

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE  
UNIVERSITE AKLI MOAND OULHADJE-BOUIRA



Faculté des Sciences et des Sciences Appliquées  
Département d'informatique

**Mémoire de fin d'étude**

Présenté par :

**ASSAM Amar**

**MOKDAD Mohamed dhia eddin**

En vue de l'obtention du diplôme de **Master 02** en :

Filière : **MATH ET INFORMATIQUE**

Option : **Ingénierie des Systèmes d'Information et de Logiciel**

**Thème :**

**Une approche multi-agents pour la simulation de la  
mobilité urbaine et de la communication V2X**

**Devant le jury composé de :**

MAHFOUD Zohra	Grade	UAMOB	Président
MESSAOUDI Oussama	Grade	UAMOB	Encadreur
DEMMOUCHE Mouloud	Grade	UAMOB	Examineur
FELLAHI	Grade	UAMOB	Examineur

Année Universitaire 2018/2019

## ***Remerciements***

*Nos remerciements vont en premier lieu notre encadreur Mr. Oussama MESSAOUDI pour la préciosité de ses conseils, son infinie disponibilité et son orientation qui ont constitué un apport considérable grâce auquel ce travail a pu être mené à bon port.*

*Nous tenons à remercier chacun des membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.*

*Nos remerciements s'étendent à tous nos enseignants et les membres du département d'informatique de l'université AKLI MOHAND OULHADJE BOUIRA*

*Ainsi qu'à tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail.*

## ***Dédicaces***

*A nos parents, pour leurs sacrifices déployés à notre égard, pour leur patience, leur amour et leur confiance. Qu'ils trouvent dans ce modeste travail, le témoignage de notre profonde affection et de notre attachement indéfectible ; nulle dédicace ne puisse exprimer ce qu'on leur doit.*

*A nos frères et sœurs et tous nos amis pour chaque mot reçu, chaque geste d'amitié, à chaque main tendue et pour toute attention témoignée.*

## ***Résumé***

Aujourd'hui Les réseaux véhiculaires représentent un intérêt certain pour l'industrie automobile, les opérateurs des réseaux, les organisations et même les particuliers. Le but des réseaux véhiculaire est d'améliorer la sécurité routière et l'efficacité des transports routiers afin de diminuer les accidents et fournir un environnement confortable aux conducteurs et à leurs passagers.

Le but de ce projet est de modéliser et de simuler la tâche de conduite ainsi que les interactions et la communication entre les véhicules et aussi entre les véhicules et les nœuds de communication de l'infrastructure. Cela permet de produire et de fournir un environnement de simulation fiable, réaliste, rentable et plus sûre.

## **ملخص**

تمثل شبكات السيارات اليوم مصلحة محددة لصناعة السيارات ومشغلي الشبكات والمنظمات وحتى الأفراد. هدف شبكات المركبات هو تحسين السلامة على الطرق وكفاءة النقل البري من أجل تقليل الحوادث وتوفير بيئة مريحة للسائقين والركاب.

الغرض من هذا المشروع هو نمذجة ومحاكاة مهمة القيادة وكذلك التفاعلات والتواصل بين المركبات وأيضاً بين المركبات وعقد الاتصال الخاصة بالبنية التحتية. وهذا يجعل من الممكن إنتاج وتوفير بيئة محاكاة موثوقة وواقعية وفعالة من حيث التكلفة وأكثر أماناً.

## ***Abstract***

Today Vehicular networks represent a definite interest for the automotive industry, network operators, organizations and even individuals. The goal of vehicular networks is to improve road safety and the efficiency of road transport in order to reduce accidents and provide a comfortable environment for drivers and their passengers.

The purpose of this project is to model and simulate the driving task as well as the interactions and communication between the vehicles and also between the vehicles and the communication nodes of the infrastructure. This makes it possible to produce and provide a reliable, realistic, cost-effective and safer simulation environmen

