

Université Paris-Est Créteil Val-De-Marne  
Institut Universitaire de Technologie Vitry-sur-Seine



## Rapport de Fin d'Etude

Pour l'obtention du

**Diplôme de Master en Ingénierie des Systèmes Complexes**

*Parcours : Systèmes cyber-physiques, technologies de l'Information, de l'intelligence et du contrôle*

réalisé par :

**Souhir Helali**

SUJET :

***Modélisation Multi-Agent Des comportement latéraux  
des véhicules légers***

Supervisé par :

**INSA**  
ROUEN  
Dr. Julien SAUNIER

 **IFSTAR**  
Dr. Sio-Song IENG

Année Universitaire : 2015-2016

# Table des matières

<b>Introduction Générale</b>	<b>1</b>
<b>1 La simulation de Trafic</b>	<b>3</b>
1.1 Approche macroscopique . . . . .	3
1.2 Approche microscopique . . . . .	4
1.3 Synthèse . . . . .	6
<b>2 Les Systèmes Multi-Agents</b>	<b>9</b>
2.1 Les Agents . . . . .	9
2.1.1 Définition . . . . .	9
2.1.2 L'environnement . . . . .	11
2.2 Les systèmes Multi-Agents . . . . .	12
2.2.1 Définition . . . . .	12
2.2.2 L'approche SMA . . . . .	13
2.2.2.1 Modéliser le monde . . . . .	13
2.2.2.2 Modèle . . . . .	13
2.2.2.3 La modélisation . . . . .	15
2.2.2.4 La validation . . . . .	15
2.2.3 Les SMA et la simulation de trafic . . . . .	15
2.2.3.1 Vision et perception limitée . . . . .	16
2.2.3.2 Processus de Décision . . . . .	17
<b>3 Yet Another Traffic Simulator (YATS)</b>	<b>19</b>
3.1 Architecture de YATS . . . . .	19
3.1.1 Diagramme de Classe . . . . .	19
3.1.2 Initialisation de la Simulation . . . . .	20

3.1.3	Le modèle d'agent . . . . .	20
3.1.4	Les limite de YATS . . . . .	22
3.2	Contribution . . . . .	22
3.2.1	Architecture . . . . .	23
3.2.2	Méthodologie de conception . . . . .	25
3.2.2.1	Modèle des rôles . . . . .	26
3.2.2.2	Modèle d'interaction . . . . .	27
3.2.3	Modèle Multi-Agents . . . . .	28
3.2.3.1	Modèle de rôle . . . . .	28
3.2.4	Protocole d'interaction . . . . .	29
3.2.4.1	Étape de découverte . . . . .	29
3.2.4.2	Étape de génération de voisinage . . . . .	31
3.2.4.3	Étape de libération de voisins . . . . .	31
<b>4</b>	<b>Réalisation</b> . . . . .	<b>32</b>
4.1	Environnement de Développement . . . . .	32
4.2	Outils de développement . . . . .	32
4.2.1	Langage C++ . . . . .	32
4.2.2	Environnements de développement intégré « IDE » : Visual Studio 2015 Community . . . . .	33
4.2.2.1	Bibliothèque SDL . . . . .	33
4.3	Simulation . . . . .	34
4.3.1	Début de la simulation . . . . .	34
	<b>Annexe Description des classes d'origine de YATS</b> . . . . .	<b>41</b>
	<b>Bibliographie</b> . . . . .	<b>43</b>

# Table des figures

1.1	Diagramme fondamentale théorique de trafic . . . . .	4
1.2	Comparaison des approches de modélisation de trafic . . . . .	7
2.1	Schémas d'Agent . . . . .	10
2.2	Relation directe entre le monde réel et le modèle [11] . . . . .	14
2.3	Vision égo-centrée de l'environnement [13] . . . . .	16
2.4	Exemple de règles de gestion des interactions . . . . .	17
2.5	Différentes approches pour la gestion des conflits en zone d'intersection. Une microscopique simple (3.7(a)) consiste à utiliser un ordonnanceur central disposant de suffisamment d'informations pour résoudre les conflits globalement et guider les véhicules dans la zone d'intérêt. Dans l'approche SMA (3.7(b)) chaque véhicule ne dispose que d'une vision limitée de l'environnement (état de la route, véhicules avec lesquels il est en conflit...) et doit élaborer une stratégie sans savoir comment les autres véhicules envisagent de traverser l'intersection. [17] . . . . .	18
3.1	Diagramme de classe de YATS [18] . . . . .	20
3.2	Les fichiers de configuration de YATS : (a) la liste des fichiers de configuration, le chargement des fichiers se fait selon l'ordre déterminé dans le fichier everyfileyouneed ; (b) le contenu du fichier de configuration des routes . . . . .	21
3.3	Prise de décision . . . . .	22
3.4	Exemple d'application du processus de Décision : (a) en effectuant un virage e véhicule calcule les points p1, p2, p3 correspondant aux états de transition possible ; (b) Avant de tourner à gauche, le véhicule calcule sa distance d'arrêt laquelle s'avère trop petite, il continue d'avancer jusqu'à trouver la distance la plus proche possible lui permettant de tourner. . . . .	23



3.5	Modele-Vue . . . . .	24
3.6	exemple de séparation de la vue du module . . . . .	24
3.7	Diagramme de Classe . . . . .	25
3.8	Modèles d'analyse de Gaia . . . . .	26
3.9	Les opérateurs de vivacité . . . . .	27
3.10	Rôle : Vehicule . . . . .	28
3.11	Rôle : BidirectionalRoad . . . . .	29
3.12	Protocole de découverte de Voisinage . . . . .	30
4.1	Début de la simulation . . . . .	35
4.2	Première vague de Hello envoyée . . . . .	35
4.3	Changement de segment . . . . .	36
4.4	Hello du changement de segment de route . . . . .	36
4.5	Voisin visible . . . . .	37
4.6	Voisin dans la liste . . . . .	37
4.7	Véhicule à destination . . . . .	38

# Liste des tableaux

# Introduction Générale

De nos jours, la croissance de la population s'accompagne de la croissance du nombre d'usagers des routes. Ce phénomène risque d'avoir de graves conséquences sur l'environnement. En effet, il a été prouvé que l'une des principale cause de l'effet de Serre est les émissions polluantes provenant des tuyaux d'échappement des véhicules lesquels contribuent à l'appauvrissement de la couche d'Ozone. En plus, l'utilisation excessive des véhicules et la mauvaise gestion nécessite une consommation élevée de carburant et provoque une perte importante du temps, causée par les bouchons rencontrés au long de la journée. N'oublions pas les incidents et les accidents routiers qu'engendrent les conditions de circulation. Ainsi, l'intervention humaine est obligatoire pour protéger l'environnement et garantir le confort et la sécurité de l'usager de la route. Les solutions adoptées habituellement telles que la construction de nouvelles routes et l'extension des réseaux existants ne sont plus efficaces. La poursuite de la croissance de population et des contraintes spatiales et financières en sont la cause. Une telle contrainte exige la réflexion à des solutions plus performantes et plus durables permettant une meilleure exploitation des infrastructures existantes et une meilleure gestion des flux de véhicules. En vue de remédier à de tels problèmes, l'informatique propose certains outils de Simulation du Trafic de plus en plus efficaces pour tester des solutions de gestion de trafic à mettre en place. De plus, ces méthodes permet de prédire l'écoulement de ce dernier, ainsi que les problèmes de circulation routière. Elle introduit des dispositifs de régulation du trafic qui permettent de mieux l'appréhender. Il est, aussi, possible de modéliser et simuler le comportement du conducteur en interaction avec son environnement grâce aux systèmes coopératifs. Cela rend plus aisée, par la suite, la mise en œuvre de solution pour l'optimisation des infrastructures. Dans ce cadre, il y un recours accru aux Systèmes Multi-Agents (SMA). Les SMA font partie des outils de simulation axés sur la création d'entités virtuelles autonomes appelé agents. Ils sont de plus en plus développés. Cela se traduit par la création des différents modèles assez avancés tels que le modèle BDI dont les domaines d'application sont très variés.

Une architecture BDI est conçue en partant du modèle "Croyance-Désir-Intention", en anglais "Belief-Desire-Intention", de la rationalité d'un agent intelligent.

Notre projet se situe dans le cadre de la réalisation d'une Modélisation multi-agent du trafic routier laquelle est présentée par le présent document. C'est un projet qui se déroule dans le cadre du projet (Profil de la Route comme Facteur d'Influence de la Lisibilité de l'infrastructure et son effet sur la trajectoire). Il est réalisé au sein du laboratoire Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux (IFSTTAR), en affiliation avec l'Institut National des Sciences Appliquées de Rouen (INSA-Rouen). Nous allons introduire, en un premier temps, les modèles utilisés dans la simulation de trafic routier. Ensuite, nous allons introduire les Systèmes multi-agents. Une fois, l'introduction des notions importante pour notre travail, nous allons présenter le Simulateur de trafic que nous sommes en train de développer qui a été nommé YATS, acronyme de "Yet Another Traffic Simulator". Avant de conclure avec une conclusion générale, nous allons présenter ce que nous avons réussi à réaliser dans un chapitre de réalisation.

# Chapitre 1

## La simulation de Trafic

### Introduction

Les techniques de modélisation du trafic routier permettent aux gestionnaires des réseaux de transport de mieux exploiter leurs infrastructures et représentent ainsi des outils d'aide à la décision. En effet, les modèles permettent la prédiction de l'état du trafic. En prévenant les congestions et en détectant les incidents et accidents, ils offrent la possibilité de traiter et intervenir dans des délais de temps réduits. Deux grandes approches existent pour modéliser et simuler le trafic routier : macroscopique et microscopique.

### 1.1 Approche macroscopique

L'échelle macroscopique est adaptée à la description des véhicules sur des réseaux de grande taille. Le modèle historique de Lighthill et Whitham [1] et de Richards [2] est fondé sur une approche continue par analogie à la mécanique des fluides. La description d'un flot de véhicules se fait principalement à l'aide de trois variables :

- La vitesse  $v$  (km/h ou m/s)
- La densité (ou concentration)  $\rho$  (véhicules/km)
- Le débit  $Q$  (véhicules/h)

Il est représenté par une équation hyperbolique reliant la vitesse à la densité spatiale des véhicules. La version classique la plus simple est issue d'un principe physique exprimé par l'équation de conservation de la masse suivante :

$$\partial_t \rho + \partial_x (v\rho) = 0, x \in \mathbb{R} \text{ et } t \in \mathbb{R}_+$$

Lorsque la densité est faible, le débit va croître. De ce fait, les interactions entre les véhicules deviennent minimales puisque ceux-ci ne sont pas contraints et peuvent donc rouler à leur vitesse de croisière. Au fur et à mesure que la densité augmente, les interactions vont devenir de plus en plus fortes et la vitesse des véhicules va diminuer au delà d'un certain point (appelé point critique). D'où l'apparition de la congestion. Ces caractéristiques sont exprimées dans la relation  $Q(\rho) = v(\rho)\rho$  est appelée diagramme fondamental.

Dans ce modèle, on distingue deux types de trafic 1.1 :

- **Trafic fluide** : Avant d'atteindre une certaine concentration  $\rho_{crit}$  le débit croît linéairement. Suite à cet accroissement, le nombre de véhicules augmente, en adoptant la vitesse  $V$  (pente du diagramme).
- **Trafic congestionné** : Dépassant  $\rho_{critique}$ , le débit décroît, accompagné de la diminution de vitesse et de l'augmentation du nombre de véhicules. Le flux congestionné se propage le long du trafic avec une vitesse  $W$  (pente de la droite).

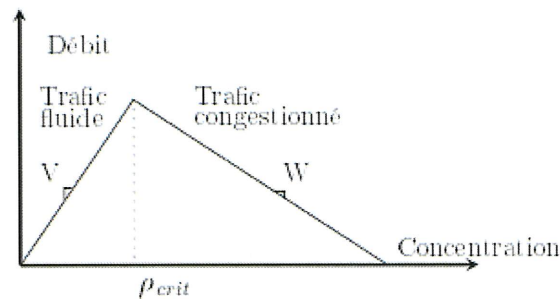


FIGURE 1.1: Diagramme fondamentale théorique de trafic

Le modèle microscopique vise à déterminer le trafic de sortie en fonction du trafic d'entrée, en représentant l'évolution des grandeurs, mentionnées au-dessus, dans un repère spatio-temporel.

