

Faculté des Mathématiques et de l'Informatique

Département d'Informatique

N° :



DOMAINE : Mathématiques et Informatique

FILIERE : Informatique

OPTION : IA

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par:

Hiyam Herizi

Intitulé

Développement d'un simulateur (à base d'agent) pour un système de transport intelligent(cas d'un véhicule autonome)

Soutenu devant le jury composé de :

Brahimi mahmoud

Université de M'sila

Président

Meliouh amel

Université de M'sila

Rapporteur

Barkat abd el basset

Université de M'sila

Examineur

Année universitaire : 2023 / 2024

اهداء

من قال أنا لها نالها

وأنا لها و إن أبت رغما عنها أتيت بها
نلتها و عاقت اليوم مجدا عظيما و لم يكن الحلم قريبا
و لا الطريق سهلا و لكن وصلت الحمد لله حبا و
شكرا و امتنانا

أهدي بكل حب مذكرة تخرجي إلى نفسي العظيمة
الفتية التي تحملت كل العثرات و أكملت رغم
الصعوبات

إلى أعظم الأشخاص و أعز الناس سندي
و ملاذي بعد الله ، فخري و اعترازي أبي و أمي
إلى من ساندني بكل حب وقت ضعفي و كان معي
في كل خطوة زوجي
إلى ضلعي الثابت و أمان قلبي إخوتي (طاهر ، محمد
،مرال)

إلى كل عائلتي و صديقاتي و كل أساتذتي
بالخصوص مشرفتي.

Sommaire

Liste de figures	4
Introduction Générale :	6
Chapitre1 :Les Systèmes Multi-Agents	8
1. <i>Introduction</i>	9
2. <i>Concept d'agent :</i>	9
3. <i>Typologies des agents</i>	9
3.2 <i>Agent cognitif</i>	11
3.3 <i>Agent hybride</i>	11
4. <i>Agents cognitifs VS Agents réactifs:</i>	12
5. <i>L'architecture d'agent</i>	12
6. <i>La communication des agents</i>	13
7. <i>Système multi-agents (SMA):</i>	14
7.1 <i>Définition d'un SMA (Système Multi-Agents) :</i>	15
7.2 <i>Organisation dans les SMA</i>	15
7.3 <i>Types de communication entre agents</i>	16
8. <i>Conclusion</i>	19
Chapitre 2:	20
Les systèmes de transports intelligents	20
1. <i>Introduction</i>	21
2. <i>Définition des systèmes de transport intelligent</i>	22
3.1 <i>Premières études (années 60années 70)</i>	23
3.2 <i>Premières applications (années 80milieu des années 90)</i>	23
3.3 <i>Les grands projets (milieu des années 90aujourd'hui)</i>	24
4. <i>Les objectifs des systèmes de transport intelligent</i>	25
➤ 4.1 <i>Un transport plusable</i>	25
4.3 <i>Une amélioration de la sécurité</i>	26
4.4 <i>Des économies de temps et gains d'ecacité opérationnelle</i>	26
5. <i>Une réduction des e ets sur l'environnement</i>	27
5.1.1 <i>Information pour les voyageurs</i>	27
6. <i>Exemple d'architecture canadienne</i>	28
6.1 <i>Gestion du transport en commun</i>	30
7. <i>Technologies intégrées dans les systèmes de transport intelligent</i>	31
7.1 <i>Communications sans fil</i>	31
7.2 <i>Technologies de calcule</i>	32
7.3 <i>Technologies de localisation</i>	32
7.4 <i>Technologie de capteurs</i>	33
7.5 <i>Capteurs vidéo</i>	34
7.6 <i>Réseaux de capteurs sans fil</i>	35
7.7 <i>Autres capteurs</i>	36
8. <i>Conclusion</i>	36
Chapitre 3 :	37

Contribution	37
1. <i>Introduction</i>	38
2. <i>La simulation informatique</i>	38
3. <i>Simulation à base d'agents</i> :.....	40
3.3 <i>Conception du modèle d'agent</i>	41
3.4 <i>Plateformes basées sur le paradigme agent</i>	42
4. <i>Conception avec UML</i> : 4.1 <i>Définition</i> :	43
4.2 <i>Utilité d'UML</i> :	43
4. 3 <i>Démarche de modélisation</i> :	44
5. <i>Présentation du système de simulation</i> :.....	49
5.1 <i>Etapes de simulation</i>	49
6. <i>Démarche d'exécution de la simulation</i> :.....	49
7. <i>Description des Algorithmes de recherche du plus court chemin</i> :.....	50
7.1 <i>Algorithme A*</i>	50
7.2 <i>Algorithme DFS (Depth-First Search)</i> :	52
8. <i>Les Principales interfaces de système simulation</i>	53
8.2. <i>Déterminez les points de début et de fin</i>	53
8.3. <i>Créez un agent voiture</i>	54
8.4. <i>Sélectionnez l'algorithme de recherche du plus court chemin</i>	55
8.5. <i>Exécutez la simulation</i>	56
9. <i>Analyse des Résultats obtenus</i> :.....	57
10. <i>Conclusion</i> :.....	58
Conclusion Générale :	59

Liste de figures

Figure 1.1 Cycle perception, action d'un agent réactif	10
Figure1. 2 Cycle Perception Délibération Action d'un agent cognitif.....	11
Figure 1.3 Structure d'agent.....	12
Figures 1.4 Positionnement des SMA	14
Figure 2.1: Exemple de deux véhicules connectés	22
Figure 2.2 Gestion du tra c routier pour un système de transport intelligent.....	27
Figure 2.3 Architecture canadienne des ITS	28
Figure 2.4 Architecture générale d'un ITS	29
Figure 2.5 Gestion de transport en commun	30
Figure 2.6: Communication sans fil.....	31
Figure 2.7 Boucle à induction magnétique(BIM)	34

Figure 3.1 Etapes de conception d'une simulation informatique.....	39
Figure 3.2. Structure d'un modèle basé sur un agent.....	41
Figure 3.3 Démarche de modélisation de l'application	44
Figure 3.4 Diagramme de cas d'utilisation	45
4.3.2 Diagramme de classe :	47
Figure 3.5 Diagramme de classe.....	47
Figure 3.6 Diagramme de séquence	48
Figure 3.8 l'algorithme A*	51
Figure 3.9 code de l'algorithme A* sur netlogo	51
Figure 3.10 le code de l'algorithme DFS sur Netlogo.....	52
Figure3.11 Créer un chemin.....	53
Figure 3.12 Déterminez les points de début et de fin.....	54
Figure 3.13 Créez un agent voiture.....	54
Figure 3.14 Programmation de création voiture	55
Figure 3.15 Sélection l'algorithme	55
Figure 3.16 exécution de simulation.....	56
Figure 3.17 courbe obtenu.....	57

Introduction Générale :

Le choix du moyen de transport adéquat par les citoyens en tenant en considération leurs besoins, leurs aspirations ainsi qu'au fonction des imprévus qui peuvent intercepter comme par exemple, la congestion, les changements climatiques, des événements spéciaux...est une décision qui ne peut trouver son efficacité qu'avec des bases solides d'information et en temps réel. Après le développement de la civilisation des transports, le transport intelligent a été réalisé, qui dominera grandement les époques à venir. Des systèmes de transport performants ont un impact fort sur le développement d'une société. Rendre efficace le déplacement des biens et des personnes participe à la croissance économique et modifie nos territoires par l'amélioration de l'accessibilité

Le transport intelligent n'est pas seulement une théorie pour l'avenir ; il est mis en œuvre aujourd'hui dans plusieurs villes, dont les succès et les échecs sont utilisés pour améliorer les systèmes dans de nouveaux endroits. Certaines des villes qui mettent en œuvre de nouvelles technologies de transport peuvent vous surprendre au premier abord. Bien sûr, des plaques tournantes mondiales comme New York a adopté le transport intelligent pour leur ville toujours plus intelligente. Cependant, l'État rural du Wyoming est également un banc d'essai de premier plan pour les véhicules connectés. Le transport autonome de marchandises à travers le pays peut améliorer considérablement l'efficacité de la chaîne d'approvisionnement et réduire le besoin de chauffeurs longue distance contraints de concilier des délais serrés et leur besoin humain de repos.

En bref, nous disons que le transport intelligent est un moyen d'assurer le confort humain, d'arriver à l'heure et d'éviter le bruit et la pollution de l'environnement. Les systèmes de transport intelligents (ITS) désignent des systèmes qui utilisent à la fois l'informatique et les technologies des télécommunications, du radiopéage et de l'automatisation afin d'améliorer la sécurité, la gestion et l'efficacité des transports terrestres.

De nos jours, l'étude et la gestion des Systèmes de transport intelligents (ITS) sont difficiles et coûteuses, surtout avec les moyens différents et nombreux de transport (véhicules privés, bus, tramway, ... etc.). De plus, la construction et les configurations de tels systèmes sont des tâches très complexes. Dans ces systèmes, l'objectif principal est d'améliorer la circulation et d'optimiser la fluidité du trafic en agissant sur le pilotage et sur les règles de fonctionnement.

L'intelligence artificielle distribuée est une approche pour contrôler les systèmes complexes en décomposant et en distribuant la prise de décision. Les systèmes complexes sont décomposés en sous-ensembles en relation. Chaque sous-ensemble est

responsable du contrôle de son domaine et de la coordination des activités avec les autres sous-ensembles. L'approche distribuée du pilotage peut contribuer à la mise en place d'organisations agiles, combinant des logiques de réseaux et de hiérarchie. Cependant, le pilotage non centralisé des STU s'appuie fondamentalement sur les interactions d'entités communicantes, autonomes et capables de prendre des décisions.

Dans les systèmes de transport intelligents, le paradigme multi-agent peut être une réponse au développement d'un STI répondant aux quatre propriétés d'autonomie, de réactivité, de communication, d'aptitude sociale et de pro-activité usuellement associées aux agents.

Dans notre travail, nous considérons les approches multi-agents pour les STI, en prenant en compte la propriété d'autonomie d'un agent voiture dans le calcul du plus court chemin dans un graphe représentant une ville.

Notre mémoire est organisé en trois chapitres : Le chapitre 1 s'intéresse à la présentation des systèmes multi agents. Nous présentons tout d'abord une description générale, puis nous abordons l'intérêt de la modélisation et de la simulation comme outils pour étudier le fonctionnement de systèmes complexes et évaluer les effets des décisions prises. Le chapitre 2 réalise un état de l'art sur les systèmes de transport intelligents. Le chapitre 3 présente notre contribution, il commence par le fonctionnement de systèmes complexes évaluer les effets des décisions prises, le chapitre 2 réalise un état de l'art sur les systèmes de transport intelligents, le chapitre 3 présente notre application et termine par les résultats obtenus.

Chapitre1 :Les Systèmes Multi-Agents

1. Introduction

Les systèmes multi-agents (SMA) forment la branche de l'Intelligence Artificielle Distribuée qui s'intéresse à la mise en interaction d'un ensemble d'agents dans le but de résoudre un problème donné. Ainsi, nous commençons ce chapitre par présenter des concepts de base de l'agent et ses caractéristiques. Ensuite, par introduire les SMA et ces caractéristiques, puis nous nous sommes tournés vers la communication dans les SMA. Par la suite, nous allons parler sur les types de communication dans les SMAs

2. Concept d'agent :

Plusieurs définitions d'agent ont été proposées. (Ferber, 1995) propose : « Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents».

Un agent présente un certain nombre de propriétés, dont certaines sont plus importantes que d'autres selon le type d'application. Voici les principales propriétés. (Wooldridge and Jennings, 1995 ; Jennings et al, 1998) :

- **Situation** : l'agent est capable de recevoir des entrées sensorielles de son environnement et peut effectuer des actions qui changent cet environnement (par exemple : systèmes de contrôle de processus, démons logiciels).
- **Autonomie** : l'agent est capable de prendre des décisions et d'agir sur son état interne sans l'intervention directe d'un tiers (humain ou un autre agent) et contrôle ses propres actions ainsi que son propre état interne.
- **Réactivité** : l'agent est capable de percevoir son environnement et de répondre dans le temps requis aux changements qui apparaissent dans cet environnement.
- **Pro activité** : l'agent ne doit pas agir seulement en réponse à son environnement, mais il doit exhiber un comportement orienté but et opportuniste avec la capacité de prendre l'initiative au bon moment.
- **Sociabilité** : l'agent doit être capable d'interagir, quand la situation l'exige, avec les autres agents et les humains via une sorte de langage de communication, afin d'accomplir ses tâches ou aider les autres dans leurs activités.

- **Mobilité** : fonctionnalité supplémentaire, elle permet à l'agent de se déplacer de manière autonome à travers le réseau.
- **Apprentissage** : l'agent est capable de changer son comportement en fonction des expériences passées (Franklin, 1996).

La figure suivante (figure 1.1) résume les caractéristiques précédemment discutées :

3. Typologies des agents

Le choix du ou des types d'agents à utiliser dépend en fait du système multi-agents le plus pertinent pour le problème à résoudre. Ainsi, certains SMA n'utilisent qu'un seul type d'agents regroupés par objectif, d'autres plusieurs types correspondant à des rôles précis nécessaires à la résolution. Nous distinguons traditionnellement deux types d'agent : l'agent réactif et l'agent cognitif.

3.1 Agent réactif

Un agent réactif possède une représentation très simplifiée de son environnement. Il communique d'une façon indirecte via l'environnement, Il se caractérise par le manque de mémoire locale (il n'est pas capable de tenir compte de ses actions passées car il ne possède pas de moyen de mémorisation), l'absence de mécanismes de raisonnement et il n'est pas nécessaire que chaque agent soit personnellement intelligent pour parvenir à un comportement intelligent de l'ensemble (Les travaux sur ces agents s'intéressent plus à la modélisation d'une société d'agents qu'à l'agent lui-même).

Un agent réactif est constitué d'un ensemble de comportements, chaque comportement est une machine à états finis qui établit une relation entre une entrée sensorielle et une action en sortie (Perception/Action) Cependant, du fait, de leur nombre, ces agents

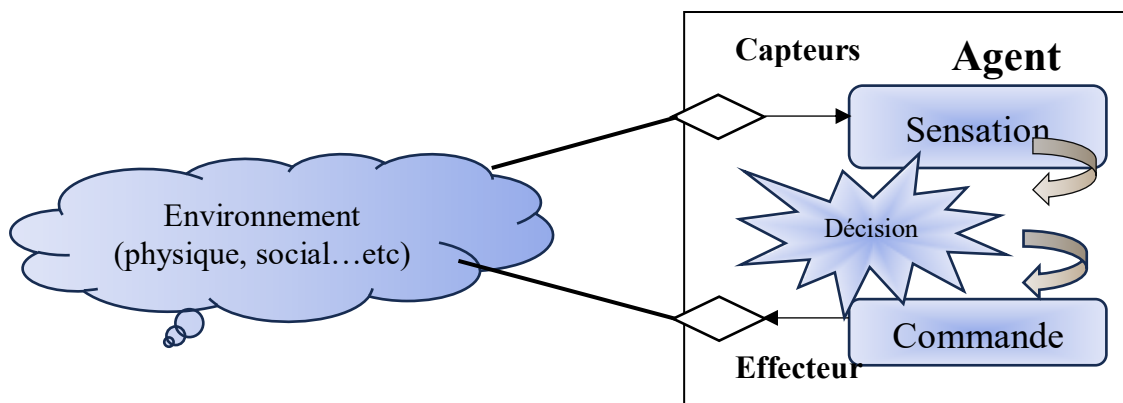


Figure 1.1 Cycle perception, action d'un agent réactif

réactifs peuvent résoudre des problèmes qualifiés de complexes. Les travaux sur ces agents s'intéressent plus à la modélisation d'une société d'agents qu'à l'agent lui-même. [1]

3.2 Agent cognitif

Contrairement à un agent réactif, un agent cognitif dispose d'une base de connaissances comprenant les diverses informations liées à son domaine d'expertise et à la gestion des interactions avec les autres agents et leur environnement. Il se caractérise par une représentation explicite de ce dernier et possède une architecture interne la plus complexe et suit un cycle Perception /Délibération/Action. [2]

Un agent cognitif est capable de planifier son comportement, de mémoriser ses actions passées, de communiquer par envoi de messages ou via des langages d'interaction élaborés, de négocier, etc.

