

République Algérienne Démocratique Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université d'Ibn Khaldoun – Tiaret

Faculté des Mathématiques et de l'Informatique

Département Informatique



Thème

Développement de mécanisme à base des colonies de fourmis pour la coopération des robots mobiles.

Pour l'obtention du diplôme de Master II

Spécialité : Génie Informatique

Option : système d'information et technologie de web

Rédigé par :

-Mr.BOUSSERIA Youcef

-Mr.BOUBEKEUR Sid Ahmed

Dirigé par :

-Dr. HATTAB Nouredine

Jury par :

- Dr.BENDAOU Mbarek

-Dr.ZIOUL Tahar

Année universitaire: 2016-2017

Dédicace

À mes parents. Que dieu leur procure bonne santé et longue vie.

À la lumière de mes jours, la source de mes efforts, la flamme de Mon cœur, ma vie et mon bonheur ; MAMAN que j'adore.

À celui que j'aime beaucoup et qui m'a soutenu tout au long de Ce projet et bien sûr À mes amis

Mon binôme Sid ahmed, Zakaria, Mohamed, Osman, Ilyes, Fawzi, Abdelmalek, Abderrahman, hocine, hicham, kamel, Jilali ...

Sans oublier mes grands-parents, mes tantes, mes oncles, mes cousins et mes cousines.

À toute ma famille, mes amis.

Et à tous ceux qui ont contribué de près ou de loin pour que ce Projet soit possible, je vous dis merci.

BOUSSERIA Youcef

Dédicace

À mes parents. Que dieu leur procure bonne santé et longue vie.

À la lumière de mes jours, la source de mes efforts, la flamme de Mon cœur, ma vie et mon bonheur ; MAMAN que j'adore.

À celui que j'aime beaucoup et qui m'a soutenu tout au long de Ce projet et bien sûr À mes amis

Mon binôme *Youcef, Missoum, Mohamed, Aminé, Zakaria, Osmané, Ilyes, Fawzi, Abderrahmané, Hociné, mohamed, Hicham, Kamel, Aminé,.....*

Sans oublié mes grands-parents, mes tantes, mes oncles, mes cousins et mes cousines.

À toute ma famille, mes amis.

Et à tous ceux qui ont contribué de près ou de loin pour que ce Projet soit possible, je vous dis merci.

BOUBEKEUR Sid ahmed

Remerciements

Nous tenons tout d'abord à remercier le bon Dieu de nous avoir guidé et donné la force et la volonté pour atteindre notre objectif.

Nous remercions nos très chers parents pour leur soutien et leur patience.

Nous tenons à exprimer nos vives gratitudee et notre profonde reconnaissance à notre encadreur Mr. HATTAB Nourddine pour ses précieux conseils qui nous ont gardés sur le droit chemin afin de réaliser ce modeste travail.

Nous remercions tous nos enseignants depuis l'école primaire jusqu'à l'université, qui ont contribué à notre formation, auxquels nous exprimons notre plus grand respect et profonde reconnaissance.

Nous plus vifs remerciements aux membres de jury qui nous font l'honneur de présider et d'examiner ce modeste travail.

Egalement, nos remerciements à tous ceux qui nous ont aidés de près ou de loin dans la réalisation de ce projet de fin de cycle.

BOUSSERIA Youcef & BOUBEKEUR Sid ahmed

Résumé

La coopération d'un ensemble d'entités (mobile ou non), n'est pas un nouveau domaine. Au début les chercheurs en science des vivants (biologistes, éthologues..), ensuite les roboticiens et les chercheurs du domaine d'automatique, et aujourd'hui les informaticiens, ont fait leurs études sur la compréhension de la coopération des insectes, robots, etc.

De nos jours la modélisation à base d'agent est en réelle effervescence, une effervescence résulte particulièrement au nombre important de scientifique qui utilisent ce concept.

L'idée de base de cette technique est résumée par le fameux slogan '**tout composant est un agent**'. A ce stade, n'importe quel phénomène est modélisé par un ensemble d'entités (agents) qui le composent et qui interagissent entre eux.

Le but de notre travail consiste à modéliser un système de participation de robots mobiles afin d'atteindre une/des cible(s) (fixe ou mobile) avec un langage de modélisation basé sur les agents et la coopération entre eux, en tenant compte des contraintes temporelles et environnementales du système étudié.

Le travail de recherche mené dans ce mémoire consiste à proposer des mécanismes basés sur les colonies de fourmis, une architecture en couche pour l'agent (robots) a été proposée afin de modéliser cette coopération, Nous avons opté pour l'approche UML pour décrire les principales fonctionnalités de notre système.

Finalement, nous avons implémenté notre approche pour la valider expérimentalement.

Mots clés: Modélisation, Simulation SMA/UML, Net Logo.

Abstract

The cooperation of a set of entities (mobile or not) is not a new domain. At the beginning, researchers in the science of the living (biologists, ethologists), then roboticists and researchers in the field of automation, and now computer scientists, have studied understanding of the cooperation of insects, robots, etc.

Nowadays agent-based modeling is a real effervescence, an effervescence results particularly to the large number of scientists who use this concept.

The basic idea of this technique is summarized by the famous slogan 'every component is an agent'. At this stage, any phenomenon is modeled by a set of entities (agents) which compose it and which interact with each other.

The aim of our work is to model a system of participation of mobile robots in order to reach target (s) (fixed or mobile) with a modeling language based on the agents and the cooperation between them, taking into account the Temporal and environmental constraints of the system studied.

The research work carried out in this paper consists of proposing mechanisms based on ant colonies, a layered architecture for the agent (robots) has been proposed in order to model this cooperation. We have opted for the UML approach to describe the main features of our system

Finally, we implemented our approach to validate it experimentally.

Keywords: Modeling, Simulation SMA / UML, Net Logo.

Table des matières

1 Introduction générale.....	1
1.1 Contexte & Motivation.....	1
1.2 Problématique.....	2
1.3 Objectif.....	4
1.4 Structure de mémoire	5
Chapitre 1 : Modélisation & Simulation	7
Introduction	7
1 La modélisation	7
1.1 Un modèle	7
1.2 La modélisation	7
1.2.1 Modélisation des processus d'affaires.....	7
1.2.2 Modélisation du système d'information.....	7
2 Les paradigmes de modélisation	10
2.1 Un paradigme	10
3 MERISE	10
4 Orienté Objet	10
4.1 Un langage unifié pour la modélisation objet (UML).....	10
5 Paradigme composant.....	10
6 Paradigme service.....	11
7 Paradigme Agent	11
7.1 La vue agent	11
7.2 L'Agent face à l'Objet.....	12
7.2.1 Des similarités	12
7.2.2 Des différences	12

7.3 Propriétés de l'agent.....	13
7.3.1 Autonome.....	13
7.3.2 Réactivité.....	13
7.3.3 Proactif.....	13
7.3.4 Flexible.....	13
7.3.5 Social.....	13
7.3.6 Situé.....	13
7.4 Classification des agents.....	13
7.4.1 Les agents Réactifs.....	14
7.4.2 Les agents Cognitifs.....	14
7.4.3 Un agent Hybride.....	15
8 System multi agent.....	15
8.1 Les caractéristiques des systèmes multi-agents.....	17
8.2 Interaction entre agents.....	17
8.3 Communication dans les systèmes multi-agents.....	18
8.4 Domaines d'application des S.M.A.....	19
8.4.1 <i>Automatique et Robotique</i>	20
9 La simulation informatique.....	20
9.3 Les objectifs de la simulation informatique.....	21
Conclusion.....	22
Chapitre 2 : Coopération dans les SMA.....	23
Introduction.....	23
1 Cadre d'étude.....	23
2 La coopération.....	24
3 La coordination.....	25
4 La collaboration.....	25

5 Etat De L'art sur la coopération	26
5.1 Des outils pour la coopération.....	27
5.2 Des approches multi agent pour la coopération:	28
5.2.1. Approche par contrôle	28
5.2.2 Approches de Construction	31
6 Coopération intentionnelle et non-intentionnelle	31
7 Les avantages de la coopération.....	32
8 Étude des différents travaux	33
8.1 L'apport de la coopération pour la résolution	33
8.1.1 Les travaux de Bouron & Ferber	33
8.2 La coopération en tant qu'attitude sociale.....	34
8.2.1 Les travaux de Sekaran & Sen	34
Conclusion.....	35
Chapitre 3 : Colonie de Fourmis	36
Introduction	36
1 La biomimétique et l'informatique.....	36
2 Les fourmis artificielles.....	37
3 Les algorithmes colonies de fourmis	37
3.1 Les traces de phéromones	39
3.2 La construction des solutions	39
3.3 Mise à jour des traces	40
3.4 Algorithme général de colonie de fourmis	40
4 L'optimisation multi-objectif par colonies de fourmis.....	41
4.1 Les traces de phéromones	41
4.2 Le processus de mise à jour des solutions.....	41
4.2.1 La mise à jour individuelle.....	42

4.2.2 La mise à jour globale	42
4.3 La visibilité.....	42
4.4 La construction des solutions	42
4.4.1 Ciblées.....	42
4.4.2 Dynamiques.....	42
4.4.3 Fixes	42
4.5 Évaluation des solutions.....	43
4.5.1 Pareto.....	43
4.5.2 Non-Pareto	43
Conclusion.....	43
Chapitre 4 : Implémentation & Expérimentation	44
Introduction	44
1 Les plateformes de simulation Multi-Agent.....	44
2 NetLogo une plateforme générique.....	44
2.1 Motivation pour NetLogo et définition	44
2.2 La répartition des agents NetLogo	45
2.3 Le modèle Net Logo, les étapes de mise en œuvre	45
2.4 Le Langage NetLogo et langage UML	46
3 Cycle de Vie d'un Logiciel Multi agent.....	46
4 Les Tâches Générique et la modélisation Orienté Agent	47
4.1 Description de la tâche étudiée (Le fourragement modifié).....	48
4.2 Processus Suivi et Approche Adopté	48
4.3 Analyse de la Tâche générique étudiée	49
5 La conception.....	49
5.1 Aperçu détaillé sur les mécanismes de coopération souhaité	51
6 Démarche	52

6.1 Identifier les variables globales.....	52
6.2 Identification des propriétés des objets à modéliser.....	53
6.2.1 Pour l'environnement	53
6.2.2 Pour les robots.....	54
6.2.3 Pour les objets	54
6.2.4 Pour les cibles.....	54
6.3 Organiser les entités entrent-elles	55
7 Implémentation.....	57
7.1 La problématique énergétique des robots-turtles	59
7.2 Présentation du simulateur "SC_SMA"	60
7.2.1 Le Tableau de Bord Global	60
7.3 Etude empirique	62
8. Expérimentation et résultats:	63
8.1 Concernent le temps	65
8.2 Concernent l'énergie	66
8.3 Concernent la distance	67
Synthèse	67
Conclusion.....	68
Conclusion Générale & Perspectives	69
Bibliographies	71
Webographie	79

Liste des Figures

Figure 1 : Notre Démarche.....	5
Figure 2 : Organisation de notre mémoire.....	6
Figure 3 : Structure générale d'un agent réactif dans un système Multi-Agents	14
Figure 4 : Structure générale d'un agent cognitif dans un système Multi-Agents.....	15
Figure 5 : Une image globale d'un Système Multi-Agents	17
Figure 6 : Communication via un Tableau Noir.....	19
Figure 7 : Communication par me envoi de messages	19
Figure 8 : La simulation informatique selon Fishwiki [6].....	20
Figure 9 : Vue d'ensemble de la simulation informatique [6].....	21
Figure 10 : Une classification pour les approches de coopération	30
Figure 11 : Méthodologie de conception en informatique.....	37
Figure 12 : Choix du plus court chemin par une colonie de fourmis	38
Figure 13 : Des fourmis suivant une piste de phéromone	39
Figure 14 : Processus de développement de SC_SMA.....	46
Figure 15 : Diagramme de classe	50
Figure 16 : Diagramme d'activité pour la tâche générique.....	50
Figure 17 : Diagramme d'activité pour les tâches de coopération.....	51
Figure 18 : La configuration de notre environnement.....	53
Figure 19 : Vue Trois Dimensions du l'environnement.	54
Figure 20 : Composant de l'environnement	55
Figure 21 : Exemple de passage UML / NetLogo.....	55
Figure 22 : Prototype conçu pour notre robot -Turtle coopérant.....	56
Figure 23 : Un Canevas pour notre programme principal.....	58
Figure 24 : Tableau de bord global de SC_SMA	61
Figure 25 : Illustration des quatre mécanismes de coopération.	62

Liste des Tableaux

Tableau 1 : Agents cognitifs vs Agents réactifs..... 15

Tableau 2 : Les différents thèmes pour analyser la tâche générique..... 57

Tableau 3 : Comportement pour les robots-turtles..... 57

Tableau 4 : Résultat de simulation de mécanisme 1 (MPD)..... 63

Tableau 5 : Résultats de la simulation selon le mécanisme (MC). 63

Tableau 6 : Résultat de la simulation selon mécanisme (MC+)..... 64

Tableau 7 : Résultat de la simulation selon le mécanisme (MC-)..... 64

Liste des histogrammes

Histogramme 1 : Le facteur temps selon nbr objet+ mécanismes!..... 65

Histogramme 2 : Le facteur énergies selon nbr objet+ mécanismes 66

Histogramme 3 : Facteur distance selon nbr objet+ mécanismes..... 67

Liste des abréviations

OMG : Object Management Group

SMA: Système Multi Agents

MERISE: Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise

UML: Unified Modeling Language

B2B: Business To Business

KQML: Knowledge Query and Manipulation Language

FIPA: Foundation of Intelligent Physical

FIPA-ACL: Foundation of Intelligent Physical Agents - *Agent Communication Language*

AS: Ant System

ACO: Ant Clony Optimization

ACS: Ant Colony System

MMAS: Max-Min Ant System

TCPO: La Tâche Coopérative de Poussée Objet

MPD: Mécanisme Par Défaut

MC: Mécanisme de Centre

MC-: Mécanisme de Coins plus proche

MC+: Mécanisme de Coins plus loin

USA : United States of America

SC_SMA: Ce simulateur est dédié à l'étude des systèmes multi-Robots en termes de planification de chemin vers des cibles fixes hétérogène

Introduction générale

1 Introduction générale

1.1 Contexte & Motivation

La modélisation d'un système ou d'un phénomène, quel qu'il soit, permet de formaliser un problème afin de le rendre fiable ou de l'optimiser pour pouvoir l'appliquer dans un cas réel. Dans la littérature, de nombreux types de modélisation ont été appliqués pour reproduire des modèles qui reflètent des systèmes cibles.

Aujourd'hui la modélisation informatique, tient une place importante dans de nombreux usages scientifiques. Les scientifiques (informaticiens) ont comme mission, de comprendre la progression du système, prévoir les événements, mais particulièrement appréhender les interactions entre les constituants du système, etc. Brièvement, ce type de modélisation dresse la construction d'un prototype d'une réalité, calqué par un programme informatique, renvoyé par une configuration exécutable par un ordinateur. Toutefois, lorsqu'on applique des entrées externes sur le prototype, et quand on analyse les conséquences produites, on parle dans ce cas de la simulation informatique, souvent la simulation résulte de la modélisation.

Depuis l'apparition du contexte « Modélisation informatique » et durant les dernières 40 années. Ce contexte est sans doute notable, n'arrête pas à offrir aux chercheurs, un nombre de modèles assez performant, pour affiner leurs recherches. Couramment ces modèles sont de plus en plus pratiques, chronologiquement on pourrait toutefois citer l'exemple de certains modèles : relationnelle, entité/association, passant par l'orienté objet standardisé par OMG¹, le langage unifié nommé 'UML' au milieu des années 90, Et terminons par l'orienté agent qui est apparu depuis les années 1990.

Durant ces dernières années la modélisation à base d'agent est en réelle ébullition, cette ébullition est due au nombre important de scientifiques qui utilisent ce concept dans les différents domaines de recherche.

Un agent (stationnaire) est défini selon Ferber, comme toute entité physique ou virtuelle, autonome, capable de percevoir et d'agir dans un environnement. Le besoin progressif de disposer des agents qui se déplacent à encourager les chercheurs à élargir ce concept par la notion de mobilité, pour définir à présent un agent mobile comme toute entité, être humain, robot ou logiciel effectuant des actions et se déplacer en même temps. D'ailleurs, cette mobilité

¹ www.omg.org

trouve un grand écho dans différents domaines tels que l'agriculture, l'exploration spatiale, la téléphonie mobiles. Etc.

La plupart du temps, un agent n'est pas seul dans son environnement, il y a d'autres agents présents autour de lui. Les agents doivent, par conséquent, être capables d'interagir entre eux. Cependant quand un environnement contient plus d'un agent, on peut déjà textuellement parler d'un système multi-agents (SMA).

Depuis l'apparition de la technologie agent (particulièrement mobiles), la coopération entre ces agents attire de plus en plus la préoccupation de nombreux chercheurs. Elle est devenue une thématique à part, aiguë par l'idée suivante : "Un groupe d'agents (Exp. robot mobile) peut effectuer des tâches plus efficacement en termes de temps, d'énergie et distance qu'un seul agent peut accomplir dans un contexte robotique coopérative". Cette efficacité inhérente, peut être vue dans plusieurs facettes, soit dans la participation des agents réalisant une tâche commune, soit dans la communication entre l'ensemble agents, ou soient dans la planification des chemins par/pour l'ensemble des agents.

Dès le début, notre vision (réflexion) inspiré pour la coopération est singularisée principalement l'usage de l'environnement (colonie de fourmis) comme un dispositif catalyseur pour notre coopération ambitionné. Pour ces raisons, surtout d'utilité de l'environnement comme un moyen catalyseur de coopération, que nous avons prédestiné la robotique mobile virtuelle comme une branche favorise notre projet, ce choix se focalise essentiellement sur la notion de coopération (Mécanismes) est non pas sur la représentation de robotique proprement dite architecture de composant, les capteurs..., etc. Cette dernière (représentation) est établit par défaut, à travers la plateforme multi-agent intuitifs choisi, pour décrire (concrétise) nos robots mobile.

De ce fait, et pour éviter toute confusion où ambiguïté durant notre rédaction, ces trois notions : agent mobile, robot mobile, robot-turtle désignent la même chose, un agent virtuel avec la faculté de mobilité physique.

1.2 Problématique

La réalisation d'un modèle ou d'un prototype pour un phénomène quelconque, recommande un procédé scientifique, qui repose principalement sur une démarche avec trois aspects incontournable:

– le phénomène réel (ou virtuel) que l'on souhaite étudier.

- le modèle de ce phénomène.
- la simulation de ce modèle par ordinateur (simulateur).

Dans ce mémoire, nous nous basons sur cette démarche, particulièrement celle liée à l'élaboration des modèles (avec nos mécanismes) et à l'implémentation de simulateur permettant de les exécuter.

De ce fait, et dans cette esprit, nous désirons découvrir par l'expression "**passage UML/ Plateforme SMA**", la réponse à "**certaines interrogations**" qui sont continuellement soulevées, mais d'une manière différente, par exemple, à travers un ensemble de méthodologies et plateformes multi agents différentes.

Par l'expression "**passage UML/ Plateforme SMA**", nous percevons la façon avec laquelle nous approcherons le problème de coopération que nous traitons dans ce mémoire à travers une tâche générique prédestiné.

Par l'expression "**Certaines Interrogation**" nous voulons dire que nous n'avons pas l'intention de répondre à toutes les questions de coopération, qui peuvent surgir dans un cadre de coopération entre des robots virtuelles, parce que les interrogations dans ce champ sont vastes, et intéresse beaucoup d'aspects qui posent beaucoup d'interrogations.

Nous nous sommes intéressés dans ce travail à deux problèmes :

- La planification du chemin des agents mobiles virtuels (robots) vers les cibles fixes, et la maîtrise de mobilité de ces robots.
- A notre avis la problématique de coopération entre agents mobiles est essentiellement une problématique de planification de chemin et de mobilité des agents.

Ces deux problèmes peuvent être exprimés en deux questions (**Q1** et **Q2**) : (cahier de charge pour notre modélisation)

Q1: Comment les robots virtuelles peuvent-ils profité de l'environnement comme un moyen de coopération tacite ? Par exemple, lorsqu'un agent mobile dépose un objet dans une cible fixe, comment cet agent devrait-il laissé des indicateurs sur l'environnement pour qu'un autre agent mobile explorateur de cibles, trouve ces cibles fixes le plus rapidement possibles! (planification déplanification)

Q2: Comment les robots peuvent-ils créés un espace virtuelle commun, pour administrer la communication, un espace consacré surtout à la gestion de mobilité des robots? L'interrogation ici stipule qu'un robot ne doit pas circuler dans l'environnement sans aucun

but. Par exemple, lorsque l'ensemble des objets à déplacer devient nul, et principalement pour des raisons énergétiques, comment l'ensemble des robots occupé par un objet informe en temps réel à l'ensemble des robots libres, qu'il n'aura pas besoin de chercher des objets (c'est mieux d'arrêter).

Ainsi, dans ce mémoire, nous adoptons un ensemble d'hypothèses, énoncées dans le dernier chapitre de ce mémoire.

1.3 Objectif

L'objectif de notre travail consiste à proposer/développer des mécanismes et des stratégies de coopération (avec une inspiration de fameuse méthode colonie de fourmis) entre un ensemble de robot mobile, afin d'atteindre un ensemble de cibles fixes avec un langage de modélisation basé sur les agents, en tenant compte des contraintes temporelles et environnementales du phénomène veulent modéliser.

Le but ultime de ce mémoire est de concevoir un environnement multi-agents, afin de montrer la coopération des agents dans un phénomène collective, achevé par le développement d'un **Simulateur de Coopération** dédié nommé " **SC_SMA** " .

Pour implémenter **SC_SMA**, nous avons adopté une démarche produite singulièrement par un passage d'un langage unifié 'UML' vers une plateforme multi-agents, ce passage inspiré par quelque ateliers référencés à la suite dans le quatrième chapitre de ce mémoire, en joignant proprement certaines contraintes temporelles pour le système globale, et des contraintes énergétique pour les robots-virtuels.

De même, nous avons adopté la plateforme de développement **NetLogo**², développé par Uri Wilenski en 1999, et dont le développement continue par le Centre « for Connected Learning and Computer-Based Modeling » de l'université de Northwestern, USA.

La **Figure 1** illustre notre démarche adoptée :

² <http://ccl.northwestern.edu/netlogo>.

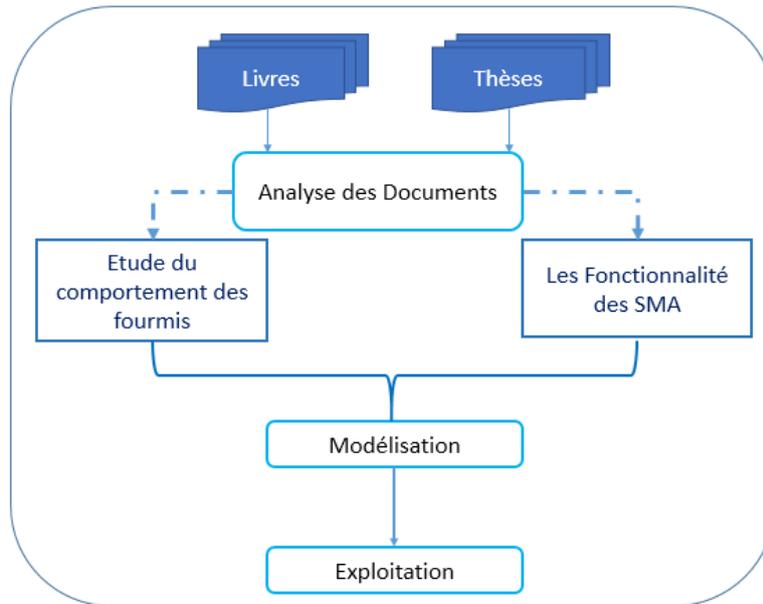


Figure 1 : Notre Démarche.