# Table des matières

<b>Table des matières</b> .....	<b>I</b>
<b>Liste des figures</b> .....	<b>IV</b>
<b>Liste des tables</b> .....	<b>V</b>
<b>Liste des abréviations</b> .....	<b>VI</b>
<b>Introduction générale</b> .....	<b>1</b>
<b>I. Modélisation et simulation de la mobilité urbaine</b> .....	<b>3</b>
I.1. Introduction .....	3
I.2. Approche de modélisation .....	3
I.2.1. Modèles microscopiques .....	3
I.2.2. Modèles macroscopiques .....	5
I.2.3. Modèles mésoscopiques .....	6
I.3. Modélisation microscopique de la tâche de conduite .....	6
I.3.1. Modèle de changement de voie .....	7
I.3.2. Modèle de suivi de véhicules .....	7
I.3.2.1. Modélisation basée sur les propriétés de la mobilité urbaine .....	8
I.3.2.2. Modélisation basée sur le comportement de conducteur humain .....	9
I.3.3. Modélisation du contrôle longitudinal .....	9
I.4. Communication v2x .....	12
I.4.1. Les réseaux véhiculaires .....	13
I.4.2. Les réseaux VANETs .....	13
I.4.2.1. Définition d'un Réseaux VANETs .....	13
I.4.2.2. Composant d'un réseaux VANETs .....	13
I.4.3. Architectures des réseaux véhiculaires .....	14
I.4.3.1. Communications de Véhicule à Véhicule (V2V) .....	14
I.4.3.2. Communications de Véhicule à Infrastructure (V2I) .....	14
I.4.3.3. Communication hybride .....	14
I.4.4. Les Applications des réseaux VANETs .....	15
I.4.4.1. Applications de gestion du trafic routier .....	15
I.4.4.2. Applications de confort .....	15
I.4.4.3. Applications de sécurité du trafic routier .....	15

I.4.5.	Simulateurs.....	15
I.4.5.1.	La Simulation dans un Réseau VANET.....	15
I.4.5.2.	Les Simulateurs de Réseaux Ad Hoc.....	16
I.4.5.3.	Les Simulateurs de mobilité et de trafic routier.....	16
I.5.	Conclusion.....	17
<b>II.</b>	<b>Les Systèmes Multi-agents.....</b>	<b>18</b>
II.1.	Introduction.....	18
II.2.	Introduction aux systèmes multi-agents.....	18
II.2.1.	Définition du concept d'agent.....	18
II.2.2.	Caractéristiques d'un agent.....	18
II.2.3.	Système multi-agents.....	18
II.2.4.	Domaines d'application des SMA.....	20
II.2.5.	Quelques exemples de SMA.....	20
II.3.	Conclusion.....	21
<b>III.</b>	<b>Une approche multi-agent pour la simulation de la mobilité.....</b>	<b>22</b>
III.1.	Introduction.....	22
III.2.	SUMO : vers une simulation microscopique à base d'agents.....	22
III.2.1.	Quel type de simulation de trafic est SUMO ?.....	23
III.2.2.	Exemples d'utilisation :.....	23
III.2.3.	Applications incluses :.....	24
III.2.4.	Critères de conception du logiciel :.....	24
III.2.5.	Quelques Contributeurs et participants.....	25
III.2.6.	Notions de base :.....	25
III.2.6.1.	XML exemple :.....	25
III.2.6.2.	Types de données référencés :.....	25
III.2.6.3.	Types de fichiers référencés.....	26
III.2.6.4.	Description de l'utilisation.....	26
III.2.7.	Les options.....	26
III.2.7.1.	Table de « Configuration ».....	26
III.2.7.2.	Table de « Input ».....	27
III.2.7.1.	Table de « Output ».....	27
III.2.7.2.	Table de « Time ».....	28
III.2.7.3.	Table de « Processing ».....	28

III.2.7.4. Table de « Traci server » .....	28
III.2.7.5. Table da « Gui only » .....	29
III.2.7.1. Table de « Batterie ».....	29
III.2.8. Traci .....	30
III.2.8.1. Introduction à TraCI : .....	30
III.2.8.2. Utiliser TraCI :.....	30
III.2.8.3. Plusieurs clients : .....	30
III.2.8.4. Utiliser SUMO comme une bibliothèque .....	30
III.2.8.5. Exemple de description .....	31
III.3. SiMMU : Simulation Microscopique de la Mobilité Urbaine .....	31
III.4. Conclusion .....	34
<b>IV. Expérimentation et résultats .....</b>	<b>35</b>
IV.1. Introduction .....	35
IV.2. Environnement de travail.....	35
V.2.1. Environnement matériel .....	35
IV.2.1. Environnement logiciel .....	35
IV.2.1.1. Eclipse .....	35
IV.2.1.2. Java .....	35
IV.2.1.3. Jade .....	36
IV.2.1.4. SUMO (Simulation of Urban MObility) .....	36
IV.3. Création des scénarios avec SUMO .....	36
IV.4. Présentation de simulateur.....	39
IV.4.1. Communication vers SUMO .....	39
IV.4.2. Interface graphique pour la simulation.....	39
IV.4.3. Lancer la simulation avec SUMO .....	40
IV.5. Exemple des modèles de suivi de véhicules .....	42
IV.6. Comparaison des résultats entre les trois modèles .....	46
IV.7. Conclusion .....	48
<b>Conclusion générale .....</b>	<b>49</b>
<b>Bibliographie.....</b>	<b>50</b>