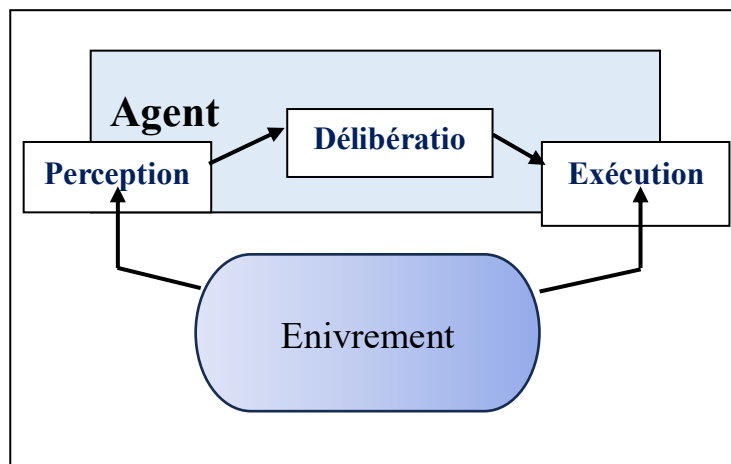


Figure1. 2 Cycle Perception Délibération Action d'un agent cognitif

Les agents cognitifs sont généralement représentés par un **état mental** ayant les attitudes mentales de l'architecture BDI : Belief (Croyance), Desire (Désir), Intention (Intention).

- **Croyances** : correspondent aux informations dont dispose l'agent sur son environnement et sur d'autres agents qui existent dans le même environnement (peuvent être incorrectes, incomplètes ou incertaines).
- **Désirs** : correspondent aux états possibles de l'environnement que l'agent aimerait atteindre.
- **Intention** : les actions que l'agent a décidé d'accomplir pour satisfaire ses désirs

3.3 Agent hybride

Les systèmes réactifs et les systèmes cognitifs peuvent ne pas convenir à la résolution de tous les problèmes. Leur différence se situe au niveau de leurs architectures internes et de la manière dont les informations perçues par ces agents sont traitées, et chacun de ces deux types de systèmes convient à certains types de problèmes et moins bien pour d'autres. Toutefois, il est maintenant possible de concevoir des systèmes hétérogènes comportant les

deux types de comportements cognitif et réactif : les agents hybrides.

Un agent hybride est composé de modules qui gèrent indépendamment la partie réactive et la partie cognitive. Il possède des capacités cognitives et réactives et conjugue la rapidité de réponse des agents réactifs ainsi que les capacités de raisonnement des agents cognitifs. Cette approche est intéressante et semble apporter une solution adéquate pour modéliser les systèmes complexes dont l'environnement est dynamique, mais elle ne résout pas clairement le problème d'interaction entre les différents modules. [3]

4. Agents cognitifs VS Agents réactifs:

Chacune de ces architectures a ses avantages et ses inconvénients. Dans notre cas, on adoptera l'architecture d'agent réactif dans la construction d'un Système d'Agents pour la résolution de problèmes complexes, vu la similarité entre un système complexe et un système formé d'agents réactifs : imprévisibilité du comportement et grand nombre d'entités (agents) en interaction. Les agents réactifs sont la solution aux problèmes de : la limite d'adaptation à son environnement, la complexité du système de communication, ajustement difficile des règles de comportement rencontrées par les agents cognitifs.

En fait, la plupart du temps un agent réactif n'est pas seul dans son environnement. Les agents doivent être capables d'interagir entre eux. Ils peuvent soit coexister, coopérer ou être en compétition entre eux. On parle alors d'un système multi agents.

5. Architecture d'agent

Dans la plupart des travaux existants Gas 87, Woo 94, Sic 95 faisant intervenir une représentation des agents, la détection de possibilité de coopération consiste en un raisonnement sur cette présentation. Il faut donc que l'agent dispose de certaines connaissances et capacités, pour pouvoir participer à la résolution des problèmes.

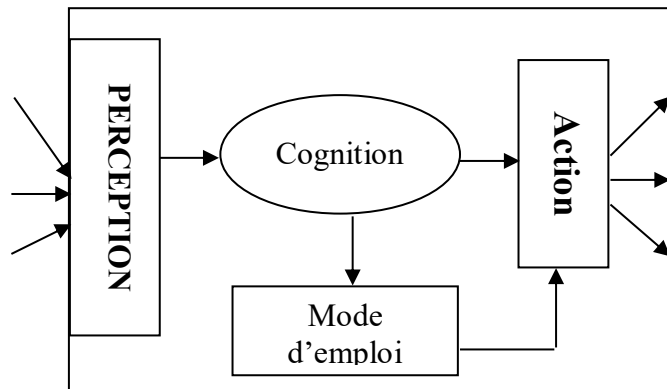


Figure 1.3 Structure d'un agent

Nous regroupons celles-ci dans un module présent dans chaque agent que nous baptisons 'Mode d'emploi'. La figure I.6 montre l'architecture de l'agent que nous avons proposée.

La modification de ce mode d'emploi est dépendante de la modification des agents du même groupe. Pour toute modification d'un agent, ce dernier informe tous ses collègues qui mettent à jour leurs représentations ce qui modifie le mode d'emploi.

A part le mode d'emploi, l'agent possède un *module de perception* lui permettant de percevoir son environnement et mettre à jour ses représentations internes de l'environnement et des autres agents. *Module d'action* : celle-ci consiste en la réalisation effective des actions qui ont été décidées. L'agent qui reçoit l'action sera désigné par le mode d'emploi. Pendant la *phase de cognition* ou de raisonnement, l'agent détermine ce qui est à faire et décide quand et comment y parvenir.

Une large palette de types de raisonnement peut être utilisée et implémentée, les extrêmes étant les raisonnements réactifs (réflexe) et cognitif (de haut niveau). On n'a pas voulu donner une architecture bien définie afin de ne pas restreindre le domaine d'application de l'architecture proposée et pour que cette dernière s'adapte à différents problèmes. Dans ce module (cognition), on peut avoir une structure basée sur un raisonnement simple composé de quelques règles de comportement. Comme il peut y avoir un raisonnement plus compliqué composé des objectifs à satisfaire, des capacités, des rôles, nous faisons la remarque que, dans notre supposition, l'agent ne nécessite pas un module d'interprétation du moment où tous les agents du même groupe parlent le même langage.

6. Communication des agents

La communication qui est la base de toute interaction. Décrite par le transfert d'une information d'un agent à un ou plusieurs autres agents à l'aide d'un langage articulé ou par d'autres codes. Comme décrit plus haut la communication peut prendre d'autres formes que l'envoi de message soit par la manipulation des objets de l'environnement.

La classification la plus pertinente de la communication est la classification qui se base sur l'intention des agents Austin en 1962 dans son article (*How To Do Things With Words*) étudie l'effet d'un énoncé sur le destinataire. Ces études ont classé la communication en trois actes locutoire illocutoire et perlocutoire.

Les actes illocutoires des langages peuvent être :

- Des actes assertifs : donner une information sur le monde en affirmant quelque chose, comme Penser, Affirmer, Dire, Informer ;
- **Des actes directifs** : donner des directives au destinataire comme *Demander, Réclamer, Exiger* ;
- **Des actes permissifs** : engagent le locuteur à accomplir certaines actions dans l'avenir comme *Promettre, Garantir, refuser* ;

- **Des actes expressifs** : donner au destinataire des indicateurs sur l'état mental du locuteur comme *Féliciter, Excuser, Approuver* ;
- **Des actes déclaratifs** : le locuteur accomplit au moment de l'énonciation l'action qu'il dit accomplir comme *Stipuler, Déclarer*.

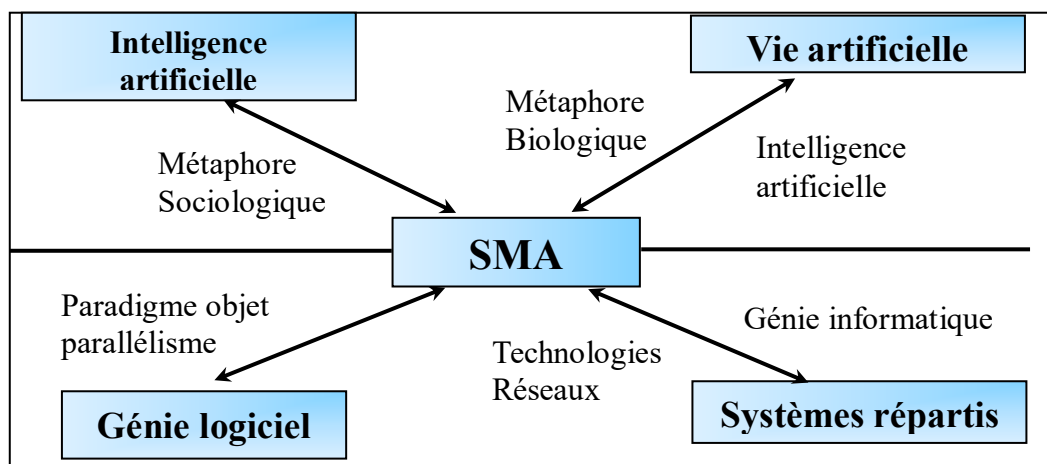
En 1995, l'organisation FIPA (Foundation For Intelligent Physical Agents) a commencé le travail sur des standards dans le domaine des agents. Le langage FIPAACL (FIPA- Agent Communication Language) est le produit essentiel de cette initiative.

Il est basé sur la théorie d'acte de langage et il n'exige pas une syntaxe spécifique pour le contenu de message. Cependant, il définit 22 performatives réparties sur les catégories suivantes :

- **Transmettre l'information** : *inform, inform-if, confirm,..etc.*
- **Demander une information** : *query, subscribe, &etc.*
- **Négocier** : *CFP, propose, accept-propose, &etc.*
- **Accomplir une action (ou distribution des tâches)** : *request, cancel, refuse, agree, &etc.*
- **Gérer un problème** : *failure, not-understood.*

7. Système multi-agents (SMA):

L'utilisation des systèmes multi-agents (SMA) sont à l'intersection de plusieurs domaines de recherches, généralement on peut distinguer les quatre axes suivants : L'intelligence Artificielle Distribuée basée sur la métaphore sociologique, le Génie Logiciel basé sur le paradigme objet, et les Systèmes Repartis basés sur les technologies réseaux et la Vie Artificielle, qui se base sur la métaphore biologique est inspirée des principes de bases sur la physique.



Figures 1.4 Positionnement des SMA

7.1 Définition d'un SMA (Système Multi-Agents) :

Plusieurs groupes de recherche proposent actuellement des définitions pour un SMA. On citera les plus connues

- [Ferber, 1995] : «Un SMA est défini comme:
- Un ensemble B d'entités plongées dans un environnement E (E est caractérisé par l'ensemble des états de l'environnement S)
- Un ensemble A d'agents avec $A \subseteq B$
- Un système d'action (opérations) permettant à des agents d'agir dans E (une opération est une fonction de $S \Rightarrow S$)
- Un système de communication entre Agents (envoi de messages, diffusion de signaux.
- Une organisation O structurant l'ensemble des agents et définissant les fonctions remplies par les agents (notion de rôle et éventuellement de groupes).

7.2 Organisation dans les SMA

Dans un SMA, les agents ont une vision locale de leur environnement, ils sont amenés à coopérer pour atteindre l'objectif global du système. Il est donc nécessaire de définir une structure organisationnelle à l'intérieur du SMA afin d'établir :

- La communication entre les agents,
- L'Interaction entre agents,
- La négociation entre les agents.

7.3 Types de communication entre agents

Les agents communiquent entre eux pour pouvoir échanger des données, des informations et coordonner leurs activités. Ils peuvent interagir de deux manières, soit par signaux, via l'environnement, soit directe par envois de messages.

Il existe deux types de communication : les communications directes et indirectes [70].

La communication indirecte : Le partage d'information dans cette démarche de communication passe généralement par l'environnement. Ainsi, pour assurer la communication, les agents doivent agir sur leur environnement. Ce mode de communication implique que les agents doivent être implémentés dans un environnement physique commun. La

communication directe : La communication directe est une démarche de communication, où l'agent s'adresse individuellement aux autres par l'envoi de messages. Ce type de communication est basé sur les trois éléments suivants :

- **Le langage de communication** : L'intérêt des langages de communication est de faciliter l'échange et l'interprétation des messages et l'interopérabilité entre les agents.

- Ces langages se focalisent essentiellement sur la manière de décrire exhaustivement

- des actes de communication d'un point de vue syntaxique et sémantique.

- **L'ontologie** : est une spécification d'objets, de concepts et de relations dans un domaine d'intérêt. L'intérêt d'une ontologie quand elle est partagée par des agents pour représenter leur connaissance est qu'ils ont les moyens de comprendre les « mots » utilisés dans une communication.

- **Les supports de communication** : sont des mécanismes utilisés pour stocker et rechercher des messages.

- **Interaction entre agents**

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

- En général, les interactions sont mises en oeuvre par un transfert d'informations entre agents ou bien entre l'environnement et les agents, soit par perception, soit par communication. L'interaction peut être décomposée en trois phases non nécessairement séquentielles [41] :

- Réception d'informations,

- Raisonnement sur les autres agents à partir des informations acquises,

➤ Emission de messages ou plusieurs actions modifiant l'environnement.

➤ **Coopération**

➤ La coopération désigne que les agents doivent s'entraider pour atteindre leurs objectifs qui peuvent être éventuellement communs. On dira que plusieurs agents coopèrent ou encore qu'ils soient dans une situation de coopération si l'une des conditions suivantes est vérifiée :

➤ L'ajout d'un nouvel agent permet d'accroître différenciellement les performances du groupe,

➤ L'action des agents sert à éviter ou à résoudre des conflits potentiels ou actuels.

➤ **Coordination**

La coordination est un processus dans lequel les agents se sont engagés en vue d'assurer une communauté d'agents individuels agissant avec cohérence et harmonie.

Les agents ont besoin de la coordination, pour être coordonnés de la même manière, puisqu'aucun agent ne possède une vue globale sur le système et parce qu'ils possèdent des capacités et expertises différentes, mais complémentaires.

La coordination comprend aussi l'allocation des tâches, qui consiste à affecter des responsabilités et des ressources nécessaires à la résolution de problèmes à un agent. Le créateur du système d'agents peut allouer toutes les tâches d'avance en engendrant ainsi une organisation de résolution des problèmes qui est non adaptable. Par contre, on peut avoir une allocation dynamique et flexible des tâches.

La planification fait également partie de la coordination. Pour un agent, elle constitue un processus de construction d'une séquence d'actions en tenant compte seulement des objectifs, des capacités et des contraintes environnementales. La planification a pour rôle d'éviter les conflits.

➤ **Négociation**

La négociation est un processus de coordination et de résolution de conflits, son objectif est d'atteindre un accord final accepté par un groupe d'agents. Il existe deux types de négociation:

les négociations compétitives et coopératives. [4]

- **Négociation compétitive** : La négociation compétitive est valable dans la situation où des agents de différents intérêts tentent de faire un choix de groupe avec des alternatives bien définies.
- **Négociation coopérative** : La négociation coopérative est utilisée dans la situation où les agents ont un objectif unique global considéré pour le système.
- **Avantages des SMA**
- L'utilisation des Systèmes Multi-Agents (**SMA**) présente une série d'avantages [5], qui se résume comme suit :
 - **Système dynamique** : Les agents sont structurés afin d'exercer une influence sur chacun pour faire évoluer le système dans son ensemble (système dynamique). On rencontre de nombreuses interactions entre agents telles que la coordination, la négociation et la coopération.
 - **Robustesse et sûreté de fonctionnement** : La mise hors fonctionnement de quelques agents ne modifie pas sensiblement le comportement global du système.
 - **Souplesse de l'outil informatique** : Qui permet de modifier le comportement des agents.
 - **flexibilité et traitement des systèmes à grandes échelles** : Nous pouvons toujours augmenter le nombre d'agents pour traiter des systèmes de plus en plus grands, sans pour autant perturber le travail des agents existants.
 - **Résolution distribuée de problèmes** : Il est possible de décomposer un problème en sous-parties et de résoudre chacune de façon indépendante pour aboutir à une solution stable.

