



République Algérienne Démocratique Populaire



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université d'Ibn Khaldoun – Tiaret

Département Informatique

Thème

**Développement des mécanismes et des stratégies a
base des SMA pour l'optimisation d'une circulation
routière**

Pour l'obtention du diplôme de Master II

Spécialité : Génie Informatique

Option : Système d'Information et Technologie Web

Dirigé par:

Mr HATTAB Nouredine

devant le jury:

Mr. ZIOUAL Tahar

Mr. BOUDAA Boudjemaa

Réalisé par :

KADI Nedjma

THAMER Nour el houda

Année universitaire : 2016-201

Dédicace

Je dédie ce travail à ...

A ma très chère mère Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

Votre prière et votre bénédiction m'ont été d'un grand secours pour mener à bien mes études.

Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

*A mon défunt père **الله يرحمه** Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous.*

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

A mes tendres frères et sœurs, pour tant de confiance, d'amour, de patience et d'abnégation.

*A ma chère Binôme **THAMER Nour el houda**, ainsi qu'à toute sa famille. Pour toutes les peines et difficultés partagées ensemble lors de l'élaboration de ce projet. A tous mes amis et en particulière **MEBKHOUT Djouhar**, **BELALIA Kamel** et **KIRIA Amine**.*

A tous les étudiants de notre promotion et à tous les gens qui m'ont aidé et soutenu tout au long de ce projet.

Kadi Nedjma

Dédicace

Je dédie ce modeste travail à ...

A mes très chers parents pour m'avoir donnés le goût aux études et m'avoir apportés un grand support moral lors de la rédaction de ce mémoire et tout au long de mes études, qu'ils trouvent ici l'expression de mon profond amour et que la réussite de ce projet soit la récompense de leurs efforts

A mes tendres frères et sœurs, pour tant de confiance, d'amour, de patience et d'abnégation.

*Il serait ingrat de ne pas songer à ma chère Binôme **KADI Nedjma** et toute sa famille*

*Je ne me permettrais surtout pas d'oublier ma très chère amie **MEBKHOUT Djouhar, Malika, BELALIA Kamel et KIRIA Amine.***

A tous mes collègues que j'ai eu l'occasion de connaître au cours de mes études. Que tous ceux qui se trouveraient omis ne me tiennent pas rigueur mais qu'ils trouvent ici l'expression de ma profonde reconnaissance.

A tous les gens qui m'ont aidé et soutenu tout au long de ce projet Et à tous ceux que j'aime. .

.

Remerciements

Au terme de deux ans d'études en master, il est une obligation de dire Merci à tous ceux qui ont contribué à la réalisation de ce travail.

Tout d'abord, nous remercions le Dieu, notre créateur de nos avoir donné les forces, la santé, la volonté et le courage afin d'accomplir ce travail.

*Je tiens à remercier et à manifester ma reconnaissance particulièrement à **Mr. HATTAB Nouredine**, d'une part pour m'avoir donné l'opportunité de participer à ce projet, qui m'a donné une capacité de recherche et d'adaptation.*

D'autre part d'avoir accepté d'être mon encadreur, avec un suivi constant et un intérêt démontré tout au long de mon travail.

Nous exprimons nos reconnaissances à tous les enseignants qui ont participé à notre formation depuis le début de nos études universitaires jusqu'au jour d'aujourd'hui. Nous remercions particulièrement les membres du Jury qui nous font honneur en acceptant d'examiner et de juger notre travail.

A toutes les personnes qui ont contribué de près ou de loin à l'aboutissement de ce travail, moralement ou matériellement,

Trouvent ici l'expression de notre gratitude et nos remerciements les plus sincères.

Résumé

Récemment la modélisation à base d'agent est en réelle effervescence, cette effervescence est due au nombre très important de chercheurs qui utilisent ce paradigme. L'idée de base de cette mode de pensée est récapitulée par cette précepte 'tout composant est un agent', derrière ce paradigme, n'importe quel système (phénomène) est modélisé par un ensemble d'entités (agents) qui le composent et qui interagissent entre eux.

De nos jours, de nombreux travaux de recherche à travers le monde portent sur des problèmes de plus en plus complexes. Parmi ces problèmes, nous nous intéressons dans ce mémoire aux problèmes de la circulation routière. La motivation vient du fait que ce secteur nécessite de nouvelles modèles, méthodes, et applications permettent de contrecarrer l'augmentation des problèmes issus de la circulation routière.

Le but ultime de notre mémoire consiste à modéliser une circulation routière basique avec un langage de modélisation basé sur les agents, en tenant compte des contraintes environnementales de la tâche générique étudiée. Nous avons adopté pour ça une approche UML/NetLogo afin de décrire les principales fonctionnalités de notre circulation routière. Finalement, nous avons implémenté notre approche pour la valider expérimentalement.

Mots clés : Circulation routière, simulation, modélisation, système multi-agents, NetLogo.

Abstract

Recently the agent-based modeling is in real effervescence, this effervescence is due to the very large number of researchers who use this paradigm. The basic idea of this way of thinking is summarized by this precept 'every component is an agent', behind this paradigm, any system (phenomenon) is modeled by a set of entities (agents) that compose it and which interact with one another.

Today, many research projects around the world are dealing with increasingly complex problems. Among these problems, we are interested in the problems of road traffic.

The motivation is that this sector requires new models, methods, and applications to counteract the increase in problems arising from road traffic.

The ultimate goal of our paper is to model a basic road traffic with an agent-based modeling language, taking into account the environmental constraints of the generic task studied. We adopted a UML / NetLogo approach to describe the main functionalities of our road traffic.

We adopted a UML / NetLogo approach to describe the main functionalities of our road traffic.

Keywords: road traffic simulation, modeling, multi-agent system, NetLogo.

Table de matière

Dédicaces

Remerciements

Résumé

Abstract

Table de matières

Liste des figures

Liste des histogrammes

Liste des tableaux

Liste des abréviations

Introduction général

1. Contexte	1
2. Problématique	2
3. Objectif de travail.....	2
4. Organisation de travail	3

Chapitre I : Les paradigmes de modélisation

1. Introduction	4
2. Qu'est-ce qu'un paradigme	4
3. Modèle et modélisation.....	4
3.1 La modélisation numérique	5
3.2 La modélisation analogique	5
3.3 Principes de la modélisation.....	5
4. Les paradigmes de modélisation.....	5
4.1 Entité.....	6
4.2 Orienté objet	6
4.2.1 Agents	6
4.2.1.1 Caractéristiques des agents	7
4.2.1.2 Types d'agent	8
4.2.1.3 Fonctionnement d'un agent	9
4.2.2 Système multi agent SMA	10

4.2.2.1	Avantage de l'utilisation des SMA.....	10
4.2.2.2	Architecture des SMA	11
4.2.2.3	Environnement.....	12
4.2.2.4	Interactions entre agents	13
4.2.2.5	Communication entre agents	13
4.2.2.6	Domaines d'application des S.M.A.....	15
5.	Conclusion.....	16

Chapitre II : Etat de l'art sur la circulation routière

1.	Introduction	17
2.	Circulation routière.....	17
2.1	Composantes de la circulation routière	17
2.2	Variables élémentaires du trafic routier	18
2.2.1	L'écart du temps inter-véhiculaire	18
2.2.2	Le débit.....	19
2.2.3	La concentration (ou la densité).....	19
2.3	La congestion routière	20
2.3.1	Pourquoi y a-t-il des bouchons sur les routes ?.....	20
2.3.1.1	Une demande	21
2.3.1.2	Une offre	21
2.3.2	Conséquences de la congestion du trafic routier.....	21
2.4	Les méthodes de gestion.....	22
2.4.1	Code de la route	23
2.4.1.1	Règles de priorité	23
2.4.1.2	Les panneaux de signalisation	24
2.4.1.3	Les feux de circulation tricolores	24
2.5	Approche dynamique.....	26
2.5.1	Contrôle dynamique par l'infrastructure.....	26
2.5.2	Quatre approches de référence	27
2.5.2.1	Limites des approches de référence	30
2.6	Synthèse des approches	31
2.7	Approche pour véhicules autonomes (IAV).....	32
2.7.1	Négociation véhicule à véhicule	32
2.7.2	Approche par réservations.....	33

3.8 Résumé des solutions citées.....	34
3. Conclusion.....	36

Chapitre III : NetLogo &UML « Aperçu et Initiation »

1. Introduction	37
2. Simulation multi-agent.....	37
2.1 Principe de la simulation multi-agents du trafic routier.....	38
2.2 Outils de développement des SMA.....	38
3. Choix de la plate-forme NetLogo.....	38
3.1 Présentation générale.....	39
3.2 Caractéristiques	40
3.3 La classification des agents	41
3.4 La mise en œuvre de la simulation.....	42
3.5 Modèle Netlogo.....	42
3.5.1 Interface graphique	42
3.5.2 Déclaration.....	43
3.5.3 Procédure Setup.....	43
3.5.4 Procédure GO	43
3.5.5 Procédures complémentaires	44
4. Le Langage NetLogo et langage UML	44
4.1 Atelier 1 [16]	44
4.2 Atelier 2 [17]	47
4.2.1 Les étapes de la démarche [17].....	48
5. Limites de NetLogo.....	49
6. Conclusion.....	50

Chapitre IV : Simulateur de Circulation "SimCirc"

Démarche & Implémentation & Expérimentation

1. Introduction	51
2. Cycle de Vie d'un Logiciel Multi agent.....	51
2.1 Tâche générique.....	52
2.1.1 Description de la tache générique	52
2.2 Analyse de la tâche.....	52
2.2.1 Voiture:	52

2.2.2	Panneaux de signal.....	53
2.2.3	Passages piétons.....	53
2.2.4	Piétons.....	53
2.3	La conception.....	54
2.3.1	Les mécanismes développés.....	54
2.4	La Démarche.....	57
2.4.1	Identification des variables globales.....	57
2.4.2	Identification des propriétés des objets à modéliser.....	58
2.4.3	Organiser les entités entre elle.....	58
2.5	. Implémentation.....	59
2.5.1	Présentation du simulateur « SimCirc ».....	60
3.	Expérimentation.....	64
3.1	L'effet de mécanisme de panneaux de signalisation sur la circulation.....	64
3.2	Interprétation des résultats.....	65
4.	Conclusion.....	67
	Conclusion générale & Perspectives	68
	Références Bibliographiques et Webographies	70

Listes des figures

Chapitre I

Figure I.1 : processus de modélisation	5
Figure I.2 : Exemple d'entité avec ses valeurs d'attribut	6
Figure I.3 : Architecture d'un agent cognitif	8
Figure I.4 : Architecture d'un agent réactif	9
Figure I.5 : fonctionnement d'un agent	9
Figure I.6 : Architecture d'un SMA a contrôle centralisé	11
Figure I.7 : Architecture d'un SMA a contrôle distribué.....	12
Figure I.8 : Communication par envoi de messages.....	14
Figure I.9 : Communication par partage d'information.....	15

Chapitre II

Figure II.1 : L'écart du temps inter-véhiculaire	18
Figure II.2 : représentation d'une circulation routière	19
Figure II.3 : La congestion d'un réseau routier.	20
Figure II.4 : les trois grandes classes présentées par Bazzan.	23
Figure II.5 : Exemple d'un plan de feu.....	26
Figure II.6 : Boucles magnétiques en amont des carrefours (SCOOT).....	29
Figure II.7 : l'organisation de la synthèse des approches.	31
Figure II.8 : Modélisation de la géométrie des intersections.....	32
Figure II.9 : Discrétisation de la zone de croisement par l'agent manager.	33

Chapitre III

Figure III.1 : interface de NetLogo	41
Figure III.2 : Différentes solutions pour modéliser la composante spatiale prairie.....	45
Figure III.3 : l'éclatement de deux composants, selon leur espèce	45
Figure III.4 : Le diagramme de classe du phénomène proie –prédateur	46

Figure III.5 : Concordance de diagramme activité & programme NetLogo	46
Figure III.6 : Diagramme d'activité pour les moutons	47
Figure III.7 : Une Ossature NetLogo.....	48
Figure III.8 : Exemple de passage UML / NetLogo	49

Chapitre IV

Figure IV.1 : Processus de développement de notre modèle	51
Figure IV.2 : diagramme de classe	55
Figure IV.3 : diagramme de séquence	57
Figure IV.4 : Diagramme de passage UML/NetLogo	59
Figure IV.5 : Organigramme pour notre simulateur	59
Figure IV.6 : présentation de simulateur « SimCirc »	61
Figure IV.7 : une vue 3D de simulateur	62
Figure IV.8 : champ de visibilité des feux de signal	64

Liste des histogrammes

Histogramme 1 : Nbr_Voitures/Mécanismes (1 ou 2).....	65
Histogramme 2 : Nbr_Voitures/Mécanismes (2 ou 3).....	66

Liste des tableaux

Tableau 1 : Etat comparatif des accidents corporels de la circulation routière.....	1
Tableau 2 : Les principales causes des accidents corporels	1
Tableau 3 : Résumé des solutions citées	35
Tableau 4 : Les différents thèmes pour analyser la tâche générique	54
Tableau 5 : présentation de simulateur « SimCirc ».....	61
Tableau 6 : les procédures de configuration dans notre simulateur ‘SimCirc’.....	62
Tableau 7 : Nbre voitures immobile/Temps	65
Tableau 8 : Nbre voitures immobile/Temps	66

Liste des abréviations

UML : Unified Modelling Language

SMA: Simulation Multi Agent

TRANSYT : TRAFIC Network STUDY TOOL

SCOOT : Split Cycle and Offset Optimisation Technique

SCATS : Sydney Co-ordinated Adaptative Traffic System

PRODYN : Programmation Dynamique

UTCS: Urban traffic control software

TRL: Traffic research laboratory

EDI : Environnement développement intégrés

CFH : Capteur feux horizontal

CFV : Capteur feux vertical

IAV : véhicule intelligents autonomes

IA : Intelligence Artificiel

Introduction générale

1. Contexte

Durant les dernières décennies, les progrès réalisés dans les domaines de l'intelligence artificielle (IA) ont conduit l'homme à traiter des problèmes de plus en plus complexe. Parmi ces problèmes, nous nous intéressons dans ce mémoire aux problèmes des transports terrestre qui est le moyen le plus utilisé aujourd'hui pour se déplacer en Algérie ainsi que dans la majorité des pays du monde, qu'ils s'agissent de transports en commun ou de véhicules personnels. Il est devenu un aspect essentiel de notre vie quotidienne.

A cause de son importante amplification ces dernières années, le transport routier confronte des problèmes qui ne cessent d'augmenter chaque année, qui se présentent dans les accidents routiers fréquents (Tableau 1et 2: statistique d'Algérie 2017)¹, et les incidents menant le flux du trafic à se rabattre sur certaines portions de route menant à des situations de congestion ou de saturation. Cela représente une problématique revêtant plusieurs aspects : social (stress), économique (retards au travail et de livraison), environnemental (pollution de l'aire), etc.

Désignation	du 18 au 24/04/2017	du 25/04 au 01/05/2017
Nombre d'accidents	286	250
Nombre de Blessés	226	343
Nombre de Décès	11	22

Tableau1 : Etat comparatif des accidents corporels de la circulation routière enregistrés durant les périodes cités ci-dessous en zones urbaines.

la cause	le nombre
le facteur humain	243
le véhicule	3
la route et l'environnement	4
Total	205

Tableau2 : Les principales causes des accidents corporels des périodes cités.

De ce fait, des chercheurs et des ingénieurs en travaux publiques veillent à trouver des solutions meilleures aux problèmes confrontés qui convergent vers des solutions basées sur l'intelligence artificielle.

¹ <http://www.dgsn.dz>

La simulation est aujourd'hui un outil efficace employé pour l'analyse, la reproduction et la prévision d'une large variété de problèmes, difficiles à étudier par d'autres moyens trop chers ou dangereux. Cet outil est désormais d'emploi courant ; il a en cela été favorisé par la démocratisation de nouvelles techniques informatiques et l'apparition de nouvelles technologies matérielles.

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 acteurs complémentaire:

- Le phénomène réel (ou virtuel) que l'on souhaite étudier.
- Le modèle de ce phénomène.
- La simulation de ce modèle par ordinateur (simulateur).

Dans le cadre de notre projet de fin d'études, nous nous intéressons à étudier l'un des problèmes de la circulation routière, principalement, la gestion des carrefours d'intersections (feux de circulation) qui est en général vu comme un problème d'optimisation à travers une tâche de circulation générique.

Dans ce travail nous n'avons pas l'intention de répondre à toutes les questions de circulation routière, parce que les interrogations dans ce champ sont vastes, et intéresse beaucoup d'aspects qui posent beaucoup d'interrogations.

Mais notre intention est orientée vers un seul problème qui est la maîtrise de la régularisation des feux rouge dans les carrefours d'une façon automatique (a travers des compteurs) puis Autonominique (Locale : au niveau d'une seul intersection, et globale : communication entre les Intersections) à travers des capteurs intégré au niveau des feux de circulation. Ainsi, dans ce mémoire, nous adoptons un ensemble d'hypothèses, énoncées dans le dernier chapitre de ce mémoire.

3. Objectif de travail

Le travail réalisé dans le cadre de notre projet de fin d'études a pour but principal, de développer un Pseudo-Simulateur nommé "**SimCirc**" pour la circulation routière afin de l'optimiser quant à diminuer le nombre de véhicules en attente dans une/ plusieurs intersections, on utilise des techniques de modélisation et de simulation basée sur le paradigme Agent et les systèmes Multi-agents (SMA).

Afin d'arriver à cet objectif, nous avons fixé un ensemble d'objectifs intermédiaires qui sont détaillé de la partie suivante.

4. Organisation du travail

Afin d'atteindre nos objectifs cités ci-dessus et pour arriver à bien répondre à la problématique nous avons décomposé notre manuscrite en quatre chapitres :

- **Chapitre I** : décrire sur les paradigmes de modélisation et les primitives basiques pour traiter des solutions aux problèmes rencontrés.
- **Chapitre II** : ce chapitre interprète un état de l'art sur la circulation routière et les systèmes de gestion d'intersections dans les réseaux de transport. Nous mettons en avant les grandes dimensions selon lesquelles ces systèmes se distinguent en insistant sur les différents types d'algorithmes utilisés.
- **Chapitre III** : nous présentons l'approche multi-agents et le langage/plateforme (multi-agents) Netlogo, et une nouvelle approche UML/SMA.
- **Chapitre IV** : cette partie a pour but d'expliquer la démarche de notre travail de simulation et son déroulement, la présentation de notre simulateur « **SimCirc** », le Déroulement des scénarios de simulation et l'interprétation des résultats.
- **Conclusion et Perspective**: Finalisé notre travail avec un résumé de ce mémoire et la rédaction de quelques perspectives envisageable de ce sujet.

Chapitre I : Les paradigmes de modélisation

1. Introduction

L'informatisation est le phénomène le plus important de notre époque. Elle s'immisce maintenant dans la plupart des objets de la vie courante et ce, que ce soit dans l'objet proprement dit, ou bien dans le processus de conception ou de fabrication de cet objet [27].

Notamment la simulation et la modélisation qui tiennent une place grandissante dans notre société actuelle, jouent un rôle prépondérant dans les technologies de la robotique, l'architecture de l'internet du futur ou la réduction des nuisances sonores dans le domaine des transports. Car la modélisation d'un système avant sa réalisation permet de mieux comprendre le fonctionnement du système. C'est également un bon moyen de maîtriser sa complexité et d'assurer sa cohérence [28].

Dans ce chapitre nous nous intéressons aux paradigmes de modélisation et les primitives basiques pour traiter des solutions aux problèmes rencontrés.

2. Qu'est-ce qu'un paradigme

Un paradigme, c'est une représentation du monde. Une manière de voir les choses.

Un modèle cohérent de vision du monde qui repose sur une base définie. C'est une forme de pensée qui ne doit pas être confondue avec celle d'un autre paradigme, qui dans le cas échéant, peut faire obstacle à 'introduction de nouvelles solutions qui seraient mieux adaptées [29].

En général est une vision du monde qui oriente notre manière de penser.

3. Modèle et modélisation

La modélisation, ou bien la construction d'un modèle, sert à créer une représentation simplifiée d'un système réel capable de reproduire son fonctionnement et qui aide à mieux comprendre les aspects clés de ce système en réalité [1].

Les modèles sont programmés sur un ordinateur. L'utilisation d'une interface graphique lors de la programmation du modèle permet un usage plus facile [2]. Les données utilisées pour la construction du modèle peuvent s'agir d'observations réelles ou de connaissances sur le domaine étudié, elles sont traduites en algorithmes qui seront exécutées par l'ordinateur dans le but de produire des données en sorties compatibles avec celles en entrée [3].

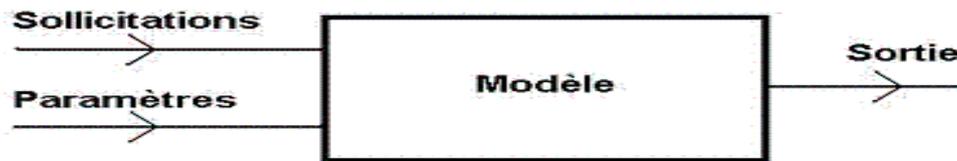


Figure I.1 : processus de modélisation

Donc la modélisation est la représentation d'un système par un autre, plus facile à appréhender. Il peut s'agir d'un système mathématique ou physique. Le modèle sera alors numérique ou analogique [30].

3.1 La modélisation numérique

La modélisation numérique consiste à construire un ensemble de fonctions mathématiques décrivant le phénomène. En modifiant les variables de départ, on peut ainsi prédire les modifications du système physique.

3.2 La modélisation analogique

La modélisation analogique consiste à construire un système physique qui reproduit plus ou moins un phénomène que l'on souhaite étudier. L'observation du comportement du modèle permet de tirer des enseignements sur le phénomène d'intérêt.

3.3 Principes de la modélisation

Objectif principal de la modélisation est maîtriser la complexité.

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 [31].

4. Les paradigmes de modélisation

A chaque fois qu'il y a eu une grande révolution informatique, cela coïncide presque systématiquement avec un changement de paradigme, c'est à dire une nouvelle façon de voir les problèmes, les solutions.

Un paradigme de modélisation est donc un style dans lequel on traite des solutions aux problèmes rencontrés. Propose un formalisme ce fonde sur des engagements ontologique. Fournit des primitives basiques (telle que le concept, la propriété,...), fixe les interprétations

qui leur sont associées, interprétation qui de fait se répercutent à leur tour sur les significations des objets présents dans le modèle [18].

4.1 Entité

Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire. « Tout objet identifiable et pertinent pour l'application » [38].