1.4 Structure de mémoire

Ce mémoire est divisé en quatre chapitres, qui sont organisés comme suit :

Le premier chapitre expose des notions générales sur les paradigmes de modélisation, ainsi que la simulation d'un système informatique. Cet aperçu nous permet de d'une part de montrer l'utilité du paradigme agent dans le contexte informatique actuel, d'autre part, de mieux cerner les besoins relatifs à notre problématique abordée par ce mémoire.

Le deuxième chapitre traite sur le concept de coopération, on ne s'intéresse pas à la coopération au niveau de la résolution de problème distribué, mais au bénéfice de la coopération au niveau de la collectivité. En premier lieu, il présente la notion de coopération et ses différentes catégories et se focalise en dernier lieu sur les différents travaux étudiés qui présente l'apport de la coopération dans la résolution de problème.

Le troisième chapitre est consacré aux différents travaux inspirés par les comportements collectifs des fourmis. Après une introduction générale sur les algorithmes de colonies de fourmis, le chapitre se focalise sur les fourmis artificielles et sur les algorithmes orientés optimisation multi-objectif par colonies de fourmis.

Le quatrième chapitre a pour objectif de visualiser et de présenter l'environnement de développement (Netlogo). Ainsi que la conception de notre simulateur **SC_SMA**, et les mécanismes développés, ensuite, l'implémentation qui présente l'interface graphique de notre

application, Subséquemment, une comparaison entre les résultats expérimentaux et l'efficacité de chaque mécanisme

A la fin, une conclusion générale sur l'ensemble de nos travaux ainsi qu'aux améliorations qui pourrait être apportées et aux perspectives qu'elle offre.

La **Figure** suivante illustre, la structure organisationnelle de notre mémoire :

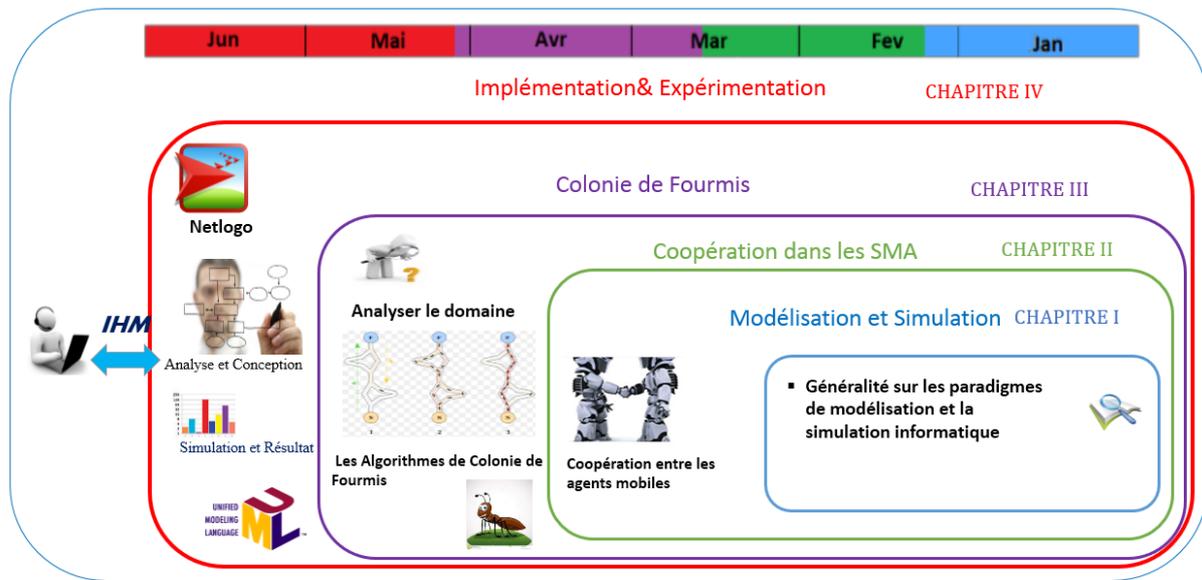


Figure 2 : Organisation de notre mémoire

Chapitre 1 :

Modélisation &

Simulation

Chapitre 1 : Modélisation & Simulation

Introduction

De nos jours, les applications informatiques sont de plus en plus complexes et difficiles à réaliser. Pour cette raison leur production doit être de plus en plus assistée et rigoureuse. Le génie informatique est une discipline qui traite de la production et de la maintenance des logiciels. Elle a connu une grande évolution depuis l'apparition de l'informatique. Au niveau de cette discipline, des réflexions sont menées sur :

- la philosophie à adopter pour représenter les objets du monde réel sous une forme virtuelle. Cette philosophie est introduite par les paradigmes de développement.

1 La modélisation

1.1 Un modèle

Une abstraction de la réalité permettant de mieux comprendre le système, le modèle doit être reliée au monde réel.

1.2 La modélisation

Abstraire la réalité pour mieux comprendre le système réalisé. Le processus de modélisation vise à obtenir une solution acceptable du système informatique. La solution finalement retenue n'est pas obtenue en une seule itération. Plusieurs étapes sont nécessaires, ces étapes successives permettent de raffiner le niveau de détails du système à réaliser. Les premières étapes donnent une vision à très gros grains et permettent d'avancer dans la compréhension du problème.

1.2.1 Modélisation des processus d'affaires

Pour comprendre le processus pour l'analyser, le simplifier ou l'optimiser.

1.2.2 Modélisation du système d'information

Créer une représentation abstraite des artefacts. Logiciels que nous devons construire : Classes, tables de bases de données, Processus, méthodes, routines, scripts,...Etc.

2 Les paradigmes de modélisation

2.1 Un paradigme

Est un mode de pensée qui permet de structurer nos connaissances, notre apprentissage et notre compréhension. Dans le contexte informatique, un paradigme de développement est une manière de représenter le monde dans le but de concevoir un programme informatique. Depuis l'apparition de l'informatique, plusieurs paradigmes ont été adoptés (paradigme procédural, paradigme orienté objet, etc.).

3 MERISE

(*Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise*) est une méthode d'analyse et de réalisation des systèmes d'information qui est élaborée en plusieurs étapes : schéma directeur, étude préalable, étude détaillée et la réalisation. La méthode Merise préconise plusieurs niveaux d'abstraction (conceptuel, organisationnel, opérationnel, physique).

4 Orienté Objet

La modélisation objet consiste à créer une représentation informatique des éléments du monde réel (que l'on appelle domaine) auxquels on s'intéresse, sans se préoccuper de l'implémentation, ce qui signifie indépendamment d'un langage de programmation. Il s'agit donc de représenter ces objets et leurs relations, l'interaction entre les objets via leurs relations permet de concevoir et réaliser les fonctionnalités attendues, de mieux résoudre le problème.

4.1 Un langage unifié pour la modélisation objet (UML)

UML (Unified Modeling Language) est un langage unifié pour la modélisation objet et propose donc une notation et une sémantique associée à cette notation (des modèles), mais pas de processus (de démarche proposant un enchaînement d'étapes et d'activités qui mènent à la résolution d'un problème posé).

5 Paradigme composant

Les besoins croissants des utilisateurs de systèmes logiciels, l'évolution rapide du matériel, l'explosion des réseaux informatique ont motivé l'émergence de nouvelles approches de développement d'application à grande échelle comme celles basées sur le paradigme de composants qui se rapproche du paradigme objet. Ce paradigme utilise une approche

modulaire de l'architecture d'un projet informatique, ce qui permet d'assurer au logiciel une meilleure lisibilité et une meilleure maintenance.

6 Paradigme service

Les services sont apparus comme une suite logique des composants logiciels et des approches d'intégration d'applications à base de composants. Les applications construites à base de composant sont monolithiques. L'architecture à base de composants est incompatible avec l'ouverture de l'infrastructure Internet.

Le concept de service et les architectures à base de services sont apparus en réaction aux limites des architectures à base de composants de façon à exploiter les possibilités offertes par Internet. Parallèlement, l'avènement du B2B (Business To Business) a renforcé le besoin d'interopérabilité d'applications hétérogènes, développées sur des plates-formes différentes.

[1]

7 Paradigme Agent

Depuis une dizaine d'années, de nouvelles architectures de systèmes d'information ont émergé. Elles sont caractérisées par l'absence d'autorité centrale, des composants faiblement couplés et la capacité à répondre (réactivité) et à s'adapter (flexibilité) rapidement aux nouveaux besoins et aux nouvelles contraintes selon des principes d'auto-organisation avec :

- Le manque de **méthodologies** et d'outils pour aborder la modélisation d'un système complexe à différents niveaux de granularité.
- Le manque de **standards** pour décrire les interactions (échanges de données) entre composants.
- Le manque de **flexibilité** : capacité d'adaptation et de configuration dynamique du système. La flexibilité doit permettre de simuler des scénarios complexes fondés sur les points de vue multiples de chaque acteur (simulateur, opérateur humain) où les règles, les contrats et la structure du réseau peuvent changer en cours de simulation.

7.1 La vue agent

Il n'existe pas, actuellement, une définition de l'agent qui fasse l'unanimité. Pour avoir une bonne vision du concept agent, nous confrontons les définitions les plus utilisées dans la littérature. La plupart des travaux francophones font référence à la définition fournie par Ferber [3] qui stipule qu'un agent est une entité physique ou virtuelle :

- qui est capable d'agir dans un environnement,
- qui peut communiquer directement avec d'autres agents,
- qui est mu par un ensemble de tendance (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
- qui possède des ressources propres,
- qui est capable de percevoir (mais de manière limitée) son environnement,
- qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- qui possède des compétences et offre des services,
- qui peut éventuellement se "reproduire",
- qui a un comportement qui tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit. [1]

7.2 L'Agent face à l'Objet

7.2.1 Des similarités

- Dispose d'un «état interne»,
- Comprend des structures de comportement modulaires (méthodes/compétences),
- Communiquent par envoi de messages,
- Agissent pour modifier leur état

7.2.2 Des différences

L'objet

Pas d'autonomie : l'objet est invoqué par un appel de méthode qu'il ne peut refuser (pas de réactivité)

- Faible niveau de sociabilité : interaction simple et rigide (pas d'évolution dans le temps)

L'agent

- Autonomie de contrôle : l'agent décide de son comportement en fonction de son état, croyances, connaissances, perceptions de l'environnement, requêtes des autres
- Sociabilité : composante très importante, complexité des interactions, des organisations
- Notion d'environnement : importante et complexe. [2]

7.3 Propriétés de l'agent

7.3.1 Autonome

Son comportement est fonction de ses perceptions qui agisse sur son état, et de sa représentation de l'environnement dans lequel il évolue. Aucun super contrôleur ne peut le piloter de l'extérieur.

7.3.2 Réactivité

Il maintient un lien constant avec son environnement et répond aux changements qui y surviennent.

7.3.3 Proactif

Il peut prendre des initiatives afin de satisfaire ses buts. Pour se faire, il n'est pas soumis à l'invocation d'une autre entité pour agir mais peut agir sur sa propre initiative.

7.3.4 Flexible

Il adapte son comportement à sa perception de son environnement et peut participer à des organisations (groupe) afin de mieux satisfaire son but.

7.3.5 Social

Il a la capacité d'interagir pour atteindre ses buts ou pour aider d'autres agents dans leurs activités.

7.3.6 Situé

Capacité à percevoir l'environnement au travers de métriques spatio-temporels dans lequel il peut agir de façon limitée.

Ces propriétés sont dites minimales car elles ne présupposent rien sur l'intelligence et les capacités cognitives des agents. Ces capacités cognitives permettent de distinguer des agents cognitifs et des agents réactifs. [1]

7.4 Classification des agents

La classification est un mécanisme important pour comprendre les différents types d'agents, elle dépend des critères sur lesquels elle est basée. Le choix de critères de classification d'une taxonomie est arbitraire et repose sur un besoin de nommer précisément les architectures existantes en fonction de leurs aspects opérationnel ou architectural. Selon leurs modes de fonctionnement et leurs représentations de leurs environnements, les agents peuvent être

classés en trois catégories essentielles qui sont: les agents réactifs, les agents cognitifs et les agents hybrides.

7.4.1 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.

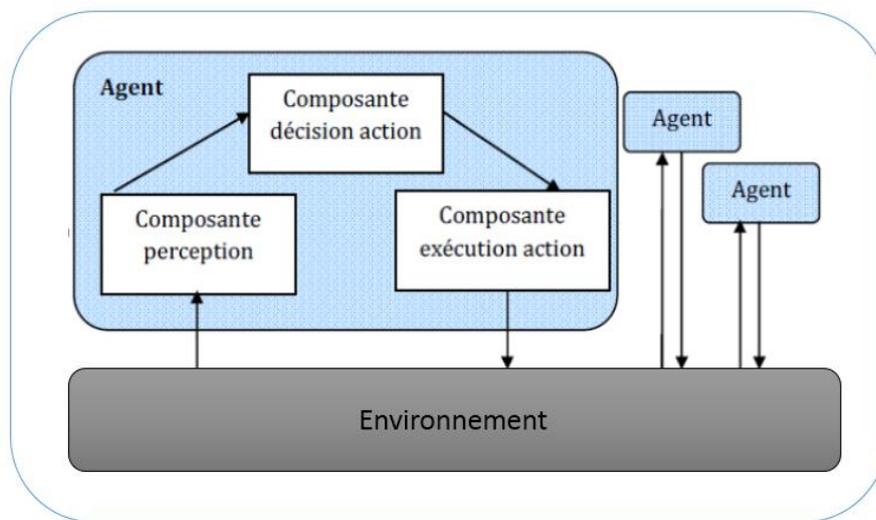


Figure 3: Structure générale d'un agent réactif dans un système Multi-Agents

Exemple : Le mécanisme de la sociogenèse de colonies de fourmis où l'ensemble des membres est capable de produire des actions évoluées (comportement collectif) mais dont les individus pris séparément ne possèdent qu'une représentation faible de leur environnement et des buts globaux.

7.4.2 Les agents Cognitifs

Ils sont parfois dits "intentionnels", leur caractéristique fondamentale est la volonté de communiquer et de coopérer, ils possèdent des buts à atteindre à l'aide d'un plan explicite. Les sociétés d'agents cognitifs contiennent communément un petit groupe d'individus de forte granularité, régit par des règles sociales prédéfinies (c'est-à-dire lors de situations conflictuelles les agents seront amenés à négocier). Ces agents sont capables à eux seuls d'exécuter des opérations complexes, ils peuvent raisonner en s'appuyant sur des bases de connaissances.

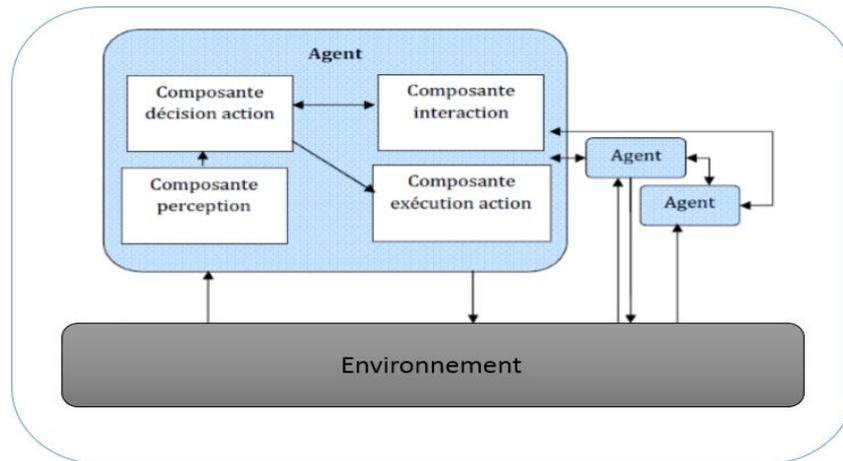


Figure 4 : Structure générale d'un agent cognitif dans un système Multi-Agents

Exemple : Pour dépasser les limitations des interfaces homme-machine à manipulation directe (rigidité, complexité, etc.), les agents assistants apportent une adaptation au profil de l'utilisateur et une capacité à anticiper ses besoins.

Selon [8] le Tableau suivant donne quelques ordres de comparaison entre les agents cognitifs et réactifs.

Systèmes d'agents cognitifs	Systèmes d'agents réactifs
Représentation explicite de l'environnement	Pas de représentation explicite
Peuvent tenir compte de leurs passé	Ne tiennent pas compte du passé
Agents exécutant des tâches complexes	Fonctionnement stimulus/réponse
Peu d'agents (petit nombre)	Grand nombre d'agents

Tableau 1 : Agents cognitifs vs Agents réactifs

7.4.3 Un agent Hybride

Combinaison des deux, il est conçu pour allier des capacités réactives à des capacités cognitives, ce qui lui permet d'adapter leur comportement en temps réel à l'évolution de leur univers. Un agent hybride est composé de plusieurs couches arrangées selon une hiérarchie.

[7]

8 System multi agent

Usuellement, un système est un ensemble organisé d'éléments concourant à la réalisation d'une tâche donnée. En suivant cette définition on peut définir immédiatement le système Multi Agents comme étant un ensemble organisé d'agents se chargeant de réaliser un but

commun. Les systèmes Multi Agents 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. [9]

Jacques Ferber [3] définit les systèmes Multi-Agents comme des systèmes composés des éléments suivants:

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

Ces systèmes possèdent les avantages traditionnels de la résolution distribuée comme la modularité, la vitesse, et la fiabilité (due à la redondance), Ils héritent aussi des bénéfices envisageables de l'intelligence artificielle (IA) comme:

- Le traitement symbolique (aux niveaux des connaissances).
- La facilité de maintenance.
- La réutilisation et la portabilité.

Mais surtout, ils ont l'avantage de faire intervenir des schémas d'interaction sophistiqués qui 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).
- La négociation (parvenir à un accord acceptable pour toutes les parties concernées).

8.1 Les caractéristiques des systèmes multi-agents

Un système Multi Agents possède généralement les caractéristiques suivantes:

- Il n'y a pas de contrôle global du système.
- Les données sont décentralisées.
- Le calcul est asynchrone.
- Chaque agent a des informations ou des capacités de résolution limitées de problème, ainsi chaque agent a un point de vue partiel. [7]

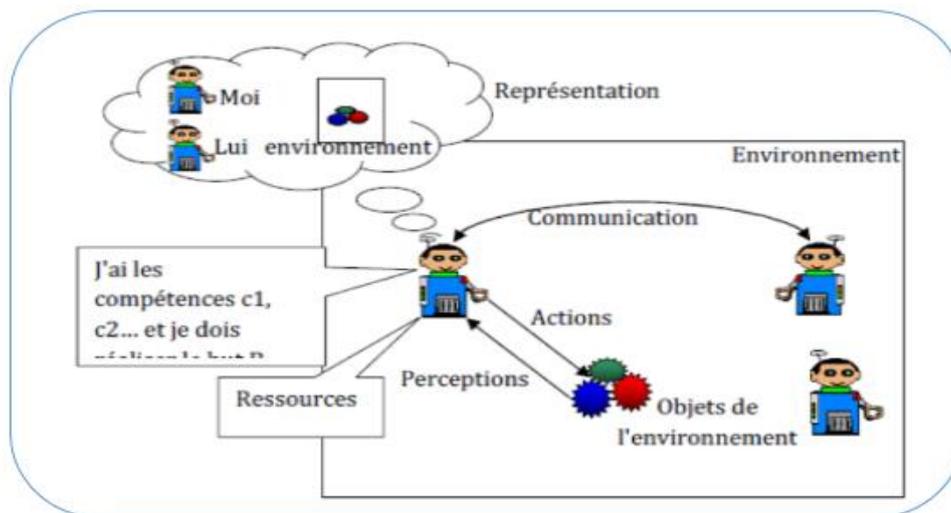


Figure 5 : Une image globale d'un Système Multi-Agents

8.2 Interaction entre agents

Un système multi agents (SMA) se distingue d'une collection d'agents indépendants par le fait que les agents interagissent en vue de réaliser conjointement une tâche ou d'atteindre conjointement un but particulier. Les agents peuvent interagir en communiquant directement entre eux ou par l'intermédiaire d'un autre agent ou même en agissant sur leur environnement.

Les interactions entre agents sont définies selon Ferber comme étant la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques.

L'interaction entre agents s'effectue par la communication, les actes de langages et les protocoles d'interaction. Les agents interagissent entre eux. Pour atteindre son objectif ou pour améliorer la coordination des actions, un agent peut demander des services à un autre agent.

Plusieurs schémas d'interactions sont nécessaires les principaux sont :

- la collaboration : manière de répartir le travail entre plusieurs agents
- la coordination d'actions : manière dont les actions des différents agents doivent être organisés dans le temps et l'espace.
- Les coopérations : forme générale d'interaction pour les agents capables d'avoir un projet explicite, Qui fait quoi, quand, ou, et avec quels moyens.

Les techniques de négociation sont utilisées pour limiter les effets des conflits qui apparaissent. Coopération = collaboration + coordination + résolution de conflits. [5]

8.3 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...

Elle est définie comme une forme d'action locale d'un agent vers d'autres agents. Les questions abordées par un modèle de communication peuvent être résumées par l'interrogation suivante: qui communique, quoi, à qui, quand, pourquoi, et comment?

Pourquoi les agents communiquent-ils ? La communication doit permettre la mise en œuvre de l'interaction et par conséquent la coopération et la coordination d'actions.

Quand est ce que les agents communiquent-ils ? Les agents sont souvent confrontés à des situations où ils ont besoin d'interagir avec d'autres agents pour atteindre leurs buts locaux ou globaux. La difficulté réside dans l'identification de ces situations.

Avec qui les agents communiquent-ils ? Les communications peuvent être sélectives sur un nombre restreint d'agents ou diffusées à l'ensemble d'agents.

Comment les agents communiquent-ils ? La mise en œuvre de la communication nécessite un langage de communication compréhensible et commun à tous les agents. Il faut identifier les différents types de communication et définir les moyens permettant non seulement l'envoi et la réception de données mais aussi le transfert de connaissances avec une sémantique

appropriée à chaque type de message. Il existe principalement deux modes de communication:

1. Communication par envoi de messages.
2. 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.

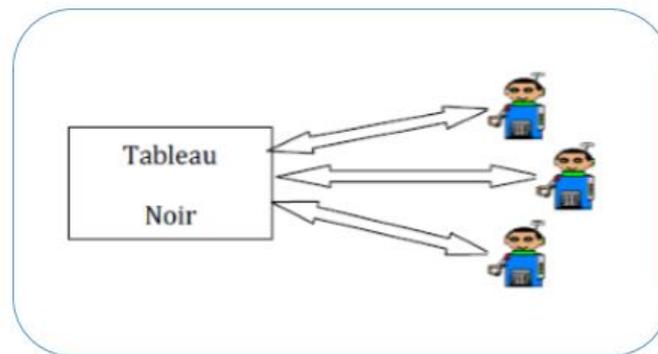


Figure 6 : Communication via un Tableau Noir

En revanche, dans les systèmes où la communication se fait par envoi de messages, les connaissances sont distribuées entre les différents agents. Chacun d'eux communique directement avec les autres par envoi de messages. [7]

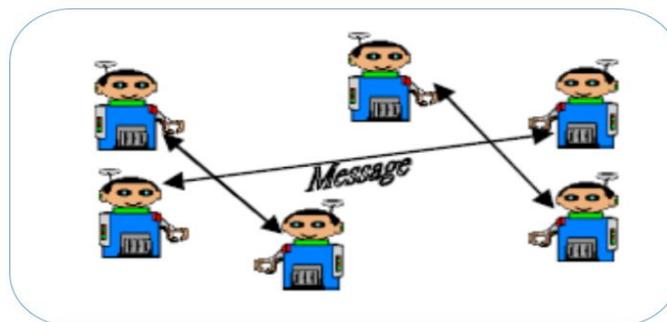


Figure 7 : Communication par envoi de messages

8.4 Domaines d'application des S.M.A

- résolution de problèmes, de décision : par planification distribuée des tâches et d'optimisation.
- Gestion de grands systèmes industriels.
- Contrôle du trafic aérien.