# Liste des figures

Figure I-1 la notation microscopique [1].	4
Figure I-2 Illustration de la variable macroscopique « le débit » [1].	5
Figure I-3 Illustration de la variable macroscopique « la densité » [1].	5
Figure I-4 Illustration de la vitesse de flot [1].	6
Figure I-5 Exemple de la situation dangereuse de changement de voie.	7
Figure I-6 Exemple de réseaux véhiculaire [26].	13
Figure I-7 communication véhicule à véhicule	14
Figure I-8 Communication véhicule à Infrastructure	14
Figure I-9 communication hybride.	14
Figure I-10 L'interface graphique du SUMO	17
Figure III-1 Sumo	23
Figure III-2 Exemple.	31
Figure IV-1 Nod.XML	36
Figure IV-2 Edge.XML	37
Figure IV-3 net.cfg	37
Figure IV-4 route file	38
Figure IV-5 type file	38
Figure IV-6 sumo.cfg.	38
Figure IV-7 la communication d'un SMA avec SUMO	39
Figure IV-8 interface graphique.	39
Figure IV-9 charger un fichier cfg	40
Figure IV-10 lancer la simulation	41
Figure IV-11 locate véhicule.	42
Figure IV-12 position véhicule	42
Figure IV-13 Modèle de Kerner[19]	43
Figure IV-14 Modèle de Krauß [13]	45
Figure IV-15 Le modèle utilisé par défaut dans SUMO [17]	46
Figure IV-16 paramètre de chaque modèle.	46
Figure IV-17 résultat de changement de vitesse	47
Figure IV-18 résultat de changement la vitesse de leader	47
Figure IV-19 résultat de changement de la distance	47
Figure IV-20 résultat de changement de l'accélération appliquée à tout instant.	48

# Liste des tables

Tableau III-1 Le paquet comprend .....	24
Tableau III-2 suite des application .....	24
Tableau III-3 quelques participants .....	25
Tableau III-4 SUMO configuration [40].....	26
Tableau III-5 Input[40] .....	27
Tableau III-6 Output[40] .....	27
Tableau III-7 Time[40] .....	28
Tableau III-8 processing[40].....	28
Tableau III-9 Traci Server[40].....	28
Tableau III-10 Gui only[40] .....	29
Tableau III-11 batterie[40].....	29
Tableau III-12 Composants du systèmes proposé.....	32

# Liste des abréviations

<b>CFM</b>	<b>Modèle De Suivi Véhicule</b>
<b>IDM</b>	<b>Intelligent Model Driver</b>
<b>LCA</b>	<b>Lane Change Assistance</b>
<b>LCM</b>	<b>Modèle Changement De Voie</b>
<b>MANET</b>	<b>Mobile Ad Hoc NETwork</b>
<b>NetSA</b>	<b>Networked Software Agents</b>
<b>Ns2</b>	<b>Network Simulator</b>
<b>OBU</b>	<b>On Board Unite</b>
<b>OTcl</b>	<b>Object Tools Command Language</b>
<b>RSU</b>	<b>Road Side Units</b>
<b>SUMO</b>	<b>Simulation For Urban Mobility</b>
<b>SMA</b>	<b>System Multi Agent</b>
<b>TA</b>	<b>Traffic Announcement</b>
<b>TraCI</b>	<b>Interface De Contrôle Du Trafic</b>
<b>VANET</b>	<b>Vehicular Ad Hoc NETwork</b>
<b>VISSIM</b>	<b>Verkehr In Städten – Smulations Modell</b>
<b>V2I</b>	<b>Vehicle-To-Infrastructure</b>
<b>V2V</b>	<b>Vehicle-To-Vehicle</b>

## **Introduction générale**

Au cours des dernières décennies, les systèmes d'aide à la conduite ont considérablement renforcé la sécurité routière et contribué à offrir une expérience de conduite confortable et agréable. Cependant, avec l'augmentation continue du nombre d'usages de la route, la demande d'une conduite plus sûre et plus efficace devient de plus en plus importante.

Les réseaux ad hoc de véhicules (VANETs) sont créés par principes des réseaux ad hoc (MANETs) sur la technologie de l'automobile afin de permettre la communication entre les véhicules (V2V) et entre les véhicules et l'infrastructures (V2I). Cette communication permet l'échange des informations entre les véhicules et l'infrastructure, mais plus important elle offre la possibilité d'exploiter les ressources et les capacités de calcul des véhicules connectés par les réseaux VANET. A cette fin, les réseaux VANETs permettent d'étendre les capacités et de compléter les ressources de véhicules, et par conséquent la mise en œuvre de ce type de réseaux est nécessaire pour produire des systèmes véhiculaires connectés capables de fournir une conduite plus sûre et plus efficace. Cependant, le déploiement et le test de tels temps, aux coûts et à la sécurité routière. Une approche alternative consiste à l'utilisation d'un simulateur.