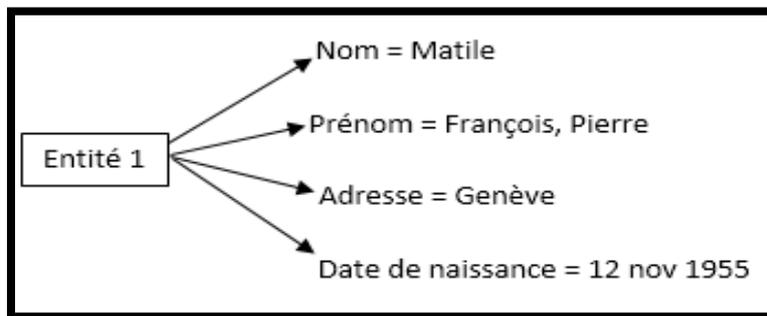


Figure I.2 : Exemple d'entité avec ses valeurs d'attribut

4.2 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.

4.2.1 Agents

Le concept d'agent a été utilisé pour plusieurs décennies dans différentes disciplines. Il a été principalement utilisé dans les systèmes à base de connaissances, la robotique, et d'autres domaines de l'intelligence artificielle [5]. Ce concept a été défini par un nombre d'auteurs, les définitions étant proches dans leurs sens, une définition commune peut être donnée:

« Un agent est une entité autonome réelle ou abstraite, poursuivant un but individuel, qui est capable d'agir sur elle-même et sur son environnement et qui peut communiquer avec d'autres agents si elle est mise dans un univers multi-agents ».

Ferber [6] précise que le comportement de l'agent tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont il dispose, et en fonction de sa perception, de ses représentations et des communications qu'il reçoit.

D'autre part, Wooldridge [7] donne une définition à l'agent informatique comme suit: Un agent est un système informatique situé dans un environnement et qui est capable d'agir de façon autonome dans cet environnement pour atteindre les objectifs pour lesquels il a été conçu.

A la différence des autres définitions, Wooldridge [7] considère l'agent comme un « système informatique » qui est « conçu » pour atteindre certains objectifs.

A partir des définitions données au concept d'agent, on peut tirer des caractéristiques communes propres à l'agent. Ces caractéristiques seront précisées ultérieurement.

4.2.1.1 Caractéristiques des agents

Un agent est caractérisé par un certain nombre de capacités. Cependant, il n'est pas obligatoire que ces caractéristiques se réunissent dans un seul agent, d'ailleurs, il existe des types d'agents suivant la présence ou l'absence de certaines caractéristiques. Les capacités qui caractérisent l'agent sont les suivantes :

- L'autonomie : un agent autonome possède ses propres buts et il est capable de prendre des décisions et ainsi de résoudre des conflits internes.
- La perception de son état interne et de son environnement.
- La sociabilité : la capacité de communiquer avec les autres agents grâce un langage et un protocole de communication.
- L'intentionnalité : elle exprime la volonté d'un agent d'atteindre un but ou d'effectuer une action.
- La rationalité : un agent rationnel dispose de critères d'évaluation de ses actions et sélectionne selon ces critères la meilleure action qui lui permet d'atteindre son but. La notion de rationalité se rapporte au comportement cognitif de l'agent.
- La planification : c.-à-d. la construction d'un plan d'actions à réaliser.
- La décision : l'agent est capable de choisir et de décider qu'une action, ou bien un plan d'actions, sera effectué par lui après l'avoir évalué.
- L'engagement : l'agent s'engage à accomplir les actions qui satisfont son but, il croit qu'il est en mesure d'exécuter toutes ces actions, ce qui le conduit à agir en conséquence.
- L'action : la capacité de l'agent à agir sur son environnement, à le modifier.
- La réactivité : un agent est réactif s'il peut réagir à un changement qui s'est effectué dans son environnement.
- La proactivité : un agent proactif agit sans que son environnement ait changé.

- L'apprentissage : l'évolution et l'amélioration de ses connaissances et l'adaptation de ses comportements suite à des événements ou des communications et en fonction des expériences passées.
- L'adaptativité : la flexibilité et la capacité d'un agent de contrôler ses aptitudes selon la situation avec laquelle il interagit.
- La reproductivité : la naissance de nouveaux agents identiques.

4.2.1.2 Types d'agent

En fonction de leur mode de fonctionnement et de leur représentation de leur environnement, les agents peuvent être cognitifs ou réactifs

- **Agent cognitif**

Les agents à capacités cognitives proviennent d'une métaphore du modèle humain. Les agents cognitifs disposent d'une base de connaissances comprenant les diverses informations liées à leur domaines d'expertise et à la gestion des interactions avec les autres agents et leur environnement. Les agents sont généralement « intentionnels » c'est-à-dire qu'ils possèdent des buts et des plans explicites leur permettant d'accomplir leurs buts. Ceci a donné lieu plus tard aux premières architectures modernes d'agents dits cognitifs.

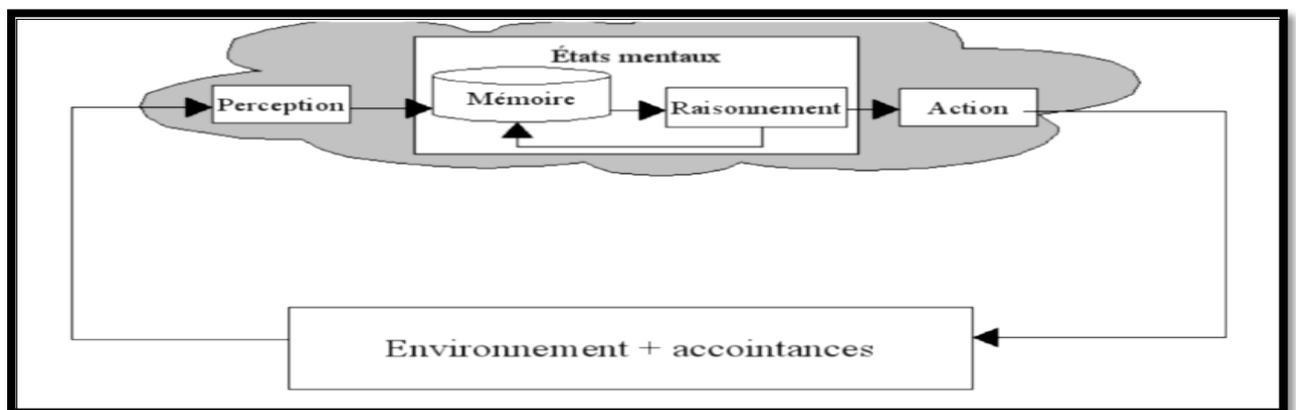


Figure I.3 : Architecture d'un agent cognitif

- **Agent réactif :**

Les agents réactifs au contraire ne sont pas « intelligents » pris individuellement .ils répond de manière opportune aux changements de son environnement. Ceux-ci peuvent être constitués de stimuli externes et asynchrones. Un tel type d'agent n'a pas besoin d'une représentation symbolique élevée de la perception son environnement.

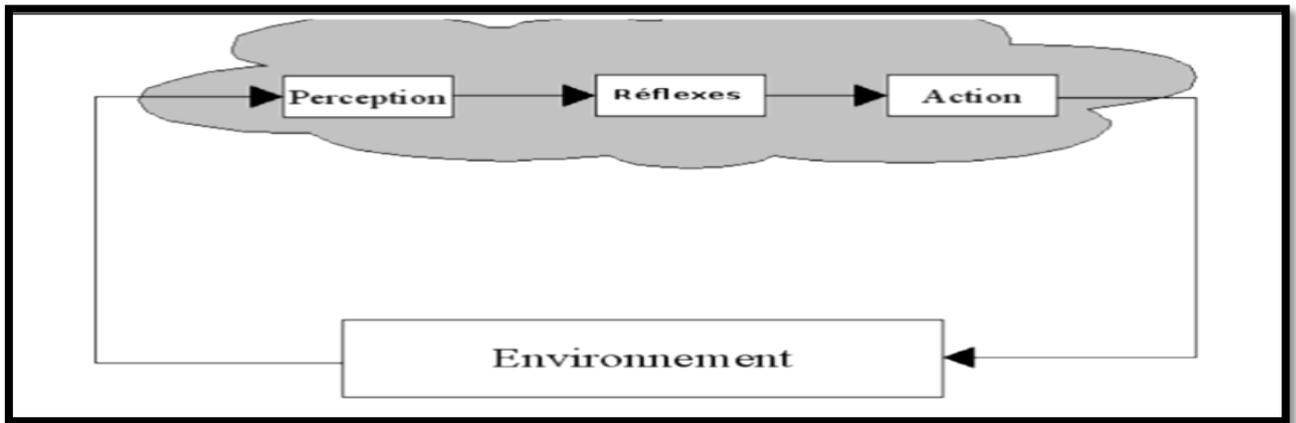


Figure I.4 : Architecture d'un agent réactif

4.2.1.3 Fonctionnement d'un agent

Pour qu'un agent arrive à effectuer une action, il passe par un processus qui se décompose en trois phases : perception, décision et action (figure I.5).

En fonction des connaissances dont l'agent dispose et des buts qu'il se fixe suite à une perception ou à une interaction avec le monde extérieur, l'agent doit décider quel serait le but à retenir et à satisfaire en premier, ensuite passer à l'exécution [8].

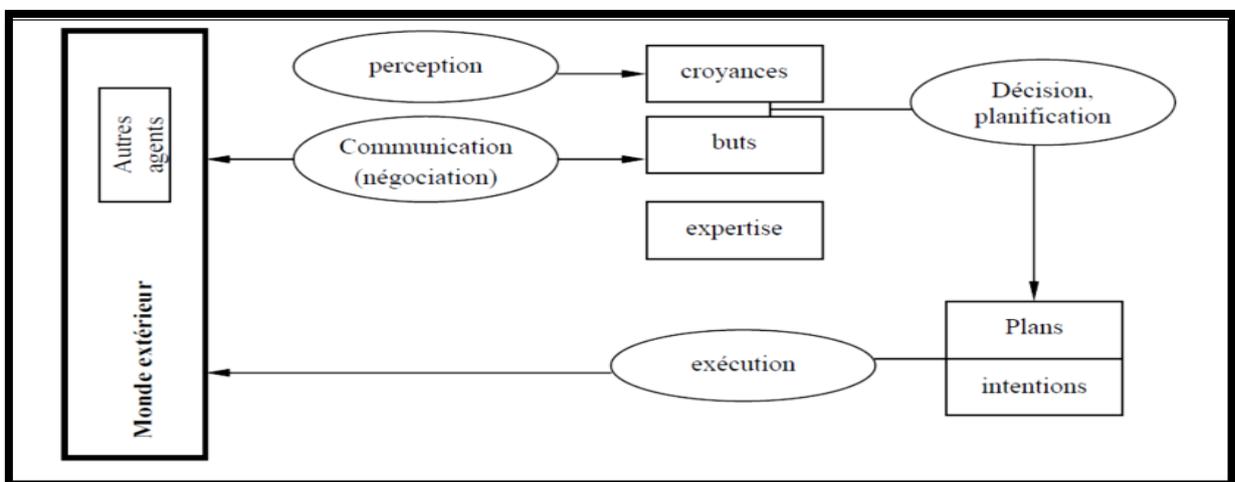


Figure I.5 : fonctionnement d'un agent

• Perception :

Cette phase permet à l'agent d'acquérir des connaissances par perception de soi-même ou de son environnement, par communication avec les autres agents, ou bien à partir de son savoir initial [8].

Dans ce dernier cas, les connaissances sont considérées comme certaines puisqu'elles n'ont subi aucune mise à jour, alors que dans les autres cas elles sont considérées incertaines puisqu'elles ont évoluées à l'extérieur avant d'arriver à cet agent. On peut pour cela associer à chaque connaissance son origine afin d'en évaluer la crédibilité et d'en permettre la vérification [8]. Dans certains cas, l'agent se construit une représentation symbolique de son environnement à partir des informations perçues, ou la met à jour [9].

• **Décision :**

La phase de décision correspond à une phase de raisonnement qui permet à l'agent de prendre sa décision et de planifier les actions à réaliser pour atteindre son objectif [9]. Durant cette phase, l'agent doit sélectionner le but à satisfaire en premier, et pour chaque but, l'action qui permet de l'atteindre. Pour ce faire, l'agent analyse les différentes alternatives en termes d'utilité et de certitude [8]. Ensuite, l'agent détermine l'ordre d'exécution des actions sélectionnées, c'est créer ce que l'on appelle un plan d'actions.

• **Action :**

La phase d'action correspond à la réalisation effective des décisions prises lors de l'étape précédente [9], c.-à-d. l'exécution du plan d'action établie.

4.2.2 Système multi agent SMA

Un système multi-agent est un système distribué composé d'un ensemble d'agents est caractérisé par :

Chaque agent a des informations ou des capacités de résolution de problèmes limités ; il n'y a aucun contrôle global du système et les données sont décentralisées et le calcul est asynchrone.

Selon Ferber « Un S.M.A est un ensemble d'entités (physiques ou virtuelles) appelées agents, partageant un environnement commun (physique ou virtuel), qu'elles sont capables de percevoir et sur lequel elles peuvent agir. Les perceptions permettent aux agents d'acquérir des informations sur l'évolution de leur environnement. Les agents interagissant entre eux directement ou indirectement, et exhibent des comportements corrélés créant ainsi une synergie permettant à l'ensemble des agents de former un collectif organisé » [21].

4.2.2.1 Avantage de l'utilisation des SMA

Ils héritent des bénéfices envisageables de l'IA comme la prise de décision automatique, la facilité de maintenance, la complexité supportée, la réutilisabilité et la portabilité [5].

En outre, les apports les plus importants des SMA peuvent être résumés dans : l'efficacité, l'adaptation des représentations et des traitements à la réalité, la simulation des fonctionnements des organisations et la prise en compte de la coopération et la négociation entre agents.

4.2.2.2 Architecture des SMA

Il existe deux types d'architectures selon le mode de contrôle du SMA : centralisé ou distribué.

- **Architecture à contrôle centralisé**

Les SMA à contrôle centralisé sont connus sous le nom de systèmes à base de tableau noir (figure I.6). L'architecture de tableau noir est la plus utilisée dans les SMA cognitifs.

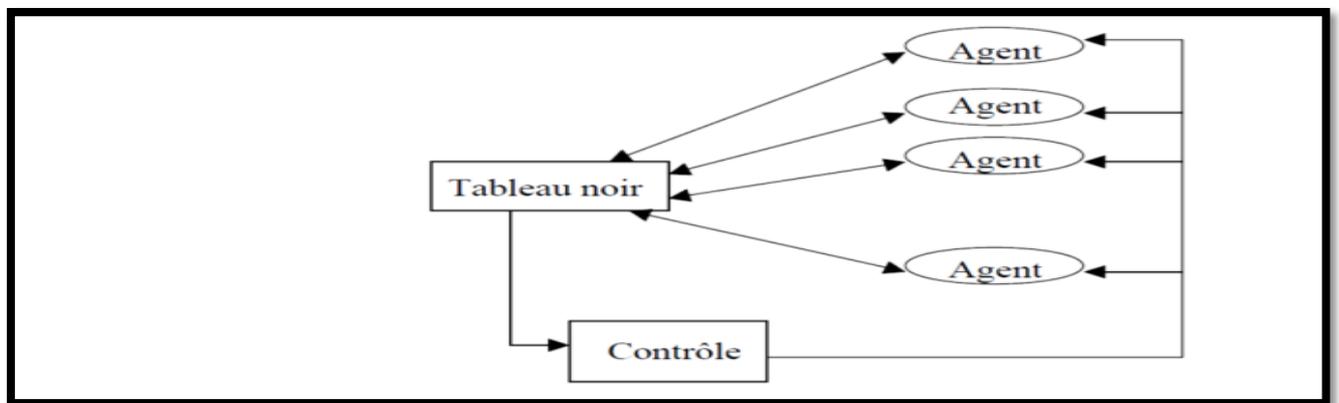


Figure I.6 : Architecture d'un SMA à contrôle centralisé

Ce type de systèmes est composé de trois éléments :

- Les sources de connaissance.
- La base partagée, ou le tableau noir, qui comprend toutes les informations que s'échangent les sources de connaissances (les états partiels d'un problème en cours de résolution, les hypothèses et les résultats intermédiaires).
- Le dispositif de contrôle qui gère les conflits d'accès entre les sources de connaissance.

Il n'y a aucune communication directe entre les composants de ces systèmes, les interactions se font via le partage d'un même espace de travail.

- **Architecture à contrôle distribué**

Dans cette architecture, les connaissances et le contrôle sont totalement distribués, le traitement se fait en local pour chaque agent, et ils communiquent entre eux par envoi de messages directs (figure I.7).

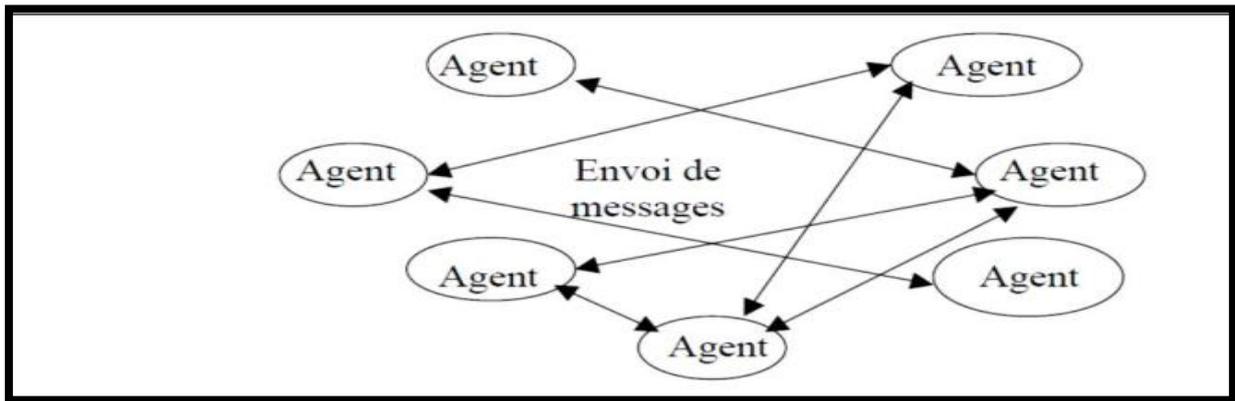


Figure I.7 : Architecture d'un SMA à contrôle distribué

4.2.2.3 Environnement

L'environnement est l'ensemble des conditions extérieures susceptibles d'agir sur le fonctionnement d'un système. Un agent autonome évolue de façon continue dans un environnement, dans lequel peuvent exister d'autres agents.

Puisqu'un agent agit sur son environnement et que l'environnement agit sur l'agent, il existe une réelle interaction entre ces deux entités.

La connexion entre un agent et son environnement peut être plus ou moins forte selon la façon dont l'agent acquiert l'information en provenance de l'environnement. A partir des informations obtenues, l'agent peut créer et modifier sa propre représentation de l'environnement.

La connexion agent-environnement est forte dans le cas d'un agent situé, c'est à-dire lorsque l'agent possède des capteurs et des effecteurs. Les capteurs sont capables de percevoir directement (mais souvent partiellement) les éléments de l'environnement ou les modifications survenues dans cet environnement. Les effecteurs permettent à l'agent d'agir sur son environnement.

Les principales caractéristiques de l'environnement sont :

- Accessible / inaccessible : l'accessibilité signifie que l'agent ait un accès à l'état complet de l'environnement
- Déterministe / non déterministe : un environnement est déterministe c'est le prochain état de l'environnement est complètement déterminé par son état courant et l'action sélectionnée par l'agent.

- Statique / dynamique : un environnement est dit dynamique s'il peut changer pendant la prise de décision de l'agent.
- Discret / continu : s'il existe un nombre limité de perceptions et d'actions possible, on parlera d'environnement discret.

4.2.2.4 Interactions 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 langage 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.

4.2.2.5 Communication entre agents

Les communications dans les systèmes multi-agents, comme chez les humains, sont à la base des interactions et de l'organisation des agents. La communication agrandit les capacités perceptives des agents en leur permettant de bénéficier des informations et du savoir-faire des autres agents. Les communications sont indispensables à la coopération et il est difficile de concevoir un système d'agents coopérants s'il n'existe pas un système permettant aux agents d'échanger des informations ou de véhiculer des requêtes. Nous distinguons essentiellement deux modes de communication :

- La communication indirecte qui est une communication par signaux via l'environnement (dans les systèmes réactifs).

- La communication directe qui procède à un échange de messages entre les agents (Dans les systèmes cognitifs).

- **Le mode de communication par échange de messages :**

Les systèmes fondés sur la communication par envoi de messages relèvent d'une distribution totale à la fois de la connaissance, des résultats et des méthodes utilisées pour la résolution du problème. Les agents sont en liaison directe et envoient leurs messages directement et explicitement au destinataire. La seule contrainte est la connaissance de l'agent destinataire : Si un agent A connaît l'agent B alors il peut entrer en communication avec lui.

Les systèmes d'acteurs en sont l'illustration parfaite.

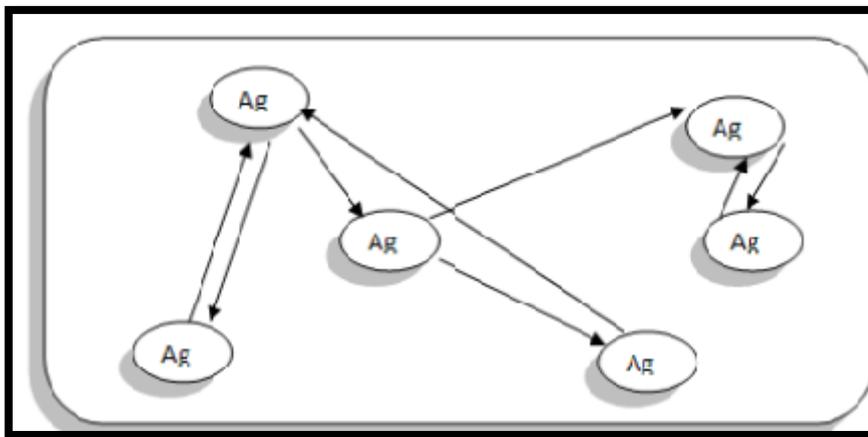


Figure I.8 : Communication par envoi de messages

- **Communication par partage d'informations :**

En intelligence artificielle la technique du tableau noir (blackboard) est très utilisée pour spécifier une mémoire partagée par divers systèmes.

Les composants (les agents) ne sont pas en liaison directe mais communiquent via une structure de données partagée, où on trouve les connaissances relatives à la résolution (état courant du problème) qui évolue durant le processus d'exécution.

Dans ce mode de communication on parle plutôt de sources de connaissances que d'agents, où le tableau noir est en général partitionné en plusieurs niveaux qui sont spécifiques à l'application. Les agents qui travaillent sur un niveau particulier peuvent accéder aux informations contenues dans le niveau correspondant du tableau noir ainsi que dans des niveaux adjacents.

Le mécanisme blackboard est constitué de trois éléments principaux :

- Les sources de connaissances : la connaissance du domaine est partagée entre les sources de connaissances.
- Le blackboard : la mémoire partagée, où on trouve les connaissances relatives à la résolution du problème.
- Le contrôle : pour gérer les conflits d'accès au tableau.