- Traitement d'image et réalité augmentée.

8.4.1 Automatique et Robotique

- Planification multi-robot.
- Robots autonomes mobiles

9 La simulation informatique

désigne l'exécution d'un programme informatique sur un ordinateur ou réseau en vue de simuler un phénomène physique réel et complexe (par exemple : chute d'un corps sur un support mou)

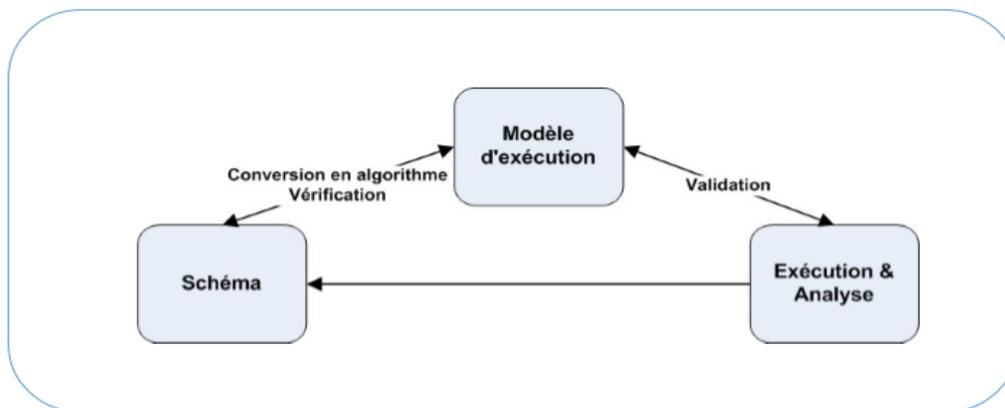


Figure 8 : La simulation informatique selon Fishwiki [6]

On peut distinguer deux types de simulations :

9.1 La simulation continue

Où le système se présente sous la forme d'équations différentielles à résoudre. Elle permet de suppléer à la résolution analytique quand celle-ci est impossible. Effectuée au départ sur des calculateurs analogiques, elle s'est effectuée aussi sur des ordinateurs ainsi que des machines hybrides, et un troisième type de calculateurs qui n'a pas eu de lendemain, les calculateurs stochastiques.

9.2 La simulation discrète

Dans laquelle le système est soumis à une succession d'évènements qui le modifient. Ce type de simulation a pour vocation d'appliquer des principes simples à des systèmes de grande taille.

9.3 Les objectifs de la simulation informatique

L'informatique permet d'effectuer la simulation de phénomènes réels sur ordinateur. Le but principal de la simulation informatique est :

- Étudier un système réel de manière à comprendre son fonctionnement interne et/ou à en prévoir son évolution sous certaines conditions.
- Concevoir un modèle informatique qui est une 'représentation' fidèle du système réel et qui est utilisé pour réaliser les expérimentations. [6]

La **Figure 9** représente une vue d'ensemble de la simulation informatique, on peut distinguer les concepts suivants :

- **Le système source** : le phénomène que l'on souhaite étudier.
- **Le cadre expérimental** : spécifications des conditions d'observation du système et des objectifs de la simulation.
- **Le modèle** : l'ensemble des instructions qui permettent de générer
 - A l'aide d'un programme informatique.
 - Le comportement du système au cours du temps.
- **Le simulateur** : le programme informatique capable d'exécuter le modèle et le produire son comportement.

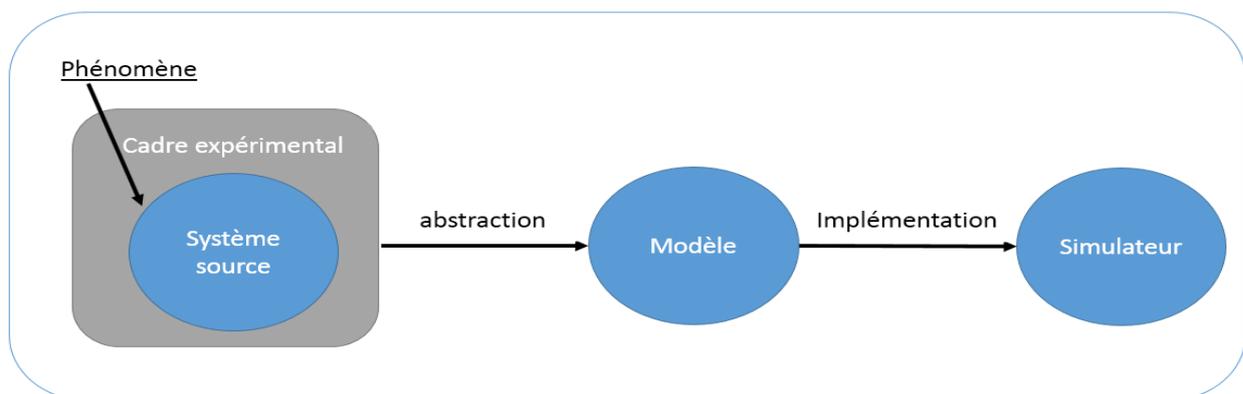


Figure 9 : Vue d'ensemble de la simulation informatique [6]

Conclusion

Dans ce chapitre, nous avons présenté des notions générales sur la modélisation et plusieurs paradigmes de modélisation, ainsi que la simulation d'un système informatique. Cet aperçu nous permet de d'une part de montrer l'utilité du paradigme agent dans le contexte informatique actuel, d'autre part, de mieux cerner les besoins relatifs à notre problématique abordée par ce mémoire.

Ce qu'on peut bien enregistrer d'après le repère des paradigmes inspiré, et d'après les axes de recherches cités du paradigme agent, c'est que ce dernier paradigme, avec ces caractéristiques claires et ses axes de recherches visibles pour beaucoup de disciplines et domaines actuels, peut être vu comme un support (brique) idéal pour la modélisation attendue de ce mémoire.

Chapitre 2 :

Coopération dans les

SMA

Chapitre 2 : Coopération dans les SMA

Introduction

La coopération nous paraît être une caractéristique essentielle dans l'apprentissage en univers multi-agent. En effet, chaque agent ne possède qu'une vue partielle du système auquel il appartient et il a la possibilité, grâce à la coopération, de réaliser plus de tâches que s'il avait travaillé seul de manière individualiste.

Les travaux les plus anciens concernant la coopération utilisent la coopération comme processus de résolution de problèmes. Le concept de coopération prend une signification plus large des lors qu'on l'utilise pour avancer qu'un système se trouve dans une situation coopérative. Il signifie toujours coopération pour la résolution du problème auquel est confronté le système, mais aussi coopération entre les entités du système, en l'occurrence les agents, pour que le système ait un fonctionnement optimal. Nous nous plaçons dans ce second cas pour étudier la coopération. Dans ce chapitre nous présentons les différentes définitions de la coopération, de la coordination, de la collaboration, nous montrons aussi l'apport de la coopération dans divers exemplesetc.

1 Cadre d'étude

L'intérêt des systèmes multi-agents ne porte pas sur la manière dont un agent résout un problème donné mais sur la manière dont un groupe d'agents arrive à résoudre ce problème. En effet, une société ne doit pas seulement être vue comme agrégation d'agents interconnectés (directement ou indirectement par le biais de l'environnement) ayant des compétences particulières (homogènes ou hétérogènes). Une société repose au contraire sur une réelle coopération des agents les uns avec les autres et non sur une simple cohabitation. L'aspect collectif lors de la résolution d'un problème et plus exactement la coopération des agents sont donc les points importants dans ce mémoire.

La coopération permet en particulier à des agents de résoudre des tâches qu'ils n'auraient pas pu résoudre seuls, d'améliorer la productivité de chacun, d'optimiser l'utilisation des ressources, d'augmenter le nombre de tâches réalisées lors d'un délai imparti... [10]. Les exemples que nous allons présenter vont dans ce sens : nous montrerons comment la coopération a été mise en œuvre dans les systèmes proposés et quelles sont les caractéristiques qui ont été accrues grâce à la coopération.

2 La coopération

Il existe plusieurs points de vue sur la coopération selon que l'on considère que la coopération est une attitude des agents qui décident de travailler en commun. Nous allons citer plusieurs définitions de la coopération et quelques définitions de la coordination, collaboration, ces notions étant parfois difficiles à différencier [11], [12], [13], [14].

La définition proposée par Petit Robert est la suivante : "Action de participer à une œuvre commune (collaboration, aide, contribution)".

Damazeau Yves, Muller Jean-Pierre [11], parlent de coopération pour une tâche locale, lorsqu'un agent a besoin de coopérer avec autrui, parce qu'il n'est pas capable d'accomplir par lui-même ou parce que les autres peuvent accomplir de manière plus efficace que lui dans un intervalle de temps plus court.

Pour Jacques Ferber [12], selon le niveau d'observation ou l'on se place, la coopération revêt deux dimensions :

- la coopération peut être une attitude des agents qui décident de travailler en commun. Dans ce cas, les agents coopèrent s'ils s'engagent dans une action commune après avoir identifié et adopté un but commun (prise de conscience de l'existence d'un but commun et engagement de chacun).
- Dans le second cas, un observateur interprète a posteriori les comportements en les qualifiant de coopératifs ou non, à partir de critères sociaux comme l'interdépendance des actions ou le nombre de communications effectuées.

Il considère que plusieurs agents coopèrent, pour atteindre un objectif commun, 'si l'une des deux conditions est vérifiée :

- 1- l'ajout d'un nouvel agent accroît différenciellement les performances du groupe.
- 2- il existe des conflits potentiels ou actuels d'accès à des ressources et l'action des agents sert à éviter ou à sortir de tels conflits. '

Brassac Christian, Pesty Sylvie [13] insistent sur la volonté individuelle et collective de coopérer chez les agents (cognitifs). En revanche, la coopération n'apparaît, selon eux, au niveau des agents réactifs que parce qu'elle est attribuée par un observateur. Le terme coopération n'est selon eux pas adapté aux agents réactifs, ils lui préfèrent la notion de co-action.

Lenay Christian [14] définit la coopération comme "un type de dynamique collective qui aboutit à un état émergent bénéfique pour la population ". Il distingue les termes coopération et co-action de Ch. Brassac et S. Pesty, par coopération intentionnelle et non intentionnelle ".

3 La coordination

La coordination est définie par le Petit Robert de la manière suivante : “agencement des parties d'un tout selon un plan logique, pour une fin déterminée (organisation, coordination des opérations d'une troupe).

“Pour garantir la cohérence globale d'un SMA et sa convergence vers une solution au problème posé, il est nécessaire que les agents coordonnent leurs différentes actions. Cette coordination sous-entend que les agents doivent négocier, décider de leur rôle respectif, échanger des informations pertinentes mais aussi construire et gérer une représentation de leur environnement et des autres agents [15]”.

Selon Durfee et Montgomery [16], “le but de la coordination est de trouver, parmi un ensemble de comportements d'agents qui interagissent, une collection de comportements qui réalise d'une façon satisfaisante les objectifs les plus importants des agents. Le degré de coordination reflète dans quelle mesure les agents peuvent éviter des activités accessoires afin d'arriver à leurs fins essentielles.”

4 La collaboration

Le Petit Robert propose la définition suivante pour la collaboration : “travail en commun, action de collaborer avec quelqu'un (coopération, aide, association).”

Selon Jacques Ferber [10], la collaboration indique “l'ensemble des techniques permettant à des agents de (se) répartir des tâches, des informations et des ressources de manière à réaliser une œuvre commune”.

Yves Demazeau et Jean-Pierre Müller [11] parlent de collaboration pour des tâches globales, et plusieurs agents prenant part à la collaboration sont capables d'effectuer individuellement chacune de ces tâches.

La collaboration fait unanimement référence à la résolution d'une tâche globale par plusieurs agents. Cette aptitude ne fait pas partie de nos préoccupations puisqu'elle présuppose de connaître la fonction globale pour laquelle les agents doivent collaborer.

5 Etat De L'art sur la coopération

Dans ces dernières années, des recherches considérables ont été dirigés vers l'étude de la coopération entre un ensemble d'entités, particulièrement mobile [17]. L'intérêt ici, est justifié par de nombreux avantages, apports et dans les différents domaines.

Le terme coopération est la pierre angulaire de ce mémoire. Derrière ce mot se profilent plusieurs domaines : celui de la biologie qui fait l'étude du comportement naturel des fourmis, poissons, oiseaux...etc. Celui de la robotique qui utilise leur comportement pour concevoir des nouvelles machines, et celui de l'informatique où ces créatures sont modélisées pour la simulation ou la création d'algorithmes.

La littérature dans ce champ est vaste (résulte l'interdisciplinarité), elle intéresse tant d'aspects. Une présentation générale des différentes méthodologies, approches de coopération serait trop longue, à la place et après une recherche bibliographique étendue, nous proposons la représentation de la **Figure 10**, une représentation qui exhibe une classification des modèles de coopération entre un ensemble d'agents, avec les différents directions possible. Cette classification est partiellement inspirée de la thèse [17]. Dans cette thèse, le doctorant Adouane présente une fragmentation de trois disciplines distinctes pour étudier la coopération.

Premièrement, les modèles recueillis par un domaine naturel (Biologie). Second, les modèles recueillis par un domaine artificiel (Robotique), et finalement les modèles recueillis dans un contexte virtuel informatique. Dans ces différentes directions l'entité autonome signifiant autant, un être vivant, un robot mobile, qu'un agent logiciel.

Contrairement à Adouane [17], qui accentue son état de l'art principalement sur les modèles de coopération dans un monde artificielle robotique. Ce mémoire, s'intéresse particulièrement au modèles virtuelle informatique interprétée indubitablement par le paradigme SMA, toute en développent avec une classification interne SMA, inspirée de la thèse [18], dans cette thèse le doctorant François Klein a divisé les approches qui tentent de maîtriser le comportement d'un SMA en deux volets, à savoir : par construction et par contrôle.

Dans ces deux thèses [18] [19], Les approches employées pour aborder le problème de coopération entre les agents, peuvent être divisées en trois classes générales : celles basées sur la communication, celles basées sur des conventions et celles basées sur l'apprentissage.

Les agents peuvent communiquer ensemble pour déterminer l'allocation des tâches. Ils peuvent aussi utiliser des conventions imposées au système par le concepteur pour assurer une action conjointe optimale. Finalement, ils peuvent apprendre des politiques de coopération (par apprentissage) lors de leurs interactions répétées avec les autres agents.

Il est fortement noter ici, comme dans [20] que les travaux de la classe virtuelle informatique incarné par les SMA, exploite et profite considérablement des études des deux autres domaines de coopération exhibé dans la thèse [17], notamment la classe naturelle (biologique).

Comme nous l'avons souligné précédemment, une exposition des différentes approches gravitent la coopération entre un ensemble d'agent notamment mobile dans le domaine des SMA, serait trop longue a la place et avec une recherche bibliographique ciblé, nous focalisons notre état de l'art que sur les approches par construction illustrées dans **la (Figure 10)**.

Cependant, et avant d'expliquer-éclairé notre représentation de coopération inspiré. Il est utile d'aborder brièvement en premier lieu, quelques outils disponible, annexer la coopération, en terme de protocole d'interaction, et de langage de communication.

5.1 Des outils pour la coopération

Pour assurer la coopération, généralement les concepteurs multi agent, s'appuient sur un ensemble de protocoles d'interaction, de langage de communication. L'intérêt des langages de communication est de faciliter l'échange et l'interprétation des messages entre les agents. Ces langages se focalisent essentiellement sur la manière de décrire exhaustivement des actes de communication d'un point de vue syntaxique et sémantique.

Le premier langage qui a été introduit est **KQML** (*Knowledge Query and Manipulation Language*) [21] a l'origine, ce langage **KQML** a été développé pour échanger des informations et des connaissances entre des systèmes à base de connaissances. Il a été ensuite repris dans [22] pour décrire les messages échangés entre les agents.

Le deuxième langage dit **FIPA-ACL** [web1] (*Foundation of Intelligent Physical Agents - Agent Communication Language*), est proposé dans le cadre d'un travail de standardisation mené au sein l'organisation **FIPA**.

Par exemple: ["inform" "sender:5" "receiver:3" "content:" "fire-at [23 15]"] Montre un message envoyé par l'agent 5 vers l'agent 3, avec le contenu : "fire-at [23.15] ", avec le

primitive 'inform' comme un message performative 'FIPA'. Dans [22] François. L, témoigne que les langages de communication sont pleinement utilisés pour la spécification des protocoles d'interaction. Alors que les protocoles d'interaction sont utilisés par les agents pour administrer leurs interactions (coopération).

A ce stade, on nomme le protocole **Contract-Net** [23], qui est l'un des protocoles d'interaction (de coopération) les plus utilisés dans les SMA, et le plus complet à ce jour.

Il consiste en un contrat élaboré entre deux participants : Le *contractant* et le *manager*. Le contractant est garant de l'exécution de la tâche et de la transmission de ses résultats au manager. Le manager, quant à lui se porte garant de la gestion de la tâche et du traitement des résultats. Ce protocole, dans sa version originale, ne présente aucun moyen d'évaluation de coût. Ce qui contraint l'environnement dans lequel il fonctionne à des hypothèses de performances, de bénévolat et de modularités vérifiées.

Les **Blackboard** (définie dans le premier chapitre).

Dans la communauté informatique, il existe d'autres outils pour organiser la communication (coopération) entre des agents, que la durée de ce mémoire, ne nous permet pas de l'expliquer distinctement, à savoir : la conversation [24], les Actes de langage [web2] La théorie de jeux [22]....etc.

5.2 Des approches multi agent pour la coopération:

A ce stade deux manières coexistent pour maîtriser la coopération entre les agents dans un monde SMA, présenté comme suite :

5.2.1. Approche par contrôle

Dans [19] Lorsqu'il existe des simulations du SMA ou le comportement global souhaité n'est pas assuré, il faut pouvoir réagir pendant l'exécution du système, en fonction de son évolution courante. C'est le but des approches par contrôle, dénommé à base d'apprentissage dans [18].

L'idée de base derrière l'apprentissage, c'est que les perceptions de l'agent ne devraient pas être utilisées uniquement pour choisir des actions, mais également pour *améliorer l'habilité* de l'agent à agir dans le futur. Une telle amélioration est une forme d'apprentissage est très importante car c'est ce qui lui permet d'évoluer, de s'adapter et de s'améliorer.

L'apprentissage est un axe de recherche de plus en plus important dans le domaine de coopération entre les agents.

Dans la thèse [25] le doctorant Valérie. Camps a décrit ce concept comme une faculté indispensable, tant pour l'amélioration du fonctionnement du système que pour faciliter le développement d'application. Selon Camps deux types d'apprentissage peuvent-ils envisager dans un cadre multi agent :

- L'apprentissage des compétences: peut s'effectuer de manière isolé (un agent crée de nouveaux corpus de connaissance), où collective (un échange de connaissance entre les agents.)
- L'apprentissage de l'organisation: il s'agit d'un apprentissage collectif qui consiste à apprendre les interactions (coopération) entre agents afin d'améliorer celles-ci.

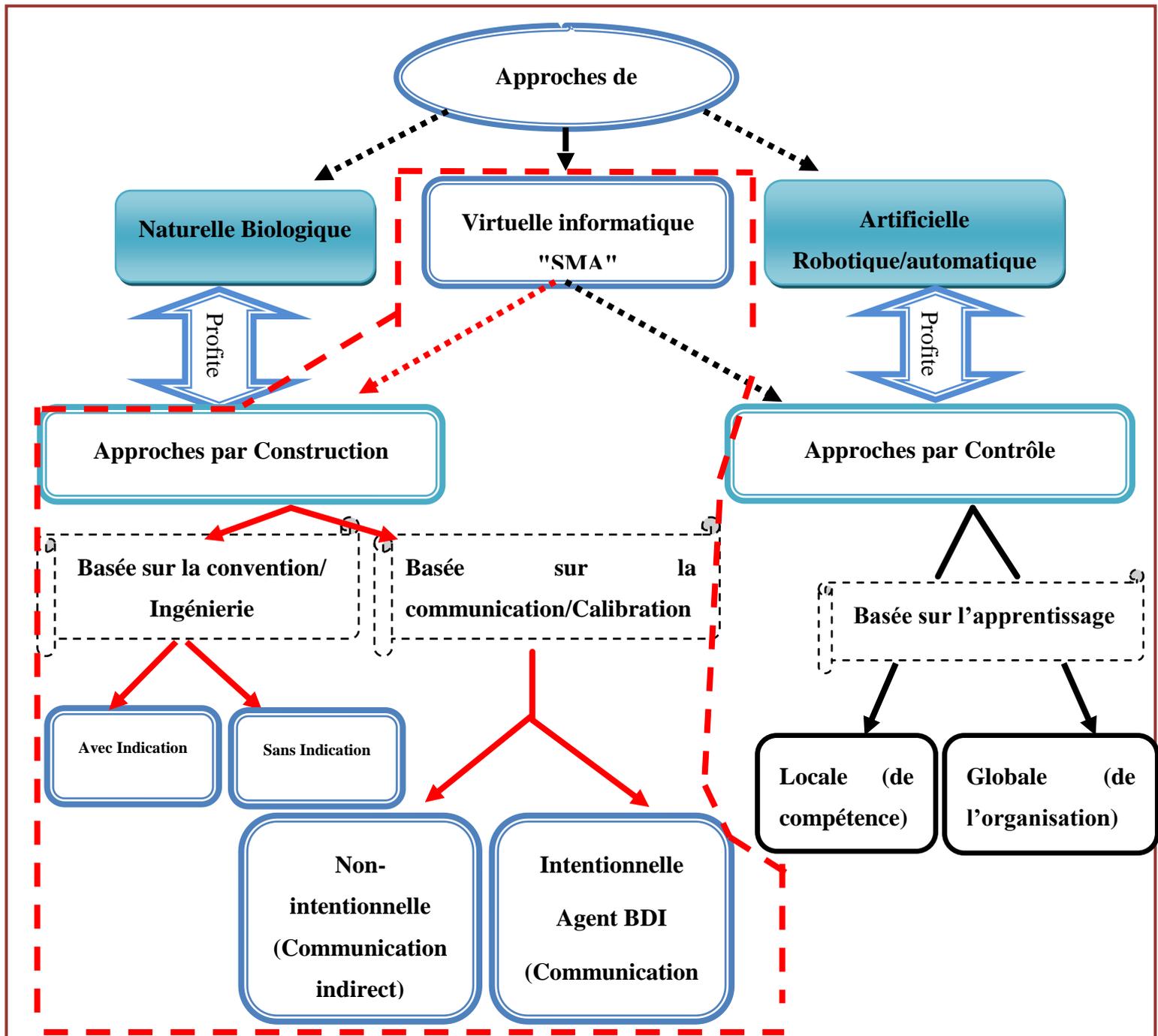


Figure 10 : Une classification pour les approches de coopération

L'apprentissage est très important pour la coopération des agents évoluant dans des environnements complexes et dynamiques, car de tels agents doivent pouvoir s'adapter à des demandes et à des opportunités changeantes [18]. Plus particulièrement, chaque agent doit pouvoir interagir avec d'autres agents ayant des habilités, des façons de raisonner et des buts communs, Pour pouvoir fonctionner efficacement. Les agents doivent donc pouvoir apprendre à connaître les autres agents et adapter leur comportement local selon la composition et les interactions.