8. Application des SMA aux systèmes de transports

La transposition des notions orientées agent au domaine des transports est en accord avec les caractéristiques des deux domaines, on y trouve des notions d'autonomie, de distribution et d'environnement partiellement observable, il est donc logique que l'on retrouve des concepts tirés du paradigme agents dans le domaine des transports. [6]

Nous considérons que les entités telles que des moyens de transport et les infrastructures peuvent être modalisés sous forme d'agents intelligents capable d'interagir dans un environnement SMA, dans le but de résoudre divers problèmes liés à la gestion des réseaux routiers. Citons comme exemple :

- **Gestion des intersections** : La gestion des intersections est un problème récurrent dans le milieu routier, la solution la plus utilisée est l'installation des feux de signalisation dans les

endroits critiques. Le cycle de feux de ces entités est généralement périodique, mais en dotant ces entités et les moyens de transport de la technologie agent et les faire coopérer, les feux de signalisation pourront adapter le cycle de feux et de le réguler par rapport à la densité du trafic dans un secteur donné.

➤ **Réduction des accidents** : L'environnement routier est un environnement partiellement observable, le conducteur a peu d'informations sur les intentions des autres conducteurs et les informations d'autres véhicules, ce qui engendre beaucoup de situations de conflits qui se termine généralement par des accidents.

La collaboration et la coopération des agents permet d'anticiper les tâches des autres entités routières, rendre l'environnement de l'agent mieux observable et synchroniser leurs comportements pour éviter les accidents routiers.

➤ **Gestion de la congestion** : Problème le plus étudié dans le système de transport, de par le nombre d'entités mobiles en circulation. L'application des SMA permet d'envisager à tout moment des modèles d'organisation permettant de fluidifier la circulation et d'anticiper des échappatoires pour réduire la densité routière dans un secteur en particulier.

➤ **Autres problèmes** : D'autres problèmes de gestion liés au système de transport sont d'actualité tels que : la régulation du trafic routier, gestion des parkings, etc. les SMA peuvent offrir diverses solutions à la résolution de ces problèmes.

8 .Conclusion

Comme conclusion de ce chapitre, nous étudions les systèmes multi agent (SMA) et la façon dont ils communiquent et les différentes plateformes de développement sur lesquelles nous avons développé la plupart des applications nous avons installé ce système spécialisé d'intelligence artificielle

Chapitre 2:
Les systèmes de transports intelligents

1.Introduction

L'homme depuis son existence a cherché des moyens pour lui faciliter ses déplacements. Aujourd'hui, nous pouvons dire que nous avons réalisé le rêve de nos ancêtres tout en construisant des puissantes infrastructures et inventant des moyens de transport diverses avec des capacités énormes. Malheureusement, ces avancées rapides couplées au désir de se déplacer de plus en plus rapidement, mènent à de nouvelles peines graves : les accidents, la pollution et les embouteillages. Pour faire face à ses problèmes et augmenter l'infrastructure actuelle, de nombreuses méthodes ont été proposées tels les panneaux à message variable ; les dos d'âne ; les décisions gouvernementales permettant de lutter contre la pollution en interdisant les jours de forte pollution, l'accès au centre-ville pour les véhicules avec des numéros immatriculés pairs ou impairs et en favorisant l'utilisation des véhicules écologiques , en donnant des avantages aux consommateurs, sous la forme d'une réduction sur les taxes du carbone et les cotisations d'assurance. A la fin du XXème siècle, les chercheurs tendent à introduire l'intelligence dans les systèmes de transport, ce qui a donné une naissance à un nouveau domaine de recherche dénommé Systèmes de Transport Intelligents abrégé STI. A leurs débuts, les initiatives étaient limitées par la télésurveillance sur les accidents de la circulation à l'aide de caméras pour informer les utilisateurs via des panneaux à messages variable. Peu après, avec l'arrivée de la communication sans fil, le monde a ouvert ses portes en direction de la recherche sur la communication inter-véhiculaires. Aujourd'hui, sans même y penser, par un simple coup d'œil le conducteur peut consulter diverses applications technologiques (niveau de carburant et d'huile, température du moteur, vitesse, kilométrage effectué, allumage des phares, pression des pneumatiques, accrochage des ceintures de sécurité, radar anti-collision, température extérieure, signal sonore d'un risque de verglas ou de brouillard ...), qui sont introduites pour lui faciliter la conduite, rendre le voyage plus sécurisé, plus confortable et même plus amusant.

2 .Définition des systèmes de transport intelligent

ITS (en anglais : intelligent transportation system) sont des applications des nouvelles technologies de l'information et de la communication au domaine des transports et de sa logistique.

C'est un système interactif de collecte, de traitement, et de diffusion d'information appliqué aux transports, basé sur l'intégration des technologies de l'information et de la communication aux infrastructures et aux véhicules utilisés de manière à améliorer la gestion et l'exploitation des réseaux de transport et des services aux utilisateurs associés. La Figure 1.1 illustre un exemple de deux véhicules connectés en interaction entre eux et avec leurs voisinages.

Des nouvelles technologies appliquées aux réseaux de transport pour en améliorer : la gestion, l'exploitation, les services aux utilisateurs. [1]

La gamme des technologies considérées comprend toutes les applications de la télématique au domaine du transport, utilisant notamment l'électronique embarquée ou xe (p.ex. : capteurs, moyens de calcul), les télécommunications, les bases de données et d'information, les systèmes de régulation, les paiements électroniques.

Tous les modes de transport routier, ferroviaire, aérien, maritime sont visés par ces applications, tant pour la sécurité ou la régulation des flux de la circulation que pour l'information des usagers des transports en commun ou des usagers du transport des marchandises. [2]

Notre étude se consacre pour les systèmes de transports routiers.



Figure 2.1: Exemple de deux véhicules connectés

3. Les origines des ITS pour la route

Selon l'histoire des ITS pour la route s'étale des années 60 jusqu'à nos jours, et elle peut être découpée en trois grandes phases :

La première phase, est déterminée par l'étude de faisabilité et de préparation des technologies de base qui servent de support aux transports intelligents.

Alors que la deuxième, est caractérisée par la mise en place des premières applications résultantes des premières études.

En fin n, la dernière phase qui non seulement mis l'accent sur l'importance des ITS pour la gestion de tra c mais de les considérer comme étant des outils de développement des Pays.

3.1 Premières études (années 60 années 70)

Les attrayants projets de cette époque c'étaient : le programme de recherche CACS "Comprehensive Automobile tra c Control System" qui s'est étendu de 1973 à 1979 , Il s'agissait du premier partenariat public-privé dans le monde ayant testé en zone urbaine un système de navigation interactif embarqué possédant un écran ; le projet ERGS "Electronic Route Guidance System" aux États-Unis et le projet similaire ALI "Autofahrer Leitund Information System" en Allemagne. Ces trois issues étaient basées sur des systèmes de communication reliés à un énorme ordinateur central. Malheureusement, vu les faibles capacités de calcul des systèmes embarqués de cette période et de l'importante puissance requise pour le serveur principal, ces projets n'ont jamais vu le jour.

3.2 Premières applications (années 80 milieu des années 90)

La période des années 80 a été marquée par plusieurs avancées technologiques (développement de la puissance des processeurs, capacités de calcul, augmentation de la taille mémoire, etc.). Le domaine des transports a largement bénéficié de ces évolutions qui ont abouti à l'essor de nombreux projets. Ces derniers s'intéressaient au développement de solutions réelles, concrètes et fonctionnelles afin de les commercialiser. Parmi ces projets, nous citons les initiatives européennes représentées par le programme EUREKA créé en 1986 et son projet PROMETHEUS "Program for Européen Traffic with Highest Efficiency and Unprecedented Safety", qui est une initiative soutenue par des constructeurs automobiles visant à développer les transports intelligents. Le programme DRIVE I "Dedicated Road Infrastructure for Vehicle safety in Europe", créé par les autorités européennes en 1988 et achevé en 1991, a été repris avec le programme DRIVE II l'année suivante. Du côté américain, cette époque s'est caractérisée par la formation d'un groupe d'étude informel dénommé "Mobility 2000" en 1988,

ainsi que la création du programme national IVHS America "Intelligent Vehicle Highway Society of America" en 1990. L'intégration du groupe d'étude au projet par le gouvernement en 1992 lors de son lancement. Simultanément, le rival japonais a créé en 1984 le projet RACS "Road/Automobile Communication System" par le ministère de la construction (MOC "Ministry Of Construction"), et en 1987 le projet AMTICS "Advanced Mobile Traffic Information and Communication Systems" de l'agence nationale de la police (NPA "National Police Agency"), qui ont aidé à la création des éléments de base des systèmes de navigation actuels. En 1991, le ministère des postes et des télécommunications a uni ces deux projets dans le projet VICS ciblant leur standardisation. En plus de ceux mentionnés ci-dessus, cette période a été marquée par la mise en place de quatre autres projets importants dans l'histoire des systèmes de transport intelligents :

Le projet ARTS "Advanced Road Transportation Systems", du ministère de la construction, qui commença en 1989 et dont l'objectif a été la conception de nouveaux types de voies.

Le projet SSVS "Super Smart Vehicle System", du ministère du commerce extérieur et de l'industrie, a été mis en place en 1990 dans le but de créer des véhicules qui interagissent entre eux et avec la route.

Le projet ASV "Advanced Safety Vehicle", du ministère des transports, qui débuta en 1991, se charge de la recherche et du développement de technologies liées à la sécurité automobile.

Le projet UTMS "Universal Traffic Management System", de la police nationale, établi en 1991 a n d'améliorer la surveillance et l'organisation du trafic. Globalement, les projets cités ci-dessus issus de nombreux endroits (privés et publics), ont aidé à accélérer les recherches dans le domaine des ITS.

3.3 Les grands projets (milieu des années 90 aujourd'hui)

À Paris, en 1994, avec l'organisation du premier congrès mondial sur les ITS, la phase actuelle de l'histoire a commencé. Cette phase est fortement marquée par sa dimension mondiale. En 1994, aux États-Unis, le projet IVHS a été renommé ITS America "Intelligent Transportation System America" pour élargir le champ de recherche ainsi que réaffirmer la volonté du gouvernement d'encourager les ITS. En Europe, les projets PROMOTE "Programme for Mobility in Transportation in Europe" et TAP "Telematics Applications Programme" ont suivi les projets PROMETHEUS et DRIVE II. Il ne faut pas omettre de mentionner la création de l'organisation public-privée ERTICO "European Road Transport Telematics Implémentation Coordination Organizations" qui a pour but de créer un réseau d'informations sur les ITS et de faciliter de nouvelles collaborations.

4 .Les objectifs des systèmes de transport intelligent

Les ITS connaissent de nombreuses applications dont :La réduction de l'encombrement et l'amélioration de la mobilité.L'amélioration de la sécurité du réseau de transports.

- L'accroissement de la productivité économique.
- la réduction du temps de déplacement et des coûts pour le gouvernement, le voyageur et l'exploitant.
- L'amélioration de l'efficacité énergétique et la réduction des effets sur l'environnement.

Nous donnons ci-dessous des exemples de certains de ces avantages tirés de projets ITS mis en œuvre au Canada, aux États-Unis, en Europe et au Japon. Réduction des accidents en zone rurale à l'aide des services 911 et autres services de gestion des véhicules d'urgence, des systèmes anticollisions, des fonctions de prévisions météo, etc.

- Accroissement des débouchés et création de nouveaux créneaux pour les fournisseurs et les usagers.
- Réduction du fardeau administratif et des coûts d'exploitation grâce à l'amélioration de l'efficacité des systèmes au moyen de fonctions automatisées et de transactions électroniques.
- Amélioration de la surveillance et de la gestion des flux de trafic et des incidents reliés au transport des marchandises dangereuses.
- Amélioration de la collecte de données sur les flux de trafic, les marchandises transportées, les transporteurs, les conducteurs et les charges marchandises par les autorités économiques, commerciales et de réglementation, les administrateurs d'installation et les fournisseurs de transport, permettant plus d'efficacité dans la planification des politiques, la conception des infrastructures et la gestion des activités. [3]

4.1Un transport plusable

Les applications de ITS ont permis un redressement de 12 à 23% de la ponctualité du transport en commun, le temps d'attente des passagers étant réduit jusqu'à 50%. Par exemple, Kansas City (Missouri) a amélioré la ponctualité de ses autobus urbains de 12%, tout en réduisant son parc d'autobus de 9%.

Les systèmes de paiement électronique mis en place ont gagné jusqu'à 90% de la faveur des habitués. Ces systèmes ont augmenté la perception du péage de 3%, passant à 30%.4.2Une productivité économique améliorée

Nous estimons que le système COMPAS (système de gestion de la circulation autoroutière au Canada) permet aux exploitants de véhicules commerciaux d'économiser 55 millions de dollars annuellement et a généré 20 millions de dollars en exportations par année depuis 1993. Le département

américain des transports estime que le déploiement des ITS peut permettre aux contribuables d'économiser 35% au chapitre des investissements d'infrastructure et de réduire les coûts de cycle de vie du réseau des transports pour la prochaine décennie de 25%, soit de 30 milliards de dollars.

Jusqu'au 2015, les États-Unis a investie en ITS 350 milliards \$US en avantages économiques directs et 600 000 emplois.

4.3 Une amélioration de la sécurité

L'expérience canadienne du système de gestion de la circulation routière COMPAS à Toronto, qui surveille la circulation sur certaines sections de l'autoroute 401, montre que les mesures des incidents de circulation et de l'encombrement ont réduit la durée des incidents entre le moment où ils surviennent et celui où ils sont éliminés de 86 à 30 minutes. Le retard moyen par incident a été réduit de 537 véhicules-heures. L'a chage de messages d'incident au moment où ceux-ci surviennent a permis de prédire environ 200 accidents par année, entraînant des économies de 10 millions de dollars.

L'expérience aux États-Unis révèle une réduction du nombre d'accidents allant de 15 à 62%. Plus précisément, le projet FAST-TRAC à Oakland (Michigan) a entraîné une réduction de 89% des accidents de virage à gauche, une réduction de 27% du nombre total de blessures et une réduction de 100% des blessures graves. Le projet Guidestar TMS à Minneapolis a permis une réduction de 25% des accidents, une augmentation de 35 de la vitesse moyenne à l'heure de pointe et un accroissement de la capacité routière de 22%.

Le comté de Fulton (Géorgie) a réduit le délai moyen d'intervention en cas d'incendie de 7,5 à 4,5 minutes.

4.4 Des économies de temps et gains d'efficacité opérationnelle

Le système de gestion routière COMPAS a permis de réduire les délais globaux de 5,3 millions de véhicules-heures par année et la consommation de carburant de 11,3 millions de litres par année.

L'expérience japonaise révèle que les mesures de gestion du tra c permettent de diminuer jusqu'à 11% la consommation annuelle de carburant.

La régulation assistée par ordinateur pour les chasse-neiges a permis à Indiana (USA) d'économiser 14 millions \$US par année en coûts d'exploitation et équipement.

Le système de perception électronique du péage (PIKEPASS) en Oklahoma a réduit les coûts d'exploitation à chaque poste de péage de 176 000\$ à 16 000\$ par année.

Le temps d'attente des véhicules dans les voies de péage de la ville de New York a diminué, passant de 15 minutes à moins de 30 secondes depuis l'introduction du système de péage E-Z Pass.

5. Une réduction des effets sur l'environnement

Le système de gestion routier COMPAS a réduit les émissions de 3 100 tonnes d'oxyde de carbone par année.