La majorité des approches macroscopique dérive du modèle Lighthill, Whitham, Richards (LWR). Le modèle LWR a pour principe la conservation de véhicules représentée par l'équation

$$\frac{\partial_t \rho(x, t)}{\partial t} + \frac{\partial_x Q(x, t)}{\partial x} = 0$$

## 1.2 Approche microscopique

En opposition au modèle macroscopique, le modèle microscopique visent à décrire le comportement du véhicule. Dans ce cadre, l'étude et la simulation s'intéressent à l'évolution des



variables permettant de caractériser chacun des véhicules évoluant sur l'infrastructure. Ces paramètres sont la position, la vitesse et l'accélération. La position est définie par deux composantes :

- **Positionnement longitudinal** : Il s'agit de l'avancement du véhicule le long de la route. Il est usuellement caractérisé par la distance  $x$  séparant le véhicule du début de la route. Le début d'une route correspond à l'une de ses extrémités généralement choisie de manière à ce qu'un véhicule se déplace sur la route en allant du début de la route vers sa fin. Ainsi,  $x(t)$  est une fonction croissante. Il est également courant de désigner cette notion comme un point kilométrique (pk) analogie aux bornes kilométriques utilisées le long des routes.
- **Positionnement latéral** sur la chaussée. Dans les cas les plus simples, il peut s'agir du numéro de la voie empruntée par le véhicule, ou dans des cas plus précis de la mesure de l'écart du véhicule par rapport au bord de la route.

D'où, les modèles microscopiques cherchent à déterminer l'évolution de ces composantes de positionnement. Pour déterminer le déplacement du véhicule deux modèles sont utilisés (un décrivant chaque positionnement) :

- **Modèle de suivi** qui permet de déterminer l'évolution de la composante longitudinale de la position des véhicules. Ces modèles sont souvent exprimés par des systèmes non linéaires donnant l'accélération d'un véhicule à un instant donné (notée  $\ddot{x}_i(t)$ , la seconde dérivée de la position  $x_i$  par rapport au temps  $t$  du véhicule  $i$ ) en fonction de la situation dans laquelle se trouve le véhicule :

$$\ddot{x}_i(t) = D(\text{Environnement})$$

En générale, les modèles de suivi tendent à exprimer la décision  $D$  prise par le conducteur, définie par un changement de l'accélération du véhicule  $\ddot{x}_i$ , en fonction de l'état de l'environnement. Dans les cas les plus simples, on décrit l'évolution de la position du véhicule en calculant la distance le séparant de son prédécesseur, notée  $\Delta x(t) = x_{i+1}(t) - x_i(t)$  où  $x_{i+1}$  est la position longitudinale du véhicule précédant le véhicule considéré et  $x_i$  la position longitudinale du véhicule simulé.

- **Modèle de changement de voie** permet de déterminer le positionnement latéral du véhicule. Généralement dans ce type de modèle, le véhicule évolue dans une infrastructure qui comporte plusieurs voies. Il s'agit, alors, de chercher à déterminer la voie de circulation adoptée. Ce processus évoque deux stades de traitement lesquels :



1) décider de changer de voie : le conducteur perçoit la nécessité ou cherche à augmenter son gain, lors d'un potentiel changement de voies.

2) la recherche d'un créneau permettant ce changement de voie : c'est la concrétisation de cette volonté de changement de voie et choisissant le moment opportun pour le faire

Si le positionnement latérale se limité à son cas générale de changement de voies, l'opération effectuée est instantanée(ce qui suppose une accélération latérale infinie).

De ce fait, on remarque que le choix de modèles dépend de la situation qu'on vise à simuler. Si la simulation, se concentre sur la détermination d'un équilibre de trafic, lequel se résume en une estimation des paramètres du diagramme fondamentale de trafic, le choix de modèle le plus adéquat serait le modèle macroscopique. Par contre, si la plus grandes importances porte sur des phénomène très localisés dans le trafic, alors le choix passe aux modèles microscopiques. Dans ce modèles, plusieurs paramètres du véhicules sont étudiés. Ce qui rend la tâche de détermination et de calibration des paramètres les plus pertinents difficiles. Deux méthodes sont proposées pour faciliter la détermination de ce jeu de véhicules :

- soit en disposant d'un jeu de paramètres uniformes commun à l'ensemble des véhicules
- soit en utilisant une loi de probabilité guidant la génération de chacun des paramètres individuels. Par exemple il est possible de considérer que sur l'ensemble des vitesses désirées -vitesse à laquelle un véhicule souhaite évoluer sur l'infrastructure- des véhicules empruntant une infrastructure suivent une loi de probabilité gaussienne de moyenne  $V_{moy}$  avec un écart type de  $E$ . Le problème se résume donc dans ce cas à estimer la loi de probabilité à utiliser ainsi que ses paramètres plutôt que les paramètres individuels, ce qui restreint l'espace de recherche durant la phase de calibration.

### 1.3 Synthèse

Une comparaison des approches macros et microscopique est présentée dans la figure 1.2. En effet, La simulation macroscopique se montre adaptée à la simulation de réseaux importants, car elle permet pour un moindre coût opérationnel d'obtenir des résultats utilisables par des exploitants d'infrastructures. Les modèles microscopiques, eux, se montrent plus coûteux sur différents points : ils nécessitent une description du réseau et de sa géométrie plus précise, des données d'entrée considérables, demandent un processus de calibration conséquent et requièrent un temps de calcul important. Ils montrent donc leurs limites pour la simulation de réseaux étendus et sur lesquels circule un nombre très important d'entités. Malgré ces limitations,

les modèles microscopiques offrent des possibilités non atteignables avec les modèles macroscopiques, surtout quand il s'agit d'étudier et comprendre des phénomènes localisés. Ainsi, ils se montrent particulièrement adaptés à l'étude des points de difficultés et de conflits sur les réseaux tels que les intersections ou les insertions. De plus, les simulations microscopiques sont fondées sur le principe d'émergence, et permettent donc de faire apparaître des situations complexes, et non anticipées par le modélisateur, au niveau du trafic généré. Cet aspect est la force principale de ce type d'outils et une raison importante pour laquelle les modèles microscopiques occupent une place significative dans la littérature. Les modèles microscopiques sont ainsi, par nature, plus adaptés à la réalisation de simulations explicatives et prospectives tandis que les modèles macroscopiques sont adaptés pour des simulations à visées reproductives et prédictives.

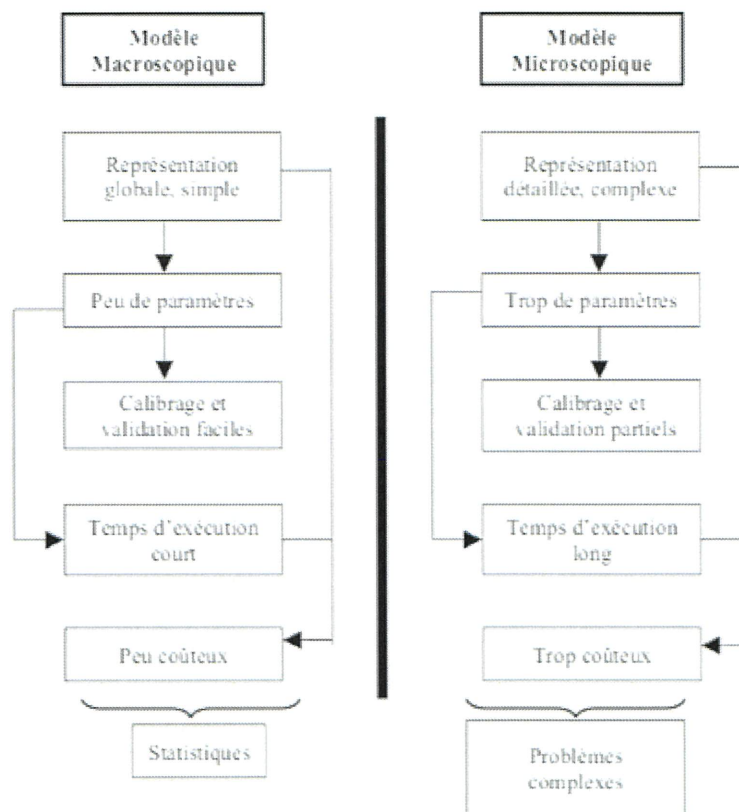


FIGURE 1.2: Comparaison des approches de modélisation de trafic

## Conclusion

Nous avons présenter dans ce chapitre deux grandes approches de modélisation : l'approche macroscopique qui vise à dégager une relation entre le trafic d'entrée et le trafic de sortie et l'approche microscopique qui a pour but de déterminer la position du véhicule simulé. Nous avons conclue le chapitre par une synthèse comparative des deux approches. Dans le chapitre suivant, nous allons présenter la notion de multi-agents et sa contribution dans la simulation de trafic.

## Chapitre 2

# Les Systèmes Multi-Agents

### Introduction

La notion d'Agent s'est présentée pour répondre aux besoins accrus des domaines qui dérivent de l'informatique. En effet, actuellement les systèmes informatiques sont de plus en plus complexes, constitués de logiciels en interaction entre eux ou avec des êtres humains. De plus, certains systèmes sont souvent répartis sur plusieurs sites. D'où un besoin s'est émergé pour assurer l'intégration, l'inter-interopérabilité et la coopération des logiciels existants. Dans ce qui suit, nous allons présenter les Agents et Systèmes Multi-Agents.

## 2.1 Les Agents

### 2.1.1 Définition

D'après le Dictionnaire de l'Académie Française :

" **AGENT** n.m. (du latin scolastique *Agens*, participe passé de *agere*, agir) Ce qui exerce une action, par opposition à ce qui la subit."

D'où un agent est par définition doté d'une capacité d'agir. De ce fait, il est capable de changer son propre état, ainsi que l'état de l'environnement où il évolue. On distingue deux types d'actions possibles :

- Actions externes : une panoplie d'actions possible dont la communication et la façon d'agir dans l'environnement.
- Actions internes : la mise à jour des connaissances de l'agent et le développement de son processus décisionnel.

De plus, selon la définition présentée au dessus, l'agent a un caractère d'autonomie peu ou pas du tout sensible aux interventions externes (tels les actions de l'homme). En effet, il a une "liberté" de choisir les actions et effectuer les décisions qu'il juge opportun.

Comme schématiser dans 2.1 agent est défini [3] comme étant une entité physique ou virtuelle qui évolue dans un environnement. Elle peut le percevoir et l'affecter par des les actions qu'elle effectue. Un agent est doté d'un degré d'autonomie en fonction de ses connaissances de l'environnement et ses interactions avec les autres agents. De ce fait, un agent est capable d'exercer un contrôle local sur ses processus (de perception, de communication, d'acquisition de connaissances et de raisonnement et de prise de décision ou d'exécution).

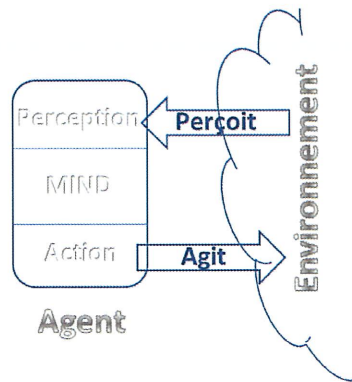


FIGURE 2.1: Schémas d'Agent

Selon Ferber,

un agent est une entité réelle ou virtuelle :

- 1- qui est capable d'agir dans un environnement,
- 2- qui peut communiquer directement avec d'autres agents,
- 3- qui est gouvernée par certaines tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
- 4- qui possède des ressources propres,
- 5- qui est capable de percevoir (mais de manière limitée) son environnement,
- 6- qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- 7- qui possède des compétences et offre des services,
- 8- qui peut éventuellement se reproduire,
- 9- dont le comportement tend à satisfaire ses objectifs, en tenant compte des res-



sources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

D'où un agent [4, 5] est caractérisé par

- Autonomie : Il n'a pas besoin de l'homme pour superviser son fonctionnement et réalisation de ses tâches.
- Pro-activité : Il est capable d'anticiper les changements de l'environnement et de réagir en fonction de ces changements.
- Réactivité : Il a la capacité de répondre à un environnement extérieur.
- Sociabilité : Un agent est capable de communiquer directement ou indirectement avec les autres agents.

Un agent est présenté de manière formelle [6] par la sextuple :  $Agent N = (\mathcal{P}, \mathcal{C}, \mathcal{A}, p, f, u)$

où :

- $\mathcal{P} = p_1, p_2, \dots, p_i$  : l'ensemble des perceptions de N,
- $\mathcal{C} = c_1, c_2, \dots, c_j$  : l'ensemble des connaissances de l'agent N,
- $\mathcal{A} = a_1, a_2, \dots, a_k$  : l'ensemble des actions réalisables par N,
- $p : P^{l*} \times C^m \rightarrow A^n$  : la fonction des possibles de N, permet d'évaluer et d'ordonner les différentes actions possibles pour choisir la mieux adaptée au contexte
- $u : F(P^{l*} \times C^m, A) \rightarrow \mathbb{R}$  : la fonction d'utilité de N, détermine l'action réalisée par l'agent, en combinant la fonction des possibles p à la fonction d'utilité u.
- $f : P^{l*} \times C^m \rightarrow A$  : la fonction de choix de N.