Notre travail aujourd'hui s'inscrit clairement au niveau de la modélisation, avec un aspect méthodologique surtout, la modélisation ambitionnée dans ce mémoire (le quatrième chapitre), prévoit que le concepteur du système a comme objectif de créer un SMA tout en assurant qu'il présente un comportement coopératif global désiré, c'est l'objectif des approches par construction.

5.2.2 Approches de Construction

On rassemble sous cette appellation comme dans [18], les travaux qui visent à concevoir et construire des SMA qui présentent un comportement désiré (coopératif pour notre étude).

Afin d'obtenir un SMA utile, deux sortes de cette approche sont exposées à la suite, une utilité qui propage à notre sens la coopération entre les agents dans un système multi agent, c'est-à-dire qui répond conformément aux spécifications fixé par le concepteur.

Ces deux types d'approches seront attaqués brièvement dans les paragraphes ultérieurs. Notant bien que ces deux types d'approches, peuvent-ils être utilisés successivement comme il déclare Klein [18].

6 Coopération intentionnelle et non-intentionnelle

Lors des discussions sur la coopération, la coopération intentionnelle est souvent opposée à la coopération non-intentionnelle sans pour autant que ces deux notions soient clairement définies. A l'issue de cette analyse générale de la coopération, il est donc intéressant d'examiner le rapport entre l'intentionnalité et la coopération. Les concepts d'intention et d'actions intentionnelles sont assez difficiles à appréhender en raison notamment de leurs significations multiples [26]. La définition de ces concepts doit s'inscrire dans l'étude d'une théorie générale de l'action qui sort du cadre de cette thèse. L'objectif de ce paragraphe n'est donc pas de définir ces concepts mais simplement de préciser les notions de coopération intentionnelle et de coopération non-intentionnelle.

La coopération se rapporte à l'activité globale d'un ensemble d'agents. Par conséquent, les critères caractérisant l'action d'une façon générale s'appliquent également à la coopération. De la même façon que les actions sont différenciées suivant leur intentionnalité [27], la coopération intentionnelle peut être opposée à la coopération non-intentionnelle.

Dans le cas de la coopération non-intentionnelle (ex., la coopération des fourmis [28]), il n'est prêté aucune intention aux agents. Les actions des agents ne sont pas déterminées à l'issue

d'une délibération, elles ne sont pas intentionnelles en ce sens qu'elles ne sont pas accomplies en vue de produire un effet particulier.

Dans le cas de la coopération intentionnelle (ex., la coopération entre les échappés d'une course cycliste dont le but est de distancer le peloton), il y a une différenciation entre l'intention de coopérer et la mise en œuvre de la coopération. L'intention peut être définie comme un but à long terme persistant mais révoquant qui guide les délibérations et les actions des agents [27] [29]. Dans la coopération intentionnelle, l'intention de coopérer se rapporte à l'accomplissement d'un but commun partagé par l'ensemble des agents coopérants. Les actions des agents sont intentionnelles puisqu'elles sont accomplies en vue de produire des effets qui vont contribuer à la satisfaction du but commun à l'origine de la coopération. Les buts génériques de coopération énoncés précédemment sont caractéristiques de la coopération intentionnelle, contrairement aux indices de coopération qui peuvent s'appliquer aux deux formes de coopération. Le modèle d'agents coopératifs développé dans cette étude met en œuvre une coopération intentionnelle. Dans ce modèle, les intentions des agents se rapportant à la coopération sont déterminées par les engagements.

7 Les avantages de la coopération

La coopération à proposerai un certain nombre d'avantages qui s'expriment soit par des améliorations qualitatifs ou quantitatifs. Dans [11] les auteurs développent quatre buts génériques de coopération :

- Augmenter la vitesse de la résolution des tâches par leur parallélisations, ce but reflète la fonction de la répartition des tâches et de leur résolution concurrente.
- Augmenter l'ensemble des tâches réalisables par le partage des ressources, ce but concerne l'utilisation des ressources et des compétences telles que l'information, les résultats, le matériel.
- Augmenter la fiabilité du système, par la duplication des tâches et si possible par l'utilisation de différentes méthodes pour les réaliser ; ce but concerne aussi l'aptitude du système à suppléer la défaillance d'un agent.
- Diminuer l'interférence entre coopération, premièrement la coordination d'action par l'ajustement de la direction des actions des agents dans le temps (synchronisation) et dans l'espace. Deuxièmement, le non persistance des conflits qui témoigne les situations bloquantes, il indique la capacité des agents à prévenir les conflits ou à défaut de les résoudre.

Les auteurs dans [11] expriment qu'il est parfois impossible de satisfaire simultanément ces buts de coopération. Par exemple, l'optimisation d'une dimension spécifique de la coopération telle que la vitesse de résolution, ne permet pas d'assurer la coopération suivant d'autres dimensions telles que la sélectivité de la communication et la fiabilité. Souvent dénommé comme des paradoxes de coopération, comme dans [12]. Cette paradoxalité résulte principalement les besoin multi-objectif de la coopération (la minimisation d'énergie et de temps), pour minimiser le temps, l'énergie des robots devraient être maximisées et vis versa, clairement les exigences sont paradoxales.

8 Étude des différents travaux

Un système multi-agent peut être considéré (d'un point de vue statique) comme un ensemble de compétences distribuées. Il est donc légitime de croire au premier abord que l'efficacité du travail collectif d'une société d'agents est proportionnelle à la quantité d'agents ayant pris part à la résolution. Ce n'est pourtant pas ce qui a été observé en pratique par Mataric [31] en robotique, par Genesereth [32] et Etzioni [33] en recherche d'information. Ce phénomène provient du fait que des agents trop individualistes provoquent d'autant plus de situations indésirables pour la collectivité (conflits, concurrences,...) qu'ils sont nombreux. Dès lors, on comprend bien l'intérêt de la coopération.

Ce paragraphe présente des travaux essentiels qui sont classés en deux grandes familles : celle qui présente l'apport de la coopération dans la résolution de problème et celle qui considère la coopération comme attitude sociale d'un agent.

8.1 L'apport de la coopération pour la résolution

Nous allons présenter dans ce paragraphe les travaux de Bouron & Ferber qui montrent l'apport de la coopération dans la résolution de problème.

8.1.1 Les travaux de Bouron & Ferber

Les trois grandes fonctions de la coopération définies par Jacques Ferber sont l'accroissement des performances, la résolution de conflits et l'amélioration de la survie. La première fonction se mesure de manière qualitative et quantitative tant d'un point de vue individuel que collectif. Les deux dernières fonctions reflètent la capacité d'adaptation d'un individu ou d'un groupe lui permettant de maintenir son intégrité fonctionnelle. Ceci est particulièrement utile en monde ouvert. Une société d'agents autonomes sera amenée à faire face aux éventuelles perturbations ou aux éventuels dysfonctionnements tendant à la détériorer. Les mécanismes sociaux mis en

place pour assurer la viabilité du groupe restreignent encore les possibilités individuelles des agents. Ils contraignent leurs comportements et les poussent à se socialiser encore plus c'est-à-dire à devenir de plus en plus spécialisés et dépendant des autres. “Les organisations émergent des interactions sociales entre individus et contraignent en retour leur comportement” [10]. Thierry Bouron [34] a défini des indices de coopération permettant de mesurer les activités de coopération entre agents. Il s'agit du degré de parallélisation, de la coordination d'actions, de la robustesse, du non redondance des actions, du non persistance des conflits et enfin du partage des ressources. Ces indices permettent de s'affranchir des caractéristiques internes des agents et de ne s'occuper que de leur comportement observable. Leurs définitions reposent d'ailleurs sur des considérations d'observabilité et de quantifiabilité.

8.2 La coopération en tant qu'attitude sociale

Les études de Sekaran et Sen que nous allons présenter considèrent la coopération comme attitude sociale chez les agents. Elles définissent plusieurs attitudes sociales, parmi lesquelles la coopération, et étudient les différences de comportements et les avantages de chacune d'entre elles.

8.2.1 Les travaux de Sekaran & Sen

Sekaran et Sen [35] s'intéressent au comportement d'une société composée d'agents plus ou moins coopératifs. Ils définissent 4 types d'agents : les agents **égoïstes** qui demandent de l'aide mais n'aident jamais personne, les agents **individuels** qui ne demandent ni ne fournissent de l'aide à autrui, les agents **philanthropes** qui aident toujours les autres lorsque que cela leur est demandé et les agents **récioproques** qui acceptent ou non (à partir de méthodes stochastiques) une demande de coopération. Ces derniers n'apportent leur aide à autrui que si l'action leur est bénéfique à court ou long terme (ils font le juste poids entre l'énergie supplémentaire que cela leur demande et le bénéfice qu'ils peuvent en retirer par la suite). Dans [36], Sen insiste sur la différence entre le comportement des agents récioproques et le comportement suivant la règle “un prêté pour un rendu”. Sen attache tout particulièrement de l'intérêt au bénéfice que peut retirer un agent de la coopération.

Selon ses propos, un agent ne doit coopérer que s'il en retire un quelconque bénéfice, d'où les agents récioproques. Il argumente sa position en disant que dans un environnement distribué ouvert, l'unique stratégie défendable qu'un agent autonome puisse suivre en décidant ses actions est celui gouverné par son intérêt. Les différents comportements des agents ont été appliqués au problème de la livraison de paquets. Chacun des N agents doit livrer T paquets

contenus dans un dépôt centralisé à une destination définie aléatoirement. Un agent peut seulement transporter un paquet à la fois par lui-même ou avec l'aide d'un autre agent. Lorsqu'il revient au dépôt un nouveau paquet lui est assigné pour livraison. A ce moment-là, il regarde si d'autres agents sont dans le dépôt ; si tel est le cas il demande alors de l'aide pour livrer le paquet. Cette demande sera ou non honorée selon les attitudes des agents. L'intérêt d'être aidé est de pouvoir se déplacer plus rapidement donc d'atteindre son but plus rapidement. Les résultats obtenus montrent, d'une part, que la performance des agents réciproques est quasiment identique à celle des philanthropes. D'autre part, la performance moyenne du groupe repose sur la performance des agents égoïstes et réciproques, et évoluent de manière très proche de la performance des agents égoïstes lorsque le pourcentage de ces derniers est augmenté. Enfin, lorsqu'il y a peu de dépôts à faire, les agents égoïstes peuvent exploiter les agents réciproques pour la plupart de leurs dépôts ce qui leur confèrent momentanément des capacités meilleures.

Leur conclusion est qu'un comportement réciproque peut servir l'intérêt personnel tout autant que l'efficacité globale. Puisque, un comportement réciproque produit, à long terme, une performance meilleure qu'un comportement égoïste ou exploiteur, il est très intéressant pour tous les agents d'être réciproques.

Conclusion

Dans ce chapitre traitant de la coopération, nous ne nous sommes pas intéressés à la coopération au niveau de la résolution de problème distribué, mais au bénéfice de la coopération au niveau de la collectivité. Les différents travaux étudiés, montrent à l'unanimité, que la coopération est bénéfique pour obtenir un comportement proche de l'optimal. Ce résultat n'est pas trivial, car il est possible de croire que l'augmentation du nombre d'agents (non coopératifs certes) dans une société est suffisante pour accroître les performances du système. Cependant, divers travaux ont montré que le caractère individualiste des agents allait à l'encontre de ce phénomène. Dès lors, nous comprenons immédiatement le caractère prépondérant de la coopération.

Faire coopérer les agents en leur attribuant une attitude sociale coopérative nous semble également une bonne méthode. Les agents n'ont alors pas le choix d'avoir un comportement autre que celui imposé par leurs attitudes sociales. Le résultat qui en découle est donc à coup sûr bénéfique pour la collectivité.

Chapitre 3 :

Colonie de Fourmis

Chapitre 3 : Colonie de Fourmis

Introduction

Le comportement des fourmis réelles représente une grande source d'inspiration dans le domaine informatique. L'intérêt accordé aux fourmis a conduit à l'élaboration de quantités d'algorithmes basés sur des populations de fourmis artificielles et ceci notamment dans les domaines de la classification et de l'optimisation.

Les algorithmes de colonies de fourmis sont inspirés par le comportement d'insectes sociaux, mais ce ne sont pas les seuls algorithmes à être issus de l'étude du comportement animal (éthologie). En effet, l'optimisation par essaim particulière ("Particle Swarm Optimization" [37],) est issue d'une analogie avec les comportements collectifs de déplacements d'animaux, tels qu'on peut les observer dans les bancs de poissons ou les vols d'oiseaux, il existe également des algorithmes inspirés des comportements des abeilles [38],[39]. On trouve encore des algorithmes s'inspirant de certains aspects du comportement des insectes sociaux, bien que ne faisant pas usage des caractéristiques classiques des algorithmes de colonies de fourmis (voir par exemple [40], [41]).

1 La biomimétique et l'informatique

Lorsqu'un nouveau problème se pose en informatique, et plus habituellement en ingénierie, il faut parfois définir de nouvelles méthodes de résolution car les techniques existantes ne sont pas précisément adaptées au cas traité. Ainsi, lorsque l'on veut inventer une nouvelle méthode de résolution de problème, il faut généralement une source d'inspiration. La source d'inspiration que constitue la biologie a de plus en plus de succès dans une branche de l'intelligence artificielle que l'on peut nommer informatique biomimétique.

L'objectif du processus de création est de concevoir un modèle résolvant un problème ou une catégorie de problèmes en s'inspirant de schémas de comportements mis au point en éthologie. La première étape se base sur des études menées en biologie et consiste à extraire de ces études un modèle réalisable du point de vue informatique.

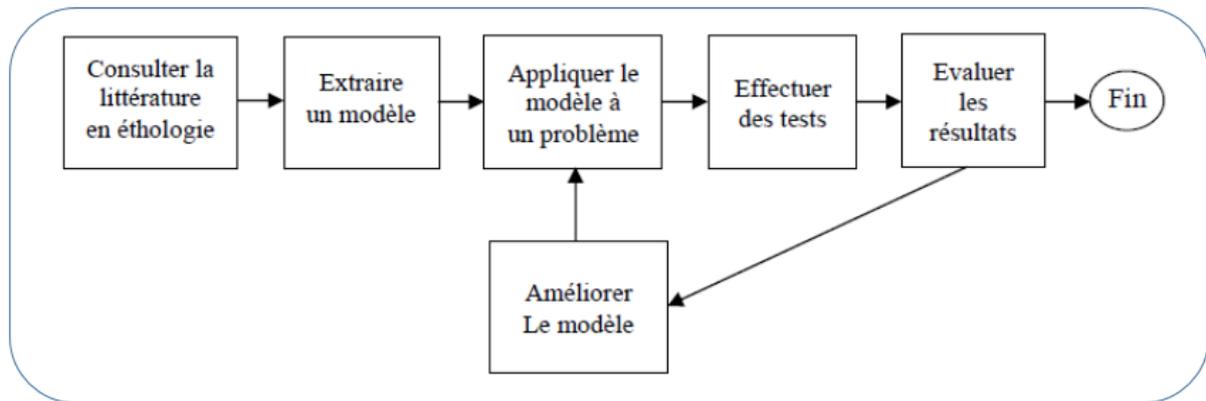


Figure 11 : Méthodologie de conception en informatique

2 Les fourmis artificielles

L'étude sur le comportement des fourmis a donné une naissance à plusieurs nouvelles méthodes de résolution de problèmes. Certains mécanismes de résolution collective de problèmes chez les fourmis ont été transformés par des informaticiens en méthodes utiles pour l'optimisation et le contrôle qui peuvent s'appliquer aujourd'hui à tout un ensemble de problèmes scientifiques et techniques [42] [43].

Les fourmis artificielles sont des agents artificiels et par facilité son nom de fourmi. La programmation informatique moderne permet de faire fonctionner en parallèle divers « agents », par exemple des agents fourmis qui interagissent entre eux et avec leur environnement. Chaque agent répond en fonction d'une ou d'un petit nombre de règles simples, sans qu'il soit nécessaire de faire appel à des capacités cognitives élaborées. Les fourmis modélisées sont alors des agents réactifs et on parle alors d'intelligence en essaim [44] [42].

La fourmi artificielle peut prendre la forme d'un processus informatique, qui possède donc ses propres zones de mémoire et ses instructions de fonctionnement. L'intérêt de la modélisation et la simulation informatique pour le biologiste est évident : cela permet de tester des modèles facilement (par rapport à l'effort expérimental de manipuler de vraies fourmis), de chercher une explication à des phénomènes émergents ou de tester la capacité de prédiction d'un modèle.

3 Les algorithmes colonies de fourmis

Les algorithmes de colonies de fourmis (ACO : Ant Clony Optimization) sont des algorithmes inspirés du comportement des fourmis et qui forment une famille de métaheuristiques

d'optimisation. Parmi les algorithmes évolutionnaires, l'un des plus récent est l'optimisation par colonies de fourmis, introduit par Marco Dorigo [45], pour la recherche de chemins optimaux dans un graphe [46]. Le premier algorithme s'inspire du comportement des fourmis recherchant un chemin entre leur colonie et une source de nourriture (**Figure 12**). L'idée originale s'est depuis diversifiée pour résoudre une classe plus large de problèmes et plusieurs algorithmes ont vu le jour, s'inspirant de divers aspects du comportement des fourmis. Les ACO sont des approches récemment proposées pour résoudre des problèmes d'optimisation avec une combinatoire énorme et complexe. La source d'inspiration des ACO est la trace des phéromones et le comportement des fourmis réelles qui utilisent les phéromones comme moyen de communication. Par analogie à la biologie, les ACO sont basés sur la communication indirecte d'une colonie de simples agents appelés fourmis artificielles en utilisant des traces de phéromones artificielles. Les fourmis artificielles dans un ACO mettent en œuvre une heuristique de construction aléatoire qui construit des décisions probabilistes en fonction des traces de phéromones artificielles et les données d'entrées du problème pour le résoudre. La **Figure 12** représente la faculté d'une colonie de fourmis à retrouver le plus court chemin entre leur nid et une source de nourriture. Des détails sur le principe de fonctionnement et les applications des ACO peuvent être trouvés dans ([47] ; [48] ; [49]).

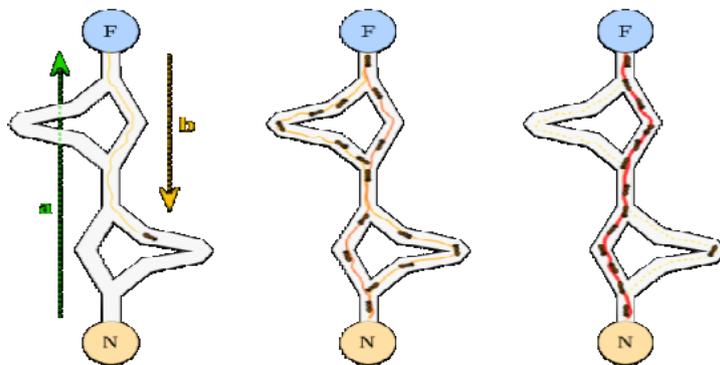


Figure 12 : Choix du plus court chemin par une colonie de fourmis

La première étape de l'application d'un ACO est la définition d'un modèle adéquat : le modèle de phéromones. En générale, tout algorithme de colonies de fourmis doit contenir les éléments suivants : *traces de phéromone, information heuristique (visibilité), règle de mise à jour des traces de phéromone, construction de solutions, probabilité de sélection et condition d'arrêt.*

3.1 Les traces de phéromones

Les fourmis ont la particularité d'employer pour communiquer des substances volatiles appelées phéromones. Elles sont attirées par ces substances, qu'elles perçoivent grâce à des récepteurs situés dans leurs antennes. Dans le cas des fourmis artificielles, les traces de phéromones sont représentées en fonction de la valeur de la fonction objective pour l'arc considéré. La définition de la quantité de phéromone est spécifique à chaque problème étudié et dépend de la variante d'ACO utilisée. Pour l'algorithme de base « *ant system* » utilisé pour le problème du voyageur de commerce



Figure 13 : Des fourmis suivant une piste de phéromone

La quantité de phéromones $\Delta\tau_{ij}^k$ déposée par une fourmi est donnée par :

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L^k} & \text{si la fourmi } k \text{ est passée par l'arc } (i,j) \\ 0 & \text{sinon} \end{cases}$$

Où Q est une constante et L^k la longueur du chemin parcouru par la fourmi k ou F^k valeur de la solution obtenue par la fourmi k (la meilleure solution).

3.2 La construction des solutions

Chaque fourmi k est placée sur un nœud source et recherche un chemin en sélectionnant les nœuds i à ajouter à sa mémoire *tabou* _{k} . À chaque fois qu'une fourmi se trouve sur un nœud i , elle choisit le nœud j suivant selon une règle probabiliste donnée dans le cas du « *Ant System* » par :

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \notin \text{tabou}_k} \tau_{il}^\alpha \cdot \eta_{il}^\beta} & \text{si } j \in \text{tabou}_k \\ 0 & \text{sinon} \end{cases}$$

Où τ_{ij} représente les traces de phéromones et η_{ij} la visibilité. α et β sont respectivement l'importance relative des traces de phéromones et de la visibilité l'une par rapport à l'autre.

3.3 Mise à jour des traces

Une fois que les solutions ont été évaluées, elles peuvent influencer la matrice de phéromones à travers un processus de mise à jour des traces de phéromones. La mise à jour des traces de phéromones est effectuée à la fois localement et globalement. Le but de la mise à jour des phéromones est d'augmenter les valeurs de phéromones associées aux solutions « bonnes » ou prometteuses et de diminuer la valeur de phéromones associées aux mauvaises solutions. Cette mise à jour est donc opérée par :

- La diminution des valeurs de toutes les phéromones à travers l'évaporation des phéromones
- L'augmentation du taux de phéromones associées à un ensemble de « bonnes » solutions

$$\tau_{ij} \leftarrow \tau_{ij} (1 - \rho) + \sum_{k=1}^m \Delta\tau_{ij}^k$$

Avec : ρ coefficient d'évaporation des traces de phéromones, $\Delta\tau_{ij}^k$ renforcement de l'arc (i, j) pour la fourmi k . L'évaporation des phéromones est nécessaire pour éviter une convergence trop rapide de l'algorithme.

Les variantes de ACO sont le « Ant System » (AS) [50], le « Ant-Q » [51], le « Ant Colony System » (ACS) [52], et le « Max-Min Ant System » (MMAS) [53], tous obtenus par l'introduction de nouvelles procédures ou par modification des paramètres de l'algorithme original.