Une récente étude commandée par la table des transports, dans le cadre du processus national sur le changement climatique du Canada, sur les effets de sept applications ITS sur les émissions de gaz à effet de serre, a estimé la réduction annuelle de ces émissions en 2010 à 763 milliers de tonnes. Cette réduction représente 0,5% des émissions totales de gaz à effet de serre attribuable au transport en 1995. Les réductions connexes dans la consommation de carburant sont estimées à près de 300 millions de litres.

5.1 Catégorisation des applications relatives aux ITS

5.1.1 Information pour les voyageurs

Informations de préparation au voyage, informations durant le voyage pour le conducteur, information durant le voyage dans les transports publics, informations routières, localisations, etc. [10]

5.1.2 Gestion du trafic

Contrôle du trafic, gestion des accidents, gestion des flux, gestion de maintenance des infrastructures, planification des infrastructures etc, comme le montre la Figure 1.2:

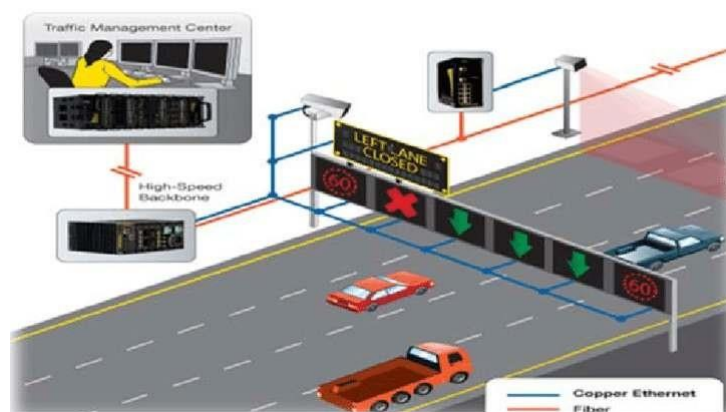


Figure 2.2 Gestion du trafic routier pour un système de transport intelligent

5.1.3 Véhicule

Amélioration de la couverture visuelle, fonctionnement automatisé du véhicule, évitement des collisions (latérales et longitudinales), sécurité, maintenance du véhicule, etc.

5.1.4 Véhicule commercial

Pré-dédouanement des véhicules utilitaires, procédures administratives des véhicules utilitaires, inspection automatique de la sécurité routière, surveillance de la sécurité à bord des véhicules utilitaires, gestion du parc automobile, systèmes de diagnostic automatisés, etc

5.1.5 Transport public

Gestion des transports publics, gestion des transports à la demande, gestion du transport partagé, gestion des horaires, localisation, etc.

5.1.6 Gestion des urgences

Avis d'urgence et sécurité personnelle des voyageurs, gestion des véhicules d'urgence, gestion du transport des matériaux dangereux et notification d'incidents, etc.

5.1.7 Paiement électronique

Opérations financières électroniques (péages).

5.1.8 Sécurité

Sécurité publique des voyages, amélioration de la sécurité pour les usagers, jonctions intelligentes, etc. [11]

6. Exemple d'architecture canadienne

L'architecture canadienne Figure 1.3 : "fournit un cadre d'intégration unique pour guider la mise en place coordonnée des programmes de ITS dans les secteurs public et privé". [11]

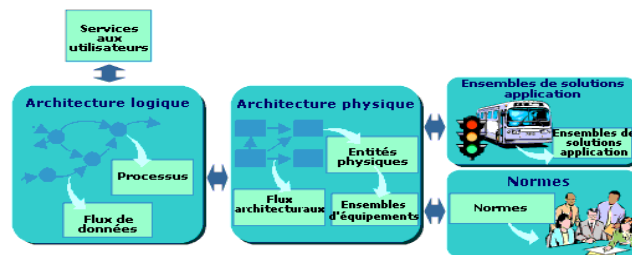


Figure 2.3 Architecture canadienne des ITS

1. Information à l'intention des voyageurs

2. Gestion du trafic

3. Gestion du transport en commun

4. Exploitation de véhicules commerciaux

5. Gestion des urgences

6. Systèmes avancés de sécurité des véhicules

7. Gestion de l'information

8. Gestion des travaux d'entretien et de construction

la Figure suivante 1.4 représente l'architecture général d'un système de transport intelligent:



Figure 2.4 Architecture générale d'un ITS

6.1 Gestion du transport en commun

Nous pouvons dire que le transport en commun est le secteur qui attire plus d'avantages des ITS, en se basant sur une architecture moderne illustrée par la Figure et qui nous offre plusieurs avantages, que nous citons:

- Gestion du transport en commun.
- Information en cours de route.
- Transport en commun adapté à la demande.
- Sécurité dans les transports en commun.
- Contexte routier (Autoroutier, Route municipaux).
- Contexte transport collectif.

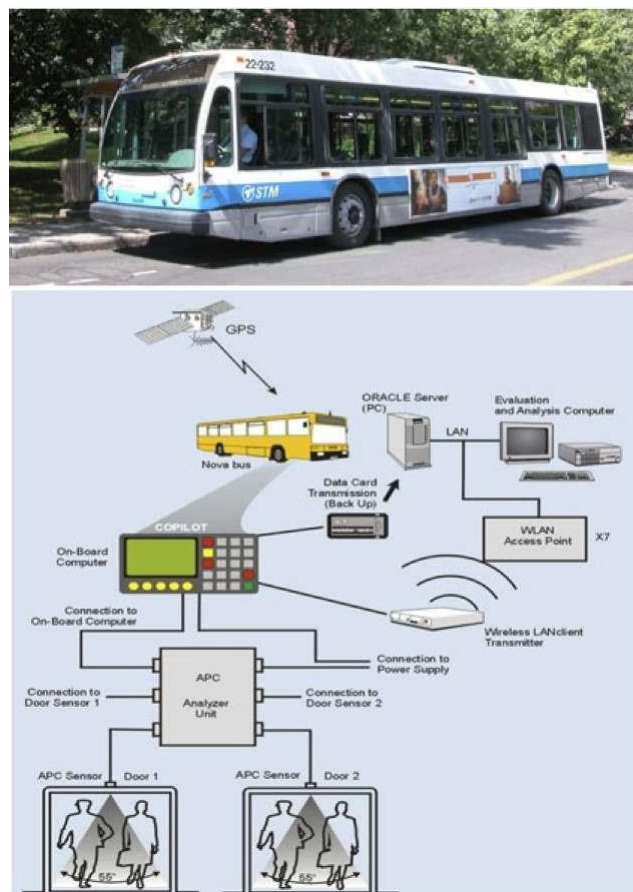


Figure 2.5 Gestion de transport en commun

7. Technologies intégrées dans les systèmes de transport intelligent

Les technologies utilisées dans les systèmes de transport intelligents varient, allant de systèmes de gestion basiques comme les systèmes de gestion des carrefours à feux, les systèmes de gestion des conteneurs, les panneaux à messages variables, les radars automatiques ou la vidéo-surveillance aux applications plus avancées qui intègrent des données en temps-réel avec retours d'informations de nombreuses sources, comme les informations météorologiques, les systèmes de dégivrage des ponts, les systèmes de navigation embarqués informant des temps de parcours en temps réel etc. De plus, les techniques prédictives sont développées pour permettre une modélisation avancée et une comparaison avec une base regroupant des données historiques de référence.

Quelques technologies typiquement implantées dans les ITS sont décrites dans les sections qui suivent :

7.1 Communications sans fil

Diverses technologies de communication sans fil Figure 1.6 sont proposées pour les systèmes de transport intelligent :

- Des communications à courte portée (moins de 350 mètres) comme le Wi-Fi.
- Des communications à plus longue portée comme le WiMAX, le GSM ou les technologies 3G, 4G et dans le futur, 5G, et LPWAN. [12]

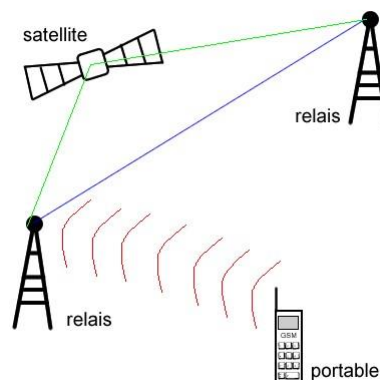


Figure 2.6: Communication sans fil

7.2 Technologies de calcul

Progrès récents dans l'électronique embarquée sur les gaines de broyeur dans les véhicules équipés de processeurs informatiques plus efficaces. Un véhicule typique du début des années 2000 aurait entre 20 et 100 modules individuels basés sur des microcontrôleurs ou des contrôleurs logiques programmables interconnectés dans un réseau avec des systèmes d'exploitation en temps non réel. La tendance actuelle est d'aller vers quelques modules à microprocesseur plus chers avec une mémoire matérielle en temps réel et une gestion du système d'exploitation. Les nouvelles plates-formes informatiques embarquées permettent la mise en œuvre d'applications logicielles plus sophistiquées, en particulier le contrôle des processus de modélisation informatique, l'intelligence artificielle et l'informatique omniprésente. L'intelligence artificielle est probablement la plus importante de ces approches pour les systèmes de transport intelligents.

7.3 Technologies de localisation

Géolocalisation par système de positionnement par satellite, Le principe du positionnement par satellites est très proche du principe de triangulation. La distance entre l'utilisateur du terminal GNSS et un certain nombre de satellites de positions connues est mesurée pour permettre une localisation de l'utilisateur à une dizaine de mètres près. La vitesse de déplacement est aussi disponible.

Cette technologie est née avec le GPS américain ce qu'on décrivons dans le chapitre 3, les constellations satellitaires commandant des systèmes de GNSS se sont multipliées (GLONASS russe, Beidou chinois et Galileo européen). Au même temps les performances des systèmes s'améliorent. De très nombreux ITS s'appuient donc sur cette technologie qui permet le traçage en temps réel et à bas coût des mobiles à savoir les personnes en déplacement, les marchandises et leurs conteneurs ou les véhicules. C'est l'un des déterminants essentiels de l'essor des STI, Téléphonie mobile.

En admettant que les voitures contiennent au moins un ou plusieurs téléphones mobiles ou cellulaires qui transmettent leur position de façon régulière au réseau - même s'il n'y a pas de communication vocale établie. Ils peuvent alors être utilisés dans les voitures comme des sondes anonymes du trafic. Quand la voiture bouge, le signal du téléphone mobile bouge également. Il est alors possible de mesurer et d'analyser par triangulation les données fournies par le réseau cellulaire - de manière anonyme - puis de convertir ces données en une information précise sur la circulation automobile.

Plus il y a de congestion, plus il y a de voitures, de téléphones et donc de sondes. En centre-ville, la distance entre les antennes est plus courte, la précision est ainsi augmentée. Il n'y a pas d'infrastructure spéciale construite le long des routes - seul le réseau de téléphonie mobile est mis en œuvre. Cette technologie FCD (floating car data) donne des données cellulaires offrant de grands avantages sur les méthodes classiques de mesure du trafic :

- Des coûts moindres par rapport aux capteurs et aux caméras
- Une meilleure couverture
- Une plus grande facilité de mise en œuvre : pas de zones de chantier, moins de maintenance des installations
- Une utilisation dans toutes les conditions météorologiques, même en cas de fortes pluies

Le gros inconvénient consiste en la précision de la localisation.

7.4 Technologie de capteurs

Les progrès des capteurs ont permis de développer les systèmes de transport intelligent sur la base de données fiables, fréquentes et en grande quantité. Ils permettent des mesures des caractéristiques de véhicules (longueur, poids, etc.), de la circulation (débit, taux d'occupation, vitesse, etc.) ou d'évènements (incidents, les d'attente, franchissement de feux rouges, etc.) pour notamment optimiser et améliorer de la sécurité routière. En général, les systèmes tendent à combiner généralement :

- Des systèmes de détection pour les ITS basés sur l'infrastructure (route intelligente) , Les capteurs d'infrastructure sont installés ou intégrés sur ou dans les routes, ou à proximité (immeubles, poteaux, panneaux, etc.). Ils peuvent être installés lors des travaux préventifs d'entretien des routes ou par un système mécanique d'injection de capteurs intégrés dans la route elle-même. Les capteurs de véhicules sont soit des dispositifs installés sur ou dans la route, soit des dispositifs disposés dans les véhicules,
 - Des systèmes mesurant les variations environnementales (vent, bruit, pluviométrie, gel...),
 - Des capteurs embarqués et systèmes reliant les véhicules à l'interne "véhicules connectés".

Boucles électromagnétiques

Les boucles à induction Figure 2.6 sont placées sous la chaussée pour détecter les véhicules qui passent sur la boucle en mesurant le champ magnétique créé par le véhicule. Les plus simples des détecteurs comptabilisent le nombre de voitures qui passent au-dessus de la boucle pendant un intervalle de temps donné (par exemple 60 secondes, la période standard aux États-Unis), tandis que les capteurs plus sophistiqués estiment également la vitesse, la longueur et le poids des véhicules ainsi que la distance qui les séparent. Les boucles peuvent être placées sur une simple voie ou au travers de plusieurs voies et fonctionnent aussi bien pour des véhicules très lents ou à l'arrêt que pour des véhicules se déplaçant à grande vitesse. [13]

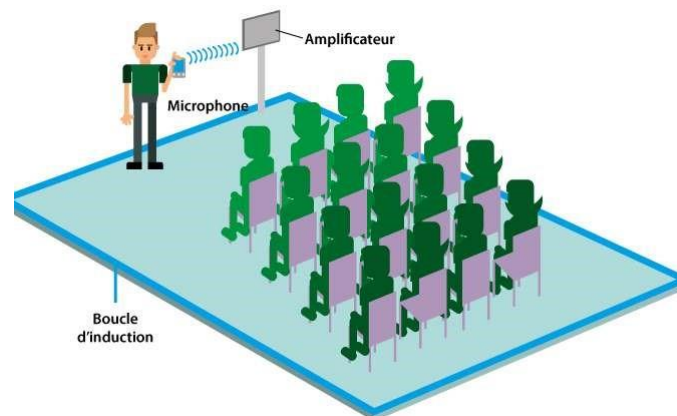


Figure 2.7 Boucle à induction magnétique(BIM)

7.5 Capteurs vidéo

La mesure du flux de trafic à base de caméra vidéo est une des autres formes existantes de détection du trafic. Comme les systèmes de détection vidéo ne nécessitent pas d'installer des composants directement à la surface ou dans la chaussée, ce type de système est dit "non intrusif". La maintenance se trouve simplifiée du fait de l'absence de contact répété entre les roues et le capteur. Les données vidéo, noir et blanc ou couleur, sont transmises à des processeurs qui analysent les changements de caractéristiques de l'image vidéo lorsqu'un véhicule passe grâce à des algorithmes de détection du mouvement. Les caméras sont installées sur des poteaux ou sur des structures adjacentes aux rues, mais toujours hors. La plupart des systèmes de détection vidéo nécessitent une configuration initiale pour apprendre au processeur l'image de fond de référence. Cela implique habituellement de préciser des

mesures connues comme la distance entre les lignes séparant les voies ou la hauteur de la caméra au-dessus de la route, mesures normalisées dans le pays. Selon la marque et le modèle, un seul processeur de détection vidéo peut détecter le trafic en simultané pour quatre à huit caméras, en fonction de la complexité des algorithmes mis en œuvre. Les données types en sortie du système de détection vidéo sont, pour chaque voie de circulation, la vitesse et le taux d'occupation de la voie. D'autres systèmes fournissent des données additionnelles tels les intervalles entre véhicules, la progression du trafic, les véhicules arrêtés et peuvent déclencher des alarmes lorsqu'un véhicule roulant à contresens est détecté par exemple.

7.6 Réseaux de capteurs sans fil

Des réseaux de capteurs sans fil Figure 1.8 sont grandement développés ces dernières années. Constitués de petits capteurs utilisant la technologie sans fil pour communiquer, ces réseaux semblent adaptés au cas de la gestion du trafic routier urbain. Outre leur réactivité et leur logique de conception naturellement distribuée, ces réseaux possèdent l'avantage d'être facilement intégrables à l'infrastructure urbaine et à faible coût, en comparaison aux boucles électromagnétiques dont le prix et l'installation sont moins accessibles.