Cette définition est générique. Chaque modèle d'agents y apporte des adaptations, notamment au niveau des fonctions p et u.

### 2.1.2 L'environnement

L'environnement où évolue l'agent est défini par l'ensemble des ressources externes mise à la disposition de l'agent et les règles qui guident son évolution. Il s'agit de tous les éléments externes à l'agent. De ce fait on peut avoir plusieurs types d'environnement.

- **Environnement Accessible/Inaccessible** : L'accessibilité aux informations de l'environnement. C'est ce la possibilité ou pas de percevoir son environnement, qui est affectée par la limitation à la vision d'agent ou la quantité d'informations perçues.
- **Déterministe/Stochastique** : L'environnement est dit déterministe, si son état est connue tout le temps. Toutefois, les interventions simultanées des agents impose le non-déterminisme

de l'environnement. D'où, les environnements complexes ou pas sont stochastique.

- **Épisodique/Séquentiel** : Un environnement est dit épisodique, si l'exécution d'une action par l'agent est indépendante de la séquence d'actions qui le précède. En opposition, si une action influe la séquence des actions qui le suivent, l'environnement devient séquentiel (non-épisodique).
- **Statique/Dynamique** : Dans un environnement statique, seules les interventions des agents peuvent changer son état. Si d'autre processeurs interviennent dans le changement d'état, il est dit dynamique.
- **Discret/Continu** : Un environnement est discret, si l'ensemble d'actions possibles sur ce dernier est fini. La plus part des environnements sont continu, mais, rendu discret par l'architecture. En effet, en limitant la perception de l'agent de son environnement, l'agent il peut avoir une approximation discrète de l'environnement continu.

Par ailleurs, une caractéristique de l'agent non toujours considérée dans les définitions, est la faculté d'**apprentissage**. Il s'agit, de la capacité d'un agent de changer son état dynamiquement, en faisant évoluer sa base de connaissances. L'agent, peut changer de manière dynamique ses règles comportementales, se dotant ainsi d'un autre niveau d'autonomie.

## 2.2 Les systèmes Multi-Agents

Les systèmes multi-agents (SMA) mettent en œuvre un ensemble de concepts et de techniques permettant à des logiciels hétérogènes, ou à des parties de logiciels, qui sont les agents de coopérer suivant des modes complexes d'interaction.

### 2.2.1 Définition

Un définition possible des SMA est donnée par [7]. C'est l'ensemble de :

- **Environnement E** : espace, disposant généralement d'une métrique
- **Ensemble d'objets O** situés dans E : La position des objets dans E est connue à chaque instant. Les objets peuvent être actif ou passif.
- **Ensemble d'agents A** : les agents sont des objets actifs. ( $A \subseteq O$ ).
- **Ensemble de relations R** : une relation est un lien définit entre objets (agents ou objets passifs).
- **Ensemble d'opérations Op** : une opération vise à manipuler un agent (perception, créa-



tion, transformation, etc).

- **Les lois d'univers** : un ensemble d'opérateurs qui définissent les résultats de l'application des opérations de l'ensemble Op.

Les SMA sont définis autrement[8], comme étant un ensemble d'agents en interactions qui visent à atteindre un but ou réaliser des tâches. L'interaction peut être directe en communiquant (par exemple, via un échange de messages) entre agents où indirectement, en percevant les résultats des actions des autres agents sur l'environnement. Selon le types d'interactions et le but à atteindre, nous pouvons catégoriser les agents comme suite.

- **Agent réactif** : il peuvent être des agents à réflexe simple ou des agents qui conservent une trace du monde.
- **Agent délibératif** :regroupe, les agents ayant un but, ceux utilisant un fonction d'utilité et les agents BDI (Belief, Desire, Intentions)
- **Agent hybride**

### 2.2.2 L'approche SMA

Le domaine des systèmes multi-agents est issu principalement de l'intelligence artificielle distribuée. Un système multi-agents est un système dans lequel évoluent les agents. Ceux-ci interagissent et s'organisent de manière dynamique et adaptative. L'objet des SMA est l'étude des systèmes dans lesquels l'intelligence (au sens le plus large, comprenant la capacité délibérative, la connaissance ou l'intelligence collective) n'est généralement pas concentrée en un endroit unique mais répartie entre ses différents agents. L'étude des systèmes multi-agents porte donc un grand intérêt aux interactions se produisant entre les différentes entités le composant, ainsi qu'aux phénomènes résultant de ces interactions. L'un des exemples d'utilisations des SMA est la colonie de fourmis [9]. Dans cet exemple, l'agent représente une fourmi, être dépourvue d'intelligence, mais doté de capacité réflexive. L'interaction entre les différents agents et le support du caractère intelligent du système.

#### 2.2.2.1 Modéliser le monde

#### 2.2.2.2 Modèle

D'après [10], un modèle est défénit comme suite :

"To an observer  $B$ , an object  $A^*$  is a model of an object  $A$  to the extent that  $B$  can use  $A^*$  to answer question that interest him about  $A$ "

Cette définition indique que, dans les approches de modélisation classique le rôle d'un modèle est d'approximer l'objet réel dans la limite où les question de l'observateur concernant l'objet réel sont répondues. EN effet la relation entre l'objet à modéliser est son modèle est direct , comme présenter dans la figure2.2.

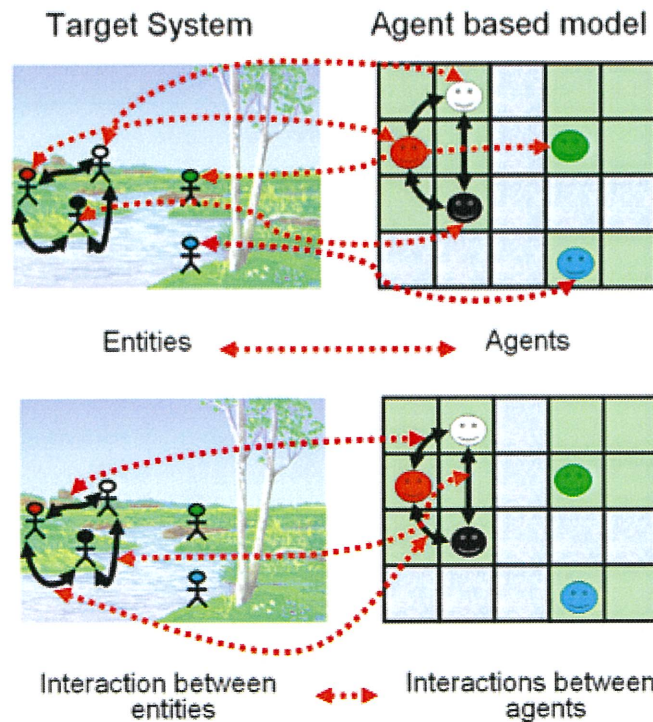


FIGURE 2.2: Relation directe entre le monde réel et le modèle [11]

Toutefois, dans l'approche multi-agent qui vise à modéliser les agents pour faire partie d'un Tout, l'objet d'étude n'est pas directement modélisé. En effet, l'objet réel est modéliser par un système composé de plusieurs modèles individuels, de relations et d'échanges. La modélisation de l'environnement dans lequel évolue l'objet fait aussi partie du système de modélisation. Opter pour une telle approche, influence à la fois la phase de modélisation que la phase de validation laquelle est rendue plus aisée en déplaçant la difficulté vers la phase de modélisation.

### 2.2.2.3 La modélisation

L'activité de modélisation est particulièrement affectée. Il est aisé de déterminer quelles sont les entités à modéliser du fait du lien direct qui existe entre les agents et les entités du monde observé. Il en va de même des relations entre ces agents qui sont directement reproduites. Malgré cette simplification de l'analyse, la modélisation réelle de chacun des agents est, elle, une tâche non triviale. En effet les acteurs modélisés ont un comportement qui n'est pas nécessairement bien connu, pas plus que la nature exacte de leurs relations. L'environnement dans lequel doivent évoluer les agents doit également être modélisé et être l'objet d'une réelle attention.

### 2.2.2.4 La validation

Si la réalisation d'un système à base d'agents n'est pas triviale, la vérification et la validation de ce dernier le sont encore moins. Dans l'approche classique, et dans la lignée de la définition de Minsky, la validation d'un modèle consiste à vérifier qu'il permet de répondre à une question que l'on se pose sur l'objet d'étude, et ce, avec une marge d'erreur acceptable. Dans le cas de simulations basées sur une approche multi-agents, cette vision ne s'applique pas directement. L'objet d'étude étant le système, et les questions d'intérêt portant sur le système dans son ensemble, seul celui-ci peut être validé. L'objet d'étude n'étant pas l'agent lui-même, la notion de validation de celui-ci, et ce, indépendamment du reste du système n'existe pas. C'est donc la conjonction de nombreux modèles qui est étudiée comme un tout et non chacun de ses composants. Chacun des composants- c'est à dire les modèles d'agents, leurs communications ainsi que l'environnement- ne peut être validé indépendamment au regard de la question originale, celle-ci portant sur le système.

Même si un modèle multi-agent est développé pour répondre à un besoin ou atteindre un but, ses composantes sont ré-utilisables dans un différent contexte.

### 2.2.3 Les SMA et la simulation de trafic

Les SMA pour le trafic, font partie des modèles microscopique particuliers. Dans ce qui suit, nous allons présenter les apports de cet outil à la simulation.



2.2.3.1 Vision et perception limitée

Les SMA, permettent de développer une vision centralisée du conducteur simulée, qui discrétise l’environnement de développement de ce dernier, de façon similaire à ce qu’il le perçoit. En effet, comme présenté dans [12], l’être humain développe une vision égo-centrée de son entourage, où les position des objets qui l’entoure sont relative à sa propre position. Dans ce cadre, le champs de vision du conducteur se limite à son voisinage proche et sa perception du monde est découpée en parties condensées regroupant chacune des propriétés différentes des autres. Ces parties sont dites zones et sont généralement définies selon les axes longitudinaux et latéraux comme présentée dans la figure 2.3.

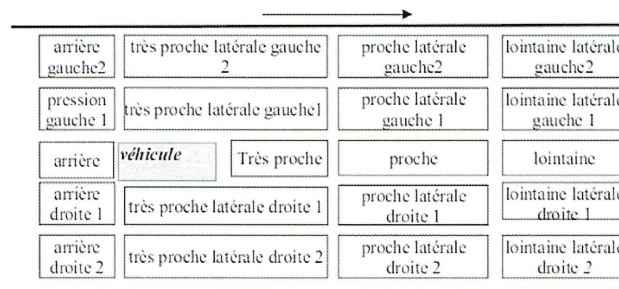


FIGURE 2.3: Vision égo-centrée de l’environnement [13]

De ce fait, l’objet de perception du conducteur est les caractéristiques qui distinguent chaque zones (ex le flux de véhicules, la vitesse du flux, sa densité, etc). Sa perception de son environnement, intervient par la suite dans le processus décisionnel du conducteur, principalement, le changement de voies. L’influence apportée par les phénomènes émergents (des propriétés du trafic sur des zones) sur le comportement individuel est dite imergence.

Dans la vie réelle, la perception du conducteur est limitée. En effet, il ne peut percevoir qu’un nombre limités de phénomènes à la fois. Cette perception est influencée par :

- 1- Le but à atteindre : l’information détectée dans ce cadre est utile pour réaliser ce but.
- 2- La saillance de l’information : par exemple un obstacle dans la route.

La limitation apportée au processus de perception, apporte des limitations sur le processus décisionnel, dues aux manques de connaissances résultants des percepts (un percept est l’élément unitaire de perception).

### 2.2.3.2 Processus de Décision

La description du comportement de conducteur ne se limite, désormais, plus à la définition d'un jeu de paramètres en vue de résolutions d'équations dynamiques. Le comportement peut, alors, être décrit via plusieurs méthodologies tels la logique floue [14], la base des règles [15]. Les SMA permettent aussi d'introduire un module de gestion d'interactions (une interaction étant la prise en considération des autres conducteurs) haut niveau qui permet de décrire des comportements plus complexes. Un exemple de gestion d'interactions, proposées dans [16], est présenté dans la figure 2.4.