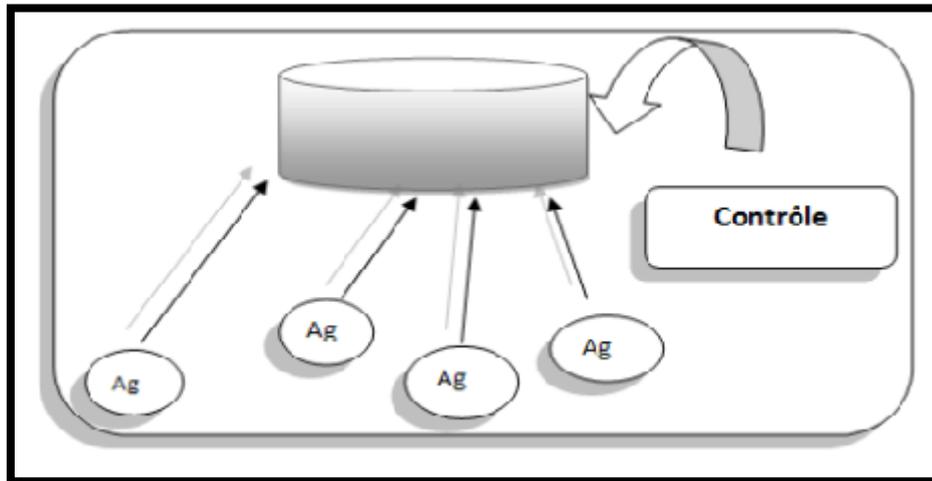


Figure I.9 : Communication par partage d'information

4.2.2.6 Domaines d'application des S.M.A

- Résolution de problèmes soit : de décision par planification distribuée des tâches ou d'optimisation.
- Modélisation et simulation d'entités distribuées : on part généralement d'une population d'individus quelconques (par exemple en éthologie ce sont des insectes, en sociologie ce sont des gens) que l'on modélise dans leur structure interne et leur comportement (réactif/cognitif), on associe à chaque individu un agent, la phase de simulation nécessite l'implémentation des lois du monde qui est simulé, c'est une phase discrète car à chaque pas de temps simulé, le système multi-agent évolue.

L'objectif est d'aboutir à la phase de visualisation qui consiste à montrer les entités qui sont censées émerger de la population sous formes de structures ou de comportements.

- Applications distribuées telle que :
 - la Gestion de grands systèmes industriels, Contrôle du trafic aérien, Applications de monitoring « surveillance médicale », Réseaux de télécommunication, Recherche de l'information, Commerce électronique, La robotique, Traitement d'image et réalité augmentée et les Applications Multimédia.

5. Conclusion

Dans ce chapitre nous avons parlé de la modélisation ; ses types et ses principes et aussi des primitives basique telle que : entité, objet, les Agents et leur environnement, la communication entre eux et leur caractéristiques principaux, en particulier le système qui se forme à partir de ces agents.

Cette vision nous permettre de mieux comprendre les paradigmes intéressent dans notre problématique abordée par ce mémoire.

Chapitre II : Etat de l'art sur la circulation routière

1. Introduction

La route occupe une place privilégiée dans le secteur du transport, et, joue un rôle central dans les déplacements des biens, et des personnes pour cela de nombreux travaux de recherche à travers le monde portent sur le transport routier. La motivation vient du fait que ce secteur nécessite de nouvelles applications de méthodes permettant de contrecarrer l'augmentation accrue des problèmes issus de la circulation routière, que ce soit, en vue de réduire le nombre d'accidents, de diminuer la circulation dans les grandes agglomérations ou de réduire la pollution. En effet, l'un des facteurs favorables à cette problématique est dû à l'accroissement du nombre exponentiel de véhicules sur la route.

2. Circulation routière

Une particularité du trafic automobile est essentiellement liée à la conception de l'infrastructure qui est conçue, selon une demande projetée, pour répondre à un optimum collectif mais que chaque individu réalise sans déplacement en cherchant à atteindre son optimum individuel, ce qui est souvent antagoniste avec l'optimum collectif. Les deux définitions ci-dessous montrent bien le double aspect individuel et collectif du trafic, l'aspect individuel étant représenté par le fait que le trafic est formé par différents véhicules et l'aspect collectif par le fait que le trafic est interprété comme un fluide. [23].

Alors le trafic routier est la circulation de nombreux véhicules sur un réseau.

2.1 Composantes de la circulation routière

En général, le trafic est constitué de deux composantes majeures : l'infrastructure et les mobiles.

- L'infrastructure est un ensemble d'éléments structuraux interconnectés qui fournissent le cadre pour supporter la totalité de la structure du trafic. Par exemple, pour les réseaux routiers et autoroutiers, l'infrastructure contient les structures : ponts, ponceaux, signalisation et marquages, systèmes électriques et les traitements de bord (bordures, trottoirs, aménagement paysager).
- Les mobiles sont les individus qui utilisent l'infrastructure, principalement dit, des véhicules. Les piétons ne sont pas considérés car leurs mouvements sont supposés dépendre de celui des mobiles.

Notons que les véhicules à l'arrêt ou en stationnement, au sens défini par le code de la route, sont considérés comme faisant partie de l'infrastructure et ne sont généralement pas pris en compte. Autrement dit, seuls les mobiles en mouvement sur la route et en interaction avec la signalisation sont considérés. Dans les sous-sections suivantes, nous introduisons les variables élémentaires utilisées et les mesures usuelles pour l'analyse du trafic. [25].

2.2 Variables élémentaires du trafic routier

Le développement considérable de la circulation automobile et l'accroissement constant de la motorisation ont rendu nécessaire l'élaboration des théories du trafic, capables de décrire, d'expliquer et de prévoir les interactions entre les véhicules ainsi que les mouvements d'ensemble sur les infrastructures routières.

Ces théories, déterministes ou probabilistes s'introduisent au moyen d'un certain nombre de variables, de relations ou de distributions caractéristiques.

2.2.1 L'écart du temps inter-véhiculaire

Le premier concept introduit est l'écart du temps inter-véhicule (en anglais, « the headway »). En un point donné de la route, c'est le temps qui s'écoule entre les instants de passage de deux véhicules successifs.

Cette variable se révèle particulièrement utile dans les études relatives à la sécurité routière et dans de nombreux problèmes de simulation du trafic. Certains algorithmes de régulation de carrefours à feux en ville, utilisent ce temps inter-véhiculaire.

Enfin, ce paramètre permet d'appréhender les effets de la composition du trafic sur les conditions d'écoulement de la circulation.

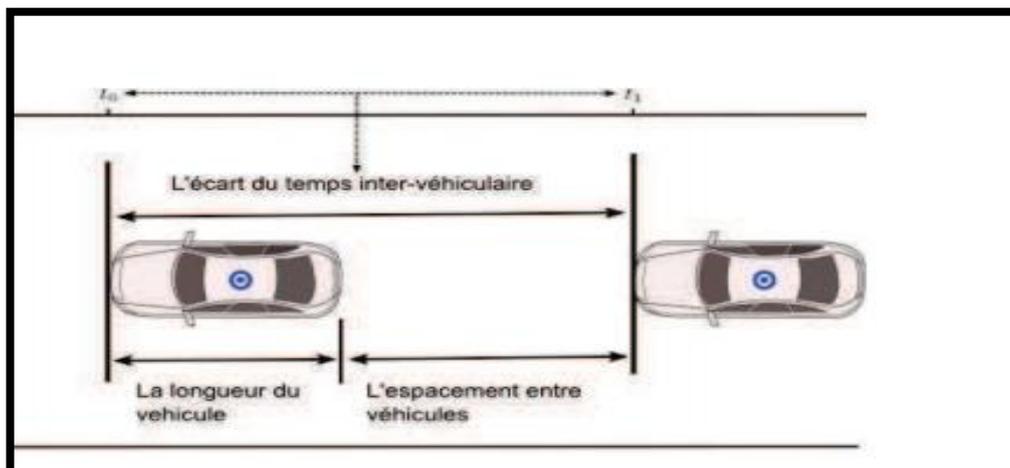


Figure II.1 : L'écart du temps inter-véhiculaire

2.2.2 Le débit

Il correspond à la répartition des véhicules dans le temps. On définit tout d'abord le débit moyen $Q(t_1, t_2, x)$ au point d'abscisse x entre les instants t_1 et t_2 par le rapport :

$$Q(t_1, t_2, x) = \frac{n(t_1, t_2, x)}{t_2 - t_1}$$

Dans lequel $n(t_1, t_2, x)$ désigne le nombre de véhicules passés en x entre les 2 instants. Expérimentalement, le débit peut être déterminé par de simples comptages sur la route.

Dans certaines théories, le flot des véhicules est parfois considéré comme continu. On définit alors le débit $Q(x, t)$ au point x et à l'instant t par :

$$q(x, t) = \lim_{\Delta t \rightarrow 0} q\left(t - \frac{\Delta t}{2}, t + \frac{\Delta t}{2}, x\right)$$

2.2.3 La concentration (ou la densité)

Elle décrit la répartition des véhicules dans l'espace.

La concentration moyenne $k(x_1, x_2, t)$ à l'instant t sur une section de route limitée par les points d'abscisses x_1 et x_2 correspond au rapport :

$$k(x_1, x_2, t) = \frac{n(x_1, x_2, t)}{x_1 - x_2}$$

Dans lequel $n(t_1, t_2, x)$ désigne le nombre des véhicules présents sur la section à l'instant t .



Figure II.2 : représentation d'une circulation routière

Malgré les progrès techniques et technologiques accomplis par l'homme dans tous les domaines de la connaissance, la circulation routière reste victime d'une congestion sans cesse croissante.

2.3 La congestion routière

La congestion du trafic est une notion intuitivement simple mais dont l'analyse rigoureuse s'avère en fait assez complexe [36]. C'est un phénomène physique concernant la façon dont des véhicules empêchent la progression des uns et des autres au fur et à mesure que la demande d'un espace routier s'approche de la capacité maximale de celui-ci [19].

La congestion d'un réseau routier est la condition dans laquelle une augmentation du trafic de véhicules provoque un ralentissement global de celui-ci. Le terme de congestion désigne la dégradation de la qualité de service quand le nombre d'utilisateurs augmente. Ce phénomène se caractérise par l'apparition de retards, voire de goulets d'étranglement en période de fort trafic, c'est-à-dire quand la capacité de l'infrastructure devient insuffisante pour réguler les flux. Le problème est fréquent localement et périodiquement, notamment dans les grandes villes et lors des grands départs pour les vacances [37].

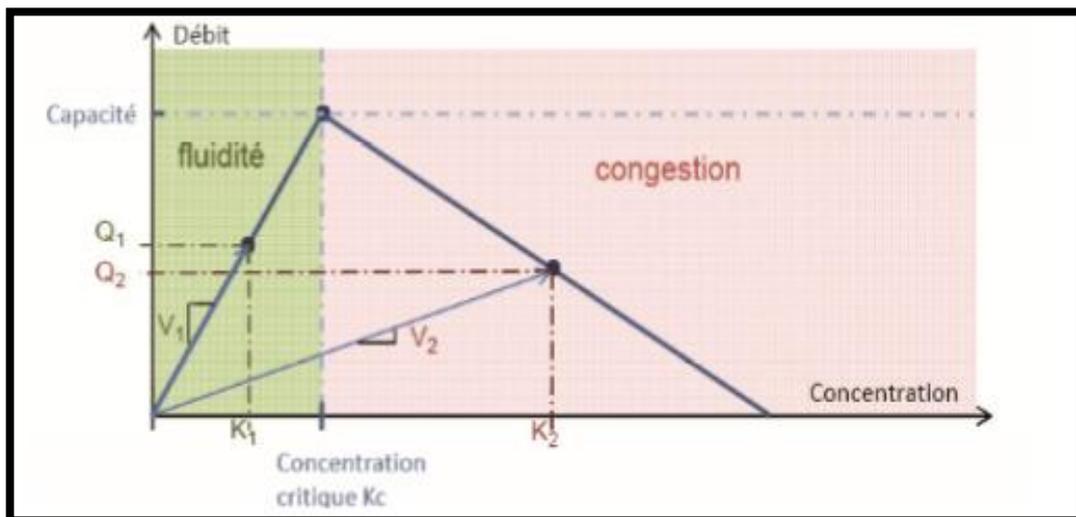


Figure II.3 : La congestion d'un réseau routier.

Le trafic est dit congestionné s'il évolue dans la partie droite du diagramme, c'est-à-dire si la concentration est supérieure à la concentration critique (concentration correspondant au débit maximum) [36].

2.3.1 Pourquoi y a-t-il des bouchons sur les routes ?

La première cause est tout simplement un volume de circulation supérieur à la capacité routière : les voitures sont trop nombreuses et le trafic s'arrête à chaque goulot d'étranglement (rétrécissement de la route, voies qui se rejoignent, croisements de routes).

La seconde cause est liée aux comportements des conducteurs. Quand les voitures sont proches les unes des autres, il suffit qu'un conducteur freine brutalement pour faire freiner en cascade tous les véhicules qui le suivent et propager ainsi un ralentissement, voire même un arrêt du trafic sur une certaine longueur. Quand le premier conducteur redémarre, de nouvelles voitures sont venues s'agglutiner et le bouchon se déplace vers l'arrière comme une onde, par un effet de dominos [37].

Avec précision le flux de la circulation qui s'écoule à un instant donné sur une voie est toujours la résultante d'une interaction entre deux éléments : [36]

2.3.1.1 Une demande

Qui représente le trafic qui cherche à s'écouler et qui s'écoulerait s'il n'était pas contraint. Cette notion est différente selon l'échelle à laquelle on se place à :

- l'échelle d'un réseau : c'est la demande globale de déplacement qui souhaite utiliser le réseau.
- l'échelle locale d'un point précis sur une voie, ce sont les véhicules présents et qui cherchent à s'écouler.

Par exemple, la demande dans une file d'attente arrêtée à un feu rouge est égale au débit maximum de la ligne de feu, c'est-à-dire au débit qui s'écoulerait si le feu passait au vert.

2.3.1.2 Une offre

Correspondant aux possibilités d'écoulement de l'infrastructure. Là encore, cette notion est dépendante de l'échelle :

- l'offre d'un réseau représente les flux qu'il est globalement capable d'écouler.
- l'offre locale en un point représente le flux qui pourrait s'écouler en ce point.

2.3.2 Conséquences de la congestion du trafic routier

Elles sont nombreuses et peuvent être classées en trois catégories : [37]

- Conséquences économiques

D'un point de vue économique, la congestion du trafic routier entraîne :

- Une perte de compétitivité d'une région, d'un pays ou d'un continent, du fait des congestions qui bloquent les artères de communication routière.

- Une perte de productivité, non seulement à cause des retards des personnes se rendant sur les lieux de leur travail, mais aussi en raison des livraisons tardives des approvisionnements ou des services rendus avec des retards importants voire annulés.

- Conséquences sociétales

Les usagers de la route se plaignent amèrement des heures qu'ils perdent dans les embouteillages (plus de 1 milliard d'heures par an pour la France) parce qu'elles empiètent sur le temps qu'ils peuvent consacrer à leur travail et à leur famille.

La congestion du trafic routier aura deux impacts importants sur :

- La qualité de vie des usagers, car les conducteurs soumis à des embouteillages fréquents sont sujet à des problèmes de stress, d'anxiété et d'énervement, entraînant un accroissement du risque d'accident.

- Le pouvoir d'achat des usagers, car la congestion du trafic a des conséquences directes sur l'augmentation de la consommation énergétique. Dans le contexte actuel d'envolée des prix du pétrole, ceci engendre une nette diminution du pouvoir d'achat des usagers.

- Conséquences environnementales

Enfin, d'un point de vue environnemental, la congestion du trafic routier a un impact néfaste et négatif à cause de l'augmentation de la pollution qui engendre des coûts économiques et écologiques en croissance exponentielle.

2.4 Les méthodes de gestion

La gestion de la circulation dans les grands réseaux routiers reste un défi important dans les systèmes de transport [24] où les congestions sont générées au niveau des intersections entre les routes, plus précisément, au niveau des parties de l'espace qui doivent être partagées par les véhicules.

Le travail, ([Bazzan](#))[10] présente un aperçu des méthodes et des approches dans les domaines de l'ingénierie du trafic pour surmonter les problèmes de congestion aux intersections, **Bazzan** choisit une classification historique, à savoir trois grandes classes (figure II.4).

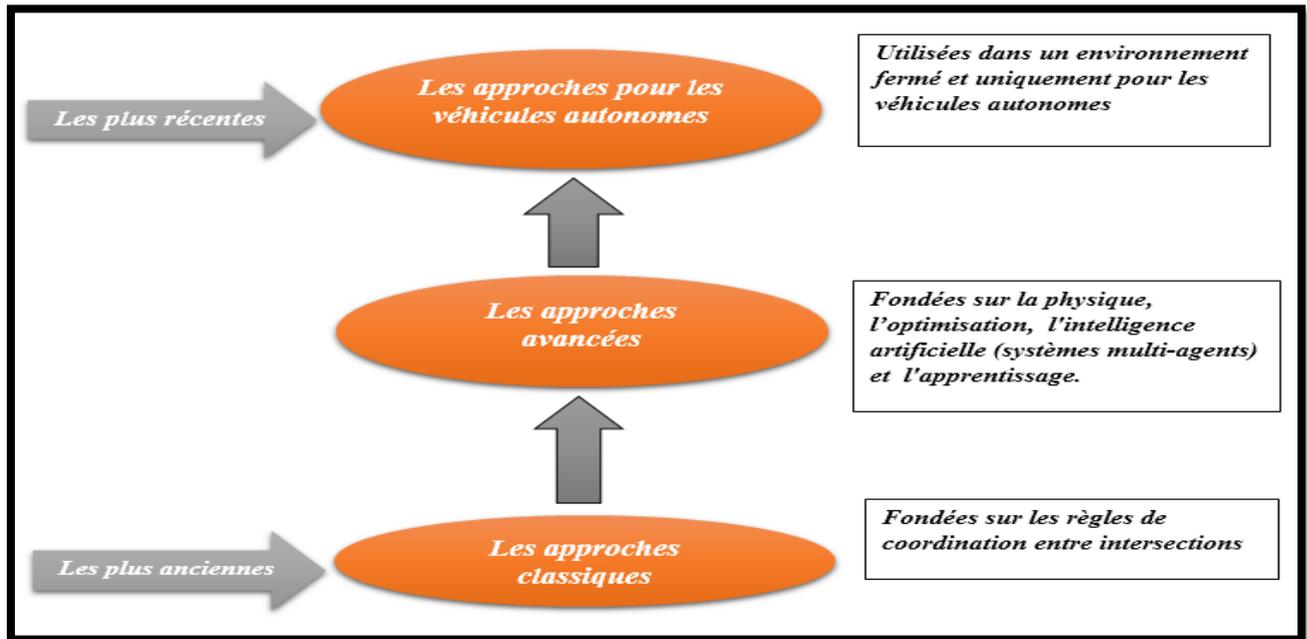


Figure II.4: les trois grandes classes présentées par Bazzan.

Pour cela nous présentons un aperçu des méthodes utilisées par le code de la route pour surmonter les problèmes de la circulation routière.

2.4.1 Code de la route

Le code de la route est le recueil, écrit ou numérisé de toute la réglementation qui concerne la circulation sur les routes et les chaussées par tous les usages qui les empruntent.

2.4.1.1 Règles de priorité

Par exemple le code de la route français : Lorsque deux conducteurs abordent une intersection par des routes différentes, le conducteur venant par la gauche est tenu de céder le passage à l'autre conducteur. Sauf dispositions différentes prévues.

Ceci implique une règle simple à respecter à chaque intersection sans signalisation souvent appelée priorité à droite.

Nous trouvons aussi : Tout conducteur abordant un carrefour à sens giratoire est tenu, quel que soit le classement de la route qu'il s'apprête à quitter, de céder le passage aux usagers circulant sur la chaussée qui ceinture le carrefour à sens giratoire. Cette règle annule donc l'effet de la première règle de priorité à droite et définit une nouvelle règle de priorité à gauche appliquée uniquement pour les carrefours à sens giratoire souvent appelés rond-point.

Ces règles ne sont pas communes à tous les pays. Elles diffèrent d'un pays à un autre et surtout dans les pays où les véhicules circulent à gauche sur la chaussée.

Par exemple, au Royaume-Uni, la priorité aux ronds-points est pour les véhicules qui viennent de droite et il n'existe pas de règle générale pour les intersections, la priorité étant généralement indiquée.

Notons que les règles de priorité du code de la route sont une forme simple de coordination réactive [24].

2.4.1.2 Les panneaux de signalisation

En France, il existe des panneaux pour la gestion des intersections. Par exemple, parmi les panneaux les plus répandus nous trouvons les panneaux de signalisation de céder le passage.

A certaines intersections indiquées par une signalisation dite "cédez le passage", tout conducteur doit céder le passage aux véhicules circulant sur l'autre ou les autres routes et ne s'y engager qu'après s'être assuré qu'il peut le faire sans danger. Il est souvent utilisé pour définir une route principale par rapport à une route secondaire et n'oblige pas les usagers de la route secondaire à l'arrêt. Il existe aussi un autre panneau très répandu dans presque tous les pays du Monde : le panneau (STOP).

Si un conducteur trouve ce panneau implanté à une intersection sur son chemin, il doit obligatoirement marquer l'arrêt avant de continuer.

Il indique que chaque véhicule sur cette voie n'est jamais prioritaire quelle que soit la direction qu'il souhaite prendre. Ce panneau est généralement utilisé pour contrôler le trafic aux intersections dangereuses, où le trafic est dense, mais pas suffisamment encombrées pour installer un système de feux de signalisation [24].

2.4.1.3 Les feux de circulation tricolores

Dans les systèmes de gestion du trafic, le système le plus connu et le plus utilisé dans le monde est le système des feux de signalisation.

Un feu de circulation routière est un dispositif permettant la régulation du trafic routier entre les usagers de la route, les véhicules et les piétons. Les feux destinés aux véhicules à moteurs sont généralement de type tricolore, auxquels peuvent s'ajouter des flèches directionnelles. Ceux destinés aux piétons sont bicolores et se distinguent souvent par la reproduction d'une silhouette de piéton.

Les feux tricolores pour cyclistes se distinguent par la reproduction d'une bicyclette. Généralement, un feu tricolore est composé d'un système électronique commandé.

Il est composé de trois couleurs principales :

- La couleur rouge indique l'obligation d'arrêt aux véhicules.

- La couleur orange qui ne dure que quelques secondes signale le passage du rouge au vert.
- La couleur verte indique aux véhicules qu'ils ont la priorité exclusive pour passer.

Ces couleurs ont été choisies parce qu'elles ont l'avantage d'être très distinctes.

Le système des feux de signalisation est le système le plus efficace pour la gestion du trafic car il évite tout malentendu entre les différents conducteurs au moment du passage [24].

Afin de régir le fonctionnement des feux de circulation, les contrôleurs utilisent des mécanismes combinatoires et temporels bien spécifiques.