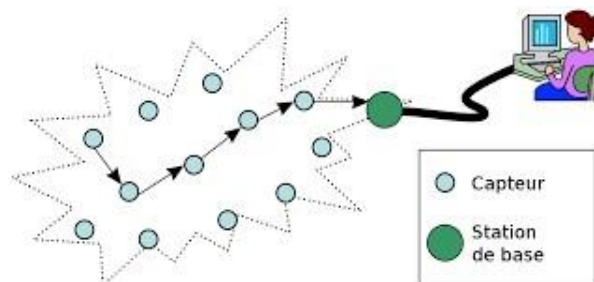


Figure 2.8 Réseaux de capteurs sans fil

7.7Autres capteurs

En fin, les boucles électromagnétiques sont les capteurs les plus utilisés. D'autres capteurs existent et sont utilisés suivant les besoins comme les capteurs piézo-céramiques, les tuyaux pneumatiques, les capteurs à hyperfréquence (radar), les bres optiques, les capteurs à ultrason, à infrarouge, etc. [14]

8.Conclusion

Comme conclusion de ce chapitre, nous avons étudié le système de transport routier et l'importance des ITS dans la vie quotidienne ensuite nous avons présenté ces différentes technologies pour améliorer la gestion et l'exploitation des réseaux de transport et des services aux utilisateurs associés.

Chapitre 3 :

Contribution

1. Introduction

Dans le vaste domaine de l'informatique et de l'intelligence artificielle, les algorithmes de recherche de chemin se distinguent comme des outils essentiels pour la navigation et la planification de trajectoires. Leur utilité s'étend à divers secteurs tels que la robotique, les jeux vidéo, la logistique, et bien d'autres. Comprendre leur fonctionnement est fondamental pour développer des systèmes autonomes performants, Cette étude se concentre sur un aspect spécifique de cette problématique, c'est la simulation de la recherche du plus court chemin par un véhicule autonome,

Ce problème, qui semble simple en apparence, pose des défis complexes en termes d'efficacité et de performance. En effet, trouver le chemin optimal tout en tenant compte des contraintes de temps et d'espace est une tâche ardue, Dans ce chapitre, nous présenterons en détail la conception et l'implémentation de notre simulateur avec la plateforme NetLogo, suivi d'une analyse approfondie des performances des algorithmes A* et de recherche en profondeur. Enfin, nous discuterons des implications de nos résultats et proposerons des pistes pour des recherches futures dans ce domaine en constante évolution.

2. La simulation informatique

2.1 Définition

La simulation par ordinateur est la discipline consistant à concevoir un modèle d'un système physique réel ou théorique, à exécuter le modèle sur un ordinateur numérique et à analyser le résultat de l'exécution [15]. C'est un processus itératif non linéaire composé de 3 tâches fondamentales, fortement interdépendantes :

- Elaboration du modèle.
- Exécution du modèle sur ordinateur.
- Analyse de l'exécution du modèle et des résultats obtenus.

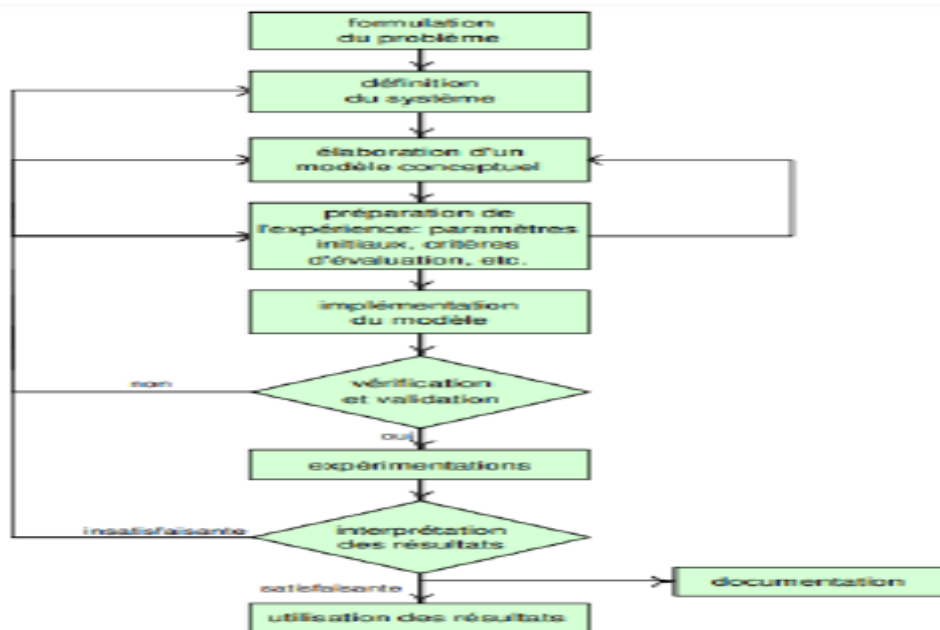


Figure 3.1 Etapes de conception d'une simulation informatique

2.2 Types de la simulation

Lorsque la simulation s'avère nécessaire pour évaluer un système réel, quatre principales méthodes de simulation peuvent être utilisées en fonction de la nature du système cible :

- **La Simulation continue** : Elle permet d'analyser de manière continue le comportement d'un système, représenté sous la forme d'équations différentielles, au cours du temps.
- **La simulation discrète** : Permet d'évaluer le comportement d'un système au cours du temps grâce à la génération d'événements.
- **La Simulation analytique** : Elle permet d'analyser des processus stochastiques à travers lesquels le système peut passer par différents états, comme par exemple les chaînes de Markov .
- Un autre type de La Simulation est celui orienté agents : les modèles à base d'agents sont généralement caractérisés par de nombreux paramètres qui déterminent la dynamique globale du système.

2.3 Intérêts de la simulation

- Parmi les intérêts de la simulation :
- Quand on ne peut pas facilement observer les états du système.

- Quand on désire analyser l'enchaînement des événements dans le système, ainsi que les relations de causes à effets.
- Quand on désire valider une solution analytique.
- Quand la complexité des interactions dans le système est telle qu'elle ne peut être étudiée qu'au travers de simulations.
- Quand on désire visualiser les états d'un système, afin d'étudier son comportement dans un contexte de surveillance ou d'aide à la décision.
- Quand on veut tester différentes optimisations pour améliorer un système déjà existant [16].

3. Simulation à base d'agents :

3.1 Modélisation et la simulation (M&S) basées sur les agents

La modélisation basée sur les agents (MBA) est un type de modélisation informatique pour simuler les actions et les interactions des agents autonomes. Ces agents peuvent être des entités individuelles ou collectives telles que des organisations ou des groupes, allant des réseaux sociaux, des groupes aux systèmes auto-organisés et aux modèles bio-inspirés [17]. MBA peut être utilisé pour développer des agents proactifs, autonomes et intelligents. La proactivité fait référence à la capacité des agents à prendre l'initiative, leur permettant de communiquer avec d'autres agents et de prendre leurs propres décisions.

Structure d'un modèle basé sur un agent Un modèle basé sur un agent comporte généralement quatre aspects :

- **Un ensemble d'agents autonomes** : chaque agent possède un ensemble d'attributs qui décrivent l'état de l'agent et un ensemble de comportements spécifiés (règles qui régissent le comportement de l'agent) qui définissent le comportement d'un agent en réponse aux changements de son environnement et, peut-être, vers un ensemble de buts ou d'objectifs
- **Un ensemble de relations d'agent** : chaque relation définit la façon dont chaque agent interagit avec les autres agents et son environnement. Cela implique également la façon dont chaque agent est «connecté» aux autres agents.
- **L'environnement de l'agent** : le monde dans lequel les agents existent, c'est-à-dire l'ensemble minimal de variables ou de structures «globales» nécessaires pour définir la façon dont les agents réagissent à leur environnement.
- **Un système** : il est composé de l'ensemble des agents, de l'environnement et de leurs relations. Un système a une frontière clairement définie avec des entrées et des sorties bien définies.

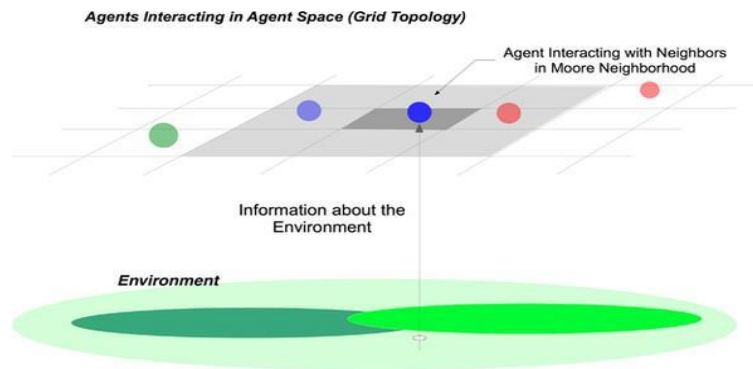


Figure 3.2. Structure d'un modèle basé sur un agent

3.3 Conception du modèle d'agent

Lors du développement d'un modèle basé sur un agent, il est utile de poser une série de questions, dont les réponses conduiront à une conception initiale du modèle :

- Quel problème spécifique devrait être résolu par le modèle ?
- À quelles questions spécifiques le modèle devrait-il répondre ?
- Quelle valeur ajoutée la modélisation basée sur les agents apporterait-elle au problème que d'autres approches de modélisation ne peuvent pas apporter ?
- Quels devraient être les agents dans le modèle ?
- Qui sont les décideurs du système ?
- Quelles sont les entités qui ont des comportements ?
- Quelles données sur les agents sont simplement descriptives (attributs statiques) ?
- Quels attributs d'agent seraient calculés de manière endogène par le modèle et mis à jour dans les agents (attributs dynamiques) ?
- Quel est l'environnement des agents ?
- Comment les agents interagissent-ils avec l'environnement ?
- La mobilité d'un agent dans l'espace est-elle un élément important à considérer ?
- Quels comportements des agents sont intéressants ?
- Quelles décisions les agents prennent-ils ? Quels comportements sont mis en œuvre ?
- Quelles mesures sont prises par les agents ?
- Comment les agents interagissent-ils entre eux ?
- Avec l'environnement ?
- Quelle est l'ampleur ou la concentration des interactions avec les agents ?
- D'où pourraient provenir les données, notamment sur les comportements des agents, pour un tel modèle ?

- Comment pourriez-vous valider le modèle, en particulier les comportements des agents ? Répondre à ces questions est un élément essentiel du processus de conception d'un modèle basé sur un agent. Il existe différentes approches pour concevoir et mettre en œuvre des modèles basés sur des agents [40].

3.4 Plateformes basées sur le paradigme agent

Dans le domaine des plateformes basées sur des agents, un certain nombre sont disponibles. Celles-ci vont des plateformes basés sur Java tels que Mason aux o plateformes basés sur Logo tels que : StarLogo, NetLogo etc. Chacune a des forces et des faiblesses différentes [37].

3.4.1 Netlogo

NetLogo est une plateforme multi-agents basé sur le langage de programmation Logo. Dans sa version actuelle, NetLogo est distribué en open source. NetLogo a été conçu et développé dès le départ pour la modélisation et la simulation de systèmes complexes évoluant dans le temps. NetLogo a été utilisé pour modéliser et simuler divers systèmes complexes dans divers domaines tels que l'économie, la biologie, la physique, la chimie, la psychologie, la dynamique des systèmes et de nombreuses autres sciences naturelles et sociales [18].

NetLogo est un outil puissant et polyvalent qui facilite la modélisation et l'étude de systèmes complexes. Sa facilité d'utilisation associée à sa capacité à gérer des simulations complexes en fait une plateforme idéale pour les enseignants, les étudiants et les chercheurs. Que vous soyez intéressé par l'écologie, En sciences sociales ou autres domaines, NetLogo fournit les outils nécessaires pour explorer et comprendre les phénomènes complexes qui caractérisent ces systèmes.

3.4.2 Pourquoi Netlogo ?

La plateforme NETLOGO est caractérisée par :

- **Interface utilisateur intuitive :**
 - NetLogo fournit une interface graphique conviviale qui permet aux utilisateurs de concevoir, d'exécuter et d'analyser des modèles de manière interactive. Les utilisateurs peuvent manipuler des curseurs, des commutateurs et des graphiques pour ajuster les paramètres de simulation en temps réel.
- **Langage de programmation intégré :**
 - Le langage de programmation NetLogo est spécifique à la modélisation basée sur agents. Il est facile à apprendre même pour les débutants, tout en étant suffisamment puissant

pour les utilisateurs avancés. Les commandes et la syntaxe de NetLogo sont conçues pour faciliter la création de modèles complexes.

➤ **Bibliothèque de modèles :**

➤ NetLogo comprend une vaste bibliothèque de modèles prédéfinis couvrant une variété de disciplines. Ces modèles peuvent être utilisés comme point de départ pour les nouveaux utilisateurs ou comme exemples de bonnes pratiques de modélisation.

➤ **Support multi-agent :**

Ces agents sont appelés « tortues » et peuvent interagir entre eux et avec leur environnement. Les interactions entre agents et règles comportementales peuvent être programmées pour simuler des systèmes complexes.

➤ **Évolutivité :**

NetLogo permet d'étendre les fonctionnalités grâce à des extensions écrites en Java. Cela permet aux utilisateurs avancés de créer et d'intégrer des fonctionnalités supplémentaires dans leurs modèles.

➤ **Visualisation et analyse :**

NetLogo fournit de puissants outils de visualisation pour observer les dynamiques de simulation en temps réel. Les graphiques, histogrammes et autres outils de visualisation vous permettent d'analyser vos résultats en détail.

4. Conception avec UML :

4.1 Définition :

UML, ou Langage de Modélisation Unifié (Unified Modeling Language), est un langage de modélisation standardisé employé dans le domaine de l'ingénierie logicielle. Il permet de spécifier, visualiser, construire et documenter les artefacts d'un système logiciel. Grâce à UML, on peut représenter graphiquement la structure, le comportement et les interactions des systèmes, ce qui facilite la compréhension et la communication entre les parties prenantes du projet.

4.2 Utilité d'UML :

UML permet de visualiser, spécifier, construire et documenter les artefacts d'un système logiciel de manière standardisée. En utilisant des diagrammes variés tels que les diagrammes de classes, de cas d'utilisation, d'activités et de séquence, UML facilite la communication entre les membres de l'équipe de développement, les parties prenantes et les clients. Cette visualisation aide à identifier et résoudre les problèmes de conception tôt dans le cycle de vie

du projet, réduisant ainsi les coûts et les délais. De plus, UML favorise une meilleure compréhension des exigences et de la structure du système, ce qui conduit à une architecture logicielle plus robuste et évolutive. En résumé, l'UML est un puissant moyen d'améliorer la qualité et l'efficacité du processus de développement logiciel.

4.3 Démarche de modélisation :

Notre démarche de modélisation comprend deux étapes essentielles : l'analyse et la conception. Dans la phase d'analyse, nous identifions les acteurs du système et examinons leurs interactions. Cette étape nous permet de comprendre les besoins et les exigences du système. Ensuite, lors de la phase de conception, nous créons des descriptions détaillées basées sur les résultats de l'analyse. Nous utilisons dans ce cas une représentation graphique. Cette approche nous permet de capturer les acteurs, les interactions et de fournir une vision claire des fonctionnalités à développer. La représentation graphique illustre les différentes étapes de la démarche de modélisation, avec des flèches indiquant la progression linéaire du processus. Chaque étape est numérotée et décrit brièvement la tâche à accomplir. Cette visualisation permet de mieux comprendre la séquence des étapes et l'enchaînement logique de la démarche de modélisation.