- Suppression d'interaction
  1. Interaction + longue durée + Possibilité de suppression => suppression de l'interaction
- Adaptation
  2. Interaction + courte durée + possibilité de suppression => adaptation de courte durée
  3. Interaction + courte durée + impossibilité provisoire de suppression => adaptation de courte durée
  4. Interaction + longue durée + impossibilité durable de suppression => adaptation de longue durée

FIGURE 2.4: Exemple de règles de gestion des interactions

La simulation Multi-Agent rend aisée la simulation de tâches de conduites complexes telles que la gestion de conflits à la traversée des intersections

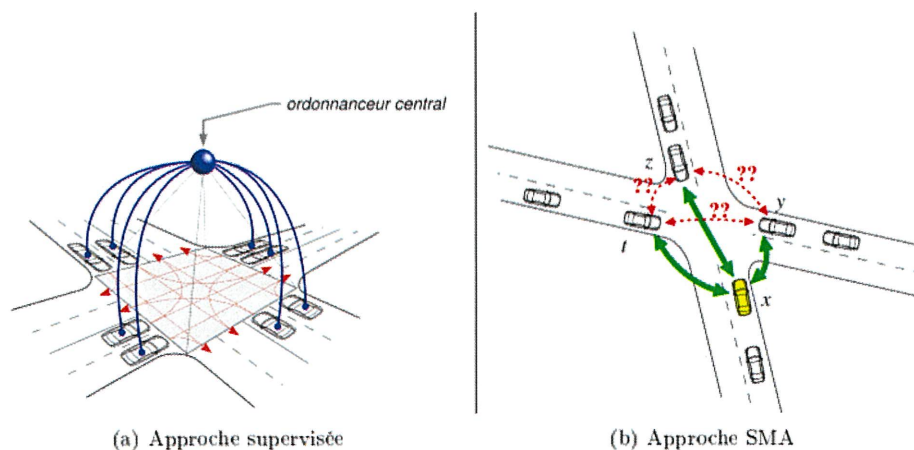


FIGURE 2.5: Différentes approches pour la gestion des conflits en zone d'intersection. Une microscopique simple (3.7(a)) consiste à utiliser un ordonnanceur central disposant de suffisamment d'informations pour résoudre les conflits globalement et guider les véhicules dans la zone d'intérêt. Dans l'approche SMA (3.7(b)) chaque véhicule ne dispose que d'une vision limitée de l'environnement (état de la route, véhicules avec lesquels il est en conflit...) et doit élaborer une stratégie sans savoir comment les autres véhicules envisagent de traverser l'intersection. [17]

## Conclusion

Dans ce chapitre nous avons présenté la notion d'agent et présenté quelques caractéristiques des Systèmes Multi-Agent. De plus, nous avons présenté quelques apports des SMA dans la simulation du trafic. Dans le chapitre suivant, nous allons présenter le simulateur de trafic YATS.

## Chapitre 3

# Yet Another Traffic Simulator (YATS)

### Introduction

YATS est un simulateur développé au sein du laboratoire de l'IFSTTAR. C'est un simulateur développé en C++ et utilise une méthodologie multi-agents pour simuler le comportement des conducteurs. Dans ce qui suit, nous allons présenter YATS et préciser notre apport à cette plateforme en cours de développement.

### 3.1 Architecture de YATS

Dans cette section, nous allons présenter l'architecture du simulateur. Ceci est dans le but de positionner notre travail dans son cadre, et réaliser un travail en cohérence avec l'existant mais qui réponds au mieux à nos besoins.

#### 3.1.1 Diagramme de Classe

Comme le montre le diagramme présenté dans la figure 3.1, le simulateur se base sur une classe de base qui est `pointSim` qui mémorise les coordonnées des éléments de simulation. Ceci suggère que chaque élément de la simulation a une connaissance de sa position dans un repère global. Les éléments pertinents pour la simulation sont :

- le véhicule : modélisé par la classe `Vehicle`. Une véhicule mémorise la position de son centre de gravité.
- la route : modélisée par une ou plusieurs classe `Road`. En effet la classe `Road` représente un modèle d'un segment de route. Ce segment est défini par une équation de droite



laquelle calculée à partir d'un point d'entrée et de sortie. De ce fait, pour chaque route, un sens est défini : le véhicule migre depuis le point d'entrée de son segment de début, modélisé par la classe SourceSim, vers le point de sortie de son segment de sortie.

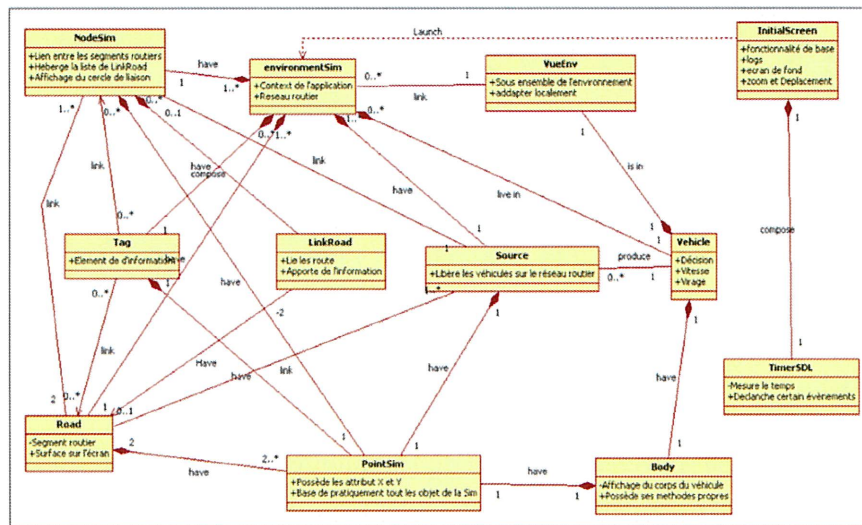


FIGURE 3.1: Diagramme de classe de YATS [18]

### 3.1.2 Initialisation de la Simulation

Au début de chaque simulation, le noyau du simulateur, crée une liste des nœuds définissant les limites de la route. Cette liste est introduite par le biais des fichiers de configuration (figure 3.2).

Une fois la liste des routes est créer (une route est définie par ses limites et son nom), le simulateur crée la liste des NodeSim. Dans ce stade, le simulateur détermine les listes des pointes d'entrée et des points de sortie. Ensuite, la liste des RoadLink est créée, pour créer après la liste des sources. Une fois les sources créés, elle génèrent les véhicules. (la description des classes est en annex).

### 3.1.3 Le modèle d'agent

L'agent dans YATS est le véhicule.

**Perception et Vision** Le module de perception de l'agent est formé par une vue de l'environnement cédé à l'agent pour pouvoir se positionner dans son entourage local. Ainsi, la percep-

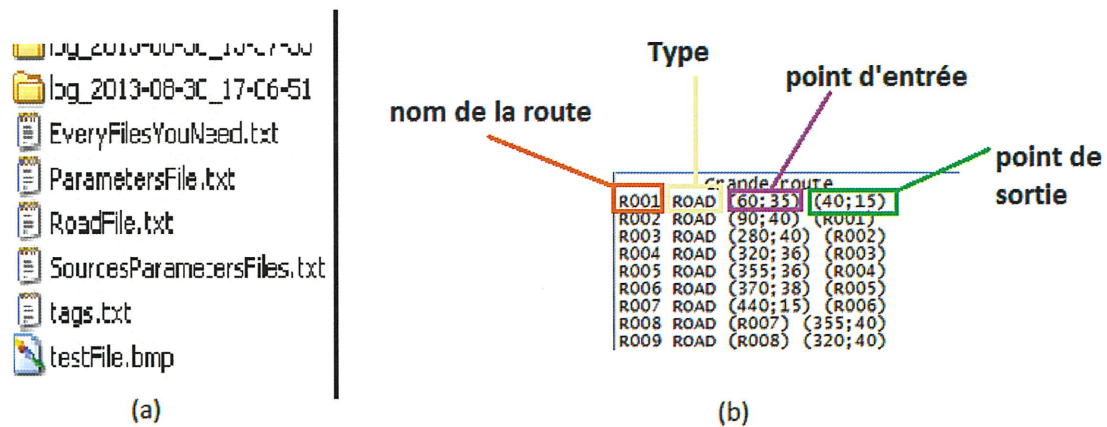


FIGURE 3.2: Les fichiers de configuration de YATS : (a) la liste des fichiers de configuration, le chargement des fichier se fait selon l'ordre déterminé dans le fichier everyfileyouneed ; (b) le contenu du fichier de configuration des routes

tion de l'agent se limite à son emplacement dans la route où il se situe.

La vision du véhicule est guidée par un point focal déterminant l'élan de son champs de vision. C'est le facteur qui décide de la direction du véhicule. En effet, ce dernier tend à atteindre son point focal(but local) lequel une fois atteint, engendre le calcul de nouveau du champs de vision.

**Processus décisionnel** Le processus décisionnel (figure 3.3) est lancer périodiquement. En effet, le véhicule teste la distance le séparant du point de sorti de chaque segment de route. SI elle est inférieur de la distance de freinage (figure 3.4.(b) ), il calcule trois état possibles (tourner à gauche, tourner à droite, avancer) et choisit l'état qui le rapproche de sa destination (figure 3.4.(a)).

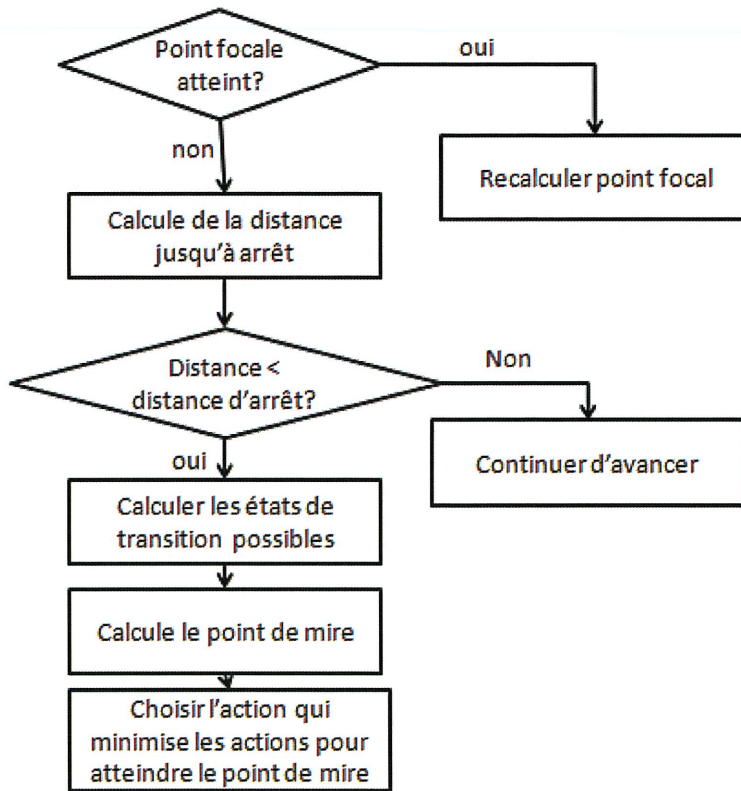


FIGURE 3.3: Prise de décision

### 3.1.4 Les limite de YATS

La première limitation dégager est par rapport à l’environnement. En effet, tel était YATS ne supportait pas une simulation en route comportant plusieurs voies. De plus, la vue que le véhicule réalise de son véhicule est restreinte à l’infrastructure. Il est incapable de détecter les autres emprunteurs de la route. Ceci rends le processus de prise de décision égoïste, et obsolète. Dans la vie réel, un tel comportement résulte toujours en accidents grave et collisions. En lisant le code du simulateur, nous nous sommes rendus compte du manque de modalités dans ce dernier et de la dépendance du code de la partie graphique.

## 3.2 Contribution

Pour cette partie , nous allons commencer par présenter une nouvelle architecture du système. Après, nous allons détailler la conception de la classe qui modélise la route bidirectionnelle. Pour conclure cette partie, nous allons présenter le nouvel modèle d’agent adopter, et

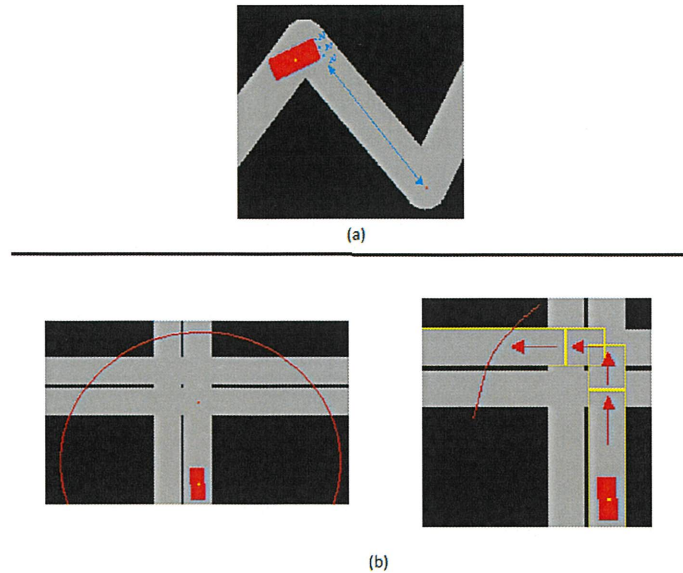


FIGURE 3.4: Exemple d'application du processus de Décision : (a) en effectuant un virage e véhicule calcule les point p1, p2, p3 correspondant aux états de transition possible ; (b) Avant de tourner à gauche, le véhicule calcule sa distance d'arrêt laquelle s'avère trop petite, il continue d'avancer jusqu'à trouver la distance la plus proche possible lui permettant de tourner.

expliquer le protocole d'interaction introduit.