3.4 Algorithme général de colonie de fourmis

1. Initialisation des *traces de phéromones*
2. **Tant que** le critère d'arrêt n'est pas atteint, **faire**
 - Pour** toutes les fourmis, **faire**
 - *Construire une nouvelle solution* en utilisant les traces de phéromones et la visibilité
 - Évaluer les solutions construites
 - Fin pour**
3. *Mise à jour* des traces de phéromones

4. Fin tant que

4 L'optimisation multi-objectif par colonies de fourmis

Compte tenu du succès des ACO pour l'optimisation de problèmes mono-objectifs, de nombreuses études sont de plus en plus menées sur leur application pour l'optimisation de problèmes multi-objectifs. La rapidité d'exécution des ACO pour les problèmes d'optimisation combinatoire a été prouvée dans García-Martínez ([54]) où il est démontré que les algorithmes de colonies de fourmis sont plus efficaces que des algorithmes génétiques existants pour la résolution de problèmes multi-objectifs. Divers algorithmes de colonies de fourmis multi-objectifs ont été proposés en modifiant les procédures de l'ACO « classique ». La plupart de ces algorithmes sont proposés pour résoudre des problèmes d'optimisation multi-objectifs concrets tels que l'ordonnancement, les problèmes de tournées de véhicule [55], ou la sélection de portefeuilles [56]. Ces algorithmes ont été construits en modifiant divers paramètres de l'ACO classique et en réadaptant certains paramètres tels que les matrices de phéromones et la visibilité. Les algorithmes présentés utilisent soit une colonie unique ou plusieurs colonies échangeant les informations sur les solutions trouvées.

4.1 Les traces de phéromones

La quantité de phéromones présente sur un élément donné représente les expériences passées de la colonie sur cet élément. Dans le cas de problèmes mono-objectifs, cette expérience est définie en fonction de l'objectif considéré. Cependant, lorsque nous avons plusieurs objectifs, nous pouvons considérer deux stratégies :

- Considérer une structure de phéromone unique ; dans ce cas, la quantité de phéromones déposée par une fourmi est fonction d'une agrégation des différents objectifs [55];[57];[58];[59].
- Considérer plusieurs structures de phéromones ; dans ce cas, on associe une matrice de phéromones différente à chaque objectif ; chaque objectif ayant sa propre structure de phéromones [60];[56] et [61];[62];[63].

4.2 Le processus de mise à jour des solutions

Pour mettre à jour les traces de phéromones, on doit décider parmi les solutions construites celles qui portent le plus de phéromones.

4.2.1 La mise à jour individuelle : dans ce cas, une des matrices de phéromones est sélectionnée pour la mise à jour [56];[63];[64].

4.2.2 La mise à jour globale : dans ce cas, plusieurs matrices de phéromones sont mises à jour. [61];[65]

4.3 La visibilité

La visibilité intervient pour beaucoup dans l'efficacité de l'algorithme, il est donc nécessaire d'en définir correctement les paramètres. Dans le cas de l'optimisation mono-objectif, la visibilité est définie comme l'inverse du critère à optimiser. Pour l'optimisation multi-objectif, deux stratégies peuvent être considérées :

- Une première stratégie consiste à agréger les différents objectifs en une seule information heuristique [61];[60];[57] les méthodes d'agrégation utilisées diffèrent d'une variante à l'autre. Doener par exemple propose une matrice définie comme étant l'inverse de la somme des critères pris en compte [61].
- Une seconde stratégie est de considérer chaque objectif séparément ; dans ce cas, une matrice de visibilité est associée à chaque objectif [55];[56];[62];[63];[58].

4.4 La construction des solutions

La clé de la construction des solutions dans le cas multi-objectif se trouve dans la combinaison de la visibilité et des valeurs des phéromones. Ces combinaisons sont de plusieurs sortes :

4.4.1 Ciblées : L'information (phéromone ou heuristique), se rapportant à un seul objectif, est utilisée dans le processus de construction des solutions [62];[56]

4.4.2 Dynamiques : L'information qui s'étend à plus d'un objectif (matrice de phéromones multiples ou multi-heuristique) est utilisée dans le processus de construction. La combinaison d'informations spécifiques de l'objectif est dynamique de sorte que les objectifs peuvent être pris en compte à différents moments de la construction ou bien des fourmis spécifiques peuvent combiner les informations sur les objectifs en utilisant la pondération [66];[55].

4.4.3 Fixes : La combinaison, des informations spécifiques de l'objectif, est fixée a priori [57]; [56].

4.5 Évaluation des solutions

L'évaluation et le classement des solutions ont été abordés de diverses manières dont les deux principales sont les méthodes d'évaluations basées sur la construction de fronts de Pareto et les méthodes non-Pareto.

4.5.1 Pareto : Les solutions sont évaluées pour tous les objectifs et se voient attribués un « score » qui reflète la façon dont la solution satisfait tous les objectifs dans le sens de Pareto en se référant classement selon la dominance [67];[63].

4.5.2 Non-Pareto : Les solutions sont évaluées pour un ou plusieurs objectifs et se voient attribuer un « score » qui reflète à quel point la solution satisfait l'un de ces objectifs ou un « score » basé sur une somme pondérée de tous les objectifs [56];[62].

Conclusion

On a voulu à travers ce chapitre de présenter quelques notions importantes, qui ont un lien avec les algorithmes de colonies de fourmis et de bien éclairer leur utilisation suivant ces notions et ces principes. Les fourmis en colonie présentent, en effet, des comportements auto-organisés.

L'objectif de ce chapitre était de présenter un aperçu sur l'optimisation et ses techniques d'un côté et de se fonder sur celle des fourmis par leurs outils (algorithmes) de l'autre côté.

Chapitre 4 :
Implémentation &
Expérimentation

Chapitre 4 : Implémentation & Expérimentation

Introduction

Ce chapitre a pour objectif de mettre en exergue notre vision de la coopération entre nos robots, apercevait les principaux éléments de conception, et d'implémentation, liées à une simulation rigoureuse de systèmes multi-robots à base d'agents (TURTLE) NetLogo coopérants.

Ces différentes modélisations vont être détaillés, discutés et analysés afin de permettre la mise en place du simulateur SC_SMA. Ce simulateur est dédié à l'étude des systèmes multi-Robots en termes de planification de chemin vers des cibles fixes hétérogènes. Finalement abordons une étude expérimentale.

Le but de ce chapitre est de familiariser le lecteur avec notre réflexion méthodologique exploitée, en termes de modélisation et d'implémentation afin de concrétiser notre simulateur de coopération ambitionné. Avec les différents mécanismes inspirés.

1 Les plateformes de simulation Multi-Agent

Dans le domaine de la simulation multi-agent [72], les plateformes sont des outils méthodiques importants, qui aident les chercheurs dans le processus de modélisation et le développement des programmes de simulation. En d'autre terme, il présente des supports d'aide à la programmation. Elles fournissent une couche d'abstraction permettant de faciliter l'implémentation des concepts orientés agents.

2 NetLogo une plateforme générique

On a travaillé avec une plateforme générique appelé Netlogo, téléchargée du site [web3]. Après l'installation, nous avons distingué trois panneaux pour le cadre Netlogo, dont deux indispensables (Procédures et Interface) et une (information) utile mais pas nécessairement capitale.

2.1 Motivation pour NetLogo et définition

Cette partie est extraite essentiellement du guide Netlogo, retrouvé dans le site [web4].

NetLogo est une plateforme multi-agent de référence, un environnement programmable permettant de modéliser et simuler des phénomènes naturels et sociaux. Il a été créé par Uri Wilenski en 1999, son développement est poursuivi de manière continue par le Center « for Connected Learning and Computer-Based Modeling », dont la dernière version 6.0.1, daté

mars 2017. En plus, est un logiciel gratuit et open source, peut être téléchargé sur le site [web5]

NetLogo étant écrit en Java, il peut tourner sur tous les systèmes d'exploitation majeurs (Mac, Windows, Linux,). 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. NetLogo est accompagné d'une documentation complète et de nombreux tutoriaux.

L'intérêt majeur de cette plateforme générique provient de l'ensemble des primitives, et les outils offerts, Comme il a été discuté ailleurs [73] [74], nous avons trouvé NetLogo convenable pour modéliser des systèmes à base des agents. À l'égard, chaque agent NetLogo:

- Perçoit son environnement et actes sur son environnement,
- Porte son propre fil de contrôle,
- Autonome, il matérialise la définition classique d'agent trouvée dans [75].

NetLogo est un outil excellent pour le prototypage initial et rapide, qui expérimente les systèmes multi-agent. Il convient particulièrement à des systèmes avec des agents situés et qui opèrent dans un espace restreint. Il représente aussi un excellent outil d'animation du système à modéliser.

2.2 La répartition des agents NetLogo

L'Univers NetLogo peut être composé de différents types d'agents. Dans [76] il existe principalement quatre types d'agents : les tortues, le monde (environnement), les patchs, les liens.

2.3 Le modèle Net Logo, les étapes de mise en œuvre

La mise en œuvre d'un modèle multi agent NetLogo se déroule en quatre étapes :

Premièrement : il faut décrire l'interface qui contient tout ce qui concerne la visualisation, ainsi que les paramètres que l'utilisateur pourra modifier.

Deuxièmement : on peut décrire des variables globales, et on peut personnaliser les agents en définissant leurs propriétés.

Troisièmement : il faut décrire ce qui se passe quand on prépare la simulation, initialiser les variables, créer les agents, préparer l'environnement. Généralement cela peut être fait en utilisant le bouton Initialiser(**Setup**).

Finalement : on doit décrire ce qui se passera dans chaque pas de simulation, et décrire ce que la plateforme affichera aux utilisateurs, évidemment cela peut être fait en employant le bouton démarrer (**Go**).

2.4 Le Langage NetLogo et langage UML

Ces dernières années, les méthodologies multi agents, et notamment les plateformes derrière eux, beaucoup de langages de développement orienté agent, ont profité du langage de spécification orienté UML, Notre objectif dans cette partie, est de citer quelques travaux étudiés, présente un rapport et un lien entre le langage UML et le langage NetLogo.

Pour concrétiser cet objectif, on a appuyé sur deux ateliers [77] [78] retrouvés dans la littérature multi-agents, afin d'éclairer les relations entre ces deux langages orientés objet (UML) et orientés agent (NetLogo).

3 Cycle de Vie d'un Logiciel Multi agent

Décrivons un simulateur orienté agent comme un logiciel multi agent, et considérons l'influence qu'a pu avoir l'univers objet sur l'agent, tout a fait comme dans [79], les cycles de vie et le processus utilisé dans la modélisation de systèmes multi-agents sont de type classique utilisé en conception orientée objets.

Selon Picard [79], Les principales phases d'un processus de développement sont : l'analyse des besoins, la conception de l'architecture, la conception détaillée, le développement (ou implémentation) et le déploiement.

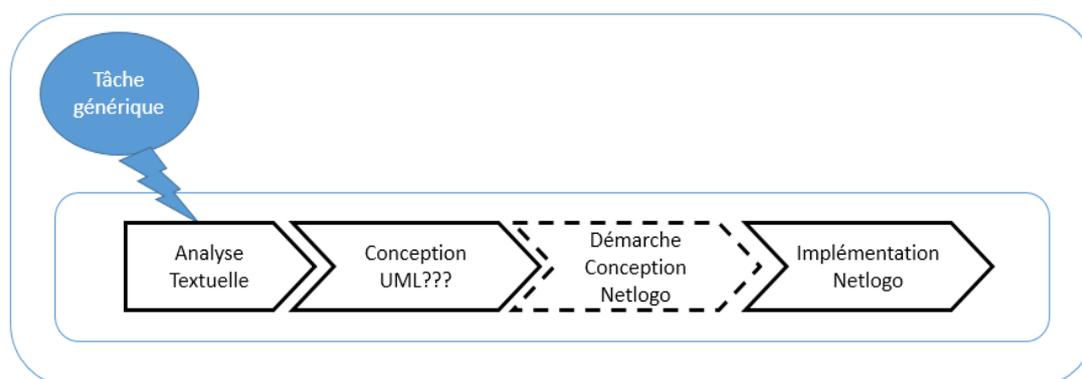


Figure 14 : Processus de développement de SC_SMA

Somme tout, notre processus suivi et illustré par la **Figure 14**.

Préalablement et avant d'expliquer les quatre étapes de notre processus de développement inspiré, il est important d'éclairer ce que signifient les points d'interrogation mentionnés dans la

partie conception de ce processus. En effet il désigne pour nous le non confirmation du modèle spécifié (au sens formelle), particulièrement en termes de diagramme de classe, et d'activité indiquent la tâche générique souhaitent modélisé.

Rejoignons maintenant à la description des différentes étapes de notre processus inspiré suivi:

4 Les Tâches Générique et la modélisation Orienté Agent

Les tâches génériques dans le domaine Multi Agent, ont comme principal objectif [80] tout comme dans d'autres domaines adjacents (Ex. Robotique), de fournir des tâches de référence, pour évaluer les architectures de contrôle proposées dans la littérature, surtout avec un contexte robotique. Alors que dans notre cas, et pour un contexte de modélisation et développement multi agent, ces tâches génériques sont censées comme animateur de l'ensemble des étapes d'un processus de développement. i.e, manipuler les quatre étapes de notre processus, surtout la première étape d'analyse, afin d'inciter le processus.

C'est on peut montrer une épithète opérationnel, on peut dire que les tâches générique indique l'essence pour n'importe quelle type de modélisation multi agent.

Fréquemment, parmi les tâches génériques les plus utilisées dans la littérature multi agent, toute comme dans [81], nous pouvons indiquons :

- **Le mouvement en formation** : il y une contrainte de contrôle sur un ensemble d'agent mobile, c.-à-d. il faut qu'un ensemble d'agents mobile préserve certaines formes de cohérence, l'aspect coopérative ici est en terme de coordination entre les agents. [82]
- **La Tâche Coopérative de Poussée Objet TCPO** : il y une contrainte sur l'objet poussé par les robots, i.e un objet, n'est pas déplaçable par plus d'un robot, c'est l'une des situations de coopération les plus difficile, l'aspect de coopération ici est en terme de coordination et de participation. [83]
- **Le fourragement** : il n'y pas de contrainte sur la coordination entre les agents, ou sur les objets poussé, mais la coopération est aperçu ici en terme de planification de chemin entre l'ensemble des agents mobiles (fourmis), afin d'assister les autres fourmis pour atteindre des cibles fixes (source de nourriture). [84]

Cette dernière tâche est largement utilisée dans la littérature, pour cela nous nous servons aujourd'hui, comme une base dans laquelle on incite notre processus inspiré, afin d'exhiber

notre coopération considéré entre nos robots mobiles, mais avec quelques modification dans la description de la tâche de fourragement.

Une modification provient particulièrement, par la fusion de cette tâche de fourragement avec la tâche "TCPO". La description de cette tâche de fourragement modifié est énoncée dans le paragraphe suivant.

4.1 Description de la tâche étudiée (Le fourragement modifié)

Il s'agit de réécrire notre tâche destiné dans notre modélisation attendu, décrit comme suite : "Dans un terrain borné se trouvent des robots virtuelles (fourmis), des objets et des cibles. La tâche collective décrit que les robots cherchent les objets dans l'environnement, une fois le robot détecte l'objet, il le porte (chaque robot peut porter un objet contrairement aux tâches TCPO, où la nécessité de plus qu'un robot est indispensable pour déplacer l'objet), ensuite le robot porteur cherche une cible non pleine ou des trace sur l'environnement pour déposer l'objet. Cette dernière opération signifie que les cibles ont un attribut 'capacité', pour lequel, il ne peut pas héberger des objets plus que sa capacité". Afin de posé l'objet dans la cible par l'agent, ce dernier commence la coopération pour agents en laissant des traces statique sur l'environnement.

NB: Dès maintenant, et pour lever toute ambigüité conceptuelle rencontrée ultérieurement, les agents mobiles, les robots-fourrageurs, ou des robots (virtuels) tout court, signifie la même chose qu'un robot-Turtle.

4.2 Processus Suivi et Approche Adopté

Dont le but d'obtenir un modèle de coopération de nos robots, nous avons appuyé sur la tâche de fourragement modifié étendu au-dessus, quant à mettre en œuvre notre processus de développement inspiré, pour peaufiné finalement notre simulateur **SC_SMA**.

Si on peut situer notre approche de coopération employée aujourd'hui, par rapport à notre état de l'art de coopération du deuxième chapitre, On peut dire que c'est une approche de coopération par construction, basée sur deux types de communication :

1. Une communication non intentionnelle (mécanisme basé sur la colonie de fourmis), pour maitriser la planification des chemins de nos robots-turtle, et
2. Une communication intentionnelle, pour maitriser la mobilité des robots-turtle.

Maintenant, on va déclencher notre processus inspiré :

4.3 Analyse de la Tâche générique étudiée

Il s'agit d'une de relecture de la tache générique définit comme suite :

Dans un **terrain borné** (**grille régulière**) se trouvent des **robots**, **objets**, et **cibles** : Les robots **se déplacent** en **parallèle**, avec une incrémentation du **temps** et d'**énergie** pour l'ensemble des robots.

1) Si un robot **détecte** un objet **non occupé** dans son **champ de vision**, il **porte** l'objet (L'objet devient occupé par un robot **porteur**)

2) Sinon le robot **circule librement**, en continuant la recherche des objets.

3) Si un robot occupe un objet **détecte** une cible fixe non plein, **dépose** l'objet, **mettre à jour** la cible, il devient **robot libre**, pour chercher un autre objet.

4) Sinon il continue la recherche des cibles non plein.

Jusqu'à l'ensemble des objets distribué dans l'environnement **devient nul**

Une relecture attentive de notre tâche, nous permet d'éclater:

- Quatre thèmes (Robots, Objets, Cibles, Environnement).
- Des attributs pour les différents thèmes.
- Des relations entre les thèmes.
- Des Méthodes pour le thème robot.

5 La conception

Dans cette section de notre processus inspiré, on doit présenter deux diagrammes UML, un statique, et une autre dynamique, afin de schématiser, visualisé, tout simplement simplifié la tâche étudié.

Pour cela nous avons appuyé sur un diagramme de classe (statique) pour ordonnancer les différents thèmes dénombrés dans la phase analyse préliminaire. Et un autre diagramme d'activité (dynamique) pour exhiber certains comportements pour nos robots mobile.

D'autre part ces deux diagrammes développés sous l'outil **Entreprise architecte** [web6], ne sont pas forcément validé formellement. Proprement dite, se sont développés au sens d'annotation seulement, pour ordonner, différencier quelques comportements premiers pour nos robots, ainsi que les différentes agents constituent (objets, cibles).

La **Figure 15**, présente un diagramme de classe de notre tâche souhaitons modéliser. Ce diagramme nous montre une multitude comportements, peuvent accepter les robots, les cibles, et les objets à déplacer.

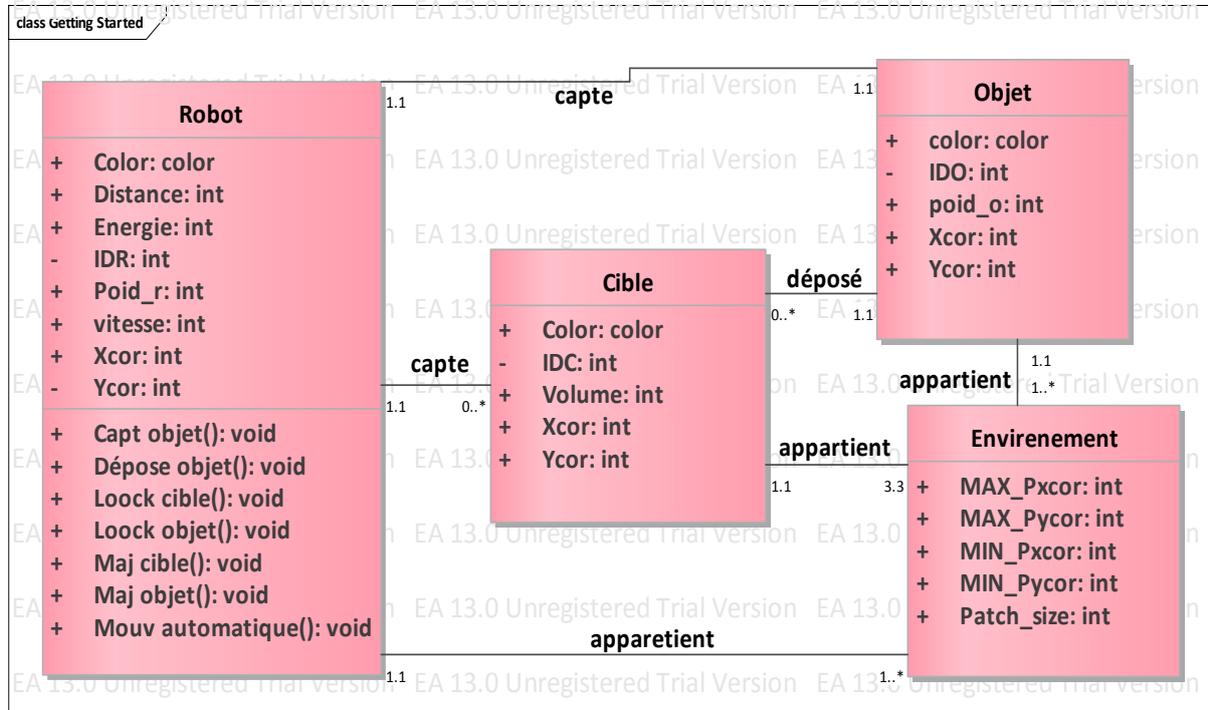


Figure 15 : Diagramme de classe

La **Figure 16** suivante présente un diagramme d'activité global pour l'ensemble des robots au commencement de la tâche générique étudié.

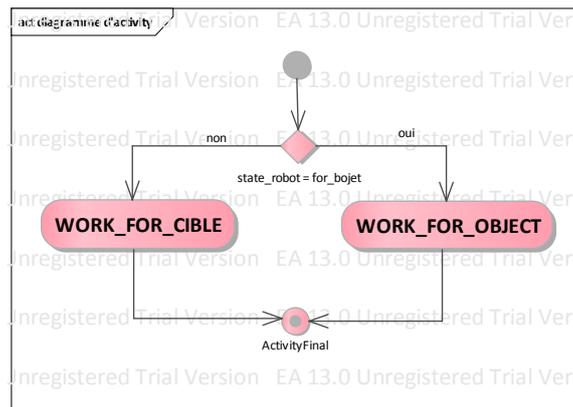


Figure 16 : Diagramme d'activité pour la tâche générique.

Ce qu'on peut ôter de notre conception **UML**, c'est qu'au début de la tâche, les robots disposent uniquement deux groupes distincts :

Un premier explorateur d'objet, possède assez d'objet pour le déplacer, et un deuxième explorateur de cible, quand il ne reste plus d'objet à transféré.

