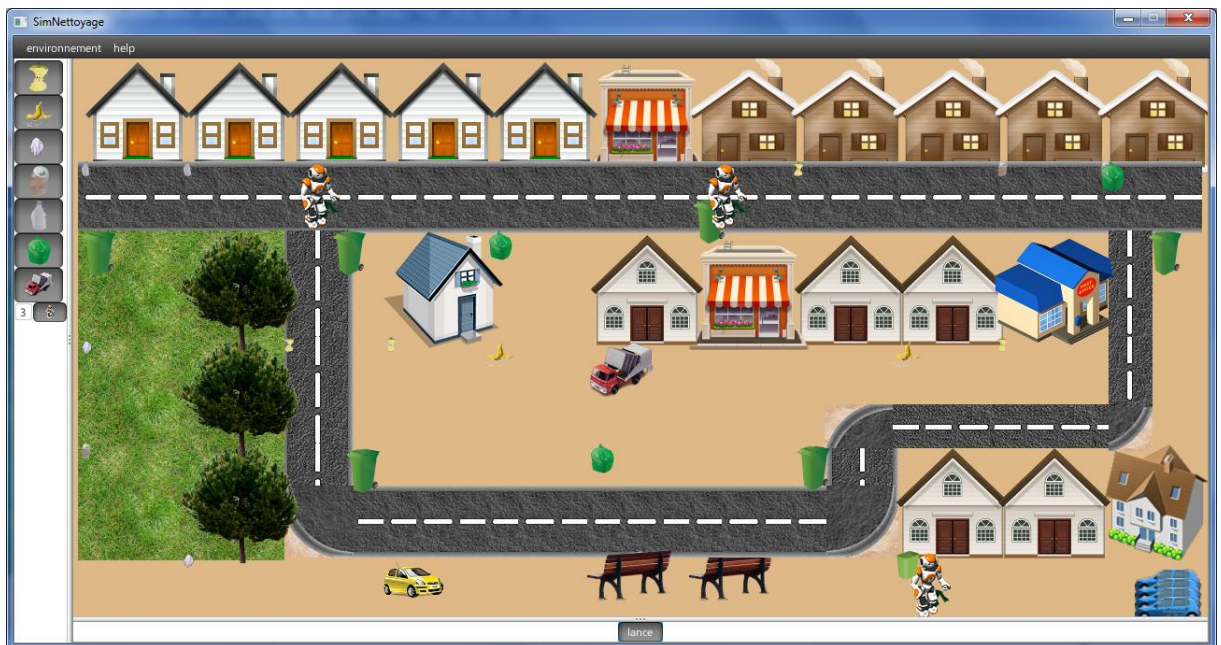


Figure IV-11 : Capture d'écran du simulateur SimNettoyage après lancement

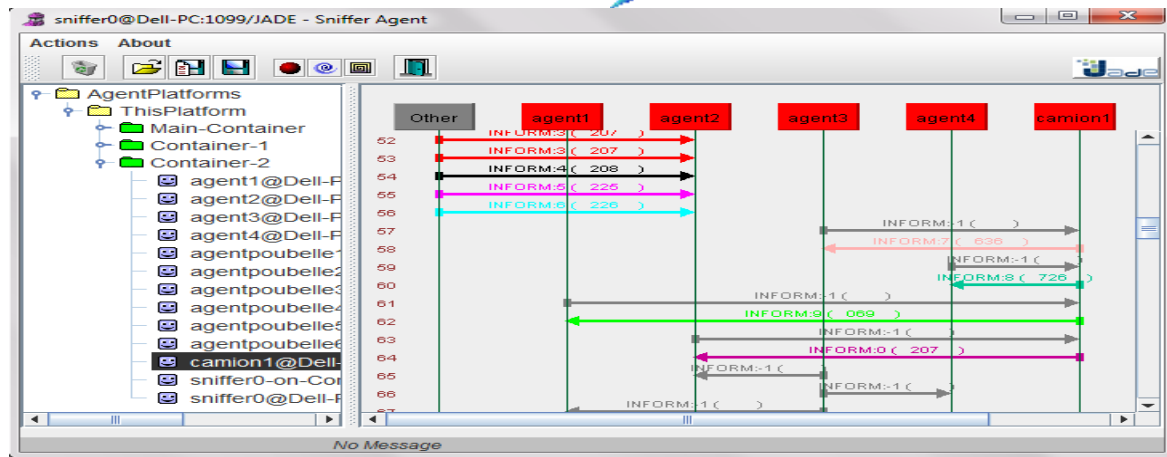


Figure IV-14 : L'interface de l'agent Sniffer pour la communication entre robotAgent et camionAgent

IV.5 Résultats des simulations :

Nous avons réalisé plusieurs simulations à partir des exemples simples de simulation de robots. Pour obtenir le tableau ci-dessous qui est présenté les nombre de déchet, le nombre de robot et le temps simulation, pour trouve une relation entre eux.