### 3.2.1 Architecture

Comme première intervention nous avons opter pour une approche Model-View(3.5). D'une part, elle assure la séparation du graphique du traitement effectué. D'autre part, elle permet d'alléger l'interdépendance des classes et augmenter la modalité du code pour des raisons de lisibilité et de manipulation.

La figure donne l'exemple de la classe de véhicule.



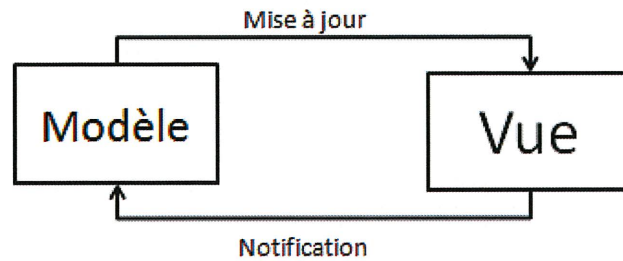


FIGURE 3.5: Modele-Vue

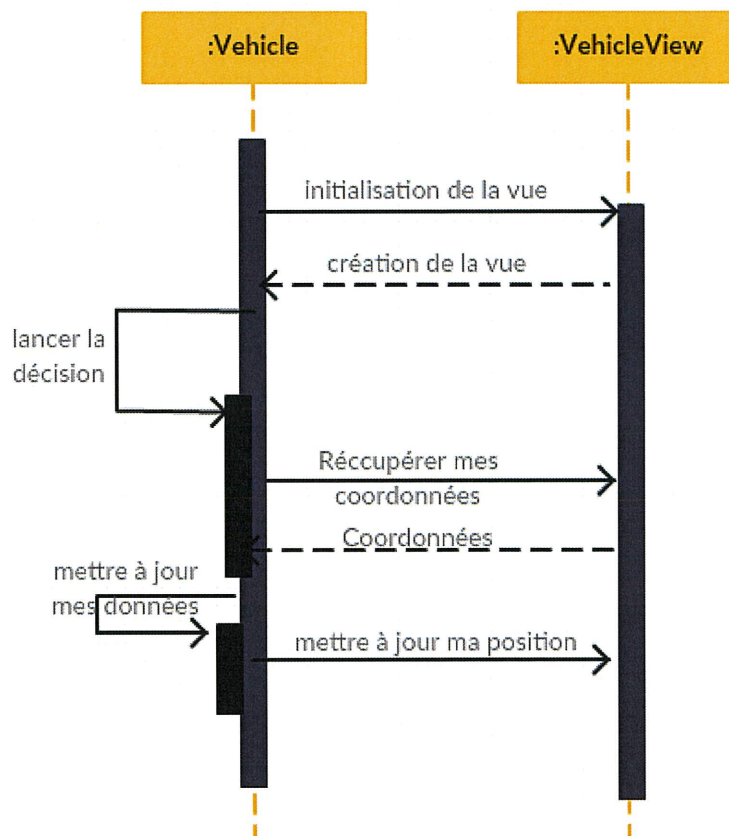


FIGURE 3.6: exemple de séparation de la vue du module

La figure 3.7 présente le diagramme de classe révisé de notre nouvelle version du simulateur. La séparation des modèles/vues est visible et les interactions entre les classes sont révisées. L'autre différence par rapport à l'ancienne architecture est la classe NeighborScoutingProtocol. C'est une classe qui se charge d'assigner le voisinage de chaque véhicule. Nous

allons détailler le protocole qui lui correspond dans la section 3.2.4. Et bien évidemment, comme l'ancienne architecture ne prend pas en considération la notion de multiple-voies, nous avons conçue une classe BidirectionalRoads qui se charge de créer un objet route bidirectionnelle comprenant une voie droite et une voie gauche.

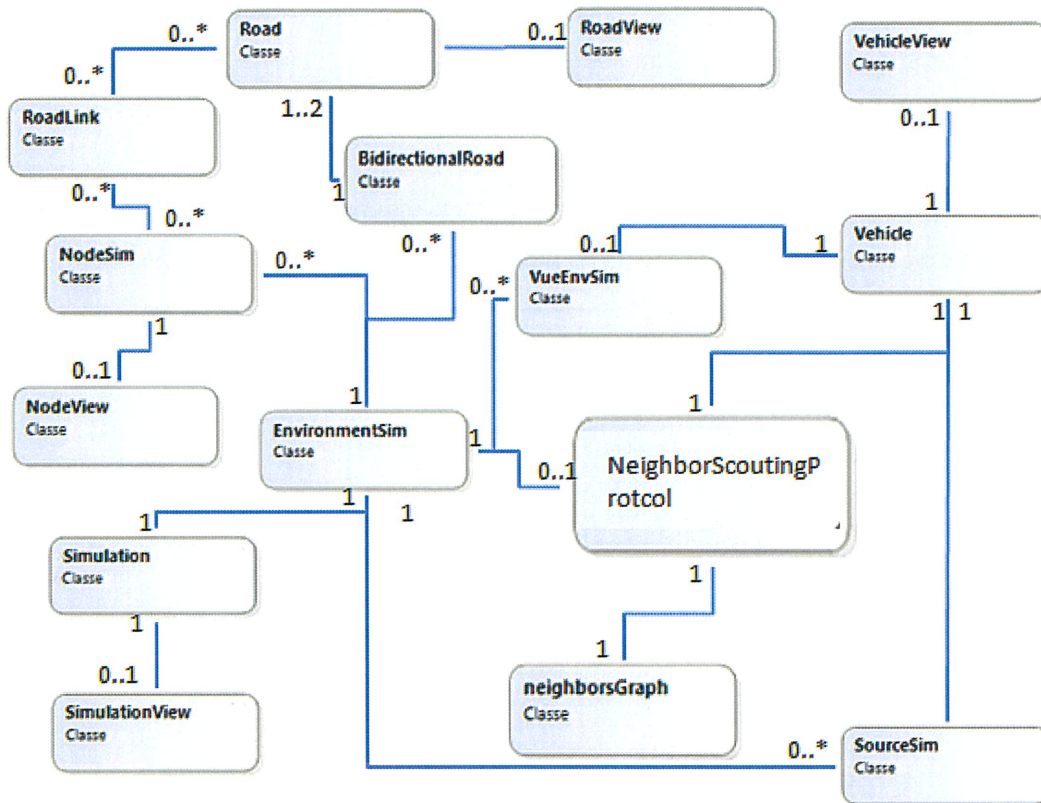


FIGURE 3.7: Diagramme de Classe

### 3.2.2 Méthodologie de conception

Pour développer notre SMA, nous avons optés pour la méthode de conception GAIA [19]. C'est une approche qui dérive des méthodologie logicielle classique, avec une large reconnaissance dans le domaine multi-agent. Gaia se veut être générale et applicable à n'importe quel domaine, et compréhensible par la distinction entre macro-niveau et micro-niveau. Les agents modélisés, pouvant être hétérogènes, sont des systèmes computationnels à gros grain, qui vont essayer de maximiser une mesure de qualité globale. Gaia manipule six modèles d'analyse et de conception différents (figure 3.8) :

- le modèle de rôle, qui identifie les différents rôles devant être joués par les agents du système
- le modèle d'interaction qui définit les protocoles de communication entre agents
- le modèle d'agent qui attribue les rôles aux différents agents du système
- le modèle de service qui définit, comme son nom l'indique, les différents services offerts par le système, et les agents tributaires
- le modèle d'organisation qui définit la structure de l'organisation grâce à des graphes orientés représentant les voies de communication entre agents
- le modèle environnemental qui décrit les différentes ressources accessibles caractérisées par les types d'actions que les agents peuvent entreprendre pour les modifier.

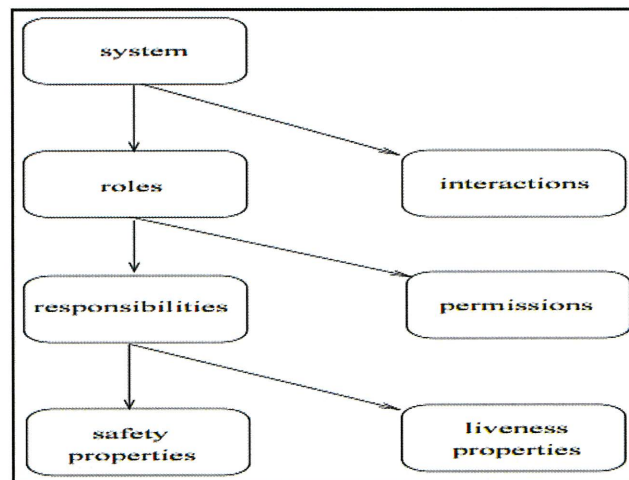


FIGURE 3.8: Modèles d'analyse de Gaia

### 3.2.2.1 Modèle des rôles

Le modèle des rôles identifie des rôles clés dans le système. Un rôle peut être considéré comme une description abstraite de la fonction d'une entité attendue. Il est caractérisé par deux types d'attributs :

- Permissions/Droits : Selon le type et la quantités des ressources à exploiter pour réaliser un rôle, certaines permission sont affectées. Ce champs permet d'identifier les ressources que le rôle en question peut utiliser et la limite d'utilisation de cette ressource. Une ressource est généralement une information ou une connaissance.
- Les responsabilités : Pour munir à bien un rôle, certaines tâches (responsabilités) doivent



être accomplies. Les responsabilités sont divisés en deux catégories :

- Vivacité : Les responsabilités de vivacité sont ceux qui , intuitivement , affirment que "quelque chose de bon va arriver " . Elles sont appelées ainsi parce qu'elles ont tendance à dire que "quelque chose sera faite" , et donc que l'agent effectuant le rôle est encore en vie . Elles ont tendance à suivre certains modèles. Par exemple, le type d'objectif de réalisation *réponse garantie* est de la forme "une demande est toujours suivie d' une réponse". La répétition infinie d'un objectif à réaliser a la forme "x souvent arrivera infiniment".
- Sécurité : Les invariants qu'un agent soit maintenir.

Opérateur	Interprétation
$x.y$	x suit y
$x y$	x ou y se produit
$x^*$	x se produit 0 fois ou plus
$x^+$	x se produit au moins une seule fois
$x^w$	x souvent se produit infiniment
$[x]$	x est optionnelle
$x  y$	x et y entrelacées

FIGURE 3.9: Les opérateurs de vivacité

### 3.2.2.2 Modèle d'interaction

Dans un système comprenant plusieurs composantes, les SMA non exclus, des liens et des relations de dépendances se manifestent entre les composantes (agents). Les liens reliant les rôles sont représentés par le modèle d'interaction. Il s'agit d'un ensemble de définitions de protocoles (pattern formalisé d'interactions). La définition d'un protocole comprends les champs suivants :

- But : La nature de l'interaction
- Initiateur : Le rôle lequel a initié l'interaction
- Répondeur : Le rôle avec lequel il est en interaction
- Entrées : Les informations utilisées par l'initiateur durant l'interaction
- Sorties : Les informations fournies au répondeur durant l'interaction
- Traitement : Description de tout traitement effectué par l'initiateur durant l'interaction

### 3.2.3 Modèle Multi-Agents

Dans cette section, nous allons présenter les modèles les plus pertinents de notre projet.

#### 3.2.3.1 Modèle de rôle

**Rôle Vehicle** Le rôle Vehicle correspond à l'agent véhicule. A chaque cycle d'exécution, le véhicule essaie de se positionner physiquement, dans la route. De ce fait, il a accès aux informations de la route et de la voie où il roule. Une fois, les informations nécessaires sont dans sa base de connaissance, il lance son processus décisionnel, dépendant de son objectif (avancer ou changer de voie).

Schémas de rôle: VEHICULE	
<b>Description:</b>	Se localise dans la route, détecte ses voisins pour avancer dans la route et atteindre sa destination.
<b>Protocoles et Activités:</b>	DetecterPosition, Decider, AvancerDansRoute, ChangerVoie
<b>Permissions:</b>	lire idRoute // l'identifiant de la route où il est localisé idVoie // l'identifiant de la voie où il est localisé idVoisin // l'identifiant du véhicule voisin, vitesseVoisin // la vitesse du véhicule voisin position vitesse
<b>Responsabilités</b>	
<b>Vivacité:</b>	VEHICLE = (DetecterPosition.Decider.AvancerDansRoute) <sup>w</sup> VEHICLE = (Decider.ChangerVoie)
<b>Sécurité:</b>	changerVoie = routeLibre = true    (routeLibre = false)^(voieLibre=true)

FIGURE 3.10: Rôle : Vehicule

**Rôle BidirectionnalRoad** Le rôle BidirectionalRoad correspond à l'agent route bidirectionnelle laquelle est déjà présentée au préalable. La Bidirectionnelle, se charge de générer deux rôles Road formant ses voies droite et gauche. Au cours de la simulation, elle répond aux interrogations du rôle Vehicle concernant ses données principales (nom, vitesse maximale supportée,

etc).