La **Figure 17** suivante présente un diagramme d'activité global pour l'ensemble des robots avec la tâche de coopération :

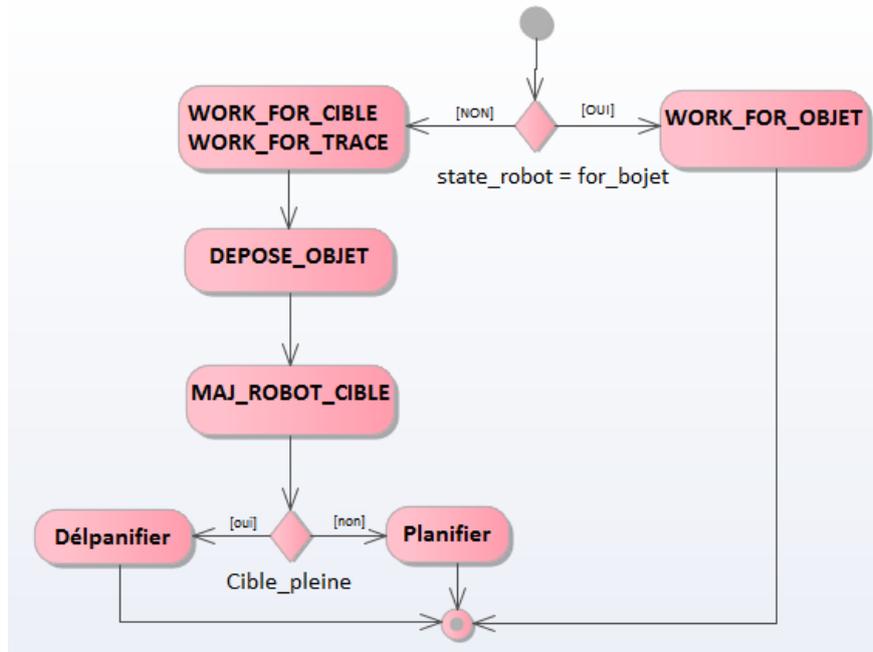


Figure 17 : Diagramme d'activité pour les tâches de coopération.

5.1 Aperçu détaillé sur les mécanismes de coopération souhaité

Notre première impression pour la coopération, s'apparaitre au niveau des robots avec le premier groupe (work_for_objet) de notre diagramme d'activité. C'est-à dire en absence du caractère coopérative entre les robots, une fois le robot dépose un objet dans la cible, et sachant qu'il y a d'autre objets a déplacé, a ce stade les agents changent son comportement et devient explorateur d'objets, mais a notre recommandation et avec un caractère coopérative, en tenant bien sur nos hypothèses imposées (**Hypo1**). C'est ici que le comportement coopératif doivent s'apparaitre, ici nos agents crée intentionnellement un comportement qui pilote ces agents vers le centre ou les coins de l'environnement (soit le plus proche(**MC+**), soit le plus loin (**MC-**)), pour laisser des marques sur ce dernier (environnement). De la même manière que l'algorithme de colonie de fourmis qui cherche à laisser des phéromones (marque dynamique) dans l'environnement, on a dessiné ces phéromones statique par des patches Netlogo, ensuite de les numérotés graduellement à partir de la cible vers le centre (ou les coins), pour orienter les robots vers les cibles le plus rapidement possible.

Jusqu'à l'ici nous avons parlé de la coopération que dans le premier groupe (`work_for_objet`), cependant et pour articulé la coopération du deuxième groupe (`work_for_cible`), Il faut qu'une gestion de mobilité doit être mis. L'obligation de cette gestion stipule qu'une fois le nombre d'objet à déplacer devient nul (non forcément tous les objets sont déposé dans les cibles, certainement occupé par d'autre robots), là il faut que l'ensemble des robots (que ça soit occupé ou non) savent cette indication. Et pour le disposent, pour que les robot-mobile changent du première groupe vers deuxième, il faut que une stratégie de partage d'information qui doivent être adoptée par l'ensemble de robots, et dont le but d'assurer ce partage, nous avons appuyé sur la technique de table noire (Blackboard), en tenant des variable globales (**Liste_objet_roboti**), pour mentionné le nombre de robots occupé par des objets en temps réel.

A ce deuxième groupe pour nos robots, est comme il ne reste plus d'objets à déplacer, notre intuition distinctive pour la coopération s'apparaisse entre nos robots par la détection des cibles non pas encore aboutis (on présuppose que nos robots connaissent dès le début le nombre de cible fixes existent dans l'environnement) ensuite laissé des traces vers le centre ou les coins de l'environnement selon le mécanisme adopté, alors que si toutes les cibles ont été aboutis, dans ce cas les robots libre tente d'éloigné des espaces ou il y a des traces (marque sur l'environnement).

Les deux groupes renommés, divers comportements dedans, ainsi que les procédés de nos robots pour passer d'un groupe (resp. comportement) à un autre, sont programmé dans la partie implémentation, convenablement passant nécessairement par la démarche de passage suivante.

6 Démarche

6.1 Identifier les variables globales

À partir des diagrammes de classe et d'activités accomplies. On essaye d'extraire les variables globales, qui peuvent être partagé par L'ensemble de composant du phénomène modélisé, par exemple dans notre application les variables globale sont (le temps, l'énergie et la distance) de notre modélisation, ces derniers indiquent le temps et l'énergie consommées, ainsi que la distance parcourue par les robots.

6.2 Identification des propriétés des objets à modéliser

Cette identification doit être en termes de structuration/organisation, ou en termes de mobilité, par exemple :

- Si la structure à modéliser est un réseau, il est fortement recommandé de modéliser le système par des composants de type 'LINK', alors que dans le cas où le système est assimilé à une grille régulière, comme celle de notre cas, il est recommandé donc d'utiliser Les 'PATCH', pour modéliser l'environnement, donc notre environnement est borné (non ouvert), il suffit de décocher les deux lignes de la fenêtre illustré par la **Figure 18**

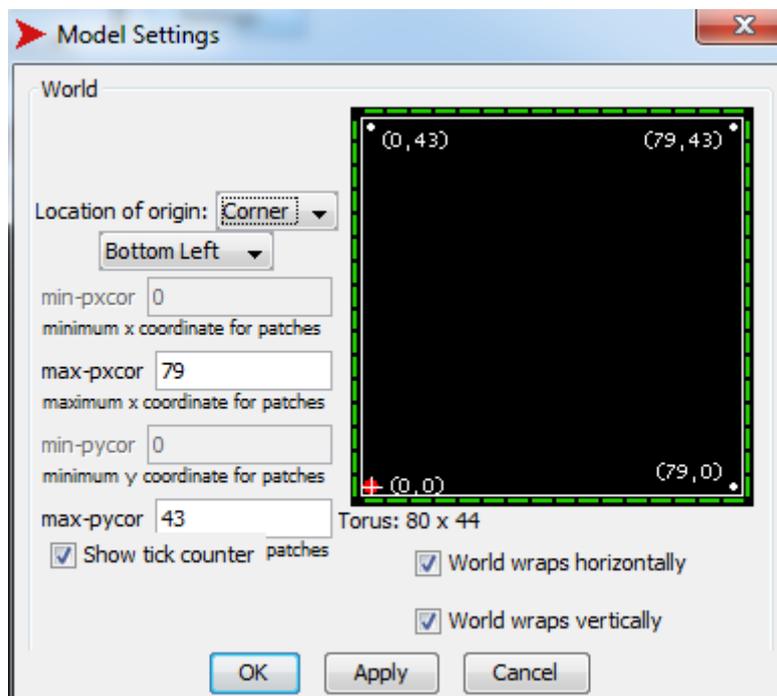


Figure 18 : La configuration de notre environnement

- Si on parle de mouvement des agents alors, on peut modéliser les agents comme des turtles, c'est le cas où ces agents sont mobiles. Sinon, il suffit de le modéliser sous forme des patches. [77]

6.2.1 Pour l'environnement

Définit comme un ensemble de patches, Modélisé par défaut dans la plateforme Netlogo.

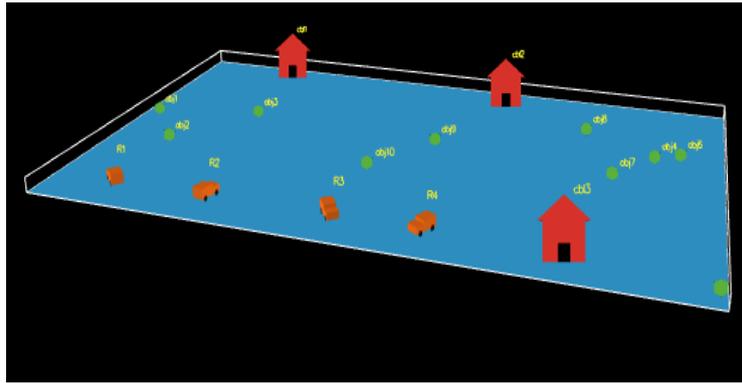


Figure 19 : Vue Trois Dimensions de l'environnement.

6.2.2 Pour les robots

Défini comme des composants totalement mobiles, continuellement supposés comme des Turtles Netlogo, afin de profiter l'ensemble de primitive, Netlogo accordé aux turtles, à savoir le déplacement 'Forward' pour avancer les Turtles, 'right' pour tourner à droite....etc.

6.2.3 Pour les objets

Défini comme des composants pseudo-mobiles, ou la mobilité de ces objets est liée fortement à la mobilité de ces robots porteurs, pour cela on peut considérer nos objets aussi comme des agents Turtles Netlogo, mais la primitive de déplacement, de rotation de ces agents objets-Turtles, est forcément piloté par le Turtle robots porteur.

6.2.4 Pour les cibles

Défini comme des composants immobiles. On peut les définir comme des patches Netlogo seulement, mais d'une côté pour des besoins d'ergonomie de notre simulateur (on peut pas définir des shapes pour les patches Netlogo, contrairement aux turtles), et d'un autre côté notre vision comme perspective d'étudié, le cas des cibles mobile plus tard. Pour ces raisons, nos cibles modélisé sont présumés également comme des Turtle netlogo, quoique sans aucune mobilité (déplacement), n'a été pris en compte.

NB: Il faut bien différencier entre les trois types Turtles employés (robots, objets, cibles) par les primitives breed et la primitive Shape. Pour cela dans notre simulateur nous avons défini trois breed distinct :

Breed [robots robot] avec set Shape "car".

Breed [objets objet] avec set Shape "box".

Breed [cibles cible] avec set Shape "house".

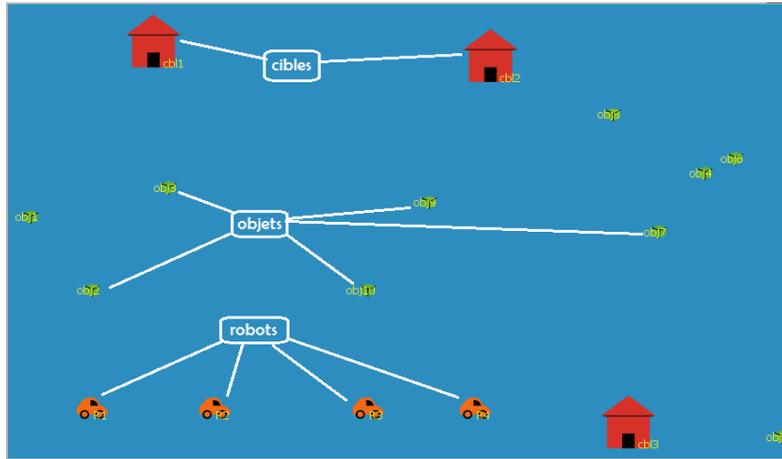


Figure 20 : Composant de l'environnement

6.3 Organiser les entités entrent-elles

C'est l'étape clé de ce passage inspiré, respectent les deux contraintes suivant :

- Un seul cran/degre d'héritage possible à partir de Turtle
- Pas de comportements attribués à une liste d'agents (AgentSet), ce sera à faire dans le programme principal.

Exemple :

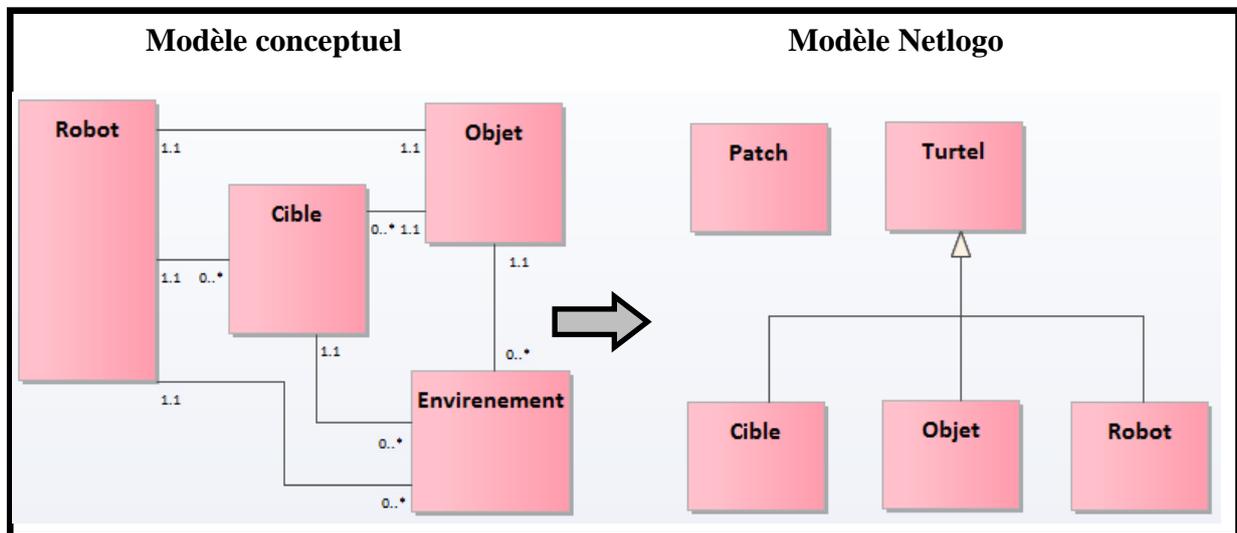


Figure 21 : Exemple de passage UML / NetLogo

Comme il souligne Amblard dans l'atelier [77], lorsqu'il n'existe pas un seul comportement possible à partir des Turtles (robots). Et lorsque il y a des fragmentations de l'ensemble des groupes de robots en plusieurs comportements, il devient nécessaire de maîtriser les attributs des différents robot-Turtle, ainsi que les procédures particulières exécutées par les différents comportements de ces robot-Turtle.

Également pour notre étude, nous distinguons plusieurs comportement pour nos robots mobiles qu'ils s'agissent non seulement d'une exploration de cible (respectivement d'objet), mais on joint l'effet des comportements robots sur les comportements objet (au moment où le robot porte objet, il y un comportement de groupe comme une seul entité ainsi que le comportement cibles. Il y a d'autre comportement, qu'on peut appeler comportement coopératifs, Présentant à titre d'exemple "robot-Turtle traceur de chemin" à partir d'une cible « rejoigne non plein », ou un comportement "effaceur du chemin" pour effacer les traces de planification à partir d'une cible « rejoigne plein ».

Pour rapprocher ces problèmes, juste après apprenne la multitude de comportement de nos robots-turtle, à l'égard des considérations et les préconisations recommandé par Amblard dans [77]. Nous avons proposé un prototype pour notre robot-Turtle coopérant, avec quatre attributs illustrés par la **Figure 22**. En plus de ces attributs que nous compléterons par quelques comportements liés exclusivement à ces quatre attributs de nos robots-Turtle coopérant.

Can_do ?	<i>La gestion de mobilité.</i>
Look_cib	<i>La gestion de coopération.</i>
Portez	<i>La gestion Allocation de tâche.</i>
State	<i>La gestion de Communication.</i>

Figure 22 : Prototype conçu pour notre robot -Turtle coopérant.

Le Tableau suivant, présente ces différents concepts utilisés :

Thèmes (Classes)		Attributs	Méthodes	Relation entre les thèmes (X et Y)
1	Robots	Etat_R, energie, champ de vision, Vitesse_R. poids	<i>déplacer ()</i> , <i>Maj_objets ()</i> , <i>Maj_cibles ()</i> .	<i>porter_objet</i> (1 et 2). <i>Depose_objet</i> (1 et 3). <i>Situer_dans</i> (1 et 4)
2	Objets	Etat_O, poids		<i>Porter_par</i> (2 et 1). <i>Situer_dans</i> (2 et 4). <i>Hébergé_dans</i> (2 et 3)
3	Cibles	Volume,		<i>Situer_dans</i> (3 et 4) <i>Atteigne par</i> (3 et 1) <i>Logé</i> (3 et 2)

4	Environnement	Max xcor		
		Max ycor		

Tableau 2 : Les différents thèmes pour analyser la tâche générique.

Pour la **Figure 22**, le sens des deux flèches de raisonnement et d'exécution exhibe la manière avec laquelle nos robots-Turtle coopérant réagissent pendant la réalisation de la tâche. Une manière qui exhibe par conséquence la hiérarchie globale de notre programme principale.

Attribut Robots	Valeur D'attributs		
State	"For_objet"		"For_cible"
	Work_Robot_For_Objets		Work_Robots_For_Cibles
Portez	Nobody		Not Nobody
	CIRCULER_ROBOT_NON_OCCUPE		CIRCULER_ROBOT_OCCUPE
Look_cib	0	1	-1
	MOVE_ROBOT_LIBRE	MOVE_ROBOT DIAGONALE_IN	MOVE_ROBOT_ DIAGONALE_OUT
Can_do	0	1	2
	Move_Atomique (Normale)	Move_Atomique (Lente)	Move_Atomique (Bloque)

Tableau 3 : Comportement pour les robots-turtles

Pour terminer cette partie, on peut dire que suivant ces quatre attributs, en plus de deux autres attributs, un pour les objets (porteur), et un pour les cibles (marqué). Que nous avons maîtrisé l'ensemble des comportements de nos robots-turtles.

Toutes les procédures concrétisant ces comportements sont directement codé avec le langage NetLogo (**Tableau 3**), dans la partie implémentation suivante.

7 Implémentation

Le diagramme d'activité suivant illustré par la **Figure 23** récapitule l'ossature globale de notre implémentation distingué.

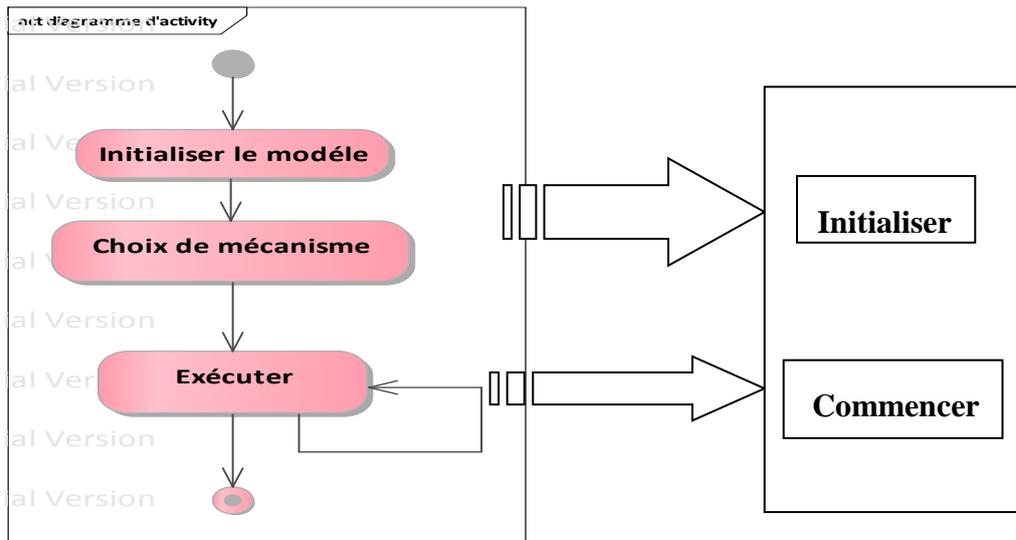


Figure 23 : Un Canevas pour notre programme principal.

Concernent le sujet '**initialiser**', reproduit l'étape d'initialisation des différents paramètres de notre modèle de choix de coopération développé. Par exemple pour notre tâche générique étudiée dans ce mémoire, il a été déraciné dans une fiche Netlogo à part (setup.nls). Et passent maintenant au deuxième corps de notre implémentation qu'il s'agit de la partie **exécuter**, ou de démarrer,

Dans notre travail exhibé par ce mémoire, on focalise majoritairement toute notre attention sur la planification/déplanification des chemins vers les cibles fixes, on essaye d'optimiser cette planification, par l'utilisation de l'environnement comme un moyen de communication. En conséquence est pour réaliser notre simulateur, nous avons adopté sept hypothèses :

Hypo1 : Chaque turtles (robot) à une connaissance inhérente au centre/coins de l'environnement (point (0.0), (-25.15) (25.15) (-25.-15) (25.-15)).

Hypo2: Chaque robot dispose des capteurs sur ces huit voisinages.

Hypo3: chaque robot sait évaluer (estimé) la distance euclidienne qui le sépare avec d'autres (objet, cible) dans l'intervalle (champ de visibilité).

Hypo4: Une fois le robot dépose un objet dans une cible non pleine, l'objet déposé devient vide (Nobody).

Hypo5 : la somme de volume de l'ensemble des cibles fixes dispersées, est supérieure ou égale à la somme des objets distribués dans l'environnement.

Hypo6 : L'objet à déplacer est présenté de telle sorte qu'un seul robot pourrait le déplacer tout seul.

Hypo7: l'acte coopératif des agents mobiles en terme traçage d'environnement, ou de communication entre agents mobiles ne consomme aucune énergie.

7.1 La problématique énergétique des robots-turtles

A ce stade la, nous avons fragmenté cette problématique en deux volets :

Avancement: C'est-à-dire, ou il y a un déplacement réel des robots d'un point (x y) vers un point (x', y'). D'après les lois physiques retrouvées ici [web7], L'énergie cinétique des robots libre correspond à l'énergie cinétique suivante :

$$\boxed{\text{Energie Robot Libre} = \frac{1}{2} * \text{poids de robot} * \text{vitesse robots}^2} \quad (01).$$

Tandis que le robot est occupé par un objet, le poids intégral devient égal à son poids élémentaire plus le poids de l'objet qu'il porte (qu'il déplace), alors :

$$\boxed{\text{Energie Robot Occupe} = \frac{1}{2} * (\text{poids de robot} + \text{poids de l'objet}) * \text{vitesse robots}} \quad (02).$$

Rotation:

Dans ce cas, il n'y a pas un déplacement réel, mais juste une rotation des robots dans un même point (x, y).

Energie Rotation Robot = 1/ 2 * Torque * vitesse angulaire.

Sachant que /

Torque = 1/2 * poid_robots * R², ou R = 1/2 rayon du robot

Vitesse angulaire = (2 * π) / D.

D : Temp nécessaire pour qu'un robot fait un Tour sur lui-même.

On suppose que : D = 1 seconde, R = 1, on trouve

$$\boxed{\text{Energie Rotation Robot Libre} = ((\pi * \text{poid_robots}) / 8)} \quad (3)$$

$$\boxed{\text{Energie Rotation Robot Occupe} = ((\pi * (\text{Poid_robots} + \text{poid_objets})) / 8)} \quad (4)$$

Les quatre équations (1, 2, 3 et 4) sont utilisées régulièrement pendant toute notre simulation pour calculer l'énergie dépensée par les robots durant la réalisation de la tâche générique sélectionnée.

Le simulateur et le Tableau de bord de simulation, ainsi qu'une petite description de l'ensemble des indicateurs attachés sont présentés dans le paragraphe suivant.

7.2 Présentation du simulateur "SC_SMA "

Un simulateur dédié à la coopération des robots–turtles avec plus d'une cible fixe hétérogène.

7.2.1 Le Tableau de Bord Global

1 INDICATEURS BASIQUES

NBR_ROBOTS : pour indiquer le nombre de robot au début de la tâche

NBR_OBJETS : pour indiquer le nombre d'objet au début de la tâche

NBR_CIBLES : pour indiquer le nombre de cible au début de la tâche

POID_ROBOTS : pour indiquer le poids des robots,

POID_OBJETS : pour indiquer le poids des objets à déplacer.