Nombre des déchets	Nombre des robots	Temps de simulation
0	2	00 :00 :48 :434
0	3	00 :00 :53 :230
0	4	00 :01 :01 :467
0	5	00 :01 :08 :622
05	2	00 :01 :53 :274
05	3	00 :01 :43 :114
05	4	00 :01 :39 :224
05	5	00 :02 :16 :657
10	2	00 :02 :14 :567
10	3	00 :02 :20 :272
10	4	00 :02 :30 :366
10	5	00 :02 :13 :207
15	2	00 :02 :16 :748
15	3	00 :02 :25 :373
15	4	00 :02 :24 :224
15	5	00 :02 :20 :376

Tableau IV.5-1:le tableau de résultat de simulation

Les courbes ci-dessous représentent la variation du délai de nettoyage (N) en fonction du Temps Inter-Lancement nombre de (robot), pour chaque valeur de nombre de (déchet).

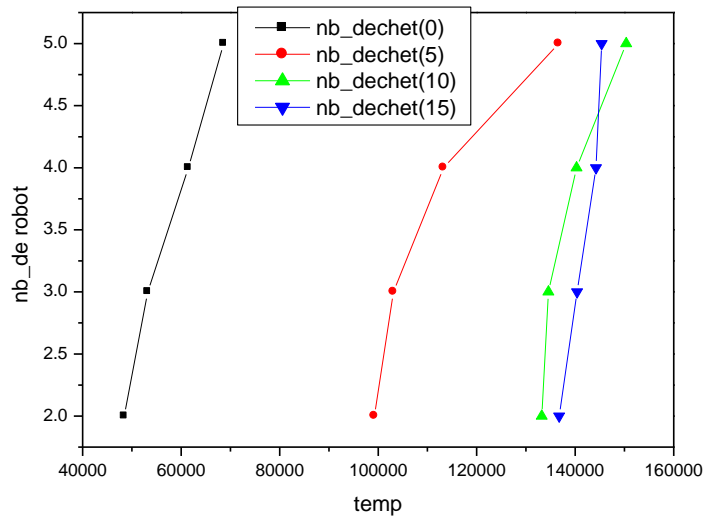


Figure IV-15 : Les courbes représentent la Temps de nettoyage Inter-Lancement nombre de (robot), pour chaque valeur de nombre de (déchet).

IV.5.1 Discussion

Grâce à la note de la courbe que plus le nombre de robots augmentation du temps passé dans le nettoyage et la raison de cette communication existe entre les robots



IV.6 Conclusion :

Dans ce chapitre, nous avons essayé de donner des éléments qui nous ont permis d'implanter notre simulateur et les outils utilisés à savoir la plate-forme Jade, le javaFX et Protégé OWL. Nous avons présenté les classes en détail et le simulateur de nettoyage que nous avons réalisé pour valider les concepts introduits dans le chapitre 3.

Conclusion générale

La modélisation et la simulation sont très utiles pour comprendre et prédire la dynamique des différents Service Nettoyage Environnement. L'approche mathématique a joué un rôle primordial dans les recherches dans ce domaine.

Les approches Multi-agents et l'ontologie sont intéressante et des outils puissant pour traiter ce type de problèmes. Grâce à ses différentes caractéristiques elle permet une élaboration facile du modèle informatique à partir d'un système réel est celui construit ce qui aide à une représentation fidèle du service étudié.

L'objectif global de ce travail était de montrer un simulateur robot (environnement multi robot) à base d'un system multi robot. L'application a concerné à présente l'aspect de coopération entre le nettoyeur, pour réalise des taches est dépend à la base de connaissance de chaque robot Intelligent.

Le simulateur fournit un environnement virtuel dans lequel évolue le nettoyage d'environnement avec l'application des règles SWRL pour réalise un tâche. Il possède une interface permettant le choix d'environnement et la possibilité d'ajoute le déchet pour nettoyer et l'équipe de nettoyage.

Les difficultés que nous avons rencontrées est de savoir comment implémente jade dans un application de JavaFX .

notre travail Utilisé par le maire, par exemple, pour déterminer le nombre des nettoyeurs nécessaires pour nettoyage d'une quartier et la paie nécessaires de chaque nettoyeur , la possibilité de remplace les nettoyeurs par des robots.