- **Une phase :**

Représente un intervalle durant lequel une combinaison de feux verts autorisés par l'unité de gestion des conflits va être activée. Les phases sont déterminées à partir des mouvements que chaque direction peut effectuer.

- **Un cycle :**

Correspond à l'enchaînement d'un ensemble de phases. Ce cycle est typiquement fixé au minimum à 45 secondes et ne dure pas plus de 90 secondes pour éviter de perdre du temps à arrêter et redémarrer le trafic.

Classiquement, un cycle déroule l'ensemble des phases et mouvements possibles, de manière à ce que toutes les voies aient au moins une fois le feu au vert. Lorsque l'intersection est suffisamment équipée, cette règle n'est pas nécessairement applicable.

- **Un plan de feux :**

Correspond à la description d'un cycle donné, et définit les différentes phases que le cycle va dérouler ainsi que leur durée [26].

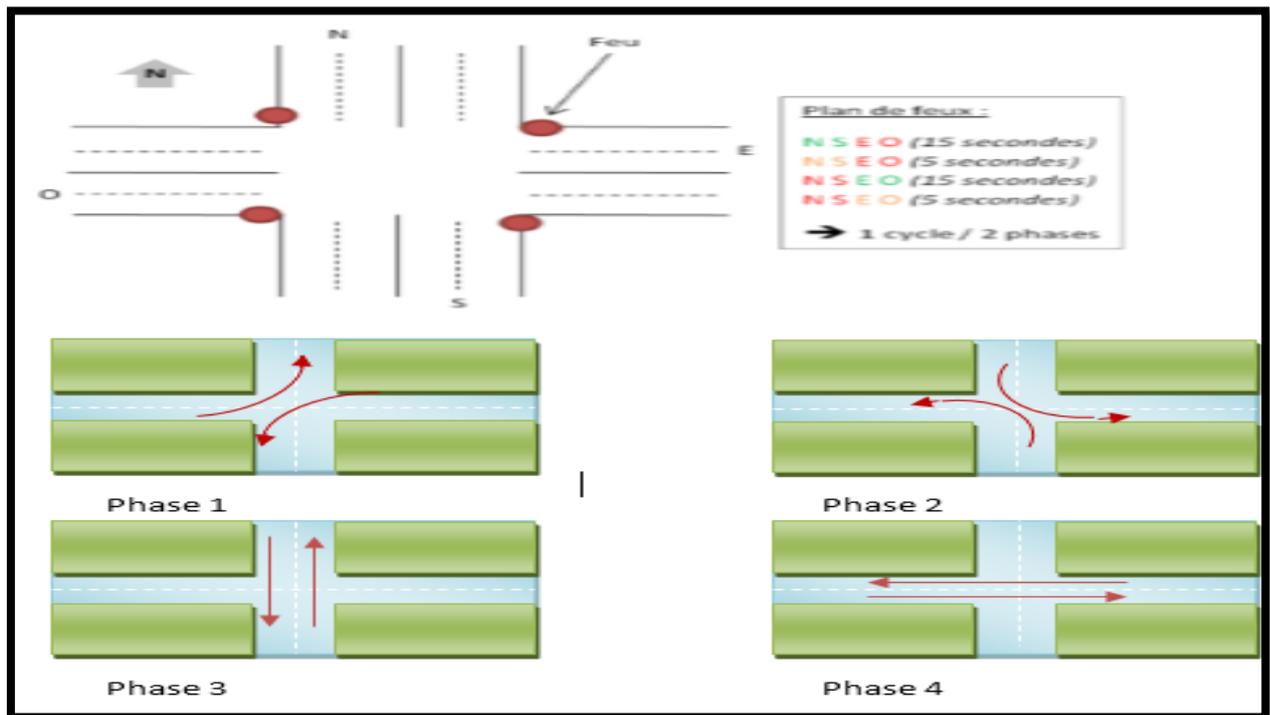


Figure II.5 : Exemple d'un plan de feu

2.5 Approche dynamique

Le développement du trafic routier urbain a amené à concevoir des systèmes intelligents. L'une des particularités de ces systèmes est leur besoin de dynamisme et de réactivité : afin de pouvoir agir sur une situation, ces derniers ont besoin de connaître des informations sur ce qu'il se déroule à des endroits bien précis, et prendre des décisions en conséquence.

2.5.1 Contrôle dynamique par l'infrastructure

La gestion des feux de circulation est un problème qui a commencé à être étudié au début des années 1970 avec l'apparition progressive de systèmes de gestion centralisés, en charge des carrefours d'une zone géographique donnée. Au fil des années, ces systèmes se sont diversifiés, et ont connu trois générations de contrôle.

Aujourd'hui, ces générations peuvent être utilisées chacune en fonction des moyens mis en place sur l'infrastructure, et de la connaissance de cette dernière :

- Première génération : contrôle à temps fixe

En fonction de l'heure et parfois du jour, le système va utiliser un plan de feux prédéfini.

- Deuxième génération : contrôle à temps dynamique :

Des détecteurs sont utilisés afin de recueillir les données du trafic toutes les X minutes. Ces données peuvent être utilisées afin d'optimiser ou mettre en place un plan de feu.

Par exemple, une fois les phases déterminées, l'un des enjeux va être de définir un temps de feu vert pour chacune d'entre elles : ce dernier est généralement constitué d'une valeur minimale et maximale, afin de ne pas provoquer des arrêts intempestifs ou d'engendrer un cycle trop long. Le feu vert minimum suffit uniquement si pendant son temps d'exécution, aucun nouveau véhicule ne franchit un détecteur.

- Troisième génération : contrôle à temps réel :

Reprend le même principe que la deuxième génération mais cette fois-ci en temps réel.

Notons que les deux dernières générations, qui introduisent un caractère dynamique au système, peuvent être chacune décomposées en deux types :

- Contrôle réactif : en fonction des données recueillies sur le terrain périodiquement (plusieurs minutes ou cycles), le système met en place une nouvelle configuration en réponse aux informations reçues. Cette méthode est le premier niveau de dynamique, et est simple à mettre en place, mais nécessite toutefois une très bonne connaissance du système afin d'être efficace. C'est également la première méthode à être apparue aux Etats-Unis à la fin des années 1980 avec l'apparition des **UTCS (Urban traffic control software)**.
- Contrôle adaptatif : ce type de contrôle va programmer dynamiquement les plans de feux en se servant des paramètres recueillis sur le terrain, ceci en calculant des valeurs telles que le temps de cycle, des phases ou encore leur ordonnancement. L'opération va être effectuée de manière adaptative, c'est à dire en quasi-temps-réel.

L'avantage de cette méthode est qu'elle peut s'adapter à de multiples situations, mais reste la plus compliquée à mettre en place (nécessité de cerner les informations à utiliser, comment les utiliser et se baser sur des théories parfois lourdes) [26].

2.5.2 Quatre approches de référence

Les approches de référence correspondent aux systèmes les plus déployés, car ils sont les plus anciens, mais aussi parce qu'ils sont simples et ne nécessitent pas beaucoup d'infrastructure comparés à d'autres. [25] A partir de 1960, plusieurs algorithmes ont été proposés pour améliorer le passage des flux de véhicules. Quatre systèmes différents, TRANSYT, SCOOT, SCATS et PRODYN, sont décrits pour montrer la diversité de ces approches classiques. Ils s'intéressent tous à la gestion du trafic dans les réseaux sans être centralisés (à l'exécution) pour autant [25]. Nous nous sommes appuyés dans cette section sur le livre [Mammar, 2007] [11] qui présente une synthèse de ces différents systèmes de gestion de trafic.

- **TRANSYT**

(TRANSYT) « TRAFIC Network STUDY TOOL » : est l'un des premiers systèmes proposés. Il repose sur une optimisation hors-ligne qui génère des plans de coordination optimaux entre les feux de signalisation d'un réseau pour une période donnée. TRANSYT exige beaucoup de paramètres d'entrées, comme par exemple la géométrie des artères des intersections, le débit des véhicules, le taux de véhicules sur chaque voie sortante de chaque intersection (fixé à l'avance), le temps de feu vert minimal, des plans de feux initiaux, et des valeurs initiales pour les durées des cycles et les déphasages.

A partir de ces paramètres, un modèle mathématique (calculé à partir de la dispersion des trains de véhicules) est simulé macroscopiquement et ses performances sont évaluées. Enfin vient l'étape de l'optimisation à l'aide d'un algorithme d'optimisation Hill-Climbing, c'est-à-dire améliorant progressivement la solution en modifiant légèrement la durée des feux verts et des décalages entre intersections adjacentes. Les plans récupérés à la fin sont des plans calculés pour une situation statique dans le temps qui dépend des paramètres d'entrée.

Pour résumer, TRANSYT est un système centralisé et n'est pas adaptatif au trafic. Une application réussie de ce modèle nécessite un calibrage approprié des paramètres du modèle. [24].

- **SCOOT**

« Split Cycle Offset Optimisation Technique », un système de contrôle à la fois réactif et adaptatif et entièrement centralisé, développé par le [TRL \(Traffic research laboratory, un centre de recherche anglais sur les transports\)](#). Dans ce système, l'ensemble des informations recueillies sur le terrain vont à un centre de gestion, qui s'occupe de traiter les informations et renvoyer des indications directement aux intersections.

Les véhicules sont détectés par des dispositifs pouvant être placés à divers endroits sur les voies au niveau des feux ou à une certaine distance afin de pouvoir mesurer le débit du trafic.

SCOOT mesure en permanence le volume de véhicules de chaque côté de l'intersection et change la durée des phases en fonction d'un indice de performance, calculé par rapport au délai d'attente moyen, de la longueur des queues et des arrêts sur le réseau. Ainsi, de la même manière que [TRANSYT](#), système sur lequel il est basé, SCOOT génère des plans de feux en fonction de la demande des utilisateurs (côté adaptatif). En plus de cela, le système utilise des informations en ligne issues de bases de données (historique ou autres, côté réactif). Notons au passage que SCOOT est l'un des principaux systèmes de contrôle adaptatif déployé mondialement [25].

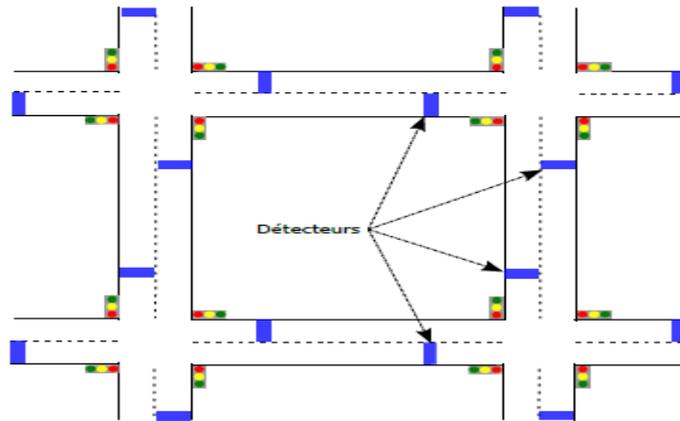


Figure II.6: Boucles magnétiques en amont des carrefours (SCOOT).

- **SCATS**

« Sydney Coordinated Adaptive Traffic System » qui a été à l'origine développé pour Sydney et d'autres villes Australiennes. Il est pour sa part entièrement adaptatif et utilise une notion de hiérarchie (ce qui forme une certaine distribution sur le réseau) : entre le recueil des données sur le terrain et le centre de contrôle, des contrôleurs intermédiaires sont insérés, permettant d'alléger la charge globale du système et d'avoir un contrôle découpé en plusieurs zones, l'ensemble des acteurs utilisant des communications synchronisées.

De manière similaire à SCOOT, ce système ajuste le temps des cycles et autres paramètres en fonction des données recueillies afin de diminuer le délai et les arrêts, mais n'utilise pas la même stratégie : les valeurs recueillies permettent la sélection de plans de feux parmi une large librairie, sur lesquels le système va se baser pour proposer des plans adaptés. De plus, contrairement à SCOOT, les détecteurs sont uniquement placés au niveau des feux de circulation [25].

- **PRODYN**

PRODYN [12][11] est un système décentralisé et adaptatif au trafic développé par le (CERT) Centre d'Etude et de Recherche de Toulouse en France.

L'objectif de PRODYN est d'optimiser la circulation en ligne en minimisant les retards aux intersections sur un horizon futur de 80 secondes. La fréquence de modification des phases et des durées des feux est fixée à 5 secondes (pas de discrétisation temporelle). Il utilise la programmation dynamique pour minimiser les retards en s'appuyant sur un modèle d'écoulement du trafic sur tout l'horizon. Ce dernier lui permet d'exprimer l'évolution des files d'attente en fonction des arrivées et des départs sur les tronçons. Pour les départs, comme SCATS et SCOOT, il utilise les données des capteurs installés sur les routes. Cependant, pour

les arrivées, il lui faudrait idéalement les données des départs des intersections en amont, eux-mêmes en cours d'optimisation, ce qui est impossible.

La solution choisie consiste à utiliser des prédictions établies au pas d'optimisation précédent tout en faisant l'hypothèse que la situation du trafic n'a pas beaucoup évolué. Etant donnée la complexité calculatoire exponentielle en fonction du nombre d'intersections, PRODYN optimise chaque intersection séparément en fonction des données reçues de ses voisines. Il est installé dans beaucoup de villes en France et en Belgique, mais il reste un système relativement complexe en termes de calcul et coûteux en communication.

2.5.2.1 Limites des approches de référence

Plusieurs aspects peuvent être perçus comme des limitations de ces systèmes. Nous présentons ici ceux qui nous paraissent les plus importants.

- Hypothèses simplicatrices

Le premier problème est que ces différents systèmes (ces quatre plus d'autres) font des hypothèses simplicatrices (prédiction du trafic pour TRANSYT, variation des durées de feux selon des pas prédéfinis pour SCOOT, bibliothèques prédéfinies de décalages et de durées de feu vert pour SCATS, etc..) qui ne reflètent pas forcément la réalité.

Ces hypothèses sont essentielles pour ces systèmes, sinon les plans des feux de signalisation deviennent complexes à calculer et donc impossibles à exécuter en temps réel.

- Approches autonomes pour véhicules non autonomes

Un autre problème de ces différents systèmes est qu'ils ont été conçus pour les véhicules conduits par les humains. En effet, l'utilisation de véhicules classiques (non autonomes) pose surtout problème parce que, si les feux évoluent (optimisation en ligne), il faut « idéalement » prévenir les véhicules pour qu'ils puissent adapter leurs profils d'accélération au mieux. En plus, nous avons remarqué que les optimisations proposées favorisent le plus souvent les routes principales, généralement les plus congestionnées, et défavorisent les routes secondaires, sachant que dans les grandes villes ces dernières sont devenues de plus en plus importantes dû à la saturation des axes traditionnels.

Enfin, d'autres aspects sont délaissés par ces systèmes. Par exemple, le trafic peut être affecté par des incidents (accidents, blocages des routes, etc..). Il est donc nécessaire d'avoir des approches robustes et flexibles pour une bonne gestion du trafic.

2.6 Synthèse des approches

Dans cette section, nous ne nous limitons pas à la synthèse des quatre approches précédentes, mais détaillons toutes les approches de gestion du trafic [20]. Nous décrivons, ici, les principales caractéristiques des approches rencontrées, puis nous donnons un tableau récapitulatif.

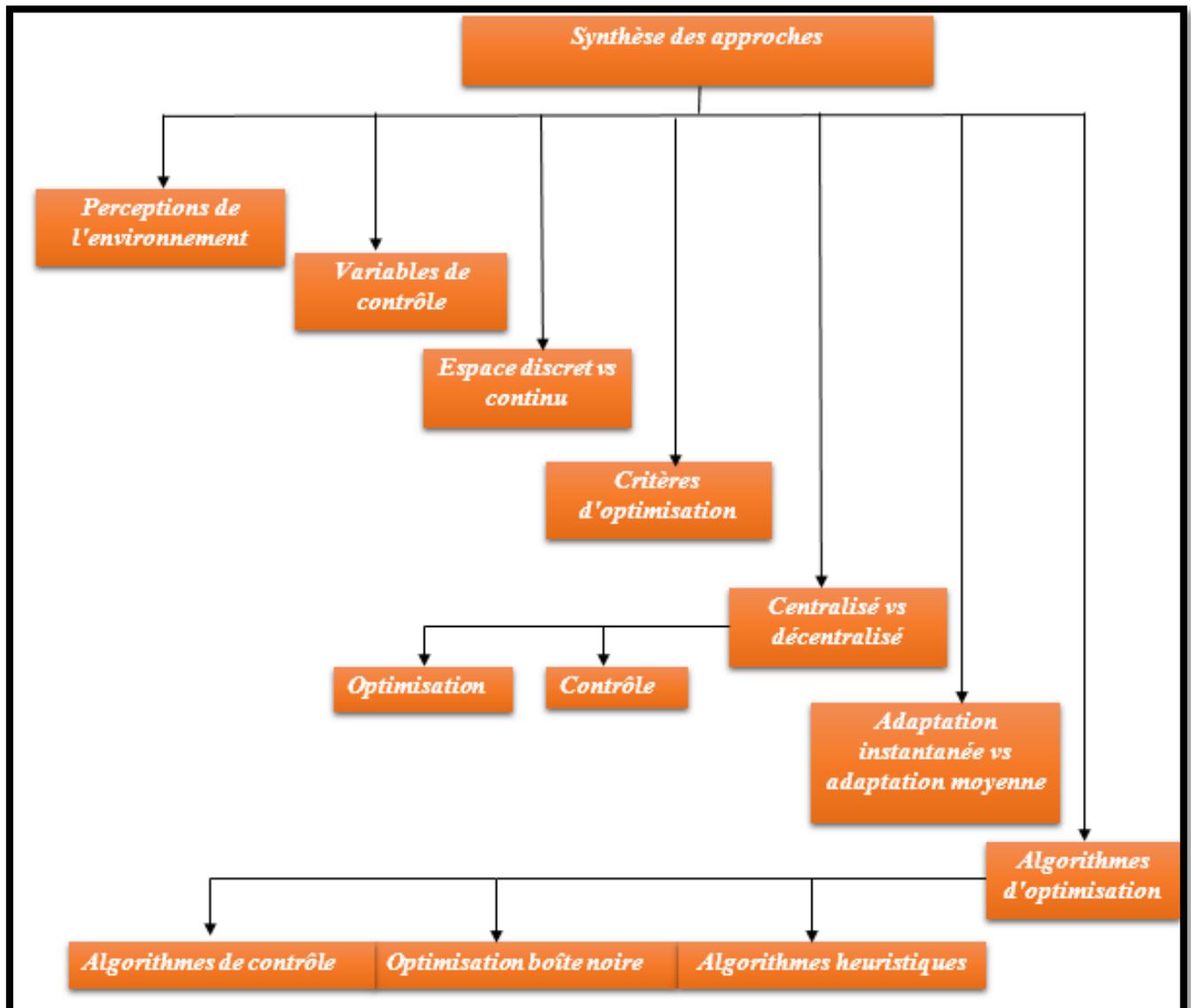


Figure II.7: l'organisation de la synthèse des approches.

2.7 Approche pour véhicules autonomes (IAV)

Nous considérons les solutions qui s'intéressent au contrôle de véhicules intelligents autonomes (IAV), Nous avons choisi de présenter les travaux spécialement dédiés aux IAV mais aussi les modèles qui, à notre sens, ne peuvent s'appliquer que sur ce type de véhicules.

2.7.1 Négociation véhicule à véhicule

L'une des premières approches pour véhicules autonomes a été proposée par [13]. Cette approche repose essentiellement sur la communication et la négociation entre les véhicules afin de déterminer l'ordre de passage et de sortie des intersections [13]. Dans ce travail, les auteurs modélisent une intersection comme un ensemble de trajectoires parcourues par les véhicules avec des zones critiques où on ne peut avoir qu'un seul véhicule à la fois. Dans la Figure 16, les points noirs représentent ces zones critiques.

L'idée de base de cette approche est l'utilisation d'un protocole de communication type Anneau à jetons (Token-ring) entre les véhicules autonomes dans une zone déterminée autour de l'intersection. Ainsi, les informations essentielles sur la position, la vitesse et l'accélération des véhicules sont échangés en vue d'une décision sur l'opportunité ou non de s'engager dans l'intersection. Pour une exécution complètement décentralisée, le véhicule le plus proche de l'intersection et qui est en possession du jeton, prend le rôle de l'agent manager.

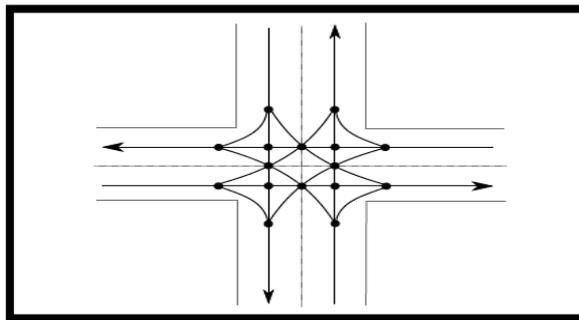


Figure II.8: Modélisation de la géométrie des intersections

Ainsi, il décidera de l'ordre de priorité de sortie de tous les autres véhicules de l'intersection. Sa tâche se termine juste avant de quitter l'intersection quand, à son tour, il doit alors passer le jeton à un autre véhicule. Par ailleurs, l'évitement de collision est assuré au moyen d'algorithmes fondés sur des sémaophores qui permettent à un seul véhicule à la fois de rester dans un segment critique d'intersection.

Cette approche est connue pour ses limites pratiques. Premièrement, elles dépendent clairement du nombre de véhicules voulant négocier leurs passages par l'intersection, car l'augmentation du nombre des véhicules implique une complexité croissante lors des négociations. Deuxièmement, elle suppose que toutes les intersections ont la même géométrie, ce qui n'est pas réaliste dans le contexte actuel (il n'y a pas que des routes avec une seule voie et des angles de 90°). Si nous avons plus de voies, nous aurions plus de calculs et plus de temps de traitement.

Enfin, cette approche dépend aussi des capacités de communication : Peut-on assurer la réception de tous les messages envoyés par les véhicules ?

2.7.2 Approche par réservations

L'approche par réservations a été introduite par (Stone K. e., 2004) [14]. Elle se passe des feux de signalisation conventionnels et à la place elle introduit un agent manager au niveau de chaque carrefour. Chaque véhicule voulant passer doit réserver un temps de passage et une route à prendre, et respecter un certain nombre de règles. Parmi ces règles, il doit garder une vitesse constante fixée à l'avance à l'approche de l'intersection et il ne doit pas changer de route (c'est à dire que les véhicules vont toujours tout droit). La réservation se déroule comme suit : Le véhicule informe l'agent manager en envoyant un message contenant sa vitesse, sa direction, son accélération maximale, sa décélération maximale, le temps qu'il lui reste avant d'arriver à l'intersection (estimé par l'IAV) et d'autres propriétés du véhicule. Chaque agent manager connaît les informations nécessaires sur la géométrie de son intersection. Il discrétise alors la zone de croisement sous forme de grille, comme le montre la Figure 17, an de pourvoir simuler la trajectoire des véhicules à l'intérieur de la zone de conflit.