FIGURE 3.11: Rôle : BidirectionalRoad

<b>Schémas de rôle:</b>		<b>BidirectionalRoad</b>
<b>Description:</b>	Créer sa voie droite, Créer sa voie gauche, détecter la liste de Véhicules qui le traverse.	
<b>Protocoles et Activités:</b>	CreateLeftLane, CreateRightLane, listVehicles	
<b>Permissions:</b>	<i>Lire</i> vehicleInformation <i>Changer</i> LaneInformation	
<b>Responsabilités</b>		
<b>Vivacité:</b>	Road = (CreateRightLane.CreateLeftLane) BidirectionalRoad = listVehicles	
<b>Sécurité:</b>	listVehicles = VehicleOnRoad	

### 3.2.4 Protocole d'interaction

Pour notre projet, nous avons conçu un protocole dont le but est de déterminer le voisinage d'un véhicule, ce dans le but de faciliter la tâche de perception de l'agent. Ce protocole est initié par un véhicule en vue d'établir sa liste de voisins. La figure 3.12 illustre la procédure d'opération du protocole.

#### 3.2.4.1 Étape de découverte

Le processus débute par une étape de découverte. Dans cette étape, le véhicule envoie un message Hello à l'agent protocole. Ce Hello est formé par les champs suivants :

- Header : identifiant du véhicule et son type
- option : Elle comprends des champs qui décrivent l'emplacement du véhicule (les identifiants de la route bidirectionnelle et de la voie où il se trouve, les identifiants de la route



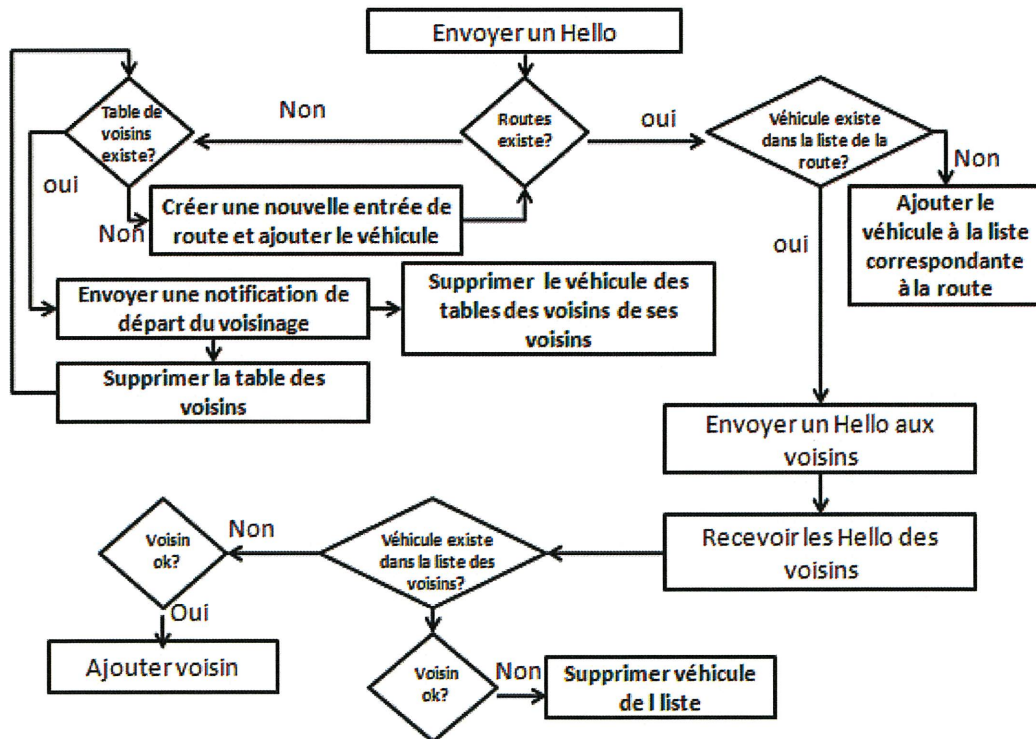


FIGURE 3.12: Protocole de découverte de Voisinage

et de la voie qu'il peut voir.

- position : les coordonnées du centre de gravité du véhicule

Une fois le message Hello est généré, le protocole se charge d'assigner le véhicule à son couple (route/voie) correspondant.

En fin de cette étape, le protocole forward le message Hello vers les potentiels voisins du véhicule. Nous définissons quatre types de potentiels voisins :

- 1- Un véhicule se trouvant sur la même route et sur la même voie
- 2- Un véhicule se trouvant sur la même route mais sur des voies différents
- 3- Un véhicule se trouvant sur des routes différentes mais sur le même type de voie
- 4- Un véhicule se trouvant sur des routes différentes et des voies différentes

Une fois la liste des potentiels voisins établie, le véhicule reçoit les messages Hello des véhicules de la liste.

### 3.2.4.2 Étape de génération de voisinage

Après avoir tester si un potentiel voisin peut faire partie du voisinage (généralement, dépendant de la distance le séparant du véhicule qui a initié le protocole), celui-ci est ajouté au graphe de voisinage. Un graphe de voisinage est défini comme un ensemble d'arcs reliant chacun deux nœuds (nom d'un véhicule) et poids formé par le triplet (*distance, sameRoad, sameLane*) où *sameRoad* = 0 si les deux véhicules appartiennent à la même route et 1, sinon, et *sameLane* = 0 si les deux véhicules appartiennent à la même route et 1 sinon.

A chaque intervalle de temps, le véhicule envoie un Hello à ses voisins. Ceux-ci vérifient si le véhicule est encore conforme à sa définition de voisins, sinon il le supprime de sa liste.

### 3.2.4.3 Étape de libération de voisins

Si un véhicule quitte une route il envoie un message de "Départ" à ses voisins les informant qu'il change de route. Suite à la réception de ce message, chaque véhicule récepteur supprime le véhicule qui a envoyé le message de sa liste de voisins, si celui-ci n'est plus conforme à sa définition de voisin. Enfin, ce dernier libère sa liste de voisins et met à jour sa liste de voisinage.

## Conclusion

Dans ce chapitre, nous avons présenté le simulateur YATS. Dans le processus, nous avons détaillé son architecture de départ et la limite de celle-ci. Après, nous avons présenté notre contribution à YATS, à savoir une nouvelle architecture et un nouveau modèle d'agent.



# Chapitre 4

## Réalisation

### Introduction

Ce chapitre vise à présenter dans un premier lieu, l'environnement matériel et logiciel utilisé, pour exposer enfin un aperçu sur le travail réalisé à travers quelques captures d'écrans commentées.

### 4.1 Environnement de Développement

- Processor Intel®*Core*<sup>TM</sup> i3-2670QM CPU @2.20Ghz 2,20 Ghz
- RAM 4.00 Go
- Operating System : Windows 7 64bits

### 4.2 Outils de développement

#### 4.2.1 Langage C++

C++ est un langage de programmation compilé, permettant la programmation sous de multiples paradigmes comme la programmation procédurale, la programmation orientée objet et la programmation générique. Le langage C++ n'appartient à personne et par conséquent n'importe qui peut l'utiliser sans besoin d'une autorisation ou obligation de payer pour avoir le droit d'utilisation. C++ est l'un des langages de programmation les plus populaires, avec une grande variété de plates-formes matérielles et de systèmes d'exploitation.

Le langage C++ est normalisé par l'ISO. Sa première normalisation date de 1998 (ISO/CEI

14882 :1998). Le standard a ensuite été amendé par l'erratum technique de 2003 ISO/CEI 14882 :2003. Le standard actuel a été ratifié et publié par l'ISO en septembre 2011 sous le nom de ISO/IEC 14882 :2011 (aussi appelé C++11)<sup>2</sup>. Une mise à jour mineure a été publiée en 2014 sous le nom de ISO/CEI 14882 :2014. Le prochain standard devrait être publié en 2017<sup>3</sup>.

En langage C, ++ est l'opérateur d'incrément, c'est-à-dire l'augmentation de la valeur d'une variable de 1. C'est pourquoi C++ porte ce nom : cela signifie que C++ est un niveau au-dessus de C. Il existe de nombreuses bibliothèques C++ en plus de la bibliothèque standard de C++ (C++ Standard Library) qui est incluse dans la norme. Par ailleurs, C++ permet l'utilisation de l'ensemble des bibliothèques C existantes.

### 4.2.2 Environnements de développement intégré « IDE » : Visual Studio 2015 Community

Visual C++ est un environnement de développement intégré pour Windows, conçu par Microsoft pour les langages de programmation C et C++ et intégrant différents outils pour développer, compiler, déboguer un programme en C++ s'exécutant sur Windows, ainsi que des bibliothèques comme les MFC.

Il a par la suite été intégré au framework Visual Studio, qui constitue ainsi un cadre unique aux divers environnements de développements de Microsoft. Le terme de Visual C++ est toutefois toujours employé pour désigner l'ensemble constitué par Visual Studio configuré pour C et C++.

#### 4.2.2.1 Bibliothèque SDL

C'est une bibliothèque libre et gratuite écrite en C. C'est une bibliothèque bas niveau, disposant de fonctions très basiques. Comme les fonctions restent basiques, elles sont très rapides. Simple DirectMedia Layer (SDL) est une Bibliothèque logicielle. Son API est utilisée pour créer des applications multimédias en deux dimensions pouvant comprendre du son comme les jeux vidéo, les démos graphiques, les émulateurs, etc. Sa portabilité sur la plupart des plateformes et sa licence zlib, très permissive contribuent à son succès. OpenClassrooms la conseille aux programmeurs débutants<sup>2</sup> pour leur entrée dans le monde de la programmation multimédia.

Cette bibliothèque est disponible sous Windows, Windows CE, BeOS, Mac OS, et la plupart des systèmes UNIX comme Mac OS X, Linux, Android (qui est basé sur Linux), FreeBSD, NetBSD, OpenBSD, BSD/OS, Solaris, IRIX, et QNX. Elle fonctionne aussi sous quelques sys-

tèmes embarqués comme des consoles de jeu portable.

Écrite en C, elle est aussi utilisable depuis d'autres langages de programmation, comme Ada, C/C++, C# (framework .net), Ch, D, Eiffel, Erlang, Euphoria, Guile, Genie, Haskell, OCaml, Java, le SDK Android (Java), Lisp, Lua, ML, Objective C, Pascal, Perl, PHP, Pike, Pliant, Python (grâce à Pygame), Ruby, Smalltalk et Vala.

### 4.3 Simulation

Dans cette section, nous allons présenter un scénario d'exécution comportant deux véhicules se déplaçant sur une seule route bidirectionnelle. Comme notre projet, se dirige dans le cadre d'un projet appelé projet PROFIL (Profil de la Route comme Facteur d'Influence de la Lisibilité de l'infrastructure et son effet sur la trajectoire) dont l'objet d'étude est l'autoroute de Rouen formée par une seule route bidirectionnelle, la simulation répond bien aux besoins.

#### 4.3.1 Début de la simulation

La route étant formée par deux voies (deux instances de la classe Road dont la source se charge de générer les véhicules), deux véhicules seront générés, une par voie (figure 4.1).

Au début, chaque véhicule va envoyer son Hello (figure 4.2). A ce stade, le véhicule ne peut voir que le segment de route où il se localise.

A ce niveau, les véhicules se trouvent sur des segments loin l'un de l'autre. D'où l'absence de voisins, pour chacun d'eux. D'où, les tables des voisins des deux sont vides.

En s'approchant de la fin de segment (figure 4.3) (distance jusqu'au point de sortie est trois fois la distance d'arrêt), le segment de route suivant est visible.

Les champs correspondants au segment suivant sont remplis est les véhicules envoies ses nouvelles données (figure 4.4). Toutefois, il n'y pas de véhicule pour recevoir le Hello, d'où le voisinage est encore nul.

Quand le véhicule 0 :0 s'apprête à entrée dans le segment, le véhicule 0 :9 commence à le percevoir comme potentiel voisin. Les deux véhicules se repère l'une l'autre jusqu'à ce que l'une d'entre eux quitte la route qu'il partage (figure 4.5).

Durant ce temps, le véhicule 9 :0 considère le véhicule 0 :0 comme voisin (et vice-versa). Les deux véhicules se trouvent sur la même route mais dans des voies différents (figure 4.6).