Les perspectives envisagées pour améliorer ce travail sont multiples, Nous identifions plusieurs perspectives :

- Améliorer le simulateur au 3D et l'environnement au type dynamique.
 - Ajouter l'animation au simulateur comme mouvement de personne.
-

Bibliographie

- [1] **Achichi, Boubakeur.** *Convention sociale dans les SMAs Analyse, Modélisation et Simulation.* Octobre 2010.
- [2] **Alan, Moreau.** *JavaFX.*
- [3] **B, Chaib-draa, I, Jarras and B, Moulin.** *Systèmes multiagents : Principes généraux et applications .* 2001.
- [4] **BADEIG, Fabien.** *Un environnement actif pour la simulation multi-agents.* septembre 2010.
- [5] **Balaguer, Benjamin, et al.** *USARSim: a validated simulator for research in robotics and automation.*
- [6] **Boimond, Jean-Louis.** *ROBOTIQUE.*
- [7] **BOUSSEBOUGH, Imane.** *LES SYSTEMES MULTI-AGENTS DYNAMIQUEMENT ADAPTABLES.* 2011.
- [8] **BUFFET, Olivier.** *Une double approche modulaire de l'apprentissage par renforcement pour des agents intelligents adaptatifs.* 2003.
- [9] **CHATILA, Raja.** *La Robotique : une science aux multiples facettes.*
- [10] **DROGOUL, ALEXIS.** *De La Simulation Multi-Agent A La Résolution Collective de Problèmes.* .1993.
- [11] **Ferber, Jacques.** *Les Systemes Multi Agents: vers une intelligence.* 1995.
- [12] **Gourdeau, Richard.** *Modelisation des robots manipulateurs.* novembre 2010.
- [13] **HACENE, BELHADEF.** *Islam : Système d'information pour l'aide à la décision spatiale basé sur une ontologie.* 2010.
- [14] **Holger Knublauch, Ray W. Ferguson, Natalya F. Noy and Mark A. Musen.** *The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications.*
- [15] **Imane, Melle BELLIFA.** *Approche Multi-Agent pour la reconnaissance de Diabète.* 03 Juillet 2011.
- [16] **Islam, AMIRA.** *DEVELOPEMENT D'UN ROBOT JOUEUR D'ECHECS.*
- [17] **Kedad-Sidhoum, Safia.** *Plus courts chemins : Algorithme A*.* 2005.
- [18] *Locomotion d'un robot mobile.* Mai 2008.
- [19] **Matignon, Laetitia.** *Introduction à la robotique.* 2012.
- [20] **O'Connor, Martin.** *Semantic Web Rule Language.*

- [21] **Sahki, Hanine.** *Vers une Architecture d'un Système de Dialogue Multi Agents Basé sur l'Argumentation Application à la négociation dans le domaine de e-commerce.* 2008.
- [22] **SAID, LABED.** *Systèmes Complexes Adaptatifs Application au traitement des images.* 2006.
- [23] **TISSOT, Thomas.** *Espace de travail collaboratif pour un environnement de grille pervasive.* mai 2010.
- [24] **Treuil, Jean-Pierre and Drogul, Alexis.** *MOD2LISATION ET SIMULATION A BASE D'AGENTS.* 2008.
- [25] **Van, NGUYEN Nhu.** *L'interaction dans simulation multi agents .* 2007.

Webgraphie

- [26] **Adel, FERGUEN.** la plate-forme JADE. *limsi*. [Online]
http://perso.limsi.fr/jps/enseignement/examsma/2005/1.plateformes_3/index-Ferguen.html#Prot.
- [27] **CARMEN.** *carmen.sourceforge*. [Online] 2006. <http://carmen.sourceforge.net/doc/>.
- [28] **Cyberbotics Ltd.** Webots7. *cyberbotics*. [Online] 2013. <http://www.cyberbotics.com/>.
- [29] **EASY-ROB.** EASY-ROB Robotik. *easy-rob*. [Online] 2013. <http://www.easy-rob.com/>.
- [30] **équipe Java.** FAQ netbeans. *netbeans.developpez*. [Online] 2009.
<http://netbeans.developpez.com/faq/?page=Plateforme>.
- [31] Microsoft robotics developer studio. *generationrobots*. [Online]
<http://www.generationrobots.com/index.cfm>.
- [32] **ORACLE.** netbeans AID. *netbeans*. [Online] 2013. <https://netbeans.org>.
- [33] protege-owl api. *protege*. [Online] 2013. <http://protege.stanford.edu/>.
- [34] Robot Simulator. *hangsim.com*. [Online] 2012. <http://hangsim.com/rbs/index.php>.
- [35] Simbad Project Home. *simbad.sourceforge.net*. [Online] 2011. <http://simbad.sourceforge.net>.