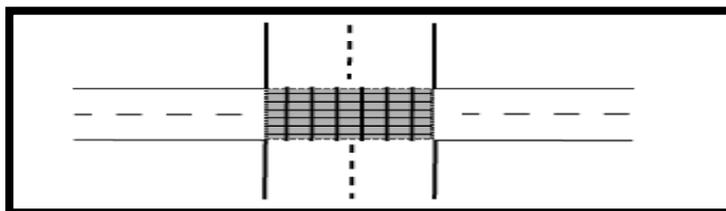


Figure II.9: Discrétisation de la zone de croisement par l'agent manager

Ensuite, l'agent manager simule le trajet du véhicule avec les autres trajets déjà réservés en fonction du temps (l'espace qui va être réservé). Si le trajet en question est en conflit avec d'autres réservations (c'est à dire l'espace discrétisé est réservé au même moment par deux véhicules), alors la réservation est rejetée. Le véhicule doit alors décélérer et demander une autre réservation. S'il arrive à l'intersection sans avoir une réservation, alors il doit s'arrêter. Si la réservation est acceptée alors le véhicule peut passer.

L'avantage de cette approche est que, si on a plusieurs véhicules qui veulent passer et que leurs trajectoires ne se croisent pas, alors on peut satisfaire tout le monde avec un passage simultané. Cependant, dans le cas contraire, on est obligé de donner la priorité à un véhicule par rapport aux autres, ce qui revient à utiliser le principe des feux de signalisation.

2.8 Résumé des solutions citées

Dans ce tableau , adapté d'après (Bazzan), les travaux sont classés d'abord selon que l'approche s'intéresse à:

- ✓ une intersection isolée à la fois
- ✓ un groupe d'intersections
- ✓ l'optimisation globale du réseau de transport.

On y distingue aussi trois grandes classes d'approches :

- ✓ les approches de référence
- ✓ les approches avancées divisées en trois sous-classes :
 - ✓ les approches fondées sur la physique, les mathématiques ou les automates cellulaires
 - ✓ les approches multi-agents (SMA)
 - ✓ les approches fondées sur l'apprentissage
- ✓ et les approches pour véhicules autonomes (IAV).

Nous pouvons constater à partir du tableau qu'il manque beaucoup de travaux de recherche s'intéressant à l'optimisation globale des intersections, et surtout il n'en existe aucune à notre connaissance dédiée aux véhicules intelligents autonomes. C'est pour cette raison que nos travaux, dans les chapitres suivants, vont désormais s'intéresser au développement d'idées pour l'optimisation du réseau de transport d'IAV via les feux de signaux tricolores.

	<i>Approche. Référence</i>	<i>Approche. Avancées</i>			<i>Approche. IAV</i>
<i>Intersections Isolées</i>	PRODYN	Physique	SMA	Apprentissage	(Stone K. e., 2004)
		(Gershenson, 2004)	(L.Bull, 2004) (A. Doniec, 2005)	(B.C. Da Silva, 2006)	(G.Balan, 2006) (R. Naumann, 1997)
<i>Intersections connectées (optimisation. locale)</i>			(Kosonen, 2003) (Sánchez, 2007) (France, 2003) (Bhourri, aug 2011) (Hounsell, 2012) (Rochner, 2006)	(E. Camponogara, 2003) (Oliveira, 2004) (Steingrover, 2005) (Wiering, 2000) (Ferreira, 2000) (S.Richter, 2007)	
<i>Optimisation . globale</i>	TRANSYT SCOOT SCATS	(Brockfeld, 2001) (Köhler,2005) (Mohring, 2006)	(D.De Oliveira, 2006) (D. De Oliveira, 2007)	(Bazzan, 2005)	

Tableau 3: Résumé des solutions citées

3. Conclusion

Ce chapitre donne un aperçu du concept de circulation routière, les composants et les variables élémentaires de cette dernière ainsi il représente la congestion et les raisons qui la conduisent Sans oublier les conséquences qui résultent.

Notamment les méthodes pour gérer ce problème « la circulation routière » tel que le code de la route les feux tricolores dans les intersections... et les approches existantes pour la gestion d'intersections. Enfin, nous donnons une synthèse des approches dans un tableau récapitulatif.

Chapitre III : NetLogo &UML « Aperçu et Initiation »

1. Introduction

Le but de la simulation est de reproduire un phénomène afin de Tester des hypothèses permettant d'expliquer le phénomène (définition d'un modèle) et Prévoir l'évolution de ce dernier.

Dans ce chapitre, nous présentons les résultats de l'étude que nous avons faite sur une plateforme multi agent générique, nommé NetLogo, afin de la découvrir et réaliser le lien entre cette plateforme et le langage de modélisation UML.

L'objectif de ce chapitre est de familiariser le lecteur avec notre réflexion méthodologique exploitée dans le chapitre suivant, en termes de modélisation et d'implémentation pour concrétiser notre simulateur de coopération ambitionné.

2. Simulation multi-agent

Les SMA apportent une solution radicalement nouvelle au concept de simulation, en offrant la possibilité de représenter directement et sous forme informatique les individus, leurs comportements, leurs interactions et le phénomène qui est le fruit de ces interactions [6].

Les simulations multi-agents sont des simulations discrètes. Il existe également des simulations discrètes par pas de temps et des simulations discrètes par événements. Pour la première catégorie, les pas de temps se définissent comme des intervalles de temps réguliers qui sont parcourus au cours de la simulation. Lors de son incrémentation, tous les agents sont mis à jour. Alors que les simulations discrètes par événements utilisent des listes ordonnées d'événements qui sont traités chronologiquement [15].

L'intérêt de la simulation multi-agents c'est qu'elle permet de représenter plusieurs niveaux de granularité dans un système homogène tout en préservant l'hétérogénéité dans ce système [6]. Elle est également adaptée à la reproduction de phénomènes naturellement localisés et distribués [9].

La simulation multi-agents est donc une technique qui permet de reproduire des phénomènes complexes qui sont difficiles à appréhender de manière analytique. Actuellement, les concepts de la simulation multi-agents sont appliqués pratiquement dans tous les domaines [9].

2.1 Principe de la simulation multi-agents du trafic routier

La simulation multi-agents, avec les possibilités qu'elle offre, s'adapte aux problèmes posés par la simulation du trafic routier [9]. La simulation multi-agents consiste à élaborer le modèle du trafic en se basant sur les principes des SMA. Il est donc nécessaire de modéliser l'environnement, les véhicules en tant qu'agents et leurs comportements de telle sorte que cette modélisation produise la simulation du phénomène du trafic.

2.2 Outils de développement des SMA

Pour assister la modélisation et le développement des SMA, il existe des environnements de développement intégrés (EDI) et des bibliothèques dédiées.

Les bibliothèques tendent à faciliter la programmation orientée agents avec les langages existants tels que Java et C++. Citons comme exemples de bibliothèques : Swarm, Repast, Cybele, JAS, Jason, MASON, SARL. Ces outils fournissent des facilités de simulation, cependant, ils sont peu accessibles et peu utilisés, car ils nécessitent une connaissance sur le langage de programmation original et sur l'interface de ces outils. Les bibliothèques sont rapides et puissantes mais moins intuitives et plus difficiles à apprendre et à utiliser que les EDI [4].

Les EDI sont à la fois des environnements de modélisation et de simulation, ce qui permet la construction du modèle, son implémentation et l'exécution de la simulation à l'aide d'un même outil et sous la même interface graphique. Ces environnements peuvent fournir de plus des niveaux d'abstractions variants [4]. Il existe de nombreux EDI pour la modélisation et la simulation multi-agents tels que : NetLogo, StarLogo, Breve, SB-MASE, SeSAM, JACK, Jade, 2APL.

3. Choix de la plate-forme NetLogo

Pour les besoins de nos travaux, nous utilisons un environnement de développement multi-agents, qui est jugé plus facile et intuitif à exploiter.

Pour le choix de l'EDI, nous mettons en considération certains critères :

- La gratuité de l'outil et des supports.
- La modélisation de l'environnement et des agents.
- Les types de simulation pris en compte.

- La facilité et la puissance du langage de programmation.
- L'ergonomie de l'interface utilisateur.
- La disponibilité de documentation.

De ce fait, notre choix se fixe pour NetLogo, téléchargée du site¹. Il s'agit d'un environnement de modélisation pour la simulation de phénomènes naturels et sociaux.



C'est une plateforme de simulation multi-agent de référence, très utilisée, dont la dernière version date de décembre 2016².

De plus des critères cités ci-avant, Netlogo possède les avantages suivants :

- Il propose un éventail de modèles pour la simulation des systèmes complexes.
- Il distingue plusieurs types d'agents dans un même modèle.
- Il manipule un grand nombre d'agents.
- Il est facile et rapide à installer et à assimiler.
- Il est extensible.

3.1 Présentation générale

NetLogo est un environnement de modélisation programmable et paramétrable créé en 1999 par Uri Wilenski et son équipe du « Center for Connected Learning and Computer-Based Modeling » de la « Northwestern University » à Evanston, Illinois, Etats-Unis.

NetLogo permet de simuler des phénomènes naturels et sociaux et il convient tout particulièrement à la modélisation de systèmes complexes évoluant au cours du temps. Il est suffisamment simple à utiliser mais il est aussi assez performant pour servir d'outil puissant pour des chercheurs dans de nombreux domaines.

NetLogo est accompagné d'une documentation complète et de nombreux tutoriaux explicatifs. Il est également livré avec une bibliothèque de modèles de simulations fonctionnelles pouvant

¹ <https://ccl.northwestern.edu/netlogo/index.shtml>

² <http://www.filecluster.fr/logiciel/NetLogo-170489.html>

être utilisées tel quelles ou modifiées selon les préférences de l'utilisateur. De nombreuses sessions d'apprentissage à base de modèles utilisant NetLogo sont également en cours de développement [18].

3.2 Caractéristiques

NetLogo offre des fonctionnalités diverses lui permettant d'être bien adapté à la modélisation de systèmes complexes composés de milliers d'agents agissant en parallèle.

Le contrôle dans NetLogo est distribué. Les traitements locaux se font pour chaque agent et ils communiquent directement entre eux et le traitement de l'environnement se fait par un contrôleur global. NetLogo est un outil multi plateformes, il tourne sur les systèmes d'exploitation Mac, Windows, Linux et autres. Il est caractérisé par une interface simple et facile à comprendre et à utiliser (figure III.1).

Les interfaces des applications développées par NetLogo peuvent contenir des contrôles d'entrée (boutons, curseurs, commutateurs, sélecteurs) et de sortie (traceurs de courbes, moniteurs, boîtes de texte, notes, etc.). De plus, l'environnement de simulation peut être en 2D ou en 3D.

Le temps de simulation dans NetLogo est discret. Les états des agents et de l'environnement sont automatiquement mis à jour après chaque pas de temps.

La vitesse de simulation peut être ajustée par l'utilisateur pour l'accélérer ou la ralentir.

Il est même possible de désactiver la vue dans le but d'augmenter la vitesse de simulation au maximum.

NetLogo possède aussi de moniteurs qui permettent d'inspecter et de contrôler les agents.

Le langage de programmation utilisé par NetLogo est Logo, il est entièrement programmable et a une structure simple. Il offre la possibilité de créer des modèles personnalisés, en utilisant un vocabulaire important de primitives et en employant divers types d'agents : mobiles et stationnaires.

NetLogo offre des fonctions d'exportation et d'importation de données, de sauvegarde et de restauration de l'état d'un modèle et de création et d'enregistrement d'une animation. De plus, les modèles développés sont enregistrables sous forme d'applets pouvant ensuite être intégrés dans des pages web [18].

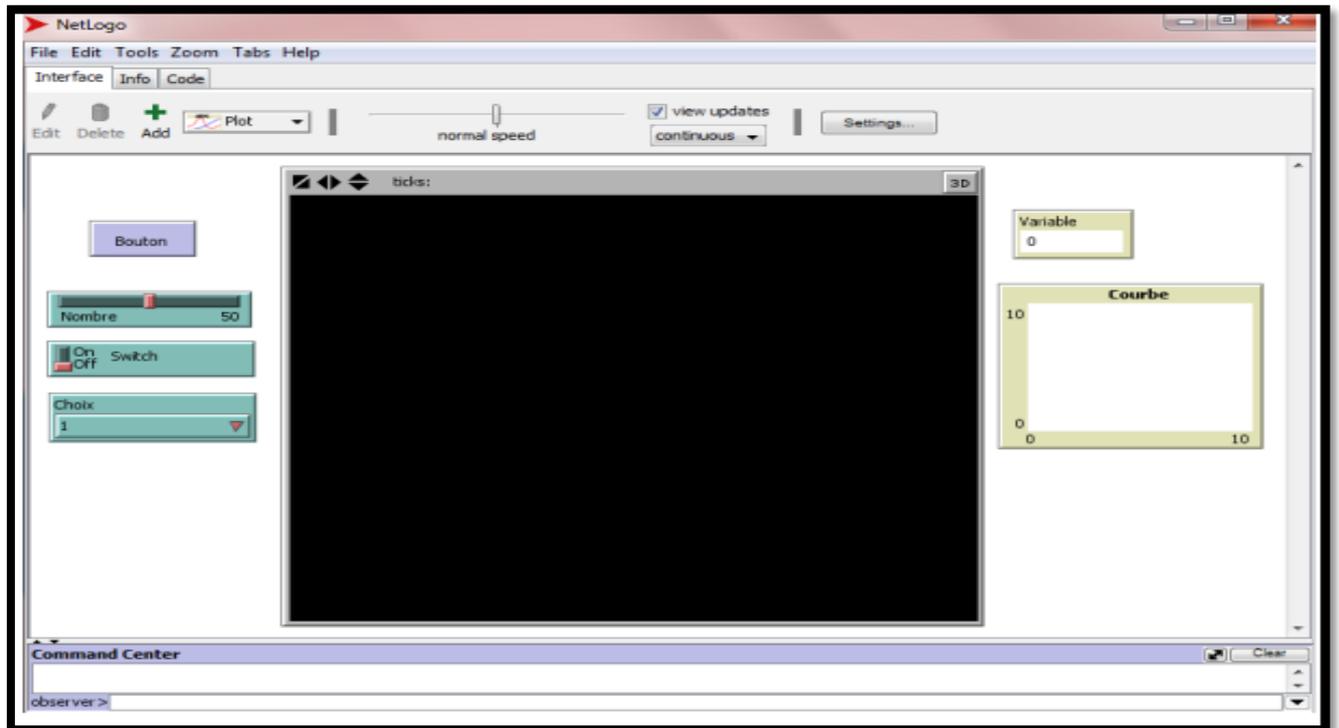


Figure III.1 : interface de NetLogo

Netlogo est formée de trois panneaux :

- **Interface** : une interface d'interaction avec le modèle et de représentation des résultats et des états du « monde »
- **Information** : une interface d'information des utilisateurs du modèle
- **Procédure** : une interface pour écrire les procédures du modèle (caractéristiques et comportements des agents et du système, leurs interactions). Les choses importantes se passent dans « Procédures » et les choses amusantes dans « Interface ».

3.3 La classification des agents

L'environnement de NetLogo où se déroule la simulation est appelé World. Ce monde est constitué d'un ensemble d'agents dont les activités s'exécutent simultanément. Il existe quatre types d'agents, chacun à sa propre fonction et des commandes propres à sa catégorie :

- **Turtle** : il représente l'agent mobile, qui se déplace dans le World.
- **Link** : c'est un lien qui connecte deux agents (Turtles) et permet de représenter les relations entre eux.

- **Patch** : c'est une cellule sur laquelle les agents (Turtles) peuvent se situer et se déplacer. L'ensemble des cellules se trouvent dans le World.
- **Observer** : il présente le contrôleur global du modèle et de l'affichage. Il aperçoit le monde de l'extérieur et peut créer des agents et contenir des variables globales.

Les agents ont plusieurs propriétés intégrées, telles que les coordonnées x et y, la couleur et l'étiquette. Les Turtles possèdent aussi un nom, une forme et des dimensions qui permettent de les distinguer physiquement.

Les fonctionnalités des agents sont données sous forme de procédures. La communication entre agents se fait en demandant à l'autre d'exécuter une action.

3.4 La mise en œuvre de la simulation

La mise en œuvre de la simulation dans NetLogo se déroule en quatre étapes :

- **Premièrement** : il faut décrire l'interface qui contient tout ce qui concerne la visualisation, ainsi les paramètres que l'utilisateur pourra changer.
- **Ensuite** : on peut décrire des variables globales, et on peut personnaliser les agents en définissant leurs propriétés.
- **Troisièmement** : il faut décrire ce qu'il 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 Setup.
- **Finalement** : on doit décrire ce qu'il se passera dans chaque pas de simulation, et décrire ce que la plateforme affichera aux utilisateurs, normalement cela peut être fait en utilisant le bouton Go.

3.5 Modèle Netlogo

Un modèle Net Logo typique est composé de plusieurs blocs :

3.5.1 Interface graphique

Interface qui contient des éléments qui permettent à l'utilisateur de :

- fixer les valeurs des paramètres du modèle.
- observer les sorties, période par période.

3.5.2 Déclaration

Déclarations des variables globales, des types d'agents et des variables spécifiques de chaque type d'agent.

Déclaration des variables globales :

Globals [nbr cars]

Les variables fixées dans l'interface sont automatiquement des variables globales

Création des types d'agents

Breed [cars car]

Création des variables individuelles des agents

Cars-own [V-max] ; chaque voiture à une vitesse max.

3.5.3 Procédure Setup

Préparer le modèle pour l'exécution par procédure (en même temps un bouton 'setup').

To setup ; execute par le bouton « setup »

(Initialisations, instructions)

End

- **Initialisation** des variables globales (lecture automatique au début de chaque exécution à de l'interface graphique ou fixation de leur valeur)
- **Création** des populations de chaque type d'agents et initialisation de leurs variables individuelles
- **initialisation** du compteur de périodes et des sorties diverses (graphiques et autres) de la simulation

3.5.4 Procédure GO

Regroupe toutes les opérations qui ont lieu pendant une « période » d'exécution du modèle et qui incrémente le compteur de périodes (ticks).

3.5.5 Procédures complémentaires

- qui gèrent les comportements des différents types d'agents.
- qui collectent des statistiques sur ces comportements et sur les résultats agrégés du modèle
- qui actualisent les sorties, notamment les graphiques.

On utilise des commentaires pour documenter le modèle « ; ceci est un commentaire ».

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 va expliquer deux ateliers ([David.Sheeren, 2009](#))[16] et ([F.Amblard, 2009](#))[17] retrouvés dans la littérature multi agent, afin d'éclairer les relations entre ces deux langages orientés objet (UML) et orientés agent (NetLogo).

4.1 Atelier 1 [16]

Dans l'atelier 1, [David.Sheeren](#) à présenter et expliquer une modélisation UML pour un phénomène collectif proie-prédateur ([Wilensky, 2006](#)) (loup-moutons), Sachant que ce modèle est déjà formalisable sur NetLogo (dans la bibliothèque NetLogo). Après la description du phénomène comme suit:

«Dans une prairie, se trouvent des moutons et des loups: les moutons et les loups se déplacent, si un mouton rencontre de l'herbe, il la mange. Si un loup rencontre un mouton, il le mange. Dès que le loup ou le mouton a dépassé son seuil de production, il se reproduit. Notons que l'herbe se régénère comme il annonce [D.Sheeren](#) ».

Le chef d'atelier [D.Sheeren](#), et après une analyse du phénomène proie-prédateur, une analyse mentionnée par des mots en italique dans notre description précédente, les mots en italique représentent soit des agents, des objets, des attributs/méthodes pour les agents.

Pour [D.Sheeren](#), est pour modéliser le composant spatial prairie il préconise la représentation de la **Figure III.2 (c)**, qui préserve les instances, seuls les états qui changent, contrairement au **Figure III.2 (a)** où il n'y a pas de représentation intra-prairie (avec et sans herbe), aussi bien la

Figure III.2 (b) où la dynamique est représenté explicitement, c à d création/destruction d’instance en continue.

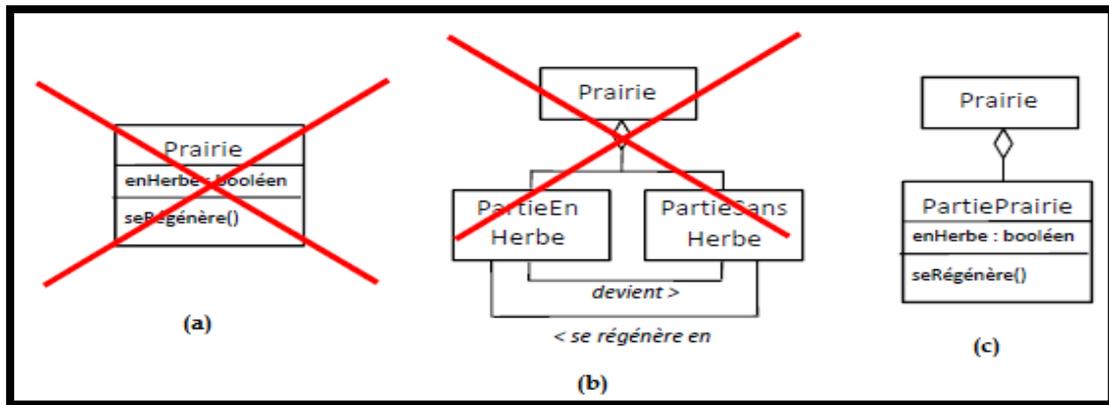


Figure III.2 : Différentes solutions pour modéliser la composante spatiale prairie

Ensuite, pour modéliser l’ensemble des agents mobile de ce phénomène, il préconise non pas rassembler les agents mobiles dans un seule classe comme dans la Figure III.3 (a), mais il recommande d’éclater deux classe différentes comme dans la Figure III.3 (b), par ce que il y a deux comportement différents (les loups et les moutons). Il y a trop de différences entre les loups et les moutons, il faut les éclater en deux classes différentes, une pour les moutons et l’autre pour les loups, malgré que les attributs et les méthodes sont similaires.

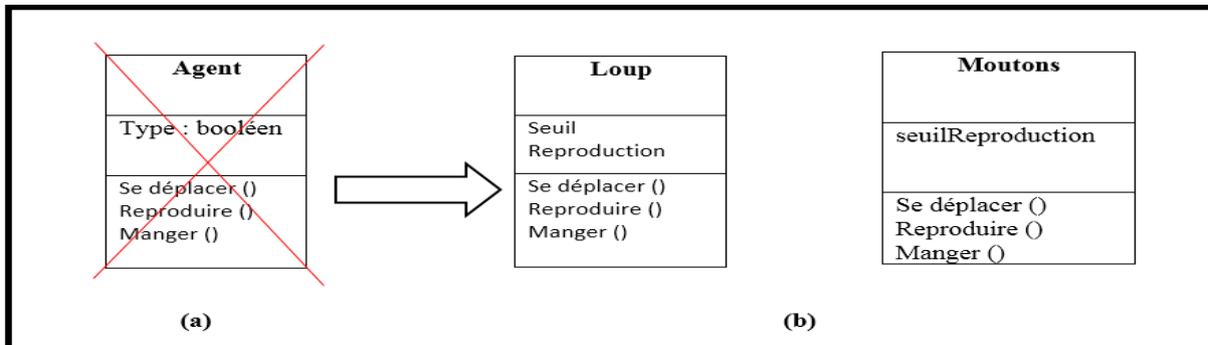


Figure III.3 : l’éclatement de deux composants, selon leur espèce

Pour modéliser le phénomène proie –prédateur, en termes de dépendances entre les classes, David a appuyé sur un diagramme de classe UML présenté dans (figure III.4), quant à exhiber la relation entre les trois composants du phénomène.