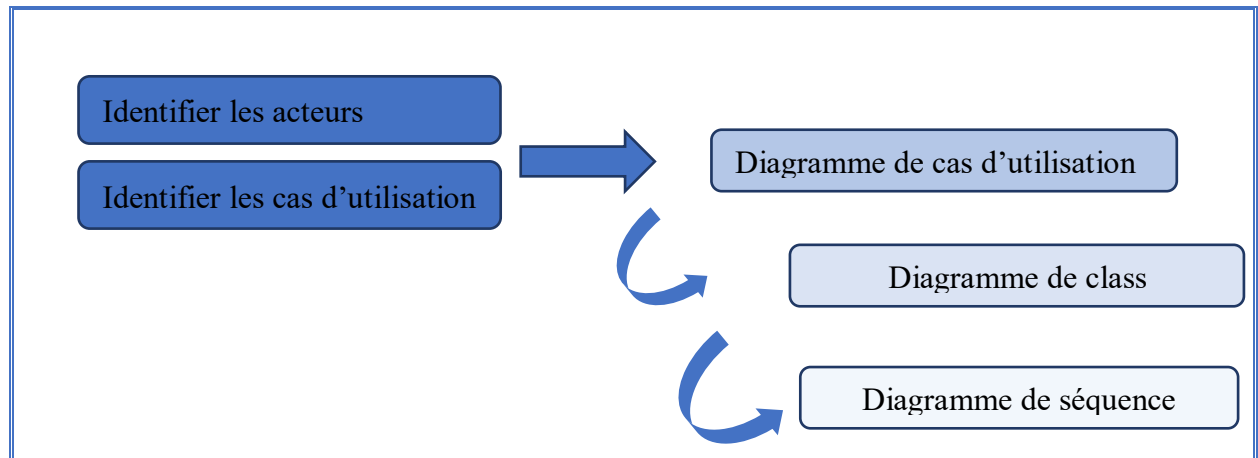


Figure 3.3 Démarche de modélisation de l'application

4.3.1 Diagramme de cas d'utilisation:



Figure 3.4 Diagramme de cas d'utilisation

Le diagramme fourni est un diagramme de cas d'utilisation qui représente les interactions entre un utilisateur (ou "Utilisateur") et un "Système de simulation".

➤ Acteurs

Utilisateur : Représente une personne qui interagit avec le système de simulation. Un utilisateur peut faire les fonctions suivantes :

1. Définir un chemin
2. Définir les points de départ et d'arrivée dans un labyrinthe
3. Créer un agent voiture
4. Sélectionner un algorithme pour trouver le plus court chemin
5. Lancer la simulation une fois que tout est configuré.

➤ **Relations entre les cas d'utilisation de l'utilisateur**

1. Sélectionner l'algorithme utilise <<extend>> vers utiliser l'algorithme : Cela signifie que le cas d'utilisation "Sélectionner l'algorithme" peut être étendu par un autre cas d'utilisation nommé "utiliser l'algorithme".

2. Lancer la simulation utilise <<include>> vers Sélectionner l'algorithme : Cela signifie que "Lancer la simulation" inclut systématiquement "Sélectionner l'algorithme". En d'autres termes, pour lancer la simulation, il faut d'abord sélectionner un algorithme.

➤ **Cas d'utilisation du système**

1. Sélectionner l'algorithme : Le système aide à sélectionner un algorithme pour la simulation.

2. Exécuter la simulation : Le système exécute la simulation basée sur les paramètres fournis.

3. Afficher les résultats : Le système affiche les résultats de la simulation.

4. Tracer une courbe de temps pour la simulation : Le système peut tracer une courbe de temps basée sur les résultats de la simulation.

➤ **Relations entre les cas d'utilisation du système**

1. Exécuter la simulation utilise <<include>> vers Afficher les résultats : Cela signifie que l'exécution de la simulation inclut systématiquement l'affichage des résultats.

2. Afficher les résultats utilise <<extend>> vers tracer une courbe de temps pour la simulation : Cela signifie que l'affichage des résultats peut être étendu pour inclure le traçage d'une courbe de temps.

4.3.2 Diagramme de classes :

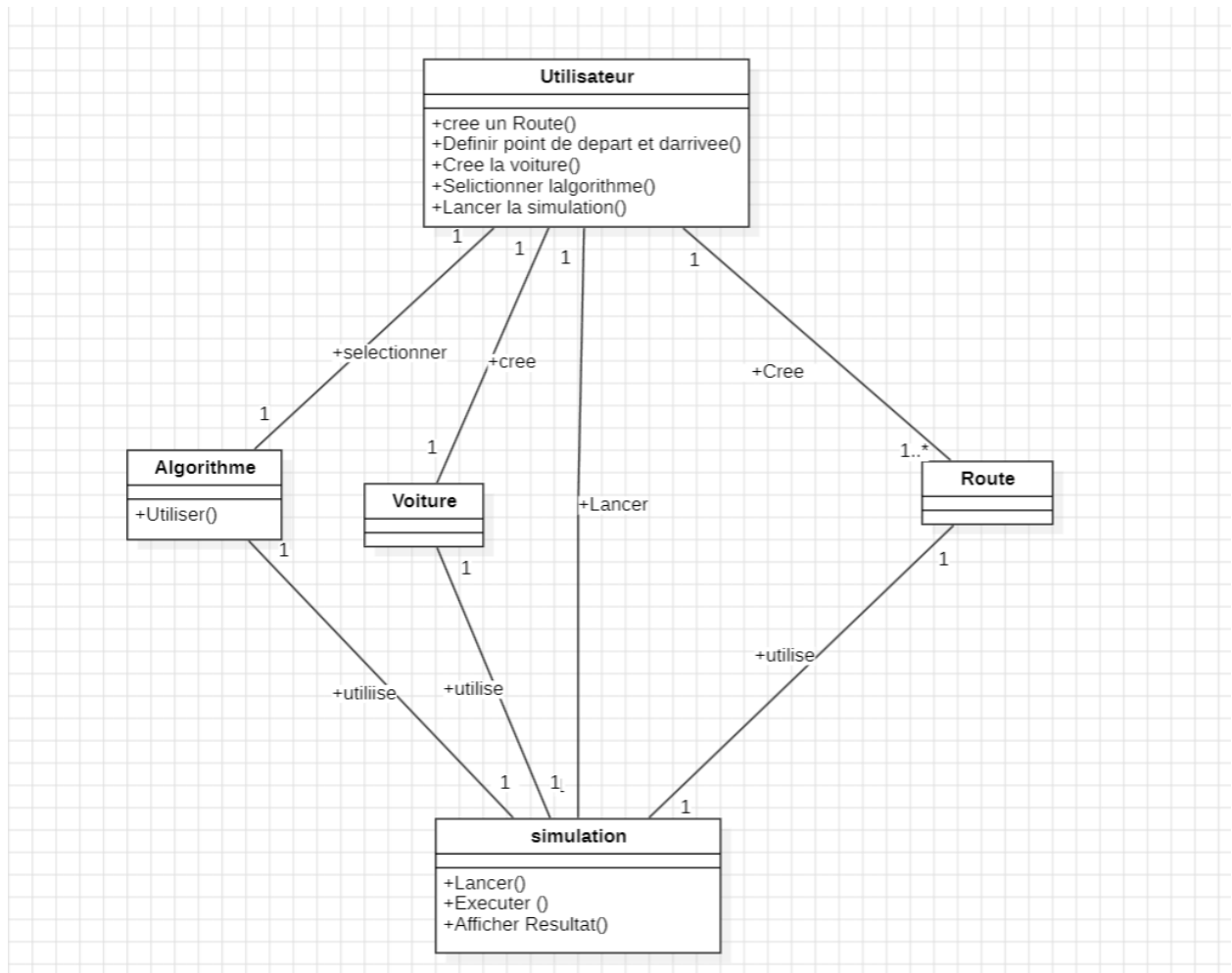


Figure 3.5 Diagramme de classes

Ce diagramme montre clairement comment les différentes composantes interagissent entre elles pour réaliser une simulation, en utilisant des algorithmes spécifiés par l'utilisateur.

4.3.3 Diagramme de séquence :

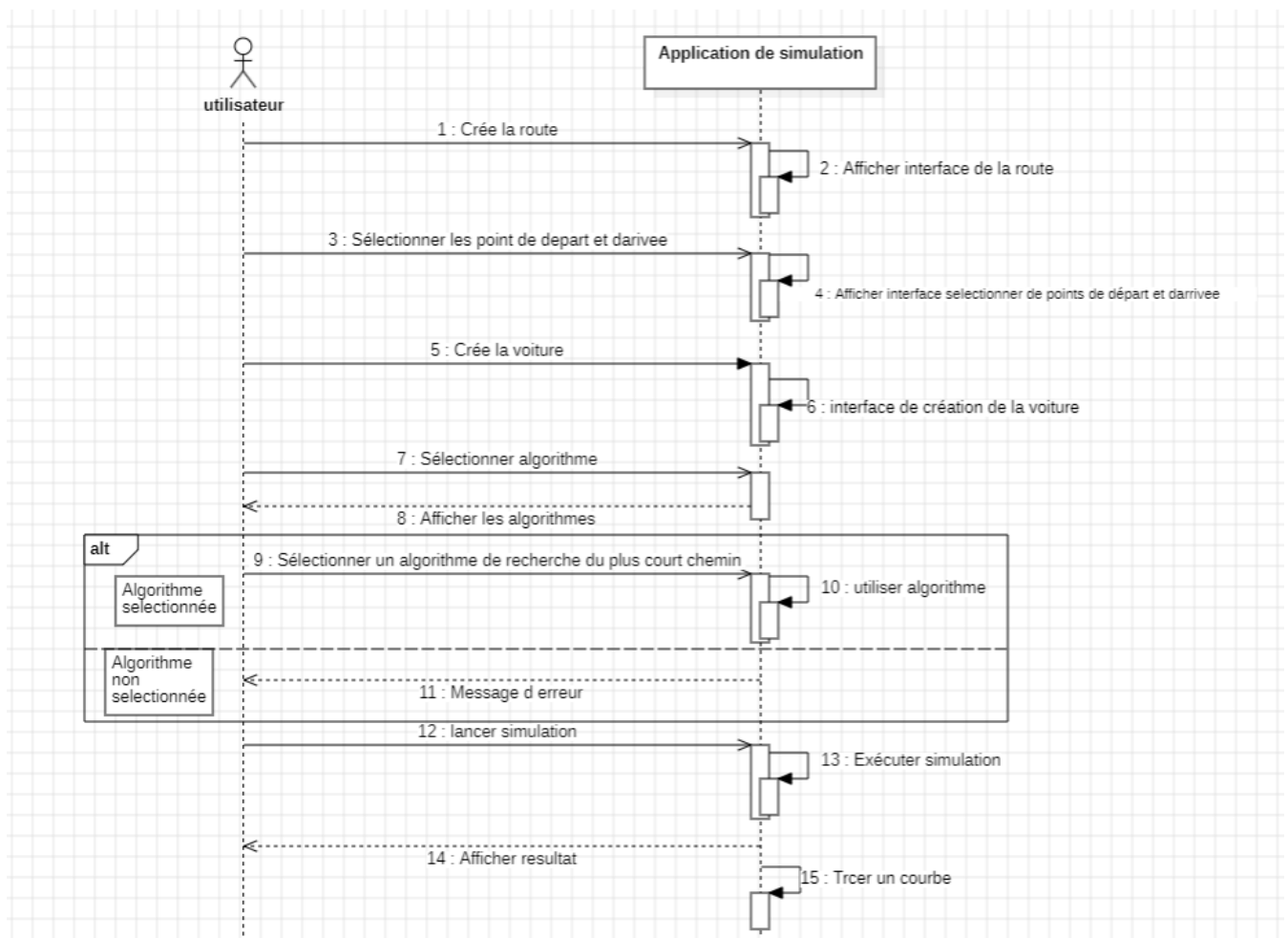


Figure 3.6 Diagramme de séquence

Ce diagramme montre comment l'utilisateur interagit avec l'application de simulation pour créer une route, définir des points de départ et d'arrivée, créer une voiture, sélectionner un algorithme, lancer la simulation et afficher les résultats. Le diagramme inclut un bloc alternatif pour gérer les cas où un algorithme n'est pas sélectionné, ce qui déclenche un message d'erreur.

5. Présentation du système de simulation :

Notre système simule un algorithme de recherche du plus court chemin dans un environnement représenté par un réseau routier. Plus précisément, il montre comment une voiture autonome (simulée par un agent), peut se déplacer d'un point de départ vers un point d'arrivée, en appliquant l'algorithme A* (ou de l'algorithme de recherche en profondeur). Les utilisateurs peuvent visualiser et comparer le comportement de ces algorithmes lors de la recherche du chemin le plus court dans un réseau routier.

5.1 Etapes de simulation

1. **Configuration** : initialisez l'espace de recherche, en créant une grille de correctifs, et définissant les variables initiales et la couleur de tous les correctifs (en vert).

2. **DrawStartAndTarget** : Permet à l'utilisateur de marquer le point de départ (rouge) et le point d'arrivée.

3. **Generate roads** : Construisez un réseau routier en ajoutant aléatoirement des murs (points blancs) dans la grille.

4. **CreateCar** : placez l'agent voiture au point de départ (point rouge) pour lancer la recherche de chemin.

5. **AlgorithmStart** : démarre l'algorithme de recherche du plus court chemin sélectionné (A* ou recherche en profondeur d'abord).

6. **AStarAlgorithm** : implémente l'algorithme A*, dans lequel un agent voiture déplace vers un point cible à l'aide d'une heuristique pour hiérarchiser les correctifs les plus proches de la cible.

6. **DepthFirstSearchAlgorithm** : implémente l'algorithme de recherche en profondeur d'abord, dans lequel l'agent voiture explore d'abord le chemin en profondeur jusqu'à ce qu'elle atteigne la destination.

6. Démarche d'exécution de la simulation :

1. Cliquez sur "Générer la route " pour générer le réseau routier et initialiser l'environnement.

2. Utilisez Marquer le début et la destination pour marquer les points de début et de fin.

3. Cliquez sur "Créer une voiture" pour placer l'agent voiture au point de départ.

4. Sélectionnez le type d'algorithme dans le menu déroulant.

5. Cliquez sur « Démarrer l'algorithme » pour démarrer l'algorithme sélectionné.

➤ **Les choses à considérer :**

1- Différences dans les stratégies de recherche de chemin entre les algorithmes de recherche A* et les algorithmes de recherche en profondeur.

2- Comment la génération de labyrinthes affecte le processus de recherche de chemin.

➤ **Quelque chose qui vaut la peine d'être essayé :**

1. Utilisez le curseur pour essayer différentes tailles de réseau routier.
2. Essayez différents points de départ et d'arrivée.
3. Comparez les performances des algorithmes de recherche A* et en profondeur pour trouver le chemin le plus court.

7. Description des Algorithmes de recherche du plus court chemin:

Il est courant, lorsque l'on cherche à se rendre d'un point à un autre dans un réseau (routier, par exemple), de chercher le plus court chemin, c'est-à-dire celui dont la distance est la plus petite. Si le nombre de trajets possibles entre le point de départ et le point d'arrivée est faible, il suffira de calculer les longueurs de chacun des trajets - en additionnant la longueur des liens qui le composent - et de comparer directement les longueurs obtenues. Mais une telle solution exhaustive devient rapidement impraticable si le nombre de trajets possibles est grand.

Heureusement, il existe des algorithmes qui évitent d'avoir à calculer tous les trajets possibles.

Pour cela, ils mettent en œuvre diverses stratégies par exemple :

7.1 Algorithme A*

7.1.1 Principe de l'algorithme A* :

Au premier abord, on pourrait se dire que pour trouver un chemin d'un point à un autre il faut commencer par se diriger vers la destination. Et bien... c'est justement cette idée qu'utilise l'algorithme A*. L'idée est très simple : à chaque itération (oui, c'est un algorithme itératif), on va tenter de se rapprocher de la destination, on va donc privilégier les possibilités directement plus proches de la destination, en mettant de côté toutes les autres.