Notre démarche consiste à modéliser un modèle de mobilité pour les VANETs qui prend en considération les contraintes de la mobilité urbaine, pour que la simulation soit proche de la réalité. et aider les chercheurs pour implémenter leurs propres modèles.

Ce mémoire est organisé en quatre chapitres :

Le premier chapitre porte sur la modélisation et la simulation de la mobilité urbaine, les approches de modélisation (approche microscopique, approche macroscopique, approche mésoscopiques), Ces modèles se divisent, selon le style de conduite, en deux classes Le modèle de changement de voie (contrôle latéral) et le modèle de suivi de véhicules (contrôle longitudinal), et la modélisation microscopique de la tâche de conduite abordés au début de ce chapitre, puis nous présentons de manière générale les réseaux véhiculaires (caractéristiques, architectures, applications), Nous concluons par une vue d'ensemble des simulateurs utilisés pour un réseau VANET (sumo, NS2).

Dans le deuxième chapitre nous avons détaillé les systèmes multi-agents, définition du concept d'agent, caractéristiques d'un agent, système multi-agents, les domaines d'application de SMA et à la fin du chapitre nous citons quelques exemples de SMA.

Le troisième chapitre est consacré à Une approche multi-agent pour la simulation de la mobilité urbaine. Nous discutons, dans la première partie sur le simulateur sumo (Quel type de simulation de trafic est SUMO, exemples de l'utilisation, notion de base...), Ainsi que nous donnons une vue générale sur ses fonctionnalités. La deuxième partie

Simulation Microscopique de la Mobilité Urbaine détaille les étapes de réalisation de notre simulateur à savoir les besoins fonctionnels et les différents diagrammes (séquences),

Dans le dernier chapitre, nous allons évaluer en premier lieu les outils matériels et logiciels utilisés pour la simulation SUMO, les résultats de la simulation, réalisées avec différents scénarios à l'aide d'une interface graphique en java. L'objectif de cette simulation consiste à étudier et tester réellement notre modèle et aider les chercheurs pour implémenter leur propre modèle dans l'évitement des collisions, une analyse comparative est présentée à la fin de ce chapitre.

Nous terminons par une conclusion générale.

# Chapitre 1

## Modélisation et simulation de la mobilité urbaine

### I.1. Introduction

Ce chapitre présente un état de l'art sur la modélisation et la simulation de la mobilité urbaine. Cette modélisation peut se faire à trois approches différents : Commenant tout d'abord par le modèle macroscopique qui décrit le comportement des véhicules de manière globale en tenant compte de certains paramètres moyennés du trafic tels que la densité, le flux et la vitesse que l'on suppose continues dans le temps. A l'autre extrême, il existe les modèles microscopiques dans lesquels on s'intéresse au comportement individuel des véhicules, ici le mouvement de chaque véhicule est déterminé au moins en fonction de l'état du véhicule qui le précède. Entre ces deux approches de modélisation nous présentons les modèles mésoscopique. Nous concluons à la fin par un exemple de certains simulateurs usuels tels que SUMO et NS2.

### I.2. Approche de modélisation

La modélisation et la simulation de la mobilité urbaine est un domaine qui traite la simplification de la circulation automobile. L'esprit général de la modélisation de la mobilité est de pouvoir reproduire la réalité physique à l'aide des modèles qui permettent d'améliorer la compréhension d'un phénomène réel complexe et de lui donner une représentation simplifiée [1].

La modélisation et la simulation de la mobilité urbaine est étudiée dont l'objectif de contrôler, de gérer la circulation et de donner des solutions pour résoudre certains problèmes (ex. la congestion, les accidents, l'évacuation, la transition énergétique, etc.).

Les modèles existants sont classifiés par la façon avec laquelle ils simulent le comportement des véhicules et reproduisent le flux du trafic urbain. Par conséquent, cette approche de classification permet de distinguer trois classes comme suit [2] :

- Modèles microscopiques,
- Modèles macroscopiques,
- Modèles mésoscopiques,

#### I.2.1. Modèles microscopiques

Dans ce type de modèles, les véhicules sont considérés uniquement de manière individuelle. Par conséquent, le véhicule est vu comme une entité indépendante de

l'environnement et des autres véhicules, plus précisément les interactions entre véhicules et avec l'environnement sont prises en compte au niveau local [3].