À la suite il propose un diagramme de classe générique, pour rapprocher le modèle d’implémentation NetLogo. Ce qu’il souligné sur ce juxtaposition de ce premier atelier], il reste en terme abstraitif seulement, et non pas en terme d’étapes organisé comme dans le deuxième atelier.

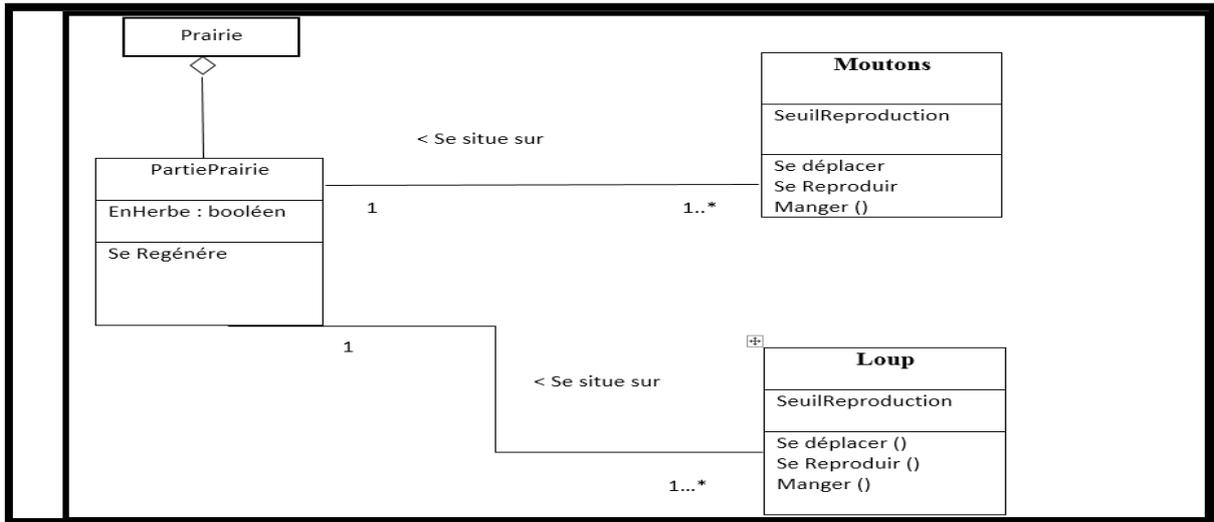


Figure III.4 : Le diagramme de classe du phénomène proie –prédateur

Pour modéliser la dynamique du phénomène dans ce premier atelier, le chef d’atelier recommande fortement l’utilisation du diagramme d’activité, pour de montrer non seulement l’ensemble d’activité réaliser par l’ensemble des agents mobiles (loups, moutons), Mais il va aller jusqu’à l’abstraction du squelette du programme Netlogo principale, comme suivant :

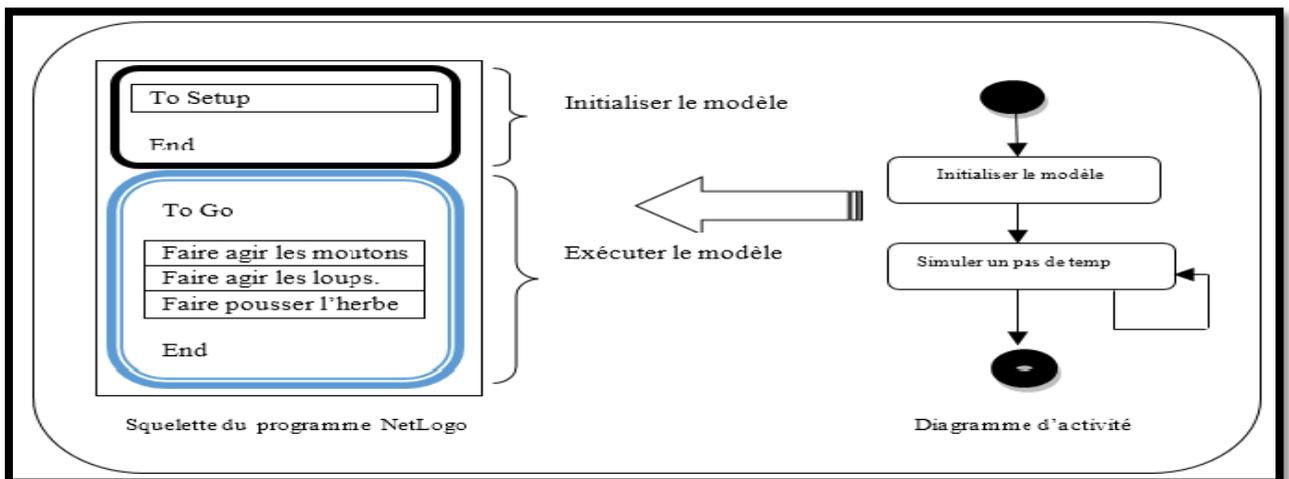


Figure III.5 : Concordance de diagramme activité & programme NetLogo.

Selon David, le dynamisme de comportement des loups/moutons, peut être représenté par les diagrammes d’activité de la Figure III.6.

Ce même diagramme est appliqué aux loups, mais avec le changement au niveau de condition (présence des moutons), sachant que même ces sous activités (manger, reproduire...etc) sont modélisés par des diagrammes d’activité d’UML pour les décortiquer.

Dans ce premier atelier, ce qu’on peut dire comme premier constat, c’est que la modélisation UML du phénomène en terme de diagramme de classe et d’activité, produit un outil

d'annotation puissant pour nous, afin d'éclairer un petit peu la structure de phénomène, les comportements (théoriquement), la dynamique et la mobilité des agents...etc.

Hélas ces différentes modélisations UML, ne s'appuient pas sur une théorie formelle, ou un outil pragmatique pour valider ces modèles, il reste seulement des diagrammes explicatifs et illustratifs.

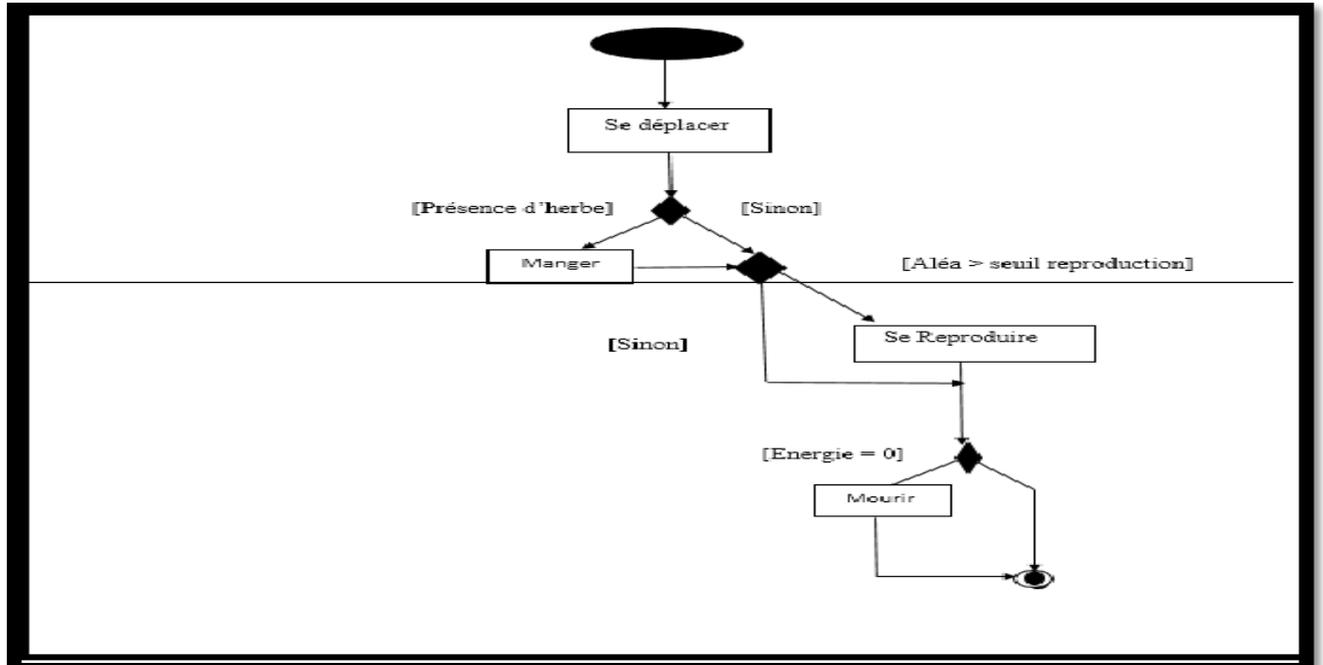


Figure III .6 : Diagramme d'activité pour les moutons.

4.2 Atelier 2 [17]

Passant maintenant au deuxième atelier, à ce stade [Amblard](#) et [Marilleau](#) ne s'arrêtent pas à une analyse, conception UML, mais il étend jusqu' à une proposition d'une démarche de passage d'un langage UML vers la plateforme NetLogo.

Au début de cet atelier, l'immense décalage existant entre la représentation conceptuelle de la tâche et la représentation d'implémentation des systèmes modélisés. Ce décalage a poussé [Amblard](#) et [Marilleau](#) d'exprimer et de s'articuler le phénomène souhaité modélisé sur des modèles conceptuels particuliers, un modèle qui doit être proche du modèle opératoire (d'implémentation) en terme d'abstraction, de simplification, de sélection des concepts pertinents.

Avant d'énumérer les étapes de la démarche de passage d'UML/NetLogo d' [Amblard](#), il est utile d'exhiber l'ossature NetLogo comme il le présente dans cet atelier.

Cette ossature NetLogo présentée par [Amblard](#), illustrer par un diagramme de classe UML, nous permet d'agencer mentalement, tout ce qui a été dit au début de ce chapitre, édifier particulièrement les différents composants NetLogo.

Dans ce qui suit, nous présentons les étapes de la démarche de [F.Amblard](#), [N.Marileau](#), qu'on peut situer comme une bonne charte pour notre approche proposée dans le chapitre quatre suivant, et qui à notre avis, constitue une remarquable assiette, à partir duquel nous pouvons implémenter notre simulateur. Cependant ce qu'il faut noter ici, c'est que le Choix d'explication et de spécification du problème qu'on souhaite modéliser, est directement influencé par les contraintes techniques imposées par NetLogo, comme le mentionne [Amblard](#).

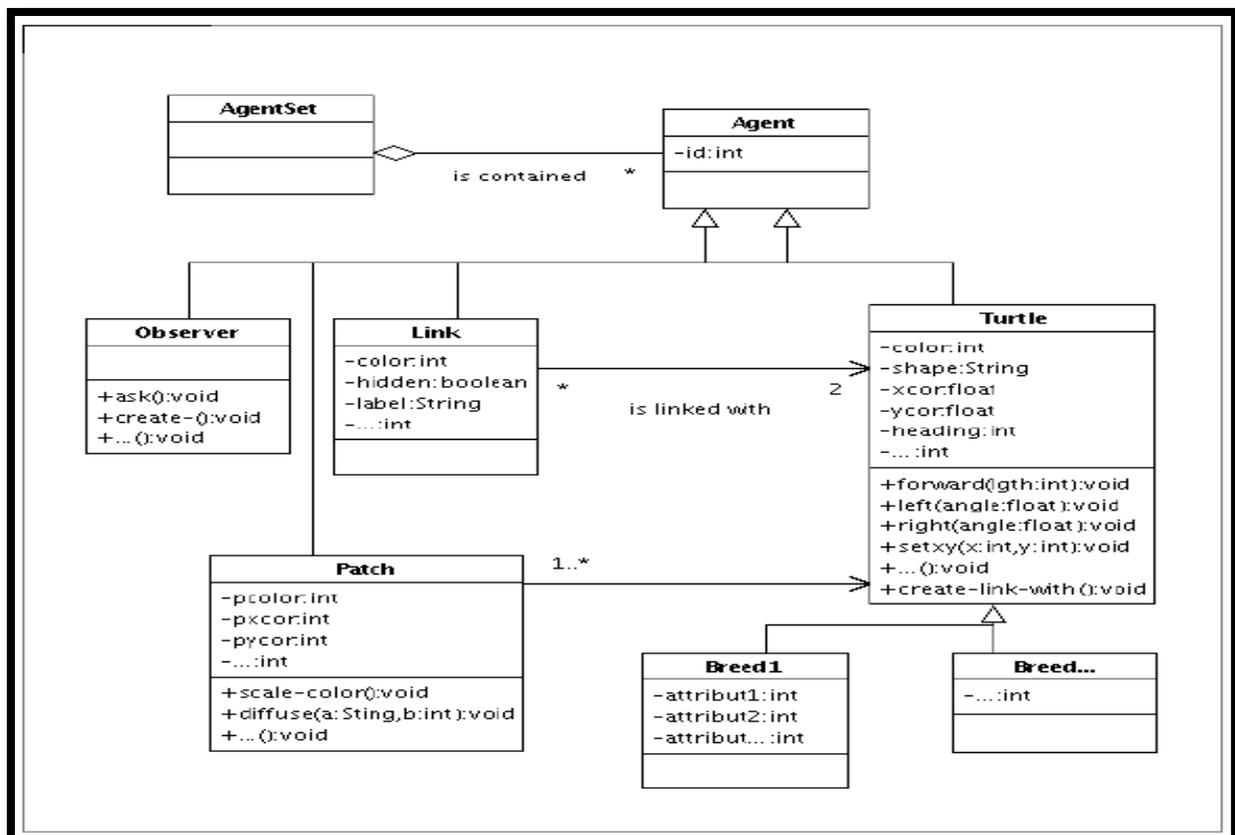


Figure III.7 : Une Ossature Netlogo.

4.2.1 Les étapes de la démarche [17]

- **Identifier les variables globales** : à partir des diagrammes de classe et d'activité accomplie. Essaye d'extraire les variables globales, qui peuvent être partagés par l'ensemble de composants du phénomène souhaité modéliser, par exemple le variable temps pour notre modélisation, ce variable globale indique le temps consommé.

- **Identifier les 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.
 - 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 de patch seulement, s'ils sont immobiles. Où même des turtles à l'exclusion des primitive de mobilité (**Forward...**etc).
- **Organiser les entités entre elles** : c'est l'étape clé de ce passage inspiré, respectent les deux contraintes suivant :
 - Un seul cran/degré 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 :

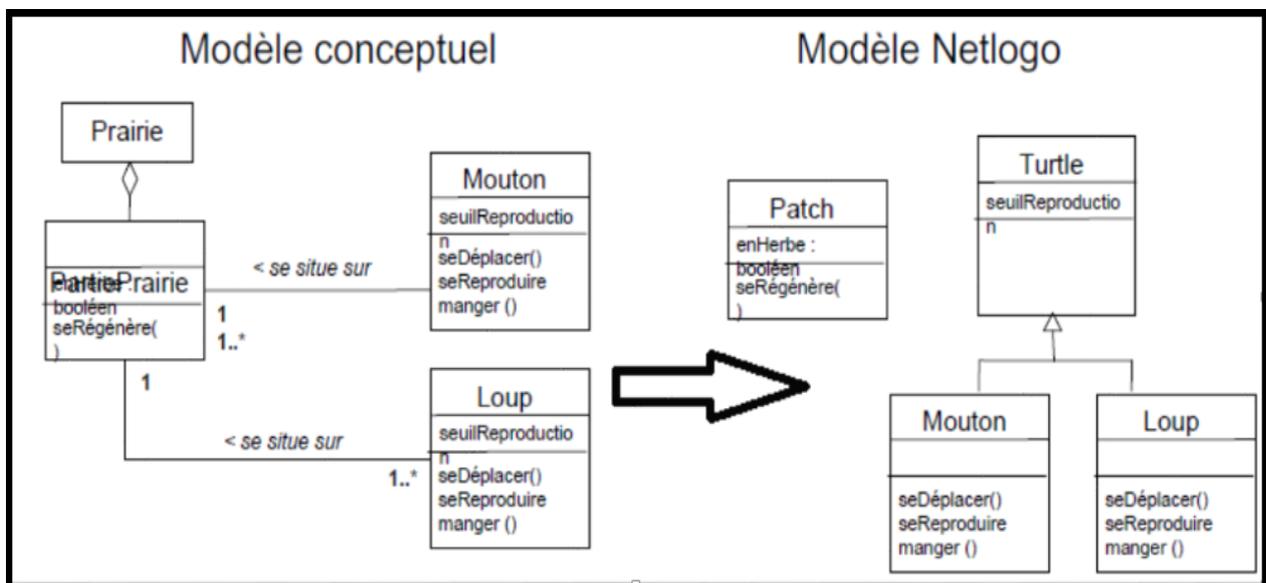


Figure III.8 : Exemple de passage UML / NetLogo

5. Limites de NetLogo

- Le paradigme de programmation utilisé n'est pas orienté objet, on perd ainsi des avantages de celui-ci. (La modularité par exemple).

- NetLogo ne s'adresse pas aux informaticiens en premier lieu, le langage utilisé est pour nous en tant qu'informaticiens relativement pauvre, on peut toujours trouver des moyens pour implanter ce qu'on veut mais cela peut être, des fois, un peu gênant, par exemple il ne gère pas les structures de données classiques comme implémenter un parcours de liste.
- La plateforme impose le modèle d'agents situés dans un environnement, donc tout modèle implémenté avec NetLogo doit être conforme avec celui-ci [20].

Avant d'achever ce chapitre trois, et après la présentation du langage/plateforme Netlogo et les deux ateliers multi agent [17] [16], on peut dire que :

NetLogo présente un langage et une plateforme pour développer des applications multi agents, ceci est découlé essentiellement de ces différentes primitives, offertes par cette plateforme, en terme de mobilité surtout de ces agents incarné par des turtles, et en terme de présentation de l'environnement, ...etc.

6. Conclusion

Dans ce chapitre, nous avons exposé une étude et un apprentissage que nous avons accompli sur la plateforme générique de modélisation/simulation multi-agents Netlogo.

Le but de cette expérimentation est de comprendre, animer, et saisir les différents types d'agents qui constituent cette plateforme, ainsi que la structure générale d'un modèle multi agent Netlogo.

Ainsi nous avons sommairement expliqué deux ateliers ([F.Amblard, 2009](#)) ([David.Sheeren, 2009](#)) retrouvé dans la littérature multi agent. A travers ces deux ateliers nous avons d'un côté montré le lien entre UML et Netlogo, et d'un autre côté, envisagé une démarche théorique quant à décortiquer la problématique d'optimisation de la circulation routière à travers la plateforme Netlogo.

Le chapitre suivant est consacré à la démarche usitée pour illustrer notre modèle multi agent, ainsi qu'à l'approche où la méthode employée pour illustrer notre simulateur.

Chapitre IV : Simulateur de Circulation "SimCirc"

Démarche & Implémentation et Expérimentation

1. Introduction

Une théorie sans pratique peut être qualifiée d'aveugle. Ce chapitre se rapporte donc à la mise en pratique des préceptes que nous avons étudiés précédemment. Il a pour vocation de présenter des illustrations pratiques de l'application réalisée à l'issue de notre mémoire.

Dans ce chapitre nous présentons la modélisation de notre stratégie selon une démarche UML/Netlogo.

La majeure partie de ce chapitre portera sur l'explication de notre simulateur « **SimCirc** » et les résultats obtenue par des simulations, en présentant les différentes interfaces qui s'affichent au cours de l'utilisation de notre simulateur.

2. Cycle de Vie d'un Logiciel Multi agent

Selon Picard [38], 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.

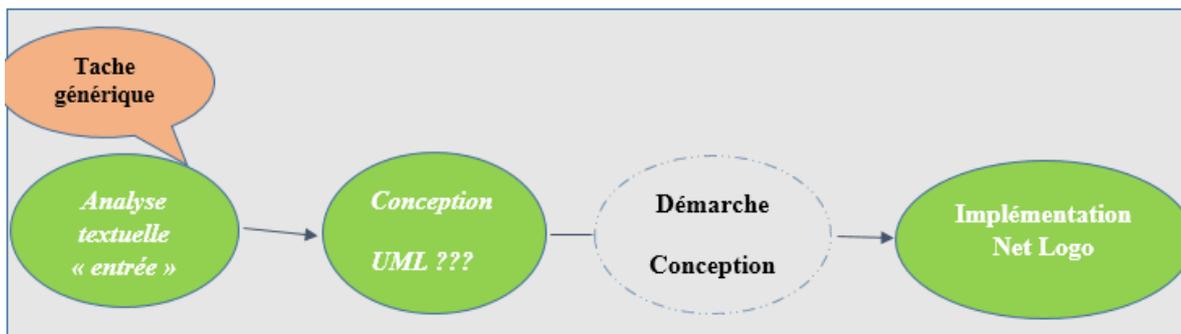


Figure IV.1 : Processus de développement de notre modèle

Somme tout, notre processus suivi et illustré par la Figure précédente (**Figure IV.1**).

Préalablement et avant d'expliquer les quatre étapes de notre processus de développement inspiré, il est important d'éclairer ce que signifie les points d'interrogation mentionné 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:

2.1 Tâche générique

Les tâches génériques fournir des tâches de référence « benchmark¹ », pour évaluer les architectures de contrôle proposées dans la littérature. Alors que dans notre cas, et pour un contexte développement des mécanismes à base des systèmes multi agent, ces tâches génériques sont censées comme animateur de l'ensemble des étapes d'un processus de développement (d'un simulateur orienté agents). I.e. manipuler les quatre étapes de notre processus, notamment la première étape 'analyse' afin d'inciter le processus.

2.1.1 Description de la tâche générique

Il s'agit de réécrire notre tâche (problème de la circulation routière) affecté dans notre modélisation attendu, décrit comme suite:

Dans un environnement se trouvent des intersections (quatre dans notre études), des routes (gauche, droite, haut, bas), panneaux de signal (feux) (changement des feux en deux couleur seulement "rouge/vert"), des voitures, des passages piétons, des piétons et des quartiers.

2.2 Analyse de la tâche

Il s'agit d'une opération de relecture du tâche souhaitons modéliser, avec un soulignement des concepts et des attributs pertinents dans un tableau à titre d'exemple, ainsi que les questionnements du concepteur. Une relecture attentive de notre tâche, nous permet d'éclater : Quatre thèmes (Voitures, panneaux signal, Passages piéton et Piétons),

Des attributs pour ces différents thèmes, des relations entre eux, et des Méthodes pour les thèmes (voiture, Feux de signal, piétons).