VITESSE_ROOTS : pour indiquer la vitesse de nos robots

VISIBILITE ROBOTS : pour indiquer la visibilité de nos robots

PAS_TEMP : pour qu'on crée notre propre horloge de simulation.

2 INDICATEUR AVANCE

MAX_TIME : pour imposer les contraintes de temps sur les robots

MAX_DISTANCE : pour imposer les contraintes de distance sur les robots

MAX_ENERGIE : pour imposer les contraintes d'énergie sur les robots

3 SIMULATION OPTION

GERER_CONTRAINTE : pour prendre en considération les contraintes imposées aux robots.

QUELLE_CONTRAINTES ? : Pour spécifier la contrainte imposée au robot, en terme d'énergie, de distance, de temps, etc

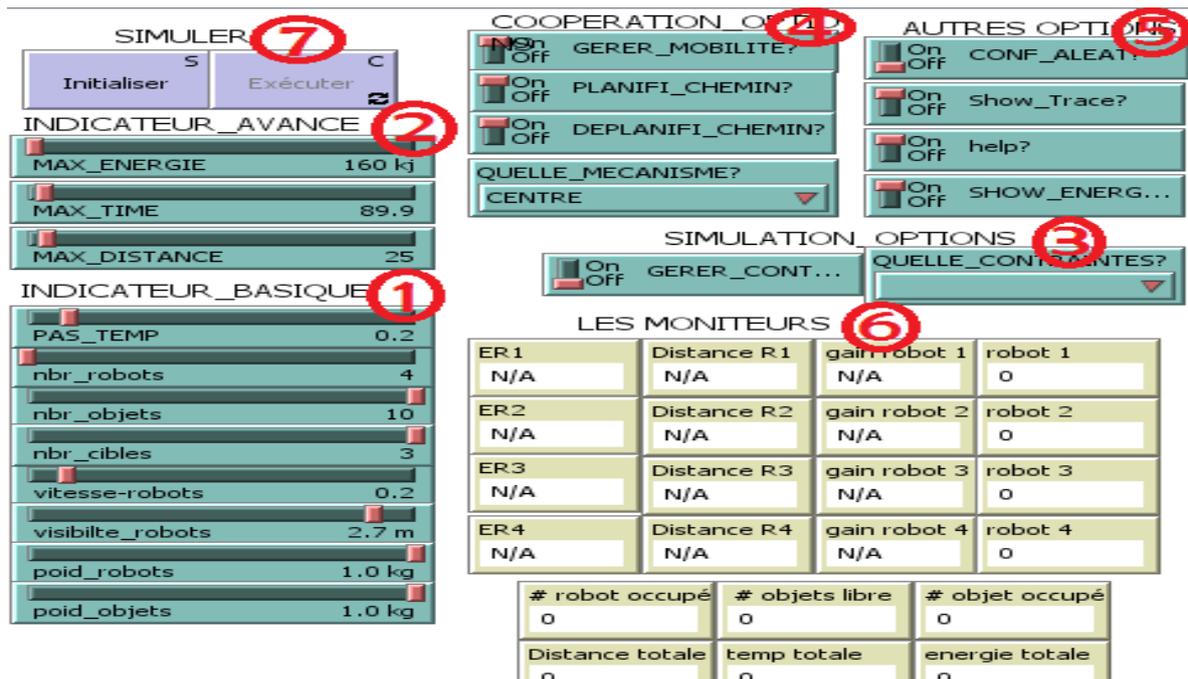


Figure 24 : Tableau de bord global de SC_SMA

4 COOPERATION OPTION

PLANIFI_CHEMIN : pour solliciter la coopération ou non.

DEPLANIFI_CHEMIN : pour solliciter la coopération ou non.

QUELLE_MECANISME ? : Pour spécifier le mécanisme imposée au robot, en terme de coopération.

GERER_MOBILITE : pour exiger à l'ensemble des robots de gérer leur mobilité ou Non, à la fin de la tâche.

5 AUTRES OPTION

SHOW_ENERGIE : pour afficher l'énergie dépensée par les robots ou non.

SHOW_TRACE : pour afficher les traces déposées par les robots ou non.

CONF_ALEAT : configuration aléatoire ou non.

6 LES MONITEURS

ER_i : Pour afficher en temps réel l'énergie consommée par le robot i.

DISTANCE R_i : pour afficher la distance parcourue par le robot i.

Gain robot R_i : pour indiquer le nombre de participation du robot i pour la planification ? Déplanification Du Chemin.

7 SIMULER

Initialiser : pour initialiser le simulateur

Commencer : pour lancer la simulation

7.3 Etude empirique

Dans cette exploration, nous essayons d'adopter trois mécanismes de coopération, auprès de tenir la politique ou le mécanisme la plus efficient pour accomplir efficacement la tâche en termes de temps et d'énergie.

Pour notre étude empirique, nous avons configuré quatre robots dans un environnement de taille 25x15 contenant (6, 8, et 10) objets, et trois cibles fixes.

NB : les simulations et l'ensemble des expérimentations, ont été réalisées sur un Micro Portable DELL avec un 8 GB de mémoire, et un processeur d'Intel (R) core (TM),i7-4500U CPU @ 1.80 GHz-2.40 GHz.

La **Figure** suivante représente une illustration de ces quatre mécanismes de coopération sur notre simulateur **SC_SMA** :

Mécanisme1 (MPD): mécanisme par défaut **(a)**.

Mécanisme2 (MC):Planification/déplanification des chemins, de cible vers le centre de l'environnement **(b)**.

Mécanisme3 (MC-): Planification et déplanification des chemins, de cible vers les coins plus proche **(c)**.

Mécanisme4 (MC+): Planification et déplanification des chemins, de cible vers les coins plus loin **(d)**.

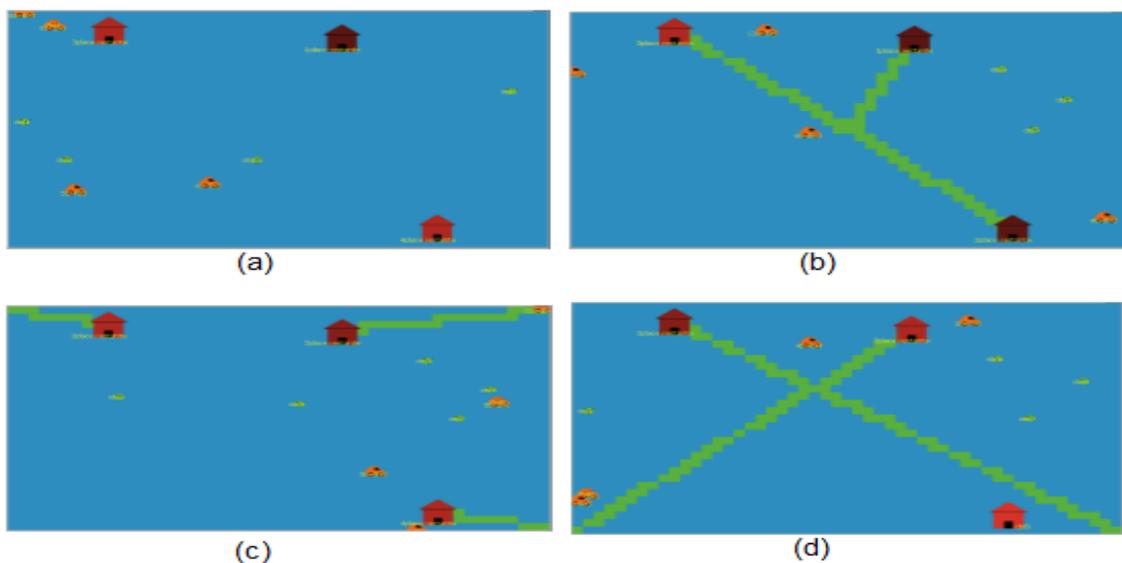


Figure 25 : Illustration des quatre mécanismes de coopération.

8. Expérimentation et résultats:

Après dix simulations, Nous avons conservés les résultats des **Tableau 4, 5, 6 et 7** retracés respectivement dans les trois histogrammes au-dessus (**Histogramme 1, Histogramme 2, et Histogramme 3**).

Mécanisme 1								
4 robots								
6 objets			8 objets			10objets		
temps	énergie	Distance	temps	Energie	Distance	temps	énergie	distance
62,4	365,961	1411,8	94,8	525,961	1652,4	92,7	463,185	1248,4
47,2	459,314	1569,314	102,8	629,805	1891,2	169,2	958,221	2421,4
103,9	327,958	1023,896	149,3	646,772	1403,4	67,5	674,053	1678
134,8	342,488	941,215	125,7	387,04	2001,6	146,8	520,421	1537,4
36,6	363,871	841,74	184	386,594	2288,6	105,8	827,765	2052,4
126,6	608,479	2268,781	148,8	563,065	848	79,3	523,736	1350,4
101,1	291,761	897,378	95,7	578,652	1843,6	103,2	560,481	1937
46	352,562	792,6	177,9	491,299	2404,8	96	534,847	2037,8
75,5	478,048	1400,724	127	363,511	1478,8	139,6	452,934	2541
96	373,744	1202,985	120,4	630,891	1397	97,1	556,727	1636,4
83,01	396,4186	1235,0433	132,64	520,359	1720,94	109,720	607,237	1844,020

Tableau 4 : Résultat de simulation de mécanisme 1 (MPD)

Mécanisme 2								
4 robots								
6 objet			8 objet			10 objet		
temps	energie	Distance	temps2	energie2	distance2	temps3	energie3	distance3
48,9	336,996	994,299	78,1	488,404	1606,545	71,2	491,03	1894,717
55,7	526,722	1632	56,9	542,094	1295,627	58,8	787,315	1714,126
43,1	363,498	1128,43	51,5	512,926	1334,543	64,2	571,475	1904,789
33,8	290,679	872,487	87,6	524,593	1335,726	83,7	536,027	2274,794
38	238,601	745,23	38,6	594,616	847,535	77,4	650,877	1414,126
41,2	433,013	1569,067	45,2	320,611	1291,943	51,3	439,359	1466,835
54,4	409,141	997,229	77,3	528,137	1776,699	75,4	576,932	1487,885
59,8	385,888	1094,857	64,2	816,826	1462,318	82,6	629,487	2207,804
94,09	264,912	826,439	83,7	430,281	1618,257	79	729,338	2213,239
76,2	316,642	887,494	98,7	402,996	1854,515	75,1	548,821	1226,444
54,519	356,6092	1074,7532	68,18	516,1484	1442,3708	71,87	596,0661	1780,4759

Tableau 5 : Résultats de la simulation selon le mécanisme (MC).

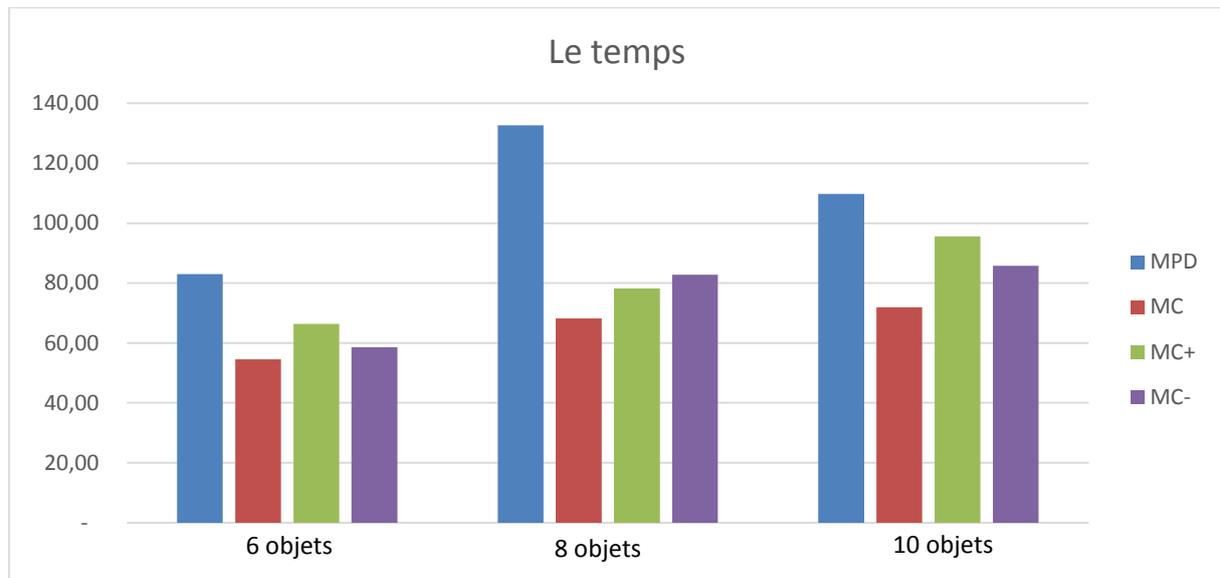
Mécanisme 3								
4 robots								
6 objets			8 objets			10objets		
temps	Energie	Distance	temps	énergie	distance	Temps	énergie	distance
58,7	439,689	1348,8	81,4	462,674	1822,854	94,6	703,652	1214,683
74,3	346,992	1075,6	145,3	435,588	1752,078	139,7	501,977	3656,805
43,8	360,467	1108,6	75,8	940,218	2042,598	74,2	707,124	2112,303
35,5	391,026	1069,6	66,6	581,552	1055,589	56,2	696,616	1590,707
124,9	298,471	975,4	38,2	353,455	1037,47	178,7	568,112	2587,705
92,5	505,543	1397,4	110,9	383,654	1970,818	56,8	639,658	1286,423
40,6	264,313	728,6	64,1	550,018	1728,214	83,9	588,879	1559,046
44,7	385,506	1307,2	54,9	445,392	1364,53	125,8	689,997	1641,461
84,7	388,155	1534,2	61,2	525,172	935,6	81	748,982	1361,557
63,6	378,539	1139,2	84	604,716	2371,144	64,6	455,396	1566,697
66,33	375,8701	1168,46	78,24	528,2439	1608,0895	95,55	630,0393	1857,7387

Tableau 6 : Résultat de la simulation selon mécanisme (MC+)

Mécanisme 4								
4 robots								
6 objets			8 objets			10objets		
temps	Energie	distance	temps	énergie	Distance	Temps	énergie	distance
96,2	537,463	1164,393	85,3	793,209	1862,264	149	724,478	937,259
87,29	458,107	1020,162	64,5	948,819	1912,45	74,4	925,388	2202,686
63,29	415,717	1112,921	100,2	721,68	1791,629	50,4	676,77	1451,299
40,1	391,049	1003,71	133,2	904,177	2490,904	95,4	990,388	2739,81
32,5	419,708	1093,107	96,4	597,021	1454,041	57,6	893,583	1763,811
68,9	378,715	1128,153	91,9	809,403	2284,643	75,1	665,368	1726,662
44,3	494,537	1324,534	82,7	823,709	1438,245	96,5	733,864	2427,565
46,9	466,695	1223,641	61	838,951	1709,669	36,2	460,349	1000,738
33,5	682,721	2181,542	71,3	773,259	1908,152	128,4	98,106	2908,305
72,7	554,933	1777,31	41,8	558,38	1223,852	94,3	945,827	1935,269
58,568	479,9645	1302,9473	82,83	776,8608	1807,5849	85,73	711,4121	1909,3404

Tableau 7 : Résultat de la simulation selon le mécanisme (MC-)

8.1 Concernent le temps



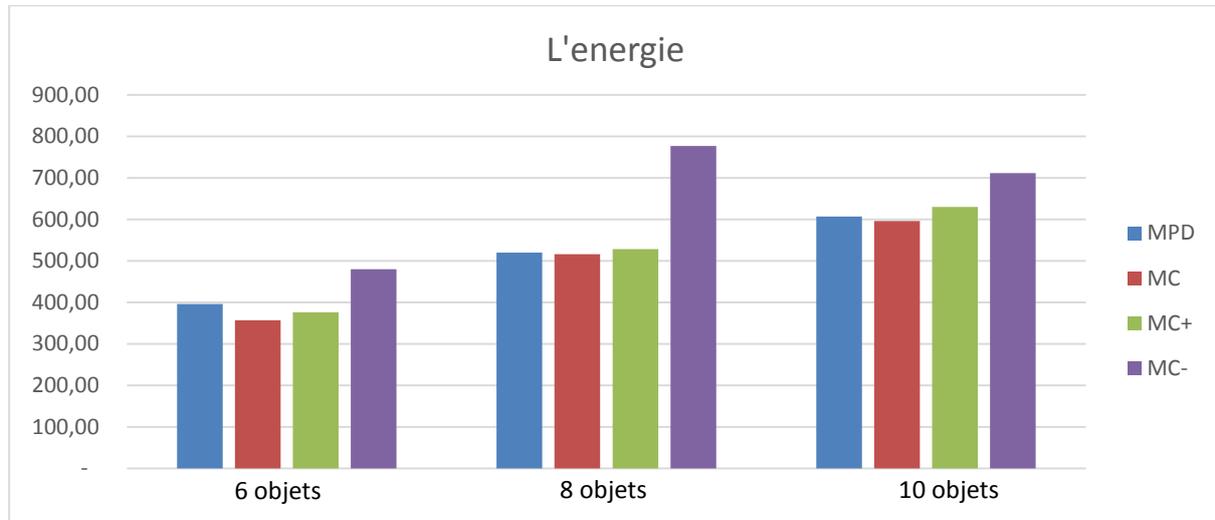
Histogramme 1 : Le facteur temps selon nbr objet+ mécanismes!

L'**histogramme 1**, démontre qu'avec un nombre d'objets quelconque, le mécanisme MPD prend assez de temps pour compléter la tâche, ce fait peut être confirmé par le temps consommé pratiquement important de l'ensemble des turtles-robots avec un nombre d'objet égal respectivement à six, huit, et dix.

Subséquentement examinons le deuxième mécanisme (MC), où il y a de stratégie de planification de chemin vers le centre, on observe un très bon impact sur le temps de la tâche par rapport au le troisième et le quatrième mécanisme (MC+ et MC-) ou il y une stratégie de planification du chemin vers les coins. **De ce fait:**

Récapitulatif_ 01: La performance de la coopération de nos robots en termes de temps dépend absolument de l'adoption d'une politique de planification de chemin vers le centre de l'environnement (**par/pours**) les robots.

8.2 Concernent l'énergie



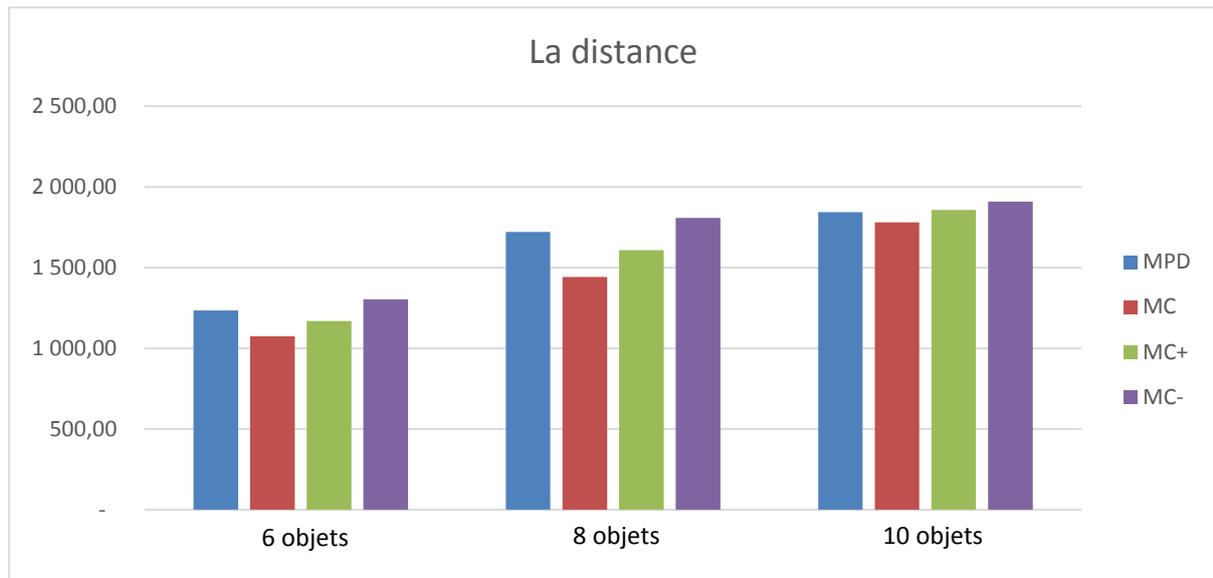
Histogramme 2 : Le facteur énergies selon nbr objet+ mécanismes

L'**histogramme 2** démontre, quel que soit le nombre d'objet, les mécanismes "MC- et MPD" prend assez du temps pour compléter la tâche par rapport aux autres mécanismes. Dans le cas où il y a (8;10) objets, presque les trois mécanismes de coopération influence négativement la tache de coopération en terme de coopération.

Cependant, nous remarquons que plus le nombre d'objet devient important, plus que la distinction entre le Mécanisme (MC-) et le mécanisme (MC) devient apparente, une apparence qui renforce la certitude que la politique trois (MC) est plus efficace en terme d'énergie que les trois autre politiques. **De ce fait,**

Récapitulatif_02 : La performance de la coopération en termes d'énergie dépend relativement de l'adoption associative d'une politique de test de mobilité, et de planification de chemin (fusion)

8.3 Concernent la distance



Histogramme 3 : Facteur distance selon nbr objet+ mécanismes

Postérieurement examinons les trois mécanismes ou il y a une stratégie de planification du chemin, on observe un très bon impact sur la distance de nos robots de la tâche avec le mécanisme "MC" par rapport à (3^{ème} /4^{ème}) mécanisme "MC-", où il y a de stratégie de planification du chemin vers les coins.

Dans le cas où il y a 4,6, et 10 objets, le mécanisme (MC) est éminent que Les autres mécanismes en termes de distance (MC- et MC+). De ce fait:

Récapitulatif_03 : La performance de la coopération en termes de distance parcouru dépend relativement de l'adoption d'un mécanisme de planification de chemin vers le centre.

Synthèse

En conséquence pour terminer notre sujet de master, et après les trois récapitulatifs antérieur, on peut résumer notre étude par le constat suivant:

Récapitulatif de l'étude : le meilleur mécanisme en terme de temps écoulé, énergie consommé et de distance parcourue c'est le deuxième mécanisme (MC), ou il y un mécanisme de planification de chemin vers le centre de l'environnement.

Conclusion

Ce qu'il faut retenir de ce dernier chapitre, c'est que la coopération entre un ensemble de robots au sein d'un environnement contraint (en termes de variété de cibles (volumes)), est une tâche complexe et paradoxale.

La difficulté s'aperçoit dans la question suivant: A qu'elle moment les robots coopèrent, avec toute la difficulté rencontrée en termes de contrainte de nos robots (accès concurrent) et surtout sur les cibles dans notre étude (distribution, capacité de stockage),

Profiter d'UML comme langage d'annotation (diagramme d'activité et de classe) et des plateformes de simulation SMA (NetLogo), devient un défi intéressant.

Dans ce refuge, la modélisation et simulation UML/SMA à travers un processus de développement inspiré, nous ont produit un support magique pour dégeler cette complexité inhérent à l'interaction (coopération) entre l'ensemble de ces robots, les rendre élémentaire pour la compréhension.