Quand le véhicule 9 :0 atteint sa destination il est supprimer et la simulation continu jusqu'à



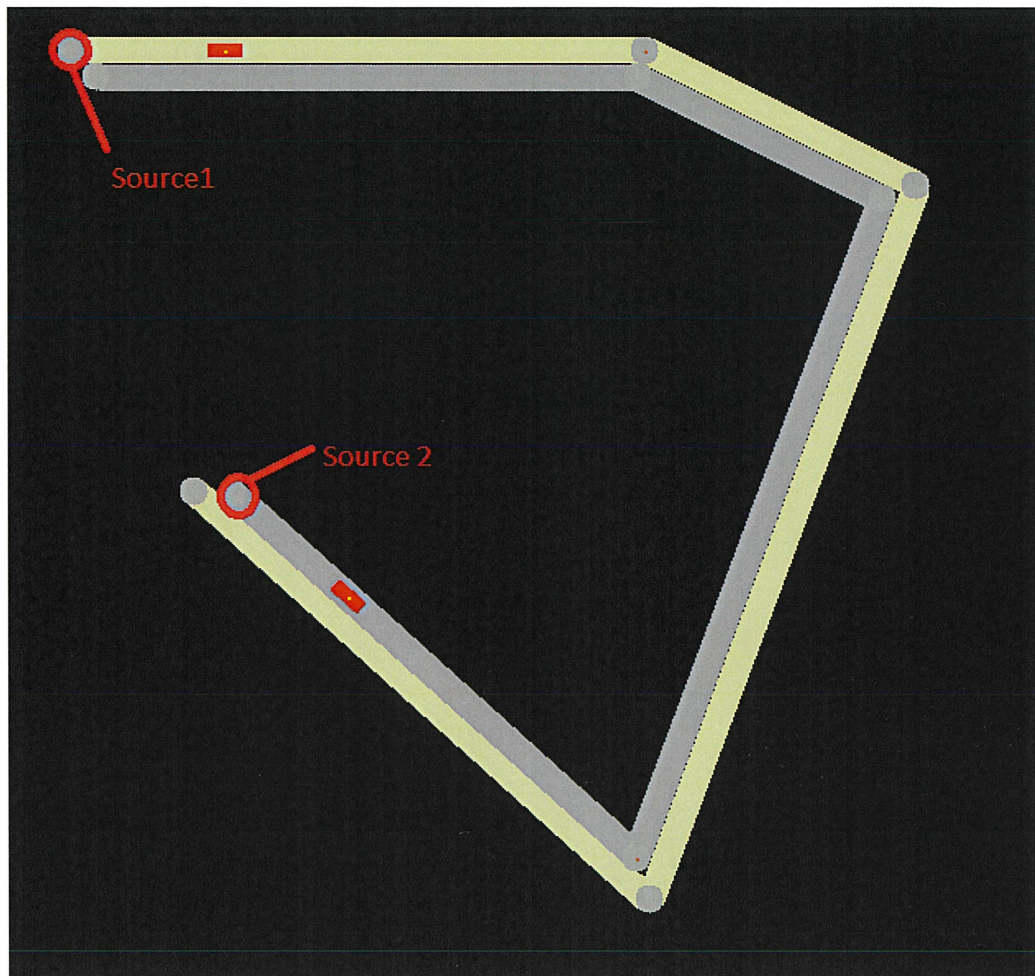


FIGURE 4.1: Début de la simulation

```

logHello.txt - Bloc-notes
Fichier  Edition  Format  Affichage ?
hello i am VEHICLE 0:0 and i am a my position is 96.2916 ; 96.2716 i am at R001RightLane lane R001 i can see
hello i am VEHICLE 9:0 and i am a my position is 229.034 ; 465.107 i am at R004LeftLane lane R004 i can see
hello i am VEHICLE 0:0 and i am a my position is 96.3316 ; 96.2716 i am at R001RightLane lane R001 i can see
hello i am VEHICLE 0:0 and i am a my position is 229.683 ; 465.134 i am at R004LeftLane lane R004 i can see
hello i am VEHICLE 0:0 and i am a my position is 96.3916 ; 96.2716 i am at R001RightLane lane R001 i can see
hello i am VEHICLE 9:0 and i am a my position is 229.728 ; 465.174 i am at R004LeftLane lane R004 i can see
hello i am VEHICLE 0:0 and i am a my position is 96.4716 ; 96.2716 i am at R001RightLane lane R001 i can see
hello i am VEHICLE 9:0 and i am a my position is 229.787 ; 465.228 i am at R004LeftLane lane R004 i can see
hello i am VEHICLE 0:0 and i am a my position is 96.5716 ; 96.2716 i am at R001RightLane lane R001 i can see
hello i am VEHICLE 9:0 and i am a my position is 229.862 ; 465.294 i am at R004LeftLane lane R004 i can see
hello i am VEHICLE 0:0 and i am a my position is 96.6916 ; 96.2716 i am at R001RightLane lane R001 i can see
hello i am VEHICLE 9:0 and i am a my position is 229.951 ; 465.375 i am at R004LeftLane lane R004 i can see
hello i am VEHICLE 0:0 and i am a my position is 96.8316 ; 96.2716 i am at R001RightLane lane R001 i can see
hello i am VEHICLE 9:0 and i am a my position is 230.055 ; 465.468 i am at R004LeftLane lane R004 i can see
hello i am VEHICLE 0:0 and i am a my position is 96.9916 ; 96.2716 i am at R001RightLane lane R001 i can see
hello i am VEHICLE 9:0 and i am a my position is 230.174 ; 465.575 i am at R004LeftLane lane R004 i can see
hello i am VEHICLE 0:0 and i am a my position is 97.1716 ; 96.2716 i am at R001RightLane lane R001 i can see
hello i am VEHICLE 9:0 and i am a my position is 230.308 ; 465.696 i am at R004LeftLane lane R004 i can see
hello i am VEHICLE 0:0 and i am a my position is 97.3716 ; 96.2716 i am at R001RightLane lane R001 i can see
hello i am VEHICLE 9:0 and i am a my position is 230.456 ; 465.83 i am at R004LeftLane lane R004 i can see
    
```

FIGURE 4.2: Première vague de Hello envoyée



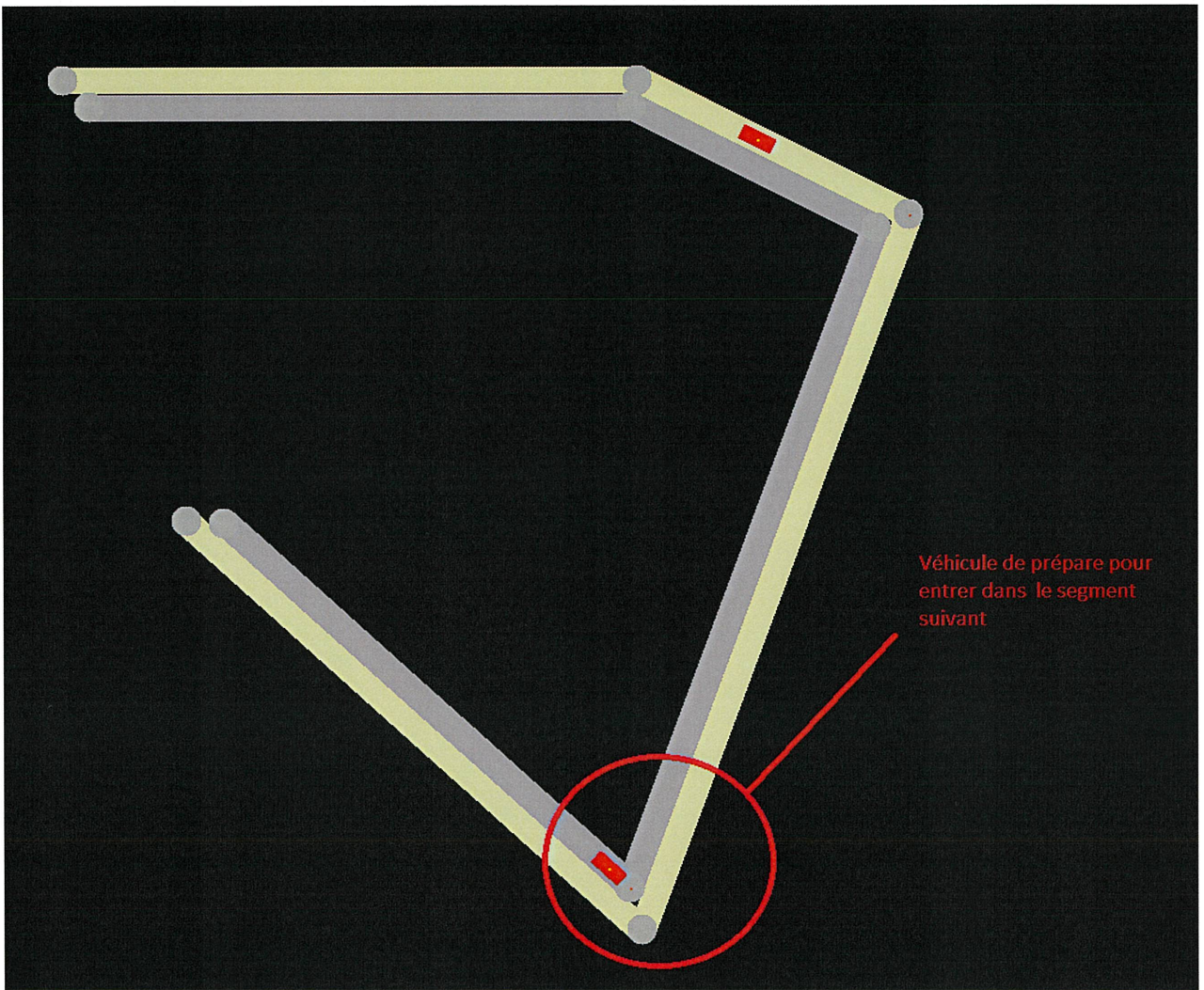


FIGURE 4.3: Changement de segment

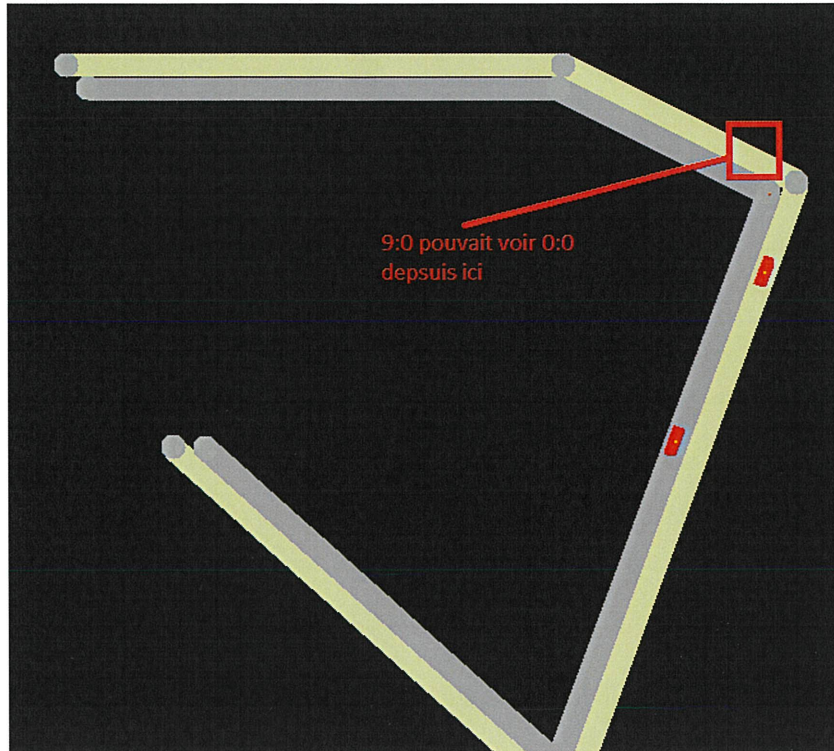
FIGURE 4.4: Hello du changement de segment de route

```

hello i am VEHICLE 9:0 and i am a 1 my position is 349.4 ; 132.72 i am at R004LeftLane lane R004 i can see R003
hello i am VEHICLE 0:0 and i am a 1 my position is 679.82 ; 148.299 i am at R002RightLane lane R002 i can see R003
hello i am VEHICLE 9:0 and i am a 1 my position is 549.553 ; 753.045 i am at R003LeftLane lane R003 i can see R002
hello i am VEHICLE 0:0 and i am a 1 my position is 681.798 ; 149.307 i am at R002RightLane lane R002 i can see R003
hello i am VEHICLE 9:0 and i am a 1 my position is 549.925 ; 753.379 i am at R003LeftLane lane R003 i can see R002
hello i am VEHICLE 0:0 and i am a 1 my position is 683.758 ; 150.305 i am at R002RightLane lane R002 i can see R003
hello i am VEHICLE 9:0 and i am a 1 my position is 550.319 ; 753.719 i am at R003LeftLane lane R003 i can see R002
hello i am VEHICLE 0:0 and i am a 1 my position is 685.701 ; 151.295 i am at R002RightLane lane R002 i can see R003
hello i am VEHICLE 9:0 and i am a 1 my position is 550.738 ; 754.059 i am at R003LeftLane lane R003 i can see R002
hello i am VEHICLE 0:0 and i am a 1 my position is 687.625 ; 152.276 i am at R002RightLane lane R002 i can see R003
hello i am VEHICLE 9:0 and i am a 1 my position is 551.188 ; 754.393 i am at R003LeftLane lane R003 i can see R002
hello i am VEHICLE 0:0 and i am a 1 my position is 689.532 ; 153.247 i am at R002RightLane lane R002 i can see R003
hello i am VEHICLE 9:0 and i am a 1 my position is 551.671 ; 754.713 i am at R003LeftLane lane R003 i can see R002
hello i am VEHICLE 0:0 and i am a 1 my position is 691.474 ; 154.211 i am at R002RightLane lane R002 i can see R003
    
```



FIGURE 4.5: Voisin visible



logNeighbors.txt - Bloc-notes

Fichier	Edition	Format	Affichage ?				
VEHICLE 9:0	VEHICLE 0:0	R003	136.366	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	141.562	1	0	1	
VEHICLE 9:0	VEHICLE 0:0	R003	146.76	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	151.962	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	157.165	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	162.372	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	167.579	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	172.79	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	178.001	1	0		
VEHICLE 9:0	VEHICLE 0:0			0	0		
VEHICLE 9:0	VEHICLE 0:0			0	0		
VEHICLE 9:0	VEHICLE 0:0			0	0		
VEHICLE 9:0	VEHICLE 0:0	R003	204.091	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	209.272	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	214.417	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	219.52	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	224.586	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	229.61	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	234.598	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	239.544	1	0		
VEHICLE 9:0	VEHICLE 0:0	R003	244.453	1	0		

de véhicule 0:0 est un voisin

Il est sur la même route mais pas la même voie

FIGURE 4.6: Voisin dans la liste

ce que 0 :0 atteint sa destination, aussi (figure ).