2.2.1 Voiture:

Un composant "agent mobile" présenté par un Turtle NetLogo. Ces voitures se déplacent d'une manière aléatoire, avec une vitesse maximum (V-max), chaque voiture "N" contrôle sa vitesse selon son emplacement (autre voiture; devant feux, passage,..., etc.)

Les comportements possibles de Nos Voitures :

- Les Voitures vis-à-vis une autre voiture :

¹ Un banc d'essai permettant de mesurer les performances d'un système pour le comparer à d'autres.

- La voiture "N" va accélérer et dépasser la voiture "N+1" lorsque la vitesse de "N+1" inférieure à "N", sinon il respecte sa vitesse et ne fait aucun dépassement.
- "N" Accélère lorsque la vitesse de "N+1" est supérieure à celle de "N".
- "N" décélère sinon arrêté, lorsque la vitesse de "N+1" Diminue (des fois vitesse devient nul) à celle "N".
- Les Voitures vis-à-vis les Panneaux de signal :
 - Avancer avec la couleur verte du panneau, et arrêter avec la couleur rouge.

NB: Il se peut que nos voiture change la direction vers la droite.

2.2.2 Panneaux de signal

Dans la plus part des pays du monde Les panneaux sont généralement composés à de trois couleurs mais dans notre modèle, et pour affaiblir la complexité du modèle souhaitais, on a soutenir uniquement le choix de deux couleurs de base seulement : le rouge pour interdire la circulation « immobiliser les voitures », et le vert pour ouvrir/autoriser le passage et la circulation des voitures.

2.2.3 Passages piétons

Ce sont des composant "agents immobiles" pour notre modélisation.il est important de souligner que ces composant ne comportent aucun mécanisme ou autonomie dans notre étude, c'est l'une de nos perspectives ambitionnés.

2.2.4 Piétons

Pareillement comme le composant passage piétons, On a ajouté les piétons dans notre environnement de travail, mais nous pouvons les considérés comme perspective pour nos futures travaux.

<i>Thèmes</i>	<i>Attributs</i>	<i>Méthodes</i>
Voitures	<ul style="list-style-type: none"> - Vitesse - V-max - Tourné? - Stop? 	<ul style="list-style-type: none"> - move-voitures - Vérifier-vitesse - peu-tourné-droit?
Panneaux de signal	<ul style="list-style-type: none"> - CFH - CFV - Position 	<ul style="list-style-type: none"> - Changé_couleur - Vision

Passage-piéton	perspective	?
Piétons	perspective	?
Environnement (Patches, Globale)	- Rôle - Temps	

Tableau 4 : Les différents thèmes pour analyser la tâche générique.

2.3 La conception

A cette étape (processus inspiré), on doit parler particulièrement sur l'ensemble des stratégies/mécanismes visée par nous afin de les concrétisé sous forme des diagrammes dynamique

2.3.1 Les Mécanismes développés

Notre impression d'optimisation de la tache de circulation routière apparaitra au niveau des panneaux de signalisation. A ce stade on a pensé à un ensemble de mécanisme pour gérer les différents Situation considérés par notre sujet, à savoir :

- Automatique : Le changement de couleur est surveillé par un Temps_fixé.
- Autonome : indique notre vision astucieuse sur ce sujet, où les panneaux de signalisation ne fonctionnent pas avec de compteurs pour inter changer les couleurs des panneaux, mais plutôt appuy sur des capteurs intégré sur ces panneaux afin qu'il puisse connaitre quelles sont les vois avec des bouchons pour qu'il le libère (capteurs aux lieux de compteurs, pour injectez assez d'autonomie a nos feux e signalisation).

Cette autonomie est décomposée en deux volets :

- Autonomie-local : pour gouverner l'encombrement dans un seul panneau de notre environnement.
- Autonomie-global : pour piloter l'encombrement dans l'ensemble des panneaux (quatre panneaux) de notre environnement.

Les voitures circulent aléatoirement (avec des paramètres d'accélération et de décélérations). Si elles détectent une intersection (panneaux), elles doivent respecter les couleurs de ses

panneaux. En plus chaque voiture peut avancer (doubler) une autre voiture selon certaine conditions tel que vitesse.

Ensuite, est pour peaufiné nos mécanisme discutés précédemment, on base sur deux diagrammes UML, l'un statique et l'autre dynamique, afin qu'on schématise, visualise, et simplifié la tâche de circulation étudiée. Pour cela nous avons appuyé sur un diagramme de classe (statique) Afin de structurer les différents thèmes associé à la tâche, Et un diagramme de séquence (dynamique) afin d'exhiber certains comportements et interactions de nos acteurs.

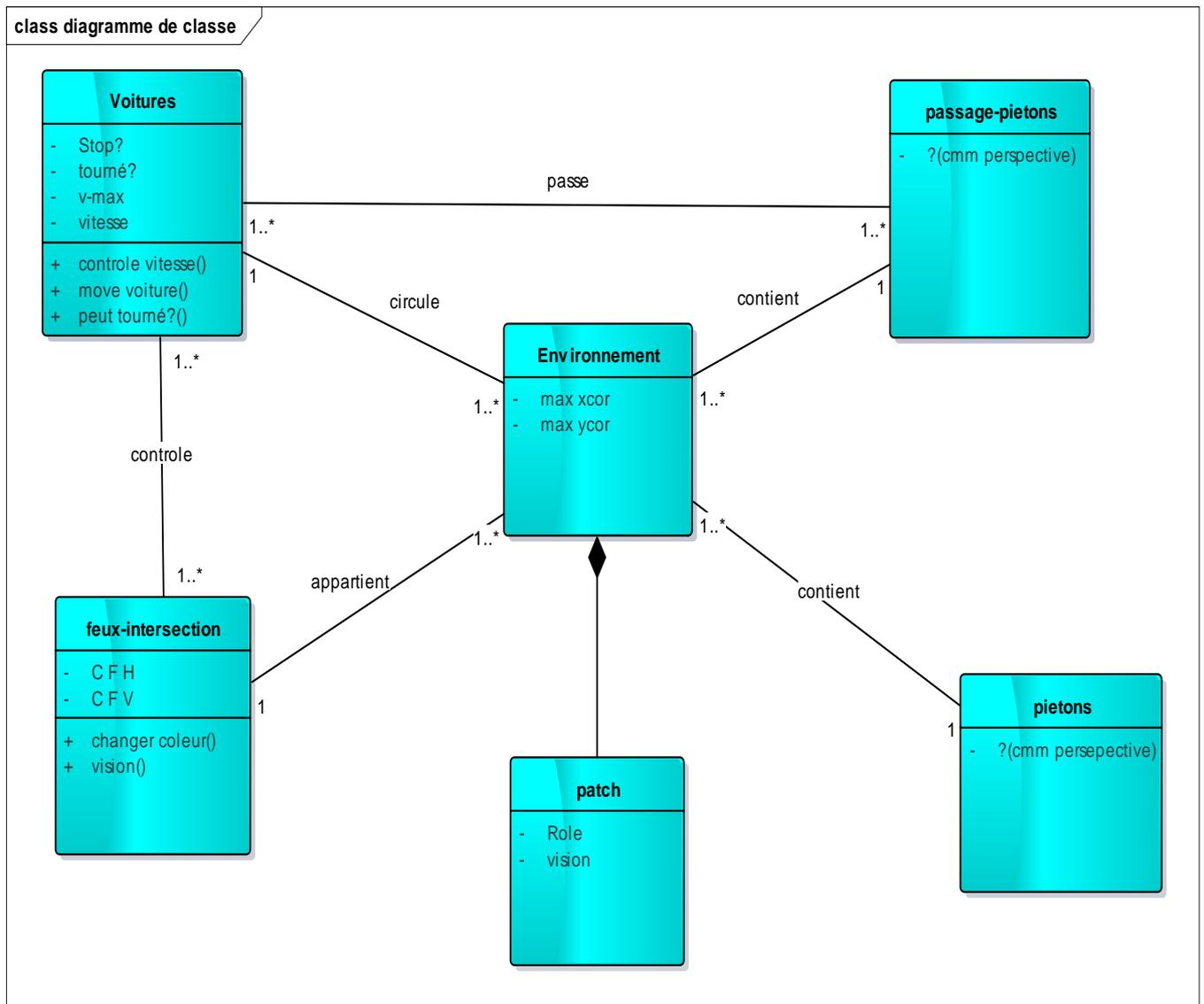
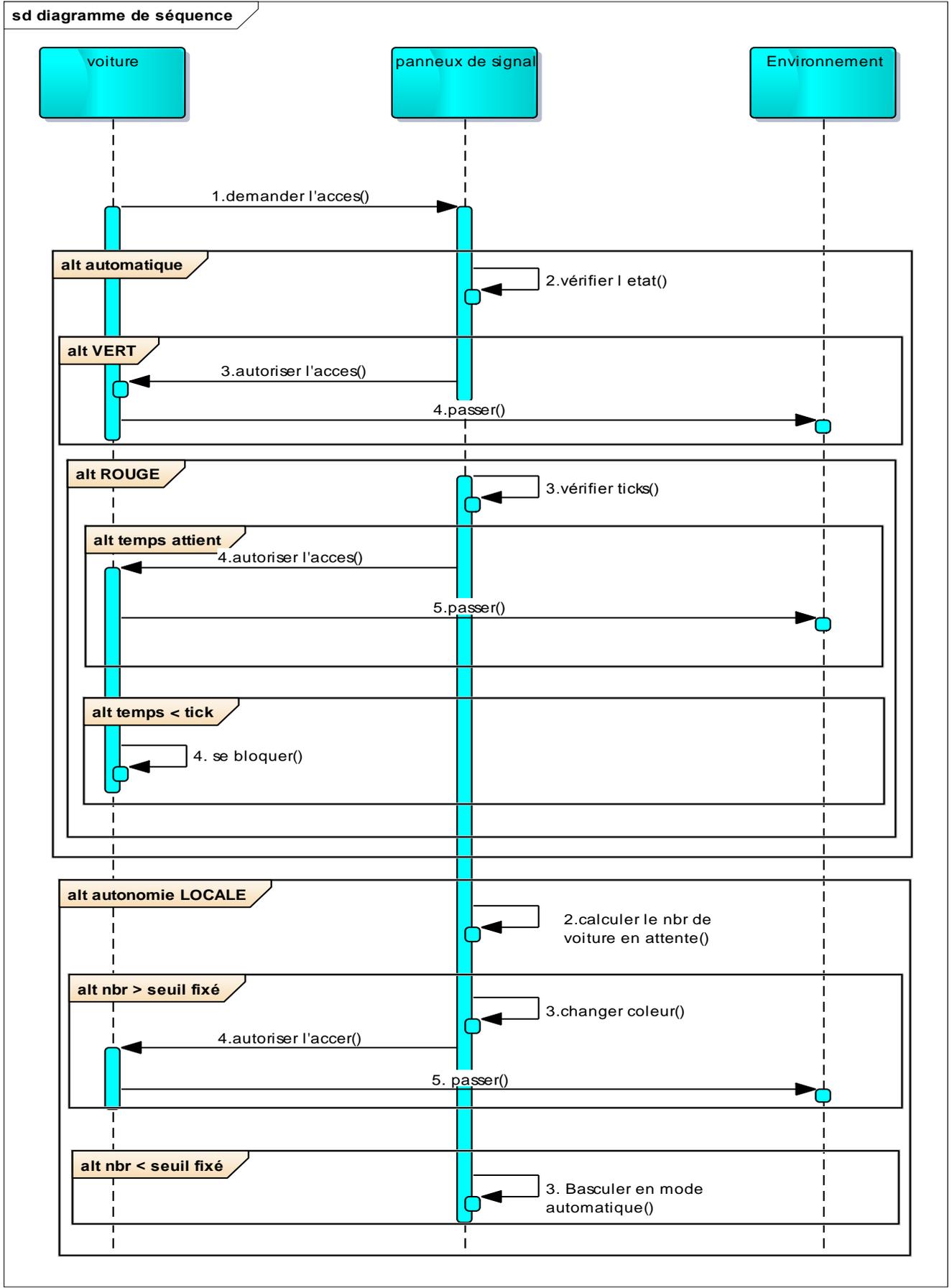


Figure IV.2 : diagramme de classe



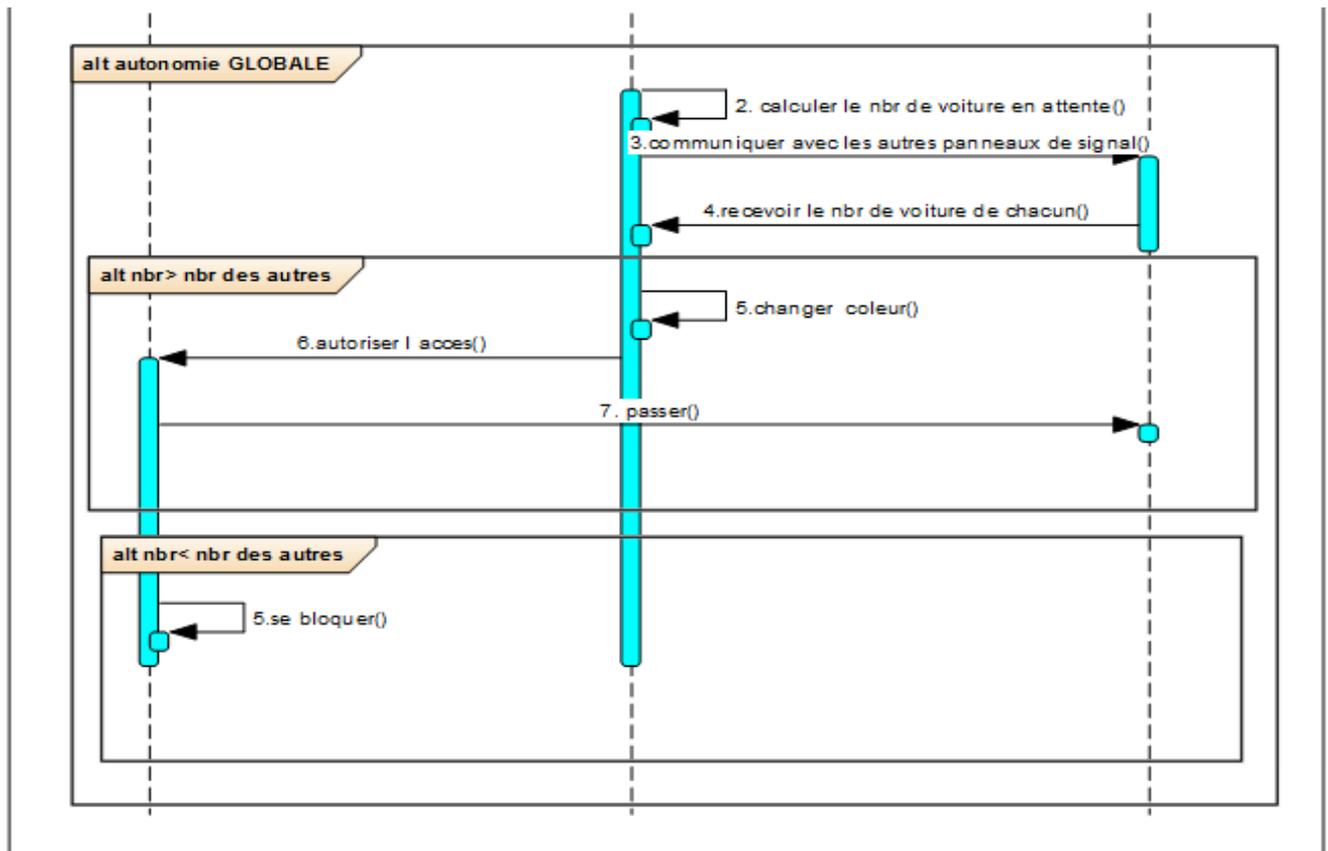


Figure IV.3 : Diagramme de séquence

2.4 La Démarche

Défini comme une étape de passage invisible, il s'agit d'appliquer l'ensemble des recommandations générées par Amblard [17] quant à relier le modèle conceptuel UML vers le modèle opératoire NetLogo. Nous réintégrons cette démarche dans notre méthodologie suivie, afin de faciliter le passage entre la conception UML aboutie et le programme Multi-Agent NetLogo inspiré :

2.4.1 Identification des variables globales

- **Temps** : une variable qui nous permet de maîtriser le fonctionnement automatique des feux de signal.
- **Vitesse Limite** : variable qui limite la vitesse dans la ville.
- **Somme-voiture en attente** : variable pour compter les voitures en attente dans les quatre intersections.
- **Voiture-en-attente** : indicateur pour compter le nombre de voiture en attente dans chaque intersection.

2.4.2 Identification des propriétés des objets à modéliser

En termes de structure d'environnement, notre univers est présenté comme une grille régulière, donc notre environnement est borné (non ouvert).

- **Voiture** : Définit comme des agents totalement mobile, continuellement supposés comme des Turtles Netlogo.
- **les feux de signal, passages piétons et les piétons** : définit comme des agents immobiles. On peut les définir comme des turtles Netlogo seulement.
- **l'environnement** : définit comme un ensemble de patches, Modélisé par défaut dans la plateforme Netlogo, toutefois sans aucune utilité de définir un « breed » comme celle des quatre autres agents (feux, voitures, passage-ps, piétons).

On différencier entre les quatre types Turtles employés par les primitives breed et la primitive Shape comme suit:

- **Breed [feux feu]** avec set Shape « circle ».
- **Breed [voitures voiture]** avec set Shape « voiture-top ».
- **Breed [passage-ps passage-p]** avec set Shape « passage-p ».
- **Breed [piétons piéton]** avec set Shape « person »

2.4.3 Organiser les entités entre elle

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

Un seul cran /degré d'héritage possible à partir de Turtle, et pas de comportements attribués à une liste d'agent (AgentSet), ce sera à faire dans le programme principal.

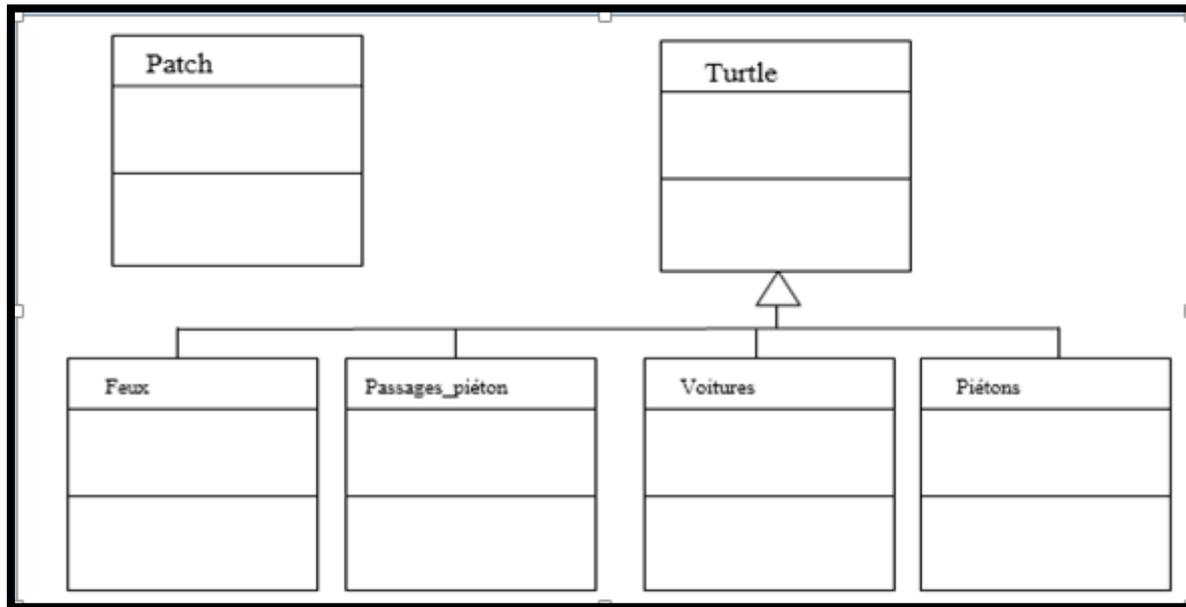


Figure IV.4 : Diagramme de passage UML/NetLogo.

2.5 . Implémentation

L’organigramme suivant récapitule l’algorithme principal de notre simulateur développé :

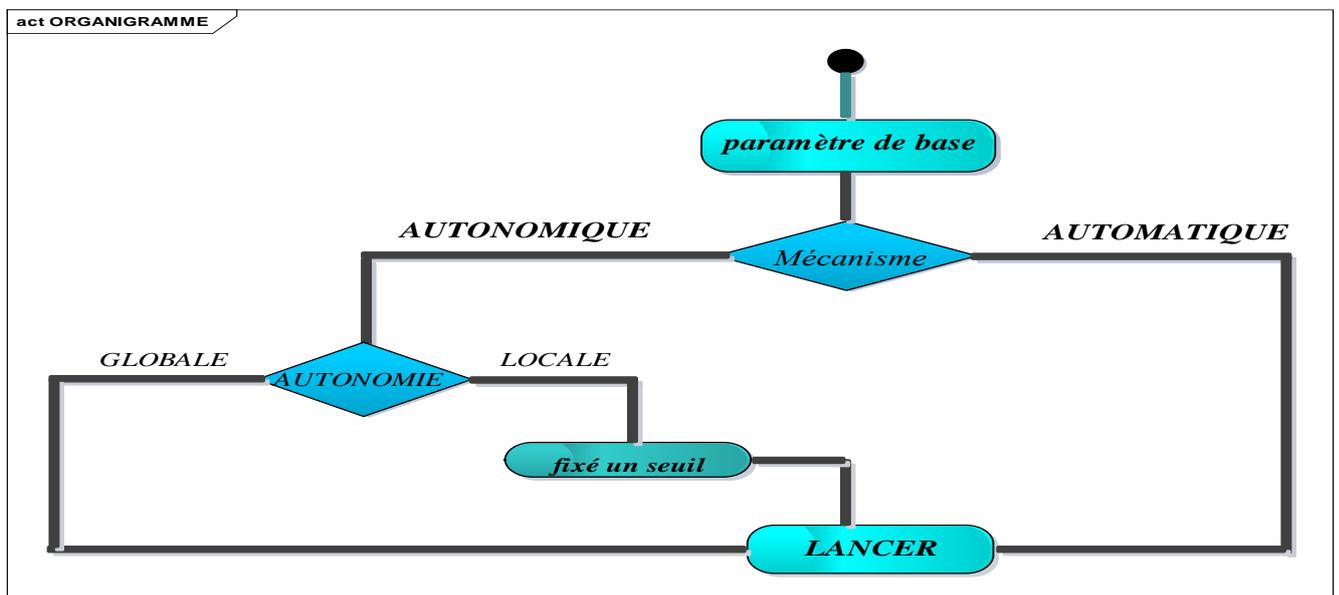


Figure IV.5 : Organigramme pour notre simulateur.