Dans ce chapitre, et après la présentation de notre simulateur, nous avons vu et proposé des mécanismes de coopération pour optimiser en termes de temps/énergie la réalisation d'une tâche générique.

Conclusion Générale & Perspectives

Conclusion Générale & Perspectives

La modélisation d'un système ou d'un phénomène, quel qu'il soit, permet de formaliser un problème afin de le rendre fiable ou de l'optimiser pour pouvoir l'appliquer dans un cas réel.

Nous avons décrit les travaux menés dans ce mémoire, dans le cadre de la modélisation par le paradigme agent avec un contexte multi-agent, afin de maîtriser la coopération entre un ensemble d'agent avec une faculté de mobilité.

L'objectif global de ce travail été de montrer l'efficacité quant à la fusion de l'approche multi-agents indiquée par le langage/platforme NetLogo, avec l'approche orientée objet reflétée par le langage 'UML', quant à modéliser la coopération entre un ensemble de robots mobiles, ces robots sont incarnés dans notre simulateur **SC_SMA**, par des robots mobile virtuelles,

-Le simulateur **SC_SMA** développé dévisage la coopération (Avec trois mécanismes) d'un ensemble de robot-mobile virtuelle, dans le but de déplacer un ensemble d'objet distribué, vers des cibles fixes disséminé dans l'environnement. Ces mécanismes de coopération ont été modélisée, programmée, afin d'illustrer par la planification de chemin vers les cibles fixes, et par la gestion de mobilité de nos robots mobiles virtuels.

Une grande partie de ce mémoire s'est concentré sur :

-La rédaction d'un état de l'art sur la modélisation et la coopération multi-agent, particulièrement la vision biologique à travers la colonie de fourmis.

-La description, l'explication, et la distinction entre les trois mécanismes de coopération inspirés dans un univers multi-agent.

-La présentation et l'apprentissage d'une plateforme multi-agent prodige nommé Netlogo.

-Le développement d'un simulateur pragmatique nommé **SC_SMA** dédié à la coopération d'un ensemble de robot-turtles mobile avec plusieurs mécanismes. Ce développement est issu d'un processus inspiré et bien détaillé dans ce mémoire, concernant une tâche générique sélectionnée. – la réalisation d'un ensemble d'expérimentation préliminaire sur le simulateur **SC_SMA** développé, pour montrer l'efficacité de la coopération entre les robots-turtles mobile employé.

Ces expérimentations ont donné des résultats prometteurs en termes d'énergie consommée par les robots virtuels et de temps écoulé ainsi que la distance parcourue pour réaliser la tache globale, quoique ces expérimentations soient assez restreintes (limités).

Cette limitation provient principalement de la taille de l'environnement, le nombre de robots-virtuels, la plateforme de simulation employés pendant les simulations effectuées, ces limitations influence négativement et considérablement la crédibilité de notre modélisation basée essentiellement sur des agents Netlogo. i.e le cadre proposé doit être testé avec un grand nombre de robots-virtuelle et dans un large environnement, et sur d'autre plateforme de simulation SMA, comme **Swarm**, **Repast**..Etc.

A court terme, les travaux les plus importants à réaliser concernent l'extensibilité du Simulateur **SC_SMA**, en termes de planification de chemin vers les objets à déplacer, et non pas seulement vers les cibles fixes. Nous envisageons d'étendre ce modèle, par l'adoption d'une stratégie de planification des chemins vers des objets, en plus en prenant en compte des contraintes sur les cibles fixes, i.e proposer des cibles mobiles. Comment les robots-turtles peuvent-ils maîtrisés la mobilité et la dynamique cibles?

Toujours à court terme, et comme notre premier modèle dans ce mémoire, est basé sur l'agent turtle (robots) comme un axe de modélisation, nous envisageons comme perspective, et pour la même tâche générique, de proposer d'autres modèles avec un autre axe conceptuel, c'est-à-dire, un deuxième modèle dirigé par les objets, un troisième modèle dirigé par les cibles, de telle sorte que notre modèle d'aujourd'hui (à base de robots), et les deux autres modèles ambitionné (à base de cible, à base d'objet) peuvent être comparés en termes de résultat de simulation (une comparaison entre ces trois modèles).

A moyen terme, et comme ce modèle reste incomplet et à parfaire. En perspective, il s'agira entres autres d'inclure d'autres styles de coopération, dans le but à moyen terme d'établir des études sur les approches de coopération basées sur l'apprentissage (qui a été brièvement expliquer dans l'état de l'art, afin d'appliquer ces approches de coopération à nos robots-virtuels.

Le travail présenté dans ce mémoire est une première pierre concernant une démarche flexible consacrée à la maîtrise de la coopération entre un ensemble d'agents mobiles (robots) pour déplacer un ensemble d'objets, afin d'atteindre un ensemble de cible fixes (**mobiles!**), Il peut se décliner en différentes extensions, que la durée d'un mémoire n'a pas permis d'aborder, et qui constituent autant de pistes de recherche à explorer.

Bibliographies

Bibliographies

- [1] Selma AZAIEZ "Approche dirigée par les modèles pour le développement de systèmes multi-agents", thèse de doctorat 11 décembre 2007, université de Savoie.
- [2] Chadli - Université Ibn Khaldoun, Tiaret - Département d'informatique--Cours Systèmes Multi-Agents.
- [3] Ferber Jacques "Les systèmes multi-agents : vers une intelligence collective" InterEditions - 1995 - ISBN : 2-72960-572-x.
- [4] Georgeff, M., Pell, B., Pollack, M., Tambe, M. and Wooldridge, M., 1999 The Belief Desire-Intention model of agency Proceedings of Agents, Theories, Architectures and Languages (ATAL).
- [5] Systèmes Complexes Adaptatifs Application au traitement des images, Université Mentouri Constantine Faculté des Sciences de l'Ingénieur, 2006
- [6] Fishwick, P. (1997) Computer simulation: growth through extension. Transactional of the societing for computer simulation international.
- [7] Benoudina Lazhar (2009), modélisation et simulation basées multi-agents du contrôle de processus industriels, Thèse de Magister en informatique de l'école Doctoral de l'informatique INI, Université 20 aout 1955 Skikda.
- [8] Labidi, S., Lejouad, W. (1993) ' De l'intelligence artificielle distribuée aux systèmes multi agents' INRIA N° 2004.
- [9] Jarras, I. et Chaib-Draa, B. (2002) Apperçu sur les systems multiagents Série scientifique du centre inter universitaire de recherché en analyse des organizations CIRRANO.
- [10] Ferber Jacques "coopération réactive et émergente" Intellectica-1994
- [11] Demazeau Yves, Müller Jean-Pierre "Decentralized Artificial Intelligence" Proceedings of the First European Workshop on Modelling Autnomous Agents in a Multi-Agent World - Cambridge, England, August 16-18, 1989 Elsevier Science Publishers 1990 - ISBN : 0-444-88705-9
- [12] Ferber Jacques "Les systèmes multi-agents : vers une intelligence collective" InterEditions - 1995 - ISBN : 2-72960-572-x

- [13] Brassac Christian, Pesty Sylvie “‘La Pelouse Fourmilière’. De la coaction à la coopération” Quatrièmes journées francophones sur l'Intelligence Artificielle Distribuée & Systèmes Multi- Agents, Port Camargue, 1-3 avril 1996, p. 251 – 264 Editions Hermès - 1996 - ISBN : 2-86601-528-2
- [14] Lenay Christian “Coopération et intentionnalité” Quatrièmes journées francophones sur l'Intelligence Artificielle Distribuée & Systèmes Multi-Agents, Port Camargue, 1-3 avril 1996, p. 265 – 272 Editions Hermès - 1996 - ISBN : 2-86601-528-2
- [15] Haton Jean-Paul “Définition de mécanismes de coopération” Bulletin de l'AFIA - n° 25 - avril 1996 - p.17
- [16] Moraitis P. “Paradigme Multi-Agent et Prise de Décision Distribuée” Thèse de Doctorat, Université Paris-Dauphine, 1994
- [17] Lounis Adouane, "architectures de contrôle comportementales et réactives pour la coopération d'un groupe de robots mobiles ". Thèse de doctorat, université de Franche-Comté .2005
- [18] François Klein,"contrôle d'un système multi-agents réactif par modélisation et apprentissage de sa dynamique globale". Thèse de doctorat à l'université de Nancy 2009. Page 23..54.
- [19] Brahim Chaib-draa Sébastien Paquet, « Apprentissage de la coordination entre agents dans unenvironnement temps-réel complexe» proposition d'un sujet de doctorat, Départementd'informatique et de génie logiciel Université Laval, SteFoy, Québec, Canada Mai 2003.
- [20] Springer Lecture Notes in Computer Science, Vol. 7154, R. Ramanujam and S. Ramaswamy(Eds.), pgs. 35-49, Proceedings of 8th International Conference on Distributed Computing andInternet Technology (ICDCIT 2012), Bhubaneswar, India, Decision Making as Optimization in Multi-Robot Teams Feb. 2-4, 2012.
- [21] T. Finin, R. Fritzson, D. McKay, and R. McEntire, "KQML as an agent communication language",In Third international conference on information and knowledge management. ACM Press,Novembre 1994.
- [22] François Legras "Coopération d'Agents et Systèmes d'Information " - support de coursENST Bretagne, 2006

- [23] R.G. Smith. "The Contract Net Protocol: High-Level Communication and Control in a DistributedProblem Solver". IEEE transactions on computers Vol. C-29, No. 12, 1104-1113. 1980.
- [24] Tarek Jarraya, Réutilisation des protocoles d'interaction et Démarche orientée modèles pour le développement multi-agents, Thèse de Doctorat en informatique, l'Université de Reims, 2006.
- [25] Valérie CAMPS, " vers une théorie de l'auto-organisation dans les systèmes multi-agents basée surla coopération " : application à la recherche d'information dans un système d'information répartie.Thèse de doctorat, université Paul Sabatier, 1998.
- [26] Philip R. Cohen, Jerry Morgan, Martha E. Pollack, 1990, *Intentions in Communication*, MIT Press.
- [27] Michael E. Bratman, 1990, What Is Intention? dans "Intentions in Communication", MIT Press, pp 15-31.
- [28] Alexis Drogoul, Jacques Ferber, Ethomodeling : a Multi-Agent Behavioral Simulation Model, Université Paris 6, LAFORIA, rapport 02.92, 1992.
- [29] Philip R. Cohen, Hector J. Levesque, 1990, Intention Is Choice with Commitment, *Artificial Intelligence* 42, pp 213-261.
- [30] Edmund H. Durfee, Thomas A. Montgomery, 1991, Coordination as Distributed Search in aHierarchical behavior Space, *IEEE Trans. on Systems, Man, and Cybernetics*, special issue onDistributed A.I.
- [31] Springer Lecture Notes in Computer Science, Vol. 7154, R. Ramanujam and S. Ramaswamy(Eds.), pgs. 35-49, Proceedings of 8th International Conference on Distributed Computing andInternet Technology (ICDCIT 2012), Bhubaneswar, India, Decision Making as Optimization in Multi-Robot Teams Feb. 2-4, 2012.
- [31] Mataric Maja J. "Interaction and Intelligent Behavior" PHD of Philosophy Massachusetts Institute of Technology May 1994
- [32] Genesereth Michael R., Ketchpel Steven P. "Software Agents" *Comm. of the ACM* - vol.37 - 7 July 1994
- [33] Etzioni Oren, Weld Daniel A Sofbot-Based Interface to the Internet. *Communications of the ACM* - vol.37 - July 1994

- [34] Bouron Thierry “Structure de communication et organisation pour la coopération dans un univers multi-agents” Thèse de doctorat de l’université Paris 6 - Novembre 1992
- [35] Sekaran Mahendra, Sen Sandip “To help or not to help” Seventeenth Annual Cognitive Sciences Conference July 22-25 - 1995 - Pitsburg Pennsylvania
- [36] Sen Sandip “Reciprocity : a foundational principle for promoting cooperative behavior among self-interested agents” in Proc. of the Second International Conference on Multi-Agent Systems - p. 322-329 – AAAI Press - Menlo Park, CA, 1996
- [37] Eberhart, R., Kennedy, J., and Shi, Y. (2001). *Swarm Intelligence. Evolutionary Computation*. Morgan Kaufmann.
- [38] Choo, S.-Y. (2000). *Genetic Algorithms and Genetic Programming at Stanford 2000*, chapter Emergence of a Division of Labour in a Bee Colony, pages 98-107. Stanford Bookstore, Stanford, California.
- [39] Panta, L. (2002). *Modeling Transportation Problems Using Concepts of Swarm Intelligence and Soft Computing*. PhD thesis, Virginia Tech, Blacksburg, Virginia.
- [40] De Wolf, T., Liesbeth, J., Holvoet, T., and Steegmans, E. (2002). A Nested Layered Threshold Model for Dynamic Task Allocation. In Dorigo, M., Di Caro, G., and Sampels, M., editors, *Proceedings of the Third International Workshop on Ant Algorithms (ANTS’2002)*, volume 2463 of *Lecture Notes in Computer Science*, pages 290-291, Brussels, Belgium. Springer Verlag.
- [41] Nouyan, S. (2002). Agent-Based Approach to Dynamic task Allocation. In Dorigo, M., Di Caro, G., and Sampels, M., editors, *Proceedings of the Third International Workshop on Ant Algorithms (ANTS’2002)*, volume 2463 of *Lecture Notes in Computer Science*, pages 28-39, Brussels, Belgium. Springer Verlag.
- [42] E Bonabeau et G. Theraulaz G., *L’intelligence en essaim, pour la science*, 282 (3): pp. 66-73, N° 271 mai 2000.
- [43] Monmarché N. (2000). *Algorithmes de fourmis artificielles : applications à la classification et à l’optimisation*. Thèse de Doctorat de l’université de Tours. Discipline : Informatique. Université François Rabelais, Tours, France, 231 p.
- [44] E. Bonabeau, M. Dorigo, G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, NEW York, Oxford University Press, 1999.

- [45] Dorigo M., Maniezzo V., Colomi A., (1991) “The Ant System: An Autocatalytic Optimizing Process”, Technical Report No. 91-016 Revised, Politecnico di Milano, Italy.
- [46] Dorigo M., Gambardella L., (1996). “A Study of some properties of Ant-Q” .In Proceedings of PPSN IV– Fourth International Conference on Parallel Problem Solving from Nature, pp. 656–665.
- [47] Dorigo M., Blum C., (2005). “Ant colony optimization theory: A survey”, Theoretical Computer Science, Vol. 344, pp. 243 – 278.
- [48] Dréo J., Pétrowski A., Siarry P., Taillard E., (2003). “Métaheuristiques pour l'optimisation difficile”. Editeur : Eyrolles.
- [49] Dorigo M., Krzysztof S., (2006). “An Introduction to Ant Colony Optimization”, IRIDIA Technical Report Series.
- [50] Dorigo M., Maniezzo V., Colomi A., (1996). “The Ant System: Optimization by a colony of cooperating agents”, IEEE Transactions on Systems, Man, and Cybernetics–Part B, Vol.26, No.1, pp.1-13.
- [51] Gambardella L. M., Dorigo M., (1995). “Ant-Q: A Reinforcement Learning approach to the traveling salesman problem”, Université Libre De Bruxelles.
- [52] Dorigo M., Gambardella L., (1997) Ant colony system: A cooperative learning approach to the travelling salesman problem, IEEE Transactions on Evolutionary Computation, Vol.1, N°1, pp.53–66.
- [53] Stützle T., Hoos H. H., (2000). “MAX – MIN Ant system”, Future Generation Computer Systems, Vol.16, N° 8, pp. 889–914.
- [54] García-Martínez C., Cordón O., Herrera F., (2007). “A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP”, European Journal of Operational Research, Vol. 180, pp. 116–148
- [55] Barán B., Schaerer M., (2003). “A multiobjective ant colony system for vehicle routing problem with time windows”, In Proceedings of the 21st IASTED international conference on applied informatics, pp. 97–102. Calgary: ACTA Press.
- [56] Doerner K., Hartl R.F., Teimann M., (2003). “Are COMPETants more competent for problem solving?-The case of full truckload transportation”, *Central European Journal of Operations Research*, Vol. 2, pp. 115–141.

- [57] Gravel M., Price W. L., Gagné C., (2002). “Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic”, *European Journal of Operational Research*, Vol. 1, pp. 218–229
- [58] Mariano C.E., Morales E., (1999). “MOAQ: An Ant-Q algorithm for multiple objective optimization problems”, in: W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Hnavar, M. Jakiela, R.E. Smith (Eds.), *Proc. of the Genetic and Evolutionary Computing Conference (GECCO 99)*, San Francisco, California, USA, July, pp. 894–901.
- [59] McMullen P.R., (2001). “An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives”, *Artificial Intelligence in Engineering*, Vol. 15, N° 3, pp. 309-317.
- [60] Bauer A., Bullnheimer B., Hartl R.F., Strauß C., (1999). “An Ant Colony Optimization Approach for the Single Machine Total Tardiness Problem”, *In CEC99: Proceedings of the Congress on Evolutionary Computation*.
- [61] Doerner K., Gutjahr W. J., Hartl R. F., Strauss C., Stummer C., (2004). “Pareto Ant Colony Optimization: A metaheuristic approach to multiobjective portfolio selection”, Kluwer Academic Publishers, *Annals of Operations Research*, Vol. 131, pp. 79–99.
- [62] Gambardella L., Taillard E., Agazzi G., (1999). “MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows”, in: D. Corne, M. Dorigo, F. Glover (Eds.), *New Ideas in Optimization*, McGraw-Hill, pp.73–76.
- [63] Iredi S., Merkle D., Middendorf M., (2001). “Bi-criterion optimization with multi colony ant algorithms”, *Lecture Notes in Computer Science (LNCS)*, Vol. 1993/2001, pp. 359–372.
- [64] Cardoso P., Jesus M., Marquez A., (2003). MONACO – « Multi-Objective Network Optimisation Based on an ACO », Proc. X Encuentros de Geometria Computacional, Seville, Spain, June 16-17.
- [65] López-Ibáñez M., (2004). *Multi-objective ant colony optimization*, Master’s thesis, Darmstadt University of Technology.
- [66] Angus D., Woodward C., (2009). “Multiple objective ant colony optimization”, *Swarm Intelligence*, Vol. 3, N° 1, pp. 69-85.

- [67] Guntsch M.G., Middendorf M., (2003). “Solving multi-criteria optimization problems with population-based ACO”, Lecture Notes in Computer Science, Vol. 2632/2003, pp. 464-478.
- [68] Merkle, D., Middendorf, M., and Schmek, H. (2000). Ant Colony Optimization for resource-constrained project scheduling. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000), pages 839{900, San Francisco, CA. Morgan Kaufmann Publishers.
- [69] Gambardella, L. and Dorigo, M. (2000). Ant Colony System hybridized with a new local search for the sequential ordering problem. *INFORMS Journal on Computing*, 12(3) :237-255.
- [70] Gambardella, L., Taillard, E., and Agazzi, G. (1999). New Ideas in Optimization, chapter MACS-VPTW : A multiple ant colony system for vehicle routing problems with time windows, pages 63{67. McGraw Hill, London, UK.
- [71] Di Caro, G. and Dorigo, M. (1998b). AntNet: Distributed stigmergic control for communications networks. *Journal of Artificial Intelligence Research*, 9 :317-365.
- [72] Selma AZAIEZ Approche dirigée par les modèles pour le développement de systèmes multiagents, thèse de doctorat 2009, université de Savoie.
- [73] Ilias Sakellariou¹, Petros Kefalas², and Ioanna Stamatopoulou² "Enhancing NetLogo to Simulate BDI Communicating Agents" ¹ Department of Applied Informatics, University of Macedonia, Thessaloniki,²Department of Computer Science, CITY College, Thessaloniki, Greece J. Darzentas et al. (Eds.): SETN 2008, LNAI 5138, pp. 263–275, 2008. Copyright Springer-Verlag Berlin Heidelberg 2008.
- [74] Vidal, J.M., Buhler, P., Guardia, H.: The past and future of multiagent systems. In: Proceedings of1st AAMAS Workshop on Teaching Multi-AgentSystems (2004)
- [75] Wooldridge, M.: An Introduction to MultiAgent Systems. J. Wiley & Sons, Chichester (2002).
- [76] Modélisation et simulation économiques. Chapitre 3 – Introduction à la plateforme NetLogo/Murat Yıldızoglu <http://yildizoglu.info> Université Bordeaux IV GRE_A (UMR CNRS 5113).

- [77] F.Amblard, N.Marileau «Modélisation multi-agents appliquée aux phénomènes spatialisés»Atelier 2 : Passage du modèle conceptuel UML à Netlogo La Vieille Perrotine, Saint-Pierred'Oléron 21-26 juin 2009.
- [78] David.Sheeren, "introduction à la modélisation UML", Atelier 1, MAPS : modélisationmultiagents appliquée aux phénomènes spatialisés, la vieille Perrotine, Saint-pierre d'Orléon 21-26 JUIN 2009.
- [79] Gauthier picard," Méthodologie de développement de systèmes multi-agents adaptatifs etconception de logiciels à fonctionnalité émergente ", thèse de doctorat, Université Paul Sabatier deToulouse III, 2004.
- [80] Lounis Adouane. " MiRoCo un simulateur pour systèmes multi-robots à forte Dynamiqued'interaction". LASMEA, UMR CNRS 6602 24 Avenue des landais, 63177 Aubière Cedex,France. Lounis.Adouane@lasmea.univ-bpclermont.fr, 2005.
- [81] Lounis Adouane, "architectures de contrôle comportementales et réactives pour la coopération d'un groupe de robots mobiles ". Thèse de doctorat, université de Franche-Comté .2005
- [82] Yamaguchi H., Arai T., Beni G.,«A distributed control scheme for multiple robotic Vehicles tomake group formations », *Robotics and Autonomous Systems*, vol. 36, 2001, pp.125-147.
- [83] Lounis Adouane. " MiRoCo un simulateur pour systèmes multi-robots à forte Dynamiqued'interaction". LASMEA, UMR CNRS 6602 24 Avenue des landais, 63177 Aubière Cedex,France. Lounis.Adouane@lasmea.univ-bpclermont.fr, 2005.
- [84] Simonin.O, 'Le modèle satisfaction-altruisme:coopération et résolution de conflits entreagents situés réactifs, application à la robotique',PhD thesis,Université Montpellier II, Déc. 2001.

Webographie

[web1] <http://www.fipa.org>(Consulté le 04/05/2017)

[web2] http://fr.wikipedia.org/wiki/Acte_de_langage. (Consulté le 29/05/2017)

[web3] [http://ccl.du.nord-ouest.edu / Netlogo / télécharger.shtml](http://ccl.du.nord-ouest.edu/Netlogo/téléchargez.shtml). (Consulté le 1/05/2017)

[web4] [http://ccl.du.nord-ouest.edu / netlogo](http://ccl.du.nord-ouest.edu/netlogo). (Consulté le 5/05/2017)

[web5] [http://ccl.du.nord-ouest.edu / Netlogo / télécharger.shtml](http://ccl.du.nord-ouest.edu/Netlogo/téléchargez.shtml). (Consulté le 4/05/2017)

[web6] <http://www.01net.com/telecharger/windows/Programmation/langage/fiches/9345.html>
(Consulté le 28/03/2017)

[web7] [http://www.msrlu/mmp/online/website/content/fiches pédagogiques/physique/
104/23_FR.html](http://www.msrlu/mmp/online/website/content/fiches_pedagogiques/physique/104/23_FR.html).(consulté le 12/05/2012)