Toutes les possibilités ne permettant pas de se rapprocher de la destination sont mises de côté, mais pas supprimées. Elles sont simplement mises dans une liste de possibilités à explorer si jamais la solution explorée actuellement s'avère mauvaise. En effet, on ne peut pas savoir à l'avance si un chemin va aboutir ou sera le plus court. Il suffit que ce chemin amène à une impasse pour que cette solution devienne inexploitable, L'algorithme va donc d'abord se diriger vers les chemins les plus directs. Et si ces chemins n'aboutissent pas ou bien s'avèrent mauvais par la suite, il examinera les solutions mises de côté. C'est ce retour en arrière pour

examiner les solutions mises de côté qui nous garantit que l'algorithme nous trouvera toujours une solution (si tant est qu'elle existe, bien sûr) ,

On peut donc lui donner un terrain avec autant d'obstacles qu'on veut, aussi tordus soient-ils, s'il y a une solution, A* la trouvera. [1]

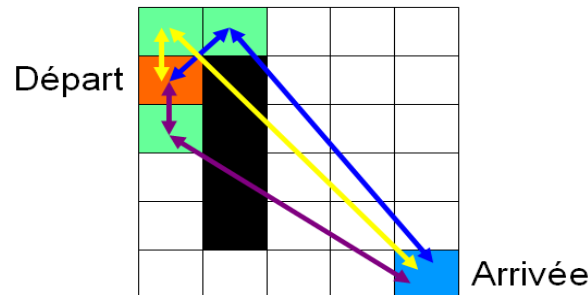


Figure 3.8 l'algorithme A*

Voici le code de cet algorithme sur netlogo:

```

to AStarAlgorithm
  while [status != "Solved"] [ ; Continue until the goal is reached
    tick
    ask Cars [
      pen-down ; make sure the pen is down so we will see where the car moved
      set pen-size 5

      set currentColor [pcolor] of patch-here ; Get the color of the current patch
      ifelse (currentColor = black) [ ; Check if the current patch is black (goal state)
        set status "Solved" ; Set status to "Solved" if goal is reached
      ] [
        ; Explore valid neighboring patches
        set validNeighborsDistances ([dis] of neighbors4 with [(pcolor = white or pcolor = black) and visited = false]) ; Get distances to valid neighboring patches
        ifelse (not empty? validNeighborsDistances) [ ; If there are valid neighbors
          set minDistance min validNeighborsDistances ; Find the minimum distance among valid neighbors
          ask patch-here [set backtrackStack lput dis backtrackStack] ; Add the distance to backtrackStack
          face one-of neighbors4 with [(pcolor = white or pcolor = black) and dis = minDistance and visited = false] ; Rotate to face a neighbor with the minimum distance
          forward 1 ; Move forward by 1 patch
          ask patch-here [set visited true] ; Mark the current patch as visited
        ] [
          ; Backtrack if no valid neighbors are found
          set lastDistance last backtrackStack ; Get the last distance from backtrackStack
          ask patch-here [set visited2 true] ; Mark the current patch as visited2
          face one-of neighbors4 with [dis = lastDistance and (pcolor = white or pcolor = red) and visited2 = false] ; Rotate to face a neighbor with the last distance
          forward 1 ; Move forward by 1 patch
          set backtrackStack but-last backtrackStack ; Remove the last distance from backtrackStack
        ]
      ]
    ]
  ]
  set status "On Progress" ; reset status for next algorithm running
end

```

Figure 3.9 code de l'algorithme A* sur netlogo

7.2 Algorithme DFS (Depth-First Search) :

Bien que l'algorithme DFS (Depth-First Search) ne soit pas idéal pour trouver le plus court chemin dans un graphe, il peut être utilisé dans certains cas, notamment lorsque les chemins sont non pondérés. Dans ce contexte, l'algorithme DFS explore tous les chemins possibles entre le nœud de départ et le nœud d'arrivée, et ensuite, parmi tous les chemins trouvés, on sélectionne celui qui est le plus court.

Voici une description détaillée de l'utilisation de DFS pour trouver le plus court chemin

- On utilise une fonction récursive DFS qui explore chaque nœud, en gardant une trace du chemin parcouru.
- Si le nœud courant est le nœud cible, on ajoute le chemin courant à une liste de tous les chemins trouvés.
- Retour en arrière :
- Lorsqu'on atteint une impasse (un nœud sans voisins non visités), on revient en arrière et on explore d'autres chemins possibles.
- Sélection du plus court chemin :
- Une fois que tous les chemins possibles entre le nœud de départ et le nœud d'arrivée ont été explorés, on parcourt la liste des chemins trouvés pour sélectionner celui qui a la longueur la plus courte.

Voici le code de cet algorithme sur netlogo :

```
to DepthFirstSearchAlgorithm
  ; Initialize stack
  let stack []
  ; Start DFS from red
  let start-patch one-of patches with [pcolor = red]; Find a patch with red color
  if start-patch != nobody [ ; Check if a red patch is found
    ask start-patch [
      set visited true ; Mark the starting patch as visited
      set stack (list self) ; Add the starting patch to the stack
    ]
    while [length stack > 0 and status != "Solved"] [ ; Continue DFS until stack is empty or goal is reached
      tick
      let current-patch last stack ; Get the last patch added to the stack
      set stack but-last stack ; Remove the last patch from the stack
      ask Cars [
        face current-patch ; Rotate the car to face the current patch
        move-to current-patch ; Move the car to the current patch

        set currentColor [pcolor] of patch-here ; Get the color of the current patch
        if (currentColor = black) [ ; Check if the current patch is black (goal state)
          set status "Solved" ; Set status to "Solved" if goal is reached
        ]
      ]
    ]
  ; Explore neighbors of the current node
  ask current-patch [
    let unvisited-neighbors neighbors4 with [(pcolor = white or pcolor = black) and visited = false] ; Find unvisited neighboring patches
    let neighborsList [self] of unvisited-neighbors ; Get a list of unvisited neighboring patches
    foreach neighborsList [
      neighbor ->
      ask neighbor [
        set visited true ; Mark the neighbor as visited

        if pcolor = white [ ; Check if the neighbor patch is white
          set pcolor yellow ; Set the color of the neighbor patch to yellow to mark it as explored
        ]
      ]
      set stack lput self stack ; Add the neighbor to the stack for further exploration
    ]
  ]
]
```

Figure 3.10 le code de l'algorithme DFS sur Netlogo

8. Les Principales interfaces de système simulation

Les étapes du processus de simulation sont illustrées à travers différents interfaces de notre application

8.1 Créer un chemin

L'utilisateur sélectionne le chemin qui sera utilisé dans la simulation.

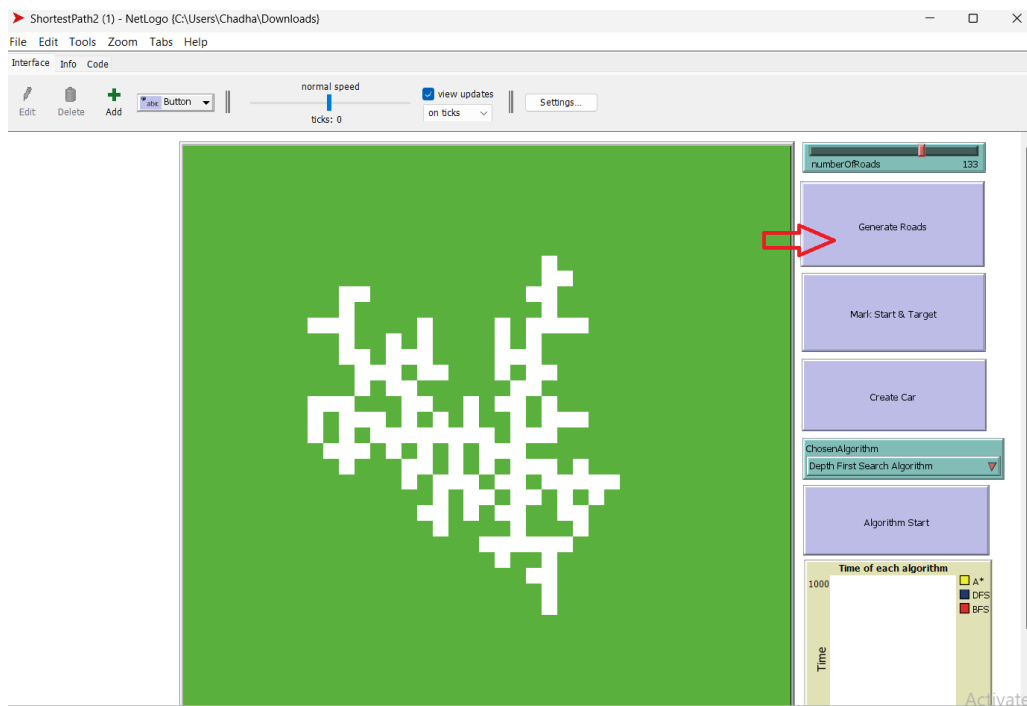


Figure3.11 Créer un chemin

8.2. Déterminez les points de début et de fin

L'utilisateur définit le point de départ et la destination de l'itinéraire.

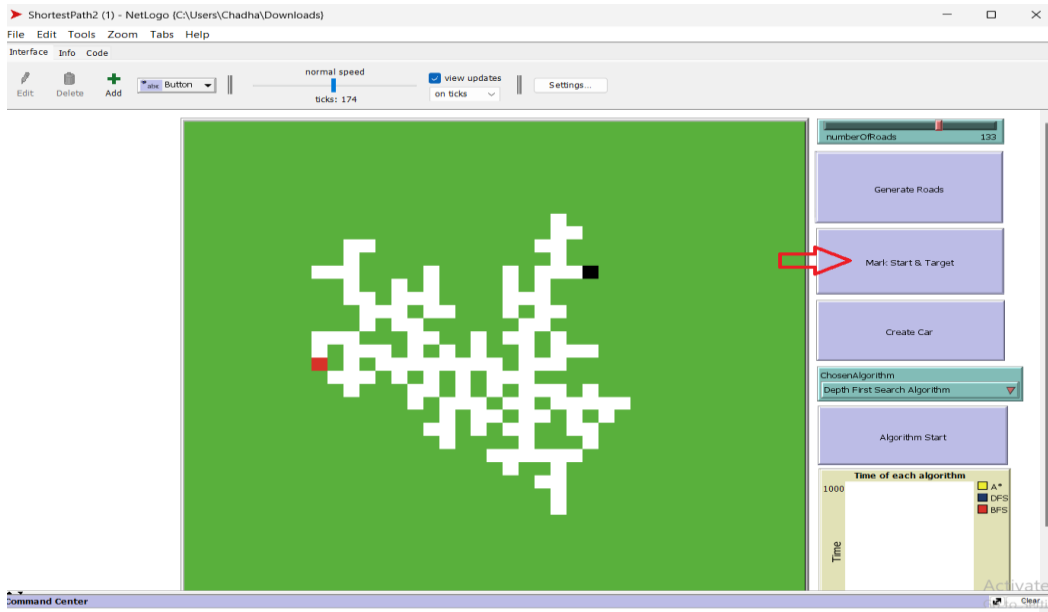


Figure 3.12 Déterminez les points de début et de fin

8.3. Créez un agent voiture

L'utilisateur crée ou sélectionne l'agent véhicule qui sera utilisé dans la simulation.

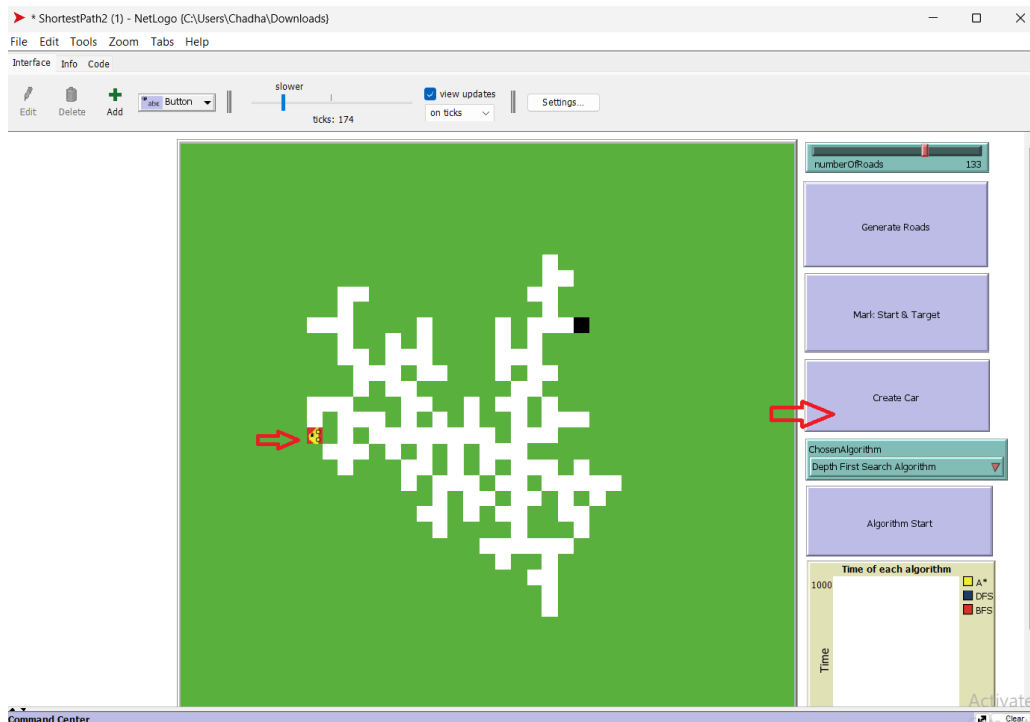


Figure 3.13 Créez un agent voiture

```

breed [Cars Car]

; Set up the initial state of the world
to setup
  clear-all
  reset-ticks
  ask patches [
    set visited false
    set visited2 false
    set dis 0 ; or any initial value based on your heuristic or distance c
  ]
  set validNeighborsDistances []
  set backtrackStack []
  set status ""

  ask patches [ set pcolor green ] ; Set all patches' color to green
end

to CreateCar
  let red-patch one-of patches with [pcolor = red]

  ifelse red-patch != nobody [
    let red-patch-x [pxcor] of red-patch
    let red-patch-y [pycor] of red-patch
    let carExist any? patches with [any? Cars] ; check if there is any car

    ifelse carExist = false [
      create-Cars 1 [
        set xcor red-patch-x
        set ycor red-patch-y
        set heading 0
        set shape "car"
        set color yellow
        pen-down
      ]
    ]
    [
      user-message "Only one car can be existed in the road"
    ]
  ]
  [
    user-message "Please mark start and target first"
  ]
]
end

```

Figure 3.14 Programmation de création voiture

8.4. Sélectionnez l'algorithme de recherche du plus court chemin

L'utilisateur choisit l'algorithme de recherche , qui sera utilisé lors de la simulation.

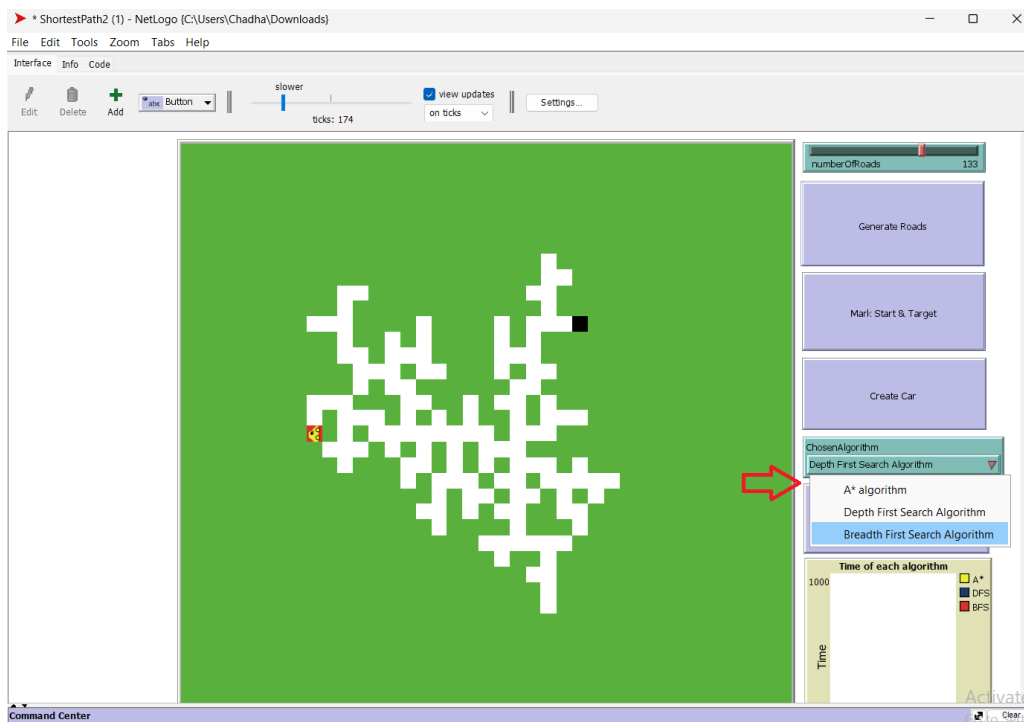


Figure 3.15 Sélection l'algorithme

8.5. Exécutez la simulation

L'utilisateur démarre le processus de simulation. Cette étape comprend les sous-étapes suivantes :

- Réalisation de la simulation
- Le système effectue la simulation en utilisant l'algorithme spécifié et les paramètres spécifiés.
- Montrer les résultats
- Lors de l'exécution, le système traite et calcule les données nécessaires.