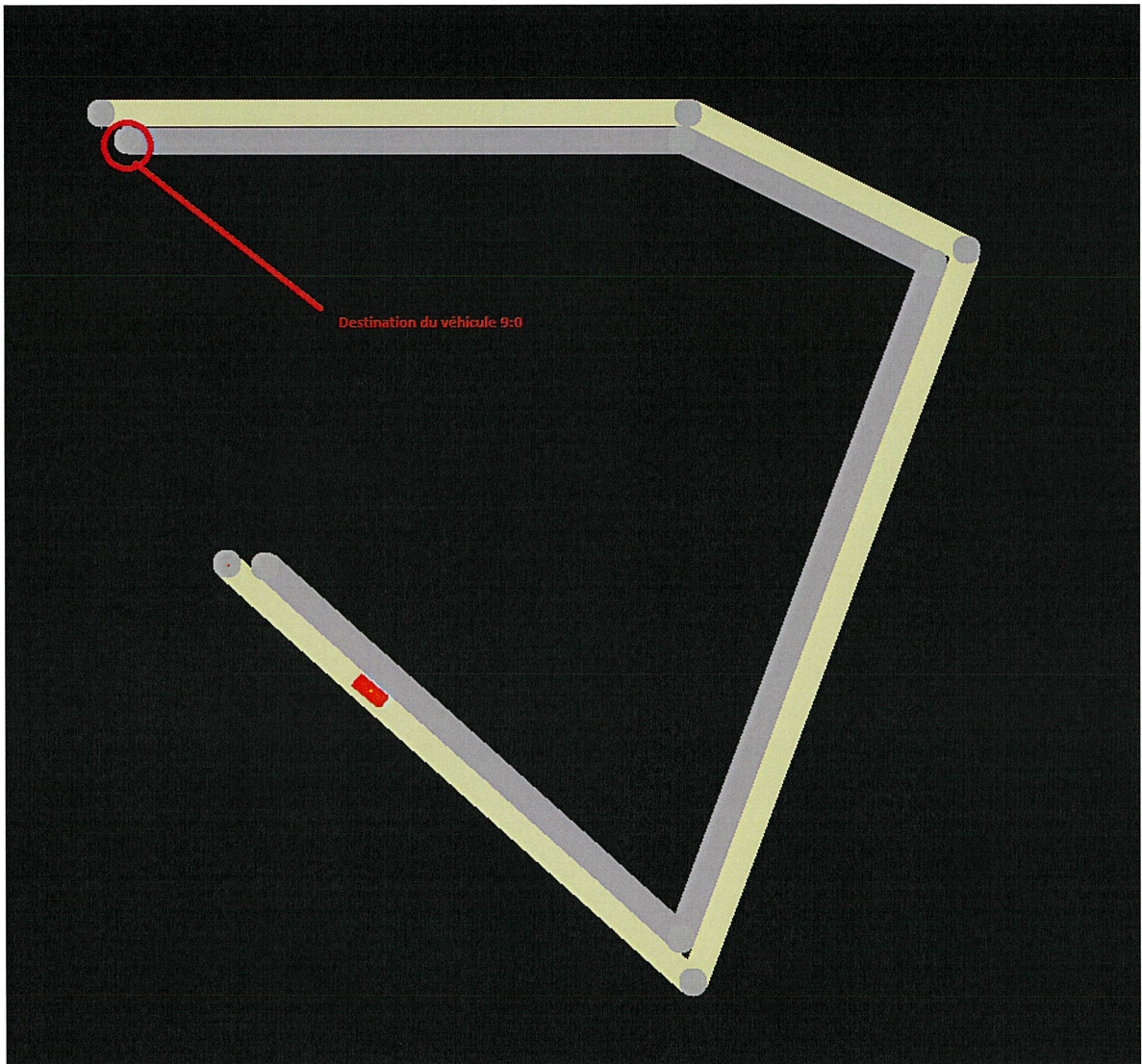


FIGURE 4.7: Véhicule à destination

## Conclusion

Nous venons de préciser l'environnement de développement de notre projet et les choix technologiques. Enfin, nous avons présenté des captures-écran illustrant le déroulement d'un simple scénario d'exécution.



# General Conclusion

L'objectif de notre projet est de réaliser un modèle multi-agents qui permet de simuler le comportement latéral des véhicules. Ceci implique d'ajouter une modalité qui se charge de contrôler le positionnement latéral du véhicule.

Le présent est le rapport de notre stage de quatre mois effectué à l'IFSTTAR sous le tutorat de l'INSA Rouen. Dans le premier chapitre nous avons commencé par présenter le principe de simulation de trafic. Ceci nous a permis de découvrir les deux principales méthodologies de simulation. La première est l'approche macroscopique, généralement, dédiée à l'étude des grands réseaux routiers. La deuxième, est l'approche microscopique, permet l'étude des influents locaux de trafic qui guide la décision d'un conducteur de se déplacer dans la route (avancer, changer de voie, accélérer, décélérer, etc).

Le deuxième chapitre servait de guide pour les systèmes multi-agents et leurs contributions au domaine de l'informatique généralement, et de la simulation de trafic en particulier.

Le troisième chapitre était une présentation du simulateur YATS (Yet Another Traffic Simulator), un simulateur que nous sommes en train de développer au sein de l'IFSTTAR. Toutefois, son architecture posait une contrainte importante du point de vue logiciel. En effet, la modalité de ce dernier était faible : nous trouvons des blocs de code assez longs qui se résolvent autour du SDL. Cette tâche générait de nombreuses bugs causés par les modifications apportées. De ce fait, il était primordial de changer l'architecture logicielle pour faciliter la tâche ensuite de concevoir notre agent. Par la suite, nous avons présenté la méthodologie Gaia pour aider à concevoir notre modèle d'agent. Nous avons conclu notre chapitre par la présentation du modèle d'agent et du premier protocole d'interaction.

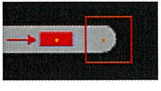
Quant au quatrième chapitre il servait de plateforme pour présenter l'avancement de la tâche de développement de la solution, en prenant l'exemple simple de deux véhicules dans le réseau. Pour conclure, les perspectives ne manquent pas à notre projet, débutant par l'optimisation jusqu'à la conception de processus de décision plus élaborés.

# Annexe

# Description des classes d'origine de YATS

Classe	Rôle	Image
Road	La Classe qui porte le nom "Road" est en fait un <i>segment de route</i> . Le réseau routier est donc composé d'un assemblage de segment de route. Cet objet possède un point en entrée et un point en sortie. Sur un segment de route un véhicule ne pourra aller que dans un seul et unique sens. Il possède d'autres attributs utiles pour d'autres objets comme l'angle par rapport à l'axe des abscisses ou la longueur.	
NodeSim	Le Nœud est ce qui relie les segments de route entre eux. Il n'a donc qu'un seul point mais peut avoir plusieurs entrées ou sorties. Le Nœud héberge aussi une liste de RoadLink, que nous verrons juste après. Cet objet possède aussi une fonction qui permet de faire un cercle plein de couleur grise afin de combler l'espace entre les segments de route, comme montré sur l'image ci-contre.	
RoadLink	Le RoadLink, comme son nom l'indique, lie deux segments de route ensemble. Il fait partie du nœud mais est à l'usage du véhicule. Il contient des informations essentielles à la décision comme l'angle entre deux segments de route, la vitesse maximale conseillée pour effectuer le virage en toute sécurité ou encore le gap à partir duquel il faudra entamer un virage. C'est principalement pour cette classe que la notion d'afforceance a été dominante.	
SourceSim	La Source propulse les véhicules sur le réseau routier en fonction d'un certain nombre de paramètres (fréquence, vitesse, intervalle, etc...). Elle correspond au point d'entrée de nos agents dans l'environnement.	



Classe	Rôle	Image
Le puit	Le puit est un objet qui n'existe par réellement dans la simulation, seulement le concept est indispensable. Lorsqu'un véhicule croise un puit il disparaît du réseau routier.	

# Bibliographie

- [1] M. H. Lighthill, G.B. Whitham, *On kinematic waves. ii. a theory of traffic flow on long crowded roads*, Proceedings of The Royal Society of London, 229 :317-345, 1955.
- [2] P.I. Richards, *Shock waves on the highway*, Operations Research, 4 :42-51, 1956
- [3] J.Ferber. *Les systèmes multi-agents ; Inter-Editions*, 1995.
- [4] F.Bellifemine, G.Caire, D.Greenwood, *Developing Multi-Agent Systems with Jade*, 2007.
- [5] R.H.Bordini, J.F.Hübner, M.Wooldridge, *Programming Multi-Agent Systems in AgentSpeak using Jason*, 2007.
- [6] S.Davier, *Etude du comportement humain grâce à la simulation multi-agents et aux méthodes de fouille de données temporelles*, 2009.
- [7] J.Ferber, *Les systèmes multi-Agents : vers une intelligence collective*, InterEditions, 1995.
- [8] G.Weiss, *Multiagent Systems : A Modern Approach to Distributed Artificial Intelligence*, 1991.
- [9] Dorigo, M. and Gambardella, L. M. (1997), *Ant colony system : A cooperative learning approach to the traveling salesman problem*, Evolutionary Computation, IEEE Transactions on, 1(1) :53-66.
- [10] Minsky, M, *Matter, mind and models*, 1965
- [11] Galán, J., Izquierdo, L., Izquierdo, S., Santos, J., Del Olmo, R., López-Paredes, A., and Edmonds, B, *Errors and artefacts in agent-based modelling*, Journal of Artificial Societies and Social Simulation, 12(1) :1, 209
- [12] A.Gomez, *Se repérer dans l'espace pour mémorier : Approche neuromatique du lien entre la mémoire et l'espace*, 2009
- [13] S. El Hadouaj, *Conception de comportements de résolution de conflits et de coordination : Application à une simulation multi-agent du trafic routier*. PhD thesis, Thèse de Doctorat de l'Université de Paris 6, France, 2004

- [14] S. Kikuchi, P. Chakroborty, *Car-following model based on fuzzy inference system*, 1992
- [15] S. Espié, F. Saad, B. Schnetzler, F. Bourlier, N. Djemane, *Microscopic traffic simulation and driver behaviour modelling : the archisim project*. Road Safety in Europe and Strategic Highway Research Program, , 1994
- [16] F. Saad, B. Schnetzler, *Un modèle conceptuel du comportement des conducteurs en conduite en file sur autoroute*, Technical report, INRETS, Convention INRETS / DRAST, 1994
- [17] A. Doniec, *Prise en compte des comportements anticipatifs dans la coordination multi-agent : application à la simulation de trafic en carrefour*, PhD thesis, Université de Valenciennes et du Hainaut-Cambrésis, 2006
- [18] J.Gagnieux, *YATS : Yet Another Traffic Simulator*, rapport de stage M2, 2013
- [19] M. Wooldrige, N.r. Jennings, D. Kinny, *The Gaia Methodology for Agent-Oriented Analysis and Design*, Journal of Autonomous Agents and Multi-Agent Systems, 2000