Avant de lancer la simulation nous passons par les étapes suivantes : Configurer les paramètres de base de "SimCirc", puis choisir le mécanisme de fonctionnement « autonome / automatique » désiré.

Si le mécanisme choisi est « AUTONOMIQUE » alors on doit limiter cette autonomie que ce soit globale, ou locale par une fixation d'un seuil (indicateur) pour les voitures en attente.

Dans notre travail exposé par ce mémoire, on focalise majoritairement toute notre attention sur la diminution du nombre de véhicules en attente, i.e on essaye de le minimiser le maximum possible.

En conséquence est pour réaliser le simulateur "**SimCirc**", nous avons adopté quelques hypothèses:

- **Hypo 1:** Chaque panneau dispose des capteurs pour capter le nombre de véhicule.
- **Hypo 2:** chaque voiture sait évaluer (estimé) la distance euclidienne qui le sépare avec d'autres (panneau, voiture) dans son champ de visibilité.

Dans une intersection on suppose qu'on a quatre voies horizontales nommées « A, B » et verticales nommées « C, D », sans oublier quatre panneaux numérotés (1 et 2) et (3 et 4) qui gèrent respectivement les quatre voies (A, B) et (C, D)

2.5.1 Présentation du simulateur « SimCirc »

Un simulateur considère le trafic d'un ensemble de véhicules (virtuel), avec un but indispensable : Autonomiser pour Optimiser c.-à-d. donner plus d'autonomie à nos feux de circulation afin d'améliorer la gestion des intersections de nos carrefours.

Considérons le bouton « Configuration » qui reproduit l'étape d'initialisation des différents paramètres de notre modèle, ainsi que la configuration de l'environnement de simulation (**tableau 6**).

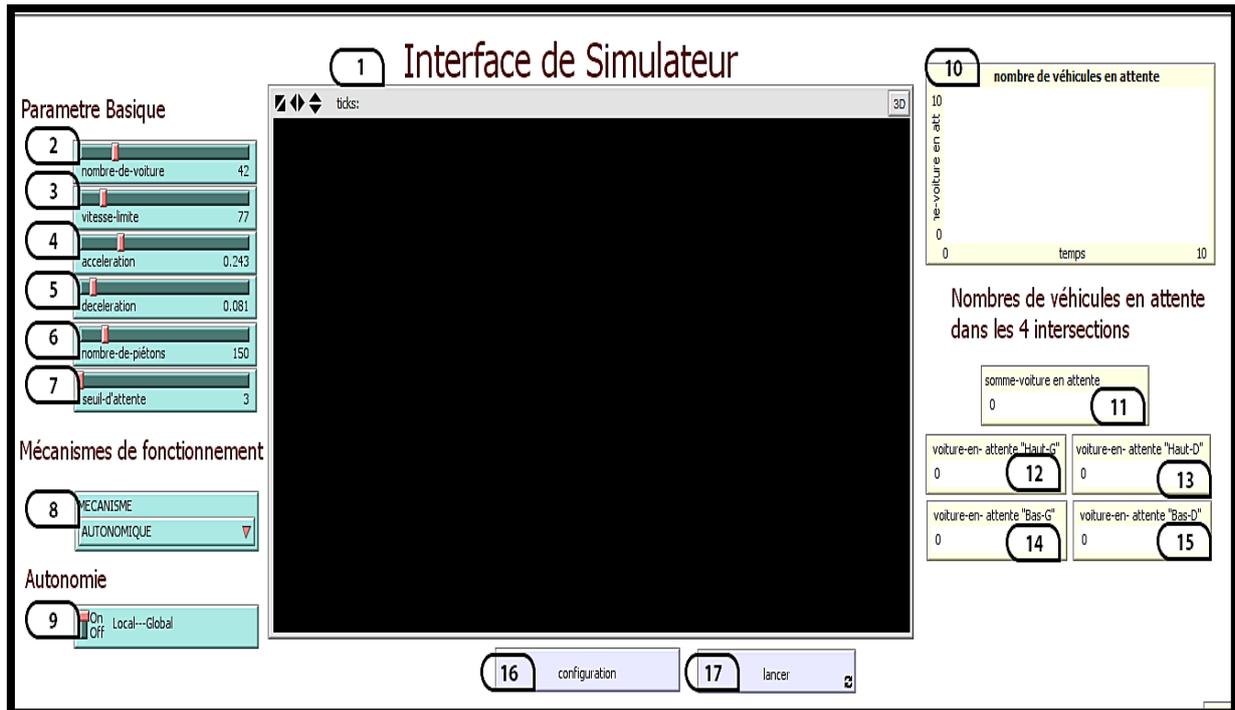


Figure IV.6 : présentation de simulateur « SimCirc ».

N°	La signification
1	Grille qui contient notre environnement
2	Pour indiquer le nombre de voitures dans l’environnement
3	Pour exiger les contraintes de la vitesse des voitures.
4	Pour exiger les contraintes de la vitesse des voitures.
5	Pour décider les contraintes de la vitesse des voitures.
6	Pour indiquer le nombre de piétons dans l’environnement
7	Indique le seuil des voitures en attente « cas autonomie LOCALE »
8	Choix du mécanisme de fonctionnement «AUTONOMIQUE/ AUTOMATIQUE »
9	Mécanisme LOCALE ou GLOBALE
10	Pour tracer le graphe des voitures en attente dans les quatre intersections
11	Pour compter le nbr des voitures en attente dans les quatre intersections
12	Pour compter le nbr des voitures en attente dans l’intersection « HAUT-G »

13	Pour compter le nbr des voitures en attente dans l'intersection « HAUT-D»
14	Pour compter le nbr des voitures en attente dans l'intersection « BAS-G »
15	Pour compter le nbr des voitures en attente dans l'intersection « BAS-D »
16	Pour initialiser le simulateur
17	Pour lancer la simulation

Tableau 5: présentation de simulateur « SimCirc ».

	Procédures
Feux	Config-feux
Voitures	Config-voiture
Piétons	Config-piéton
Passage-piéton	Config-passage
Environnement	Config-route ; config-intersection ; config-zone-urb

Tableau 6: les procédures de configuration dans notre simulateur " SimCirc".



Figure IV.7 : Une vue "3D" de Simulateur "SimCirc"

Deuxième bouton « Lancer » a pour but de démarrer La simulation ou la circulation routière, Symbolisé par le programme NetLogo suivant :



Figure IV.8 : champ de visibilité des feux de signal.

3. Expérimentation

Dans ces expériences, nous avons voulu tester l'influence du mécanisme de nos panneaux de circulation (avec/sans autonomie), sur le nombre de voitures arrêté au niveau des carrefours dans "SimCirc". En effet trois mécanismes de fonctionnement pour l'ensemble de panneaux s à été apprécié afin de découler le mécanisme le plus performant en terme de nombre de véhicule en attente.

Pour notre étude empirique, nous avons configuré cinquante voitures dans un environnement de taille 79 X 40 contenant quatre intersections (16 panneaux de circulation).

NB: les simulations et l'ensemble des expérimentations, ont été réalisées sur un Micro Portable ACER avec un 3 GB de mémoire, et un processeur d'Intel (R) pentium(R), P6200 avec deux CPUs de 2.1 GHz.

3.1 L'effet de Mécanisme de nos Panneaux de Circulation adopté sur la tâche de circulation

Dans cette exploration, nous essayons d'adopter Trois politiques de gestion pour les panneaux, auprès de tenir la politique la plus efficient pour accomplir efficacement la tâche dans un objectif de réduire le maximum de nombre de voiture en attente. En effet chaque politique report une architecture distinct de nos panneaux, à savoir Automatique pour la

première, Autonome local pour la deuxième, Autonome global pour la troisième.

Méca_1 : Panneaux avec des compteurs numérique seulement, pour gérer l'alternance de couleurs.

Méca_2: Panneaux avec des Capteurs qui gère l'alternance de couleur selon la situation local du carrefours en terme de nombre de voiture.

Méca_3: Panneaux avec des capteurs qui gère l'alternance de couleur selon la situation global de toutes les carrefours en terme de nombre de voiture dans chaque intersection a part.

3.2 Interprétation des résultats

Après dix simulations, Nous avons conservés les résultats dans **le Tableau 7 et Tableau 8**. Retraces dans les deux histogrammes au-dessus (**Histogramme 1 & Histogramme 2**).

Durant chaque simulation de chaque mécanisme on a recensé le nombre de voiture en attente toute les "60 unité de temps", et on a fait la comparaison entre des mécanismes, au début entre Méc_1 et Méca_2. Ensuite entre Méca_2 et Méca_3.

Unité de Temps	Méc_1	Méca_2
0	0	0
60	10	8
120	15	6
180	16	12
240	16	3
300	16	3
360	7	3
420	9	8
480	8	5
540	12	7
600	8	5

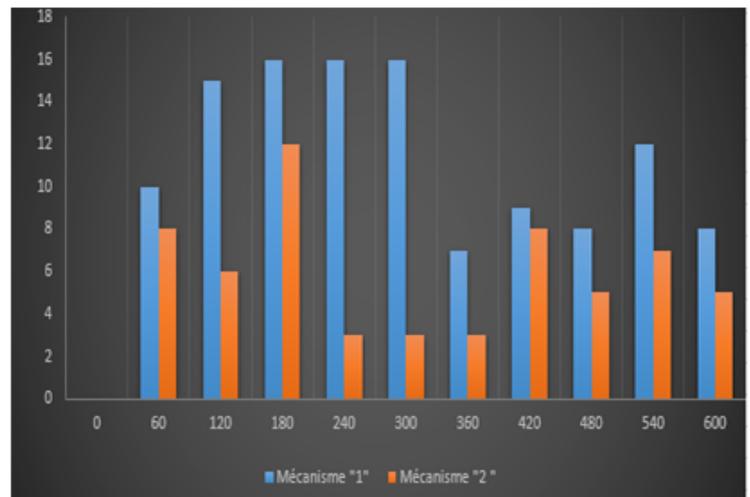


Tableau 7 : Nbre voitures immobile/Temps **Histogramme 1:** Nbr_Voitures / Mécanismes (1 ou 2)

L'histogramme 1 , démontre qu'avec les différentes unité du temps de 0 jusqu'à 600, le mécanisme de gestion (Méca_2) donne des meilleures résultats en terme de nombre de voiture arrêter au niveau de carrefours a fin de compléter la circulation, ce fait peut être confirmé par le nombre pratiquement important de l'ensemble des voitures en immobilisé au niveau des carrefours (16 voitures) avec l'unité de temps 180,240, et 300 Tiks. .

De ce fait, La performance de la circulation routière en termes d'encombrement au niveau de carrefours dépend absolument de l'adoption d'un mécanisme d'autonomie basé sur les capteurs et non pas des compteurs pour gouverné les intersections.

L'histogramme 2 , démontre qu'une autonomie local seul n'est pas suffisante pour maitriser la situation, avec les différentes unité du temps de 0 jusqu'à 600, le mécanisme de gestion (Méca_3) donne des meilleures résultats en terme de nombre de voiture arrêter au niveau de carrefours a fin de compléter la circulation que le Méca_2, ce fait peut être confirmé par le nombre pratiquement important de l'ensemble des voitures en immobilisé au niveau des carrefours (12 voitures) avec l'unité de temps 180 Tiks au lieu de 4 voitures pour le mécanisme Méca_3.

Temps	Méca_2	Méca_3
0	0	0
60	10	3
120	6	4
180	12	4
240	3	2
300	3	2
360	7	4
420	9	2
480	5	3
540	7	1
600	8	1

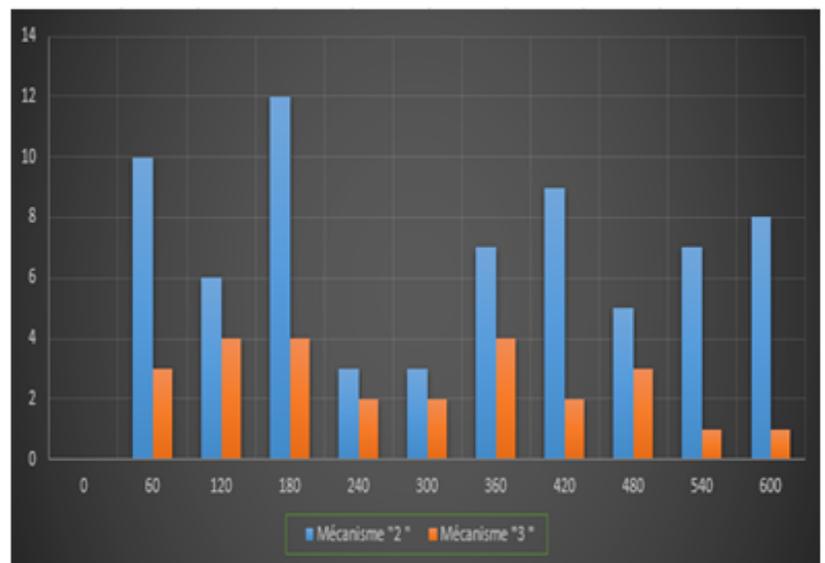


Tableau 8 : Nbre Voiture immobile/Temps **Histogramme 2 :** Nbr_Voitures / Mécanismes 2 ou 3.

De ce fait, La performance de la circulation routière en termes d'encombrement au niveau de toutes les carrefours dépend absolument, de non seulement de l'adoption d'un mécanisme d'autonomie basé sur les capteurs au niveau des panneaux mais aussi bien d'une communication en temps réel entre les panneaux de circulation pour rendre nos système de circulation d'être à la fois local et global dans l'homogénéité.

Résumé :

A partir de nos Simulations/Expériences, on peut dire, certifié le troisième mécanisme. i.e l'autonomie global (Méca_3) comme meilleur mécanisme pour notre problème de circulation qui dévisage seulement la réduction de nombre de voiture en attente au niveau des carrefours.

Remarque :

A notre avis L'approche adoptée et les mécanismes développés dans ce mémoire, peut il critiqués clairement au niveau de sa crédibilité, aussi bien en termes de modélisation sur un outil pragmatique avant de lancer dans des simulations. Cependant, cette approche adoptée, peuvent il, plus au moins restreinte, le cadre proposé doit être testé avec un grands nombre de voitures/carrefours et dans un large environnement, la gestion des accès concurrents de nos voitures, les collisions entre les voitures, et surtout sur d'autres plateformes de simulation SMA.

4. Conclusion

Ce qu'il faut retenir de ce dernier chapitre, c'est que la circulation routière au sein d'un environnement contraint (en termes de variété de voitures, vois et d'intersection), est une tâche complexe et paradoxale. Cette complexité n'est pas perçue dans le sens théorique du terme, mais au sens pratique, c'est-à-dire, si on connaît bien comment la voiture (turtle) se déplace et les panneaux de signalisation se manifestent.

La difficulté s'aperçoit dans la question suivant: A qu'elle moment les panneaux de circulation changent ces couleurs d'autorisation/interdiction.

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

Dans ce refuge, la modélisation et simulation à travers un processus de développement inspiré de trois mécanismes de gouvernance, nous ont produit un support magique '**SimCirc**' pour dégeler cette complexité inhérent à la circulation, les rendre élémentaire pour la compréhension.

Conclusion générale et 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 l'une des pistes de la circulation routière (la gestion des intersections 'carrefour').

La circulation routière est l'une des problèmes socio-économiques qui doivent être résolus pour supporter l'évolution de la société. La solution accommodée est de trouver des algorithmes de contrôle pour prévenir l'apparition de la congestion routière et pour l'optimiser.

Pour implémenter avec succès de tels algorithmes, une bonne modélisation de la dynamique du trafic routier est aussi nécessaire pour cela nous avons décrit notre travail, dans le cadre de la modélisation par le paradigme agent avec un contexte multi-agent, pour maîtriser, ensuite gouverner la complexité de ce phénomène de circulation.

Les recherches effectuées dans notre mémoire présentent des nouvelles solutions/mécanismes dans ce cadre pour améliorer et minimiser le trafic routier au niveau des intersections, par l'emploi des stratégies et des techniques de contrôle au niveau des feux de signalisation, non seulement d'une manière automatique mais aussi bien autonome.

Dans ce contexte, les principales contributions de ce travail sont présentées ci-dessous :

- Une courte présentation de l'ensemble des paradigmes de modélisation retrouvé dans la littérature.
- La rédaction d'un état de l'art sur les approches qui abordent la circulation routière, La description, l'explication, et la distinction entre les approches de trafic routière, particulièrement avec la vision courante vers des perceptions autonomes.
- La présentation et l'apprentissage d'une plateforme multi-agent nommé NetLogo.
- Le développement d'un simulateur nommé "**SimCirc**" dédié à la simulation d'une circulation routière. Ce développement est issu d'un processus inspiré (UML/Netlogo) et bien détaillé dans ce mémoire, concernant une tâche de circulation générique sélectionnée.
- La réalisation d'un ensemble d'expérimentation préliminaires sur ce simulateur "**SimCirc**" développé, pour montrer l'efficacité et l'efficience quant à l'intégration de l'autonomie aux feux de signalisation que ce soit au niveau locale ou globale.

Ces expérimentations ont donné des résultats prometteurs en termes taux de congestion (nombre de voiture en attente), quoique ces expérimentations soient assez restreintes (limités).

Cette limitation provient principalement de la taille de l'environnement, le nombre de véhicules, 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 véhicules et dans un large environnement, et sur d'autre plateforme de simulation multi-agent, comme **Swarm**, **Repast**...Etc.

Les résultats présentés dans ce mémoire peuvent être augmentés dans d'autres directions. Les perspectives de développement et d'augmentation les plus prometteuses sont :

- Les piétons et les passages piétons (traiter bientôt) qui posent des problèmes supplémentaires, en changeant totalement la dynamique de modèles.
- Les algorithmes de contrôle peuvent être améliorés en intégrant des aspects d'optimisation du trafic tels que le temps, la distance parcourue, la vitesse moyenne de déplacement et l'énergie.

A travers notre travail réalisé, la simulation a montré son intérêt dans le domaine du trafic routier. Pour notre cas, elle a paru indispensable, vu qu'elle a permis de fournir une vision future sur la situation proposée. De même, elle a permis d'anticiper les difficultés et donc de proposer des solutions pour fluidifier le trafic.

Actuellement "**SimCirc**" permet des simulations symptomatiques d'une seule tâche de circulation.

Le travail présenté dans ce mémoire est une première pierre concernant une démarche flexible avec de mécanismes basée sur les systèmes multi-agents, consacrée à la gouvernance de la circulation routière, Il peut se décliner en différentes extensions, que la durée d'une mémoire n'a pas permis d'aborder, et qui constituent autant de pistes de recherche à explorer.

Références Bibliographiques et **Webographies**



- [1] L. ARENA et C. BERARD. 'La simulation : une double-utilité en sciences de gestion', 2011, 25 pages.
- [2] C. BUISSON. 'Modélisation et simulation du trafic'. Institut national de recherche sur les transports et leur sécurité : France, 2005, 30 pages
- [3] J. SANSONNET. 'Simulation Multi-Agent', 2014.
- [4] A. LANSLOWNE. 'Traffic Simulation using Agent-Based Modelling'. University of the West of England : Angleterre, 2006.
- [5] I. JARRAS et B. CHAIB DRAA. 'Aperçu sur les systèmes multi-agents'. CIRANO : Montréal, 2002, 42 pages.
- [6] J. FERBER. 'Les Systèmes Multi Agents : vers une intelligence collective'. Inter Edition, 1995, 499 pages.
- [7] M. WOOLDRIDGE. 'Introduction to Multi Agent Systems'. Department of Computer Science, University of Liverpool: Royaume-Uni, 2002, 348 pages.
- [8] A. TLILI. 'Etude et Réalisation d'une Plate-Forme Multi-Agents'. Algérie : Thèse de Magister. Université Hadj Lakhdar de Batna Faculté des Sciences de l'Ingénieur Département de l'Informatique, 124 pages
- [9] F. VARENNE. 'Les simulations computationnelles dans les sciences sociales. Nouvelles Perspectives en Sciences Sociales'. Université de Rouen & GEMAS: France, 38 pages, 2010
- [10] Bazzan. (2009). Opportunities for multi-agent systems and multi-agent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, 18(3): (p.342_375).

- [11] Mammar. (2007). Systèmes de Transport Intelligents, modélisation, information et contrôle . Hermes science, Lavoisier,.
- [12] Henry, J. F. (1984). The prodyn real time traffic algorithm. 4th IFAC/IFIP/IFORS Conference On Control.
- [13] R. Naumann, R. R. (1997). Validation and simulation of a decentralized intersection collision avoidance algorithm. In Intelligent Transportation System (ITSC'97) (p. 818_823). IEEE
- [14] Stone, K. e. (2004). Multiagent traffic management : A reservation-based intersection control mechanism. In Proceedings of the Third International Conference on Autonomous Agents and Multiagent Systems (AAMAS).
- [15] A. ROUSSET et al. 'Etude comparative des plateformes parallèles pour systèmes multi-agents'. Université de Franche-Comté : France, 2014, 13 pages.
- [16] David.Sheeren. (21-26 JUIN 2009). introduction à la modélisation UML,. Atelier 1, MAPS : modélisation multiagents appliquée aux phénomènes spatialisés. la vieille Perrotine, Saint-pierre d'Orléon.
- [17] F.Amblard, N. (21-26 juin 2009). Modélisation multi-agents appliquée aux phénomènes spatialisés . Atelier 2 : Passage du modèle conceptuel UML à Netlogo. La Vieille Perrotine, Saint-Pierre d'Oléron .
- [18] M. Berryman et S. Angus. 'Chapter 1 Tutorials on Agent-based modelling with NetLogo and Network Analysis with Pajek'. World Scientific Review Volume, 2009, 27 pages.
- [19] Gérer la congestion urbaine Par European Conference of Ministers of Transport
- [20] Infrastructure logicielle pour la construction de plateformes de simulation multi-agent, 2008
- [21] Systèmes Complexes Adaptatifs Application au traitement des images, Université Mentouri Constantine Faculté des Sciences de l'Ingénieur, 2006
- [22] Cadre formel pour la modélisation et l'analyse des agents mobiles.

- [23] Mécanisme de coordination multi-agent fondé sur des jeux : application à la simulation comportementale de trafic routier en situation de carrefour, 2003.
- [24] Coordination locale et optimisation distribuée du trafic de véhicules autonomes dans un réseau routier ,2015
- [25] Contribution à la modélisation et à la régulation du trafic aux intersections : Intégration des communications Véhicule-Infrastructure ,2012
- [26] Contrôle du trafic routier urbain par un réseau fixe de capteurs sans fil, 2012

Les sites web consultés :

- [27] Ego.developpez.com
- [28] www.upmc.fr
- [29] Wordpress.com
- [30] www.futura-sciences.com
- [31] www-inf.it-sudparis.eu
- [32] www.maxicours.com
- [33] www.developpez.com
- [34] www.crescenzo.nom.fr
- [35] www.lemoniteur.fr
- [36] www.moodle.epfl.ch
- [37] www.mediatheque.snbpe.org
- [38] www.commentcamarche.net