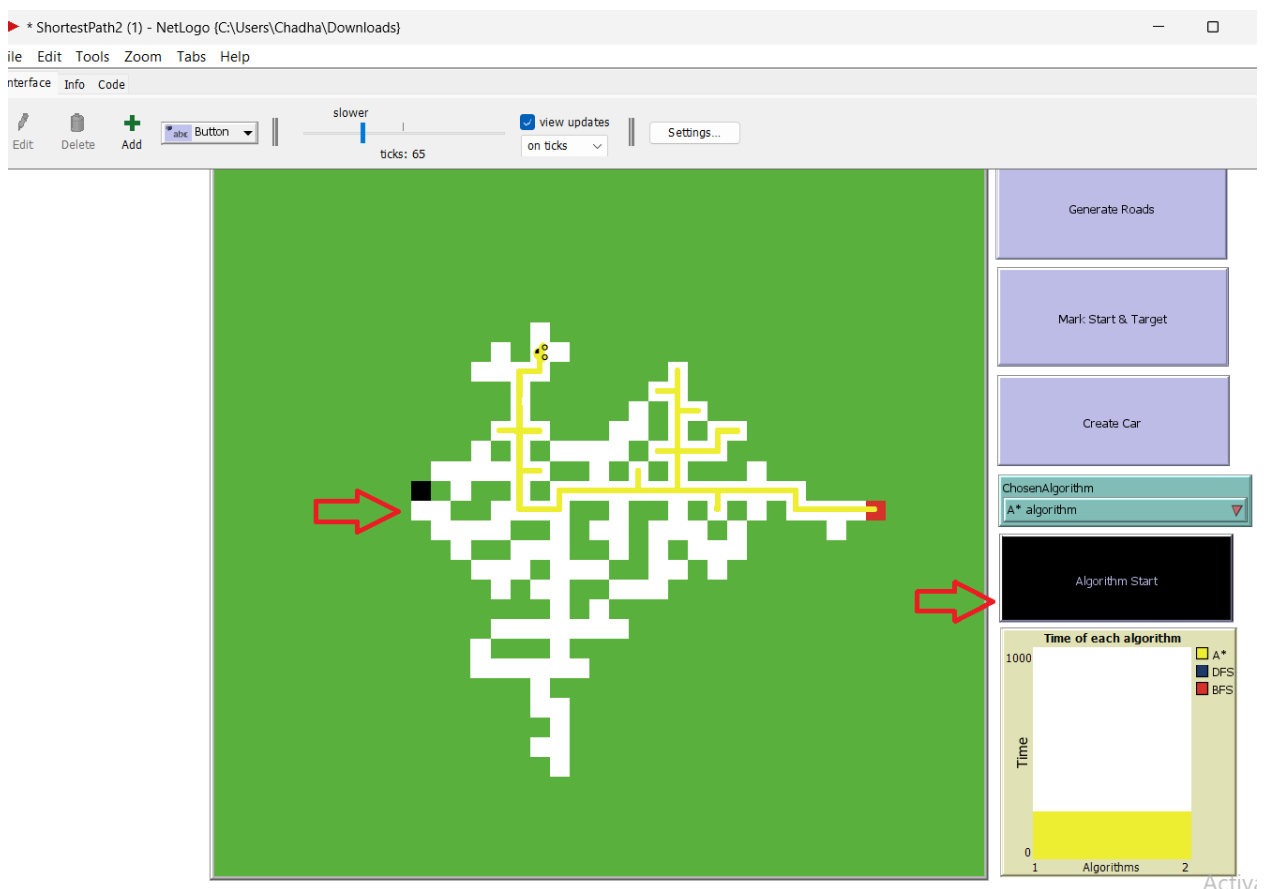


Figure 3.16 exécution de simulation

Le système affiche les résultats de la simulation.- Si nécessaire, le système peut également tracer les résultats sous forme graphique.

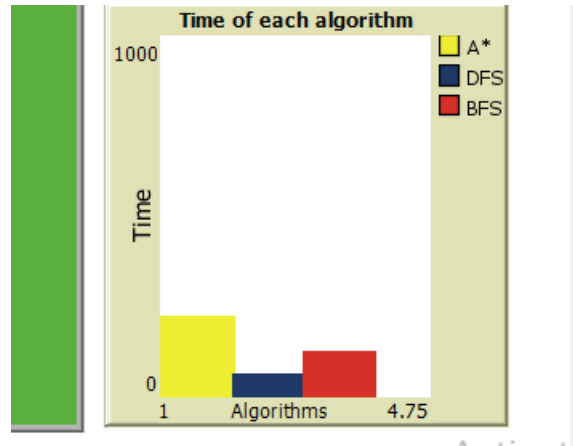


Figure 3.17 courbe obtenu

9. Analyse des Résultats obtenus :

En utilisant NetLogo, il est possible de simuler efficacement des systèmes complexes et des phénomènes émergents. La simulation de divers algorithmes de recherche de chemin dans un environnement de réseau routier permet d'explorer et de comprendre ces algorithmes de manière interactive.

Dans la résolution de problèmes de recherche de chemin, l'algorithme A* est fréquemment employé car il est performant et trouve généralement le chemin le plus court entre deux points. Par contre, la mise en œuvre de l'algorithme de DFS peut être plus facile, mais cela ne garantit pas toujours la recherche du chemin le plus court.

En comparant visuellement les performances de ces algorithmes dans la simulation, les utilisateurs peuvent mieux comprendre leurs différences en termes d'efficacité et de qualité de la solution trouvée. Cela pourrait être particulièrement utile pour les étudiants ou les passionnés d'informatique qui cherchent à approfondir leur compréhension des algorithmes de recherche de chemin.

En termes de performance, l'algorithme A* a tendance à être plus rapide et plus efficace, car il utilise une heuristique pour guider sa recherche, réduisant ainsi le nombre de nœuds explorés. En comparaison, l'algorithme de DFS explore les nœuds de manière exhaustive, ce qui peut entraîner une recherche plus longue et moins efficace, particulièrement dans de grands réseaux routiers complexes. De plus, A* trouve souvent le chemin le plus court, ce qui est un avantage significatif par rapport à DFS, qui peut parfois prendre des chemins plus longs ou même se perdre dans des recoins du réseau routier avant de trouver la sortie.

Pour les grands réseaux routiers, l'algorithme A* est généralement le plus rapide, car il exploite ses heuristiques pour minimiser le nombre de nœuds explorés. À l'inverse, pour les petits réseaux routiers ou ceux avec une structure simple, DFS peut parfois être plus rapide grâce à sa simplicité et à son approche exhaustive qui ne nécessite pas de calculs heuristiques complexes. Ces observations permettent de comprendre pourquoi A* est souvent préféré dans les applications nécessitant une recherche de chemin optimale, alors que DFS peut être utilisé dans des situations où la simplicité et la rapidité de mise en œuvre sont plus importantes que l'optimisation du chemin.

10.Conclusion :

La visualisation des algorithmes de recherche du plus court chemin dans un environnement de réseau routier permet une compréhension approfondie de ces algorithmes de recherche. Les utilisateurs ont la possibilité de tester diverses configurations et d'observer les performances comparatives des algorithmes A* et de recherche en profondeur, ce qui renforce leur compréhension des stratégies de navigation et des conséquences de la structure du réseau routier sur l'efficacité de la recherche de chemin. La visualisation des algorithmes de recherche du plus court chemin dans un environnement de labyrinthe permet une compréhension approfondie de ces algorithmes de recherche. Il est possible pour les utilisateurs de tester diverses configurations et d'observer les performances comparatives des algorithmes A* et de recherche en profondeur, ce qui renforce leur compréhension des stratégies de navigation .

Conclusion Générale :

L'objectif de cette étude est de concevoir et de tester des algorithmes qui permettraient aux véhicules autonomes de naviguer efficacement sur un réseau de routes en déterminant le chemin le plus court entre deux endroits. Les résultats obtenus démontrent la capacité des véhicules à utiliser des algorithmes d'optimisation des voies, tels que A* et Dfs, pour optimiser leurs routes en fonction de divers paramètres, y compris la distance et le temps. Les simulations effectuées sur la plate-forme NetLogo ont permis de voir et d'étudier le comportement des véhicules autonomes dans une variété de scénarios, en simulant des environnements urbains complexes. Les avantages de l'utilisation de cette approche NetLogo ont permis de modéliser des réseaux aléatoires de différentes tailles et complexité et La capacité de NetLogo à offrir des visualisations en temps réel a fait, Analyser les résultats des algorithmes et ajuster les paramètres afin d'améliorer leur efficacité. La plateforme s'est avérée extrêmement flexible pour évaluer et comparer divers algorithmes de recherche de chemin.

Finalement, cette recherche confirme que les véhicules autonomes peuvent tirer un avantage considérable des simulations sur NetLogo afin d'améliorer la planification de leurs déplacements. Des applications concrètes sont possibles grâce aux résultats obtenus dans la gestion du trafic urbain et l'amélioration des réseaux de transport. Il serait intéressant d'inclure des variables supplémentaires dans les recherches à venir, comme les conditions météorologiques et les accidents de la route, et la collaboration avec d'autres types de véhicules afin d'améliorer la réalisme et la résistance des simulations.

Bibliography

- [1] "N Kabachi. Intelligence artificielle distribuee et systemes multi-agents. support de cours : extrait de mémoire de thèse de doctorat, partie état de l'art, page 33,," 2008..
- [2] " Michael Wooldridge. Intelligent agents. Multiagent systems, 6,," 1999.
- [3] " Imane Boussebough. Les systèmes multi-agents dynamiquement adaptables. Doctoral dissertation, Thèse de doctorat,," 2011.
- [4] "Kamel Zidi. Système interactif d'aide au déplacement multimodal (SIADM). PhD thesis, Ecole Centrale de Lille ; Université des Sciences et Technologie de Lille-Lille I,," 2006.
- [5] "Marjorie Le Bars. Un Simulateur Multi-Agent pour l'Aide à la Décision d'un Collectif : Application à la Gestion d'une ressource Limitée Agro-environnementale. PhD thesis,," Université Paris Dauphine-Paris IX, 2003.
- [6] "M.Guériau. " Systèmes multi-agents, auto-organisation et contrôle par apprentissage constructiviste pour la modélisation et la régulation dans les systèmes coopératifs de trafic ". PhD thesis in Computer Science,," Université de Lyon, 2016.
- [7] "slideplayer," [Online]. Available: <https://slideplayer.fr/slide/12450268>.
- [8] "L'intelligence ambiante et les systèmes," AHMED MALEK Nada, UBM-Annaba, 2014.
- [9] "Développement d'un device IOT pour la supervision d'un système de," MAHDAD NARIMENE et LAIREDJ DJAMILA, Aboubakr Belkaïd – Tlemcen –, 2020.
- [10] "Le système de gestion routier COMPAS a réduit les émissions de 3 100 tonnes d'oxyde de carbone par année.," Le système de gestion routier COMPAS a réduit les émissions de 3 100 tonnes d'oxyde de carbone par année..
- [11] "utec.univ-saida.dz," [Online]. Available: https://butec.univ-saida.dz/admin/opac_css/index.php?lvl=notice_display&id=380264.
- [12] "LABO UNIV," [Online]. Available: <https://labos.univ-batna2.dz/lap>.
- [13] "SCRIBD," [Online]. Available: <https://www.scribd.com/document/698671661/Ms-tel-Mahdad-Lairedj>.
- [14] "Déploiement d'un système de suivi par vidéo et analyse," *THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE*, , le 26 Septembre 2022.
- [15] "F.MICHEL, « Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents ». Système multi-agents [cs.MA]. Montpellier II,," 2004.
- [16] Consulté le 30/05/2023]. [Online]. Available: [http://www-public.int-evry.fr/~gauthier/\(VINCENT, GAUTHIER : support de cours simulation\) .](http://www-public.int-evry.fr/~gauthier/(VINCENT, GAUTHIER : support de cours simulation) .)
- [17] "♣ M. NIAZI, A. HUSSAIN, « Agent-based tools for modeling and simulation of self-organization in peer-to-peer, ad hoc, and other complex networks ». .," IEEE Communications Magazine, 47(3), 166–173, 2009.
- [18] "M. BABIS, P. MAGULA, P, « NetLogo - An alternative way of simulating mobile ad hoc networks », Wireless and Mobile Networking Conference,," 2012.
- [19] Interstices. [Online]. Available: <https://interstices.info/le-plus-court-chemin/>.
- [20] <https://interstices.info/le-plus-court-chemin/>.
- [21] [Online]. Available: <https://interstices.info/le-plus-court-chemin/>.

- [22] "Houssein Ben-Ameur. Enchères multi-objets pour la négociation automatique et le commerce électronique. 2001."
- [23] H. Ben-Ameur., Artist, *Enchères multi-objets pour la négociation automatique et le commerce électronique. 2001.* [Art].
- [24] "SCRIBD," [Online]. Available: <https://www.scribd.com/document/698671661/Ms-tel-Mahdad-Lairedj>.

Résumé :

Ce travail présente la simulation de véhicules autonomes avec l'objectif de déterminer le plus court chemin, en utilisant la plateforme NetLogo. L'objectif principal est de développer et tester des algorithmes d'optimisation, tels que A* et DFS, pour la navigation autonome dans un ensemble de routes. La simulation permet d'analyser le comportement des agents (véhicules), et d'optimiser les trajets en termes de distance et de temps. Les résultats obtenus visent à améliorer l'efficacité des véhicules autonomes en conditions réelles. Cette recherche contribue à l'avancement des technologies de transport intelligent et de mobilité durable.

Mots-clés: agent , netlogo,A*,DFS, vehicule autonome

Abstract:

This work presents the simulation of autonomous vehicles with the goal of determining the shortest path, using the NetLogo platform. The primary objective is to develop and test optimization algorithms, such as A* and DFS, for autonomous navigation in various environments. The simulation allows for the analysis of agent (vehicle) behavior and the optimization of routes in terms of distance and time. The results obtained aim to improve the efficiency of autonomous vehicles in real-world conditions. This research contributes to the advancement of intelligent transportation technologies and sustainable mobility.

Key words: agent , netlogo,A*,DFS, autonomous vehicles

الملخص :

يستكشف هذا

البحث محاكاة المركبات الذاتية بهدف تحديد أقصر مسار باستخدام منصة NetLogo. الهدف الرئيسي هو تطوير واختبار خوارزميات تحسين، مثل A* و DFS، للتنقل الذاتي في بيئات متنوعة. تسمح المحاكاة بتحليل سلوك الوكلاء (المركبات) وتحسين المسارات من حيث المسافة والوقت. تهدف النتائج التي تم الحصول عليها إلى تحسين كفاءة المركبات الذاتية في الظروف الواقعية. تساهم هذه الأبحاث في تقدم تقنيات النقل الذكي والتنقل المستدام.

الكلمات المفتاحية: agent , netlogo, A*,DFS,المركبات الذاتية