Bien que les modèles microscopiques soient souvent utilisés pour une simulation réelle du flux, la limitation du comportement d'un nombre important de véhicules constitue un handicap pour ces modèles. Ces modèles se divisent, selon le style de conduite, en deux classes Le modèle de changement de voie (contrôle latéral) et le modèle de suivi de véhicules (contrôle longitudinal).

De nombreuses notations sont utilisées pour la description des modèles définissant le comportement d'un véhicule au sein d'une simulation du trafic urbain. Notons tout d'abord le temps par la variable  $t$ . Dans le cas des modèles microscopiques, nous utiliserons les notations suivantes (voir Figure I-1) [1]:

- La position du véhicule  $i$  à un instant  $t$ , est notée par  $x_i(t)$ ,
- La vitesse instantanée du véhicule  $i$  à un instant  $t$ , est notée par :  
 $v_i(t) = x'_i(t)$ ,
- L'accélération du véhicule  $i$  à un instant  $t$ , est notée  $a_i(t) = x''_i(t)$ ,
- Nous notons  $b_i(t) = -x''_i(t)$ ,  $b_i(t) > 0$  La décélération du véhicule  $i$  à un instant  $t$ .
- La longueur du véhicule  $i$  notée  $L_i$ ,
- L'inter-distance entre le véhicule  $i$  et son véhicule précédent  $leader(i-1)$  au temps  $t$ , notée  $s_i(t) = x_{i-1}(t) - x_i(t)$ ,
- La vitesse relative du véhicule  $i$  par rapport à son véhicule précédent  $leader(i-1)$  à un instant  $t$ , notée  $s'_i(t) = x'_{i-1}(t) - x'_i(t)$ .
- On note  $G_i(t) = x_{i-1}(t) - x_i(t) - L_{i-1}$  l'écart entre le véhicule  $i$  et son prédécesseur  $i-1$ ,

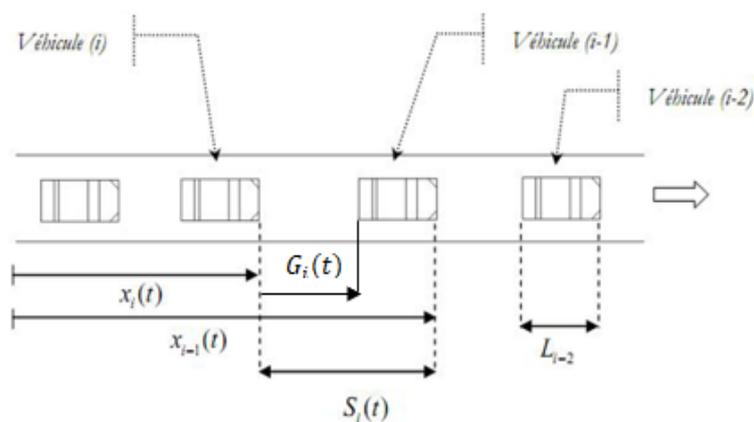


Figure I-1 la notation microscopique [1].

Un modèle microscopique classique est celui de (follow-the-leader) [4, 5] Ce modèle décrit le flux de circulation sur une route par la façon dont le conducteur suit son prédécesseur sans dépasser. Il est supposé que chaque véhicule suit son prédécesseur avec une relation de stimulation-réponse.

L'accélération  $\frac{d^2 x_n(t+T)}{dt^2}$  du véhicule  $n$  à un instant  $(t + T)$ , où  $T$  est le temps de réaction du conducteur, est donnée en multipliant par  $\lambda$  la réponse à la stimulation (voir L'Équation I-1) :

$$\frac{d^2 x_n(t+T)/dt^2}{dt^2} = \lambda \left( \frac{dx_{n-1}(t)}{dt} - \frac{dx(t)}{dt} \right) \quad \text{Équation I-2}$$

### I.2.2. Modèles macroscopiques

Ce modèle est inspiré de la mécanique des fluides et modélise l'écoulement du trafic comme l'écoulement d'un fluide continu, et décrit par les variables macroscopiques suivantes: le nombre des véhicules noté  $N(x, t)$ , la densité noté  $K(x, t)$  et le débit noté  $Q(x, t)$ .

La Figure I-2 illustre une représentation de la variable de débit. Le débit  $Q(x, t)$  correspondant au nombre de véhicules s'écoulant à un point d'abscisse  $x$  à un instant  $t$  par unité de temps :

$$Q(x, t) = N(x, t \rightarrow t + dt) / dt \quad \text{Équation I-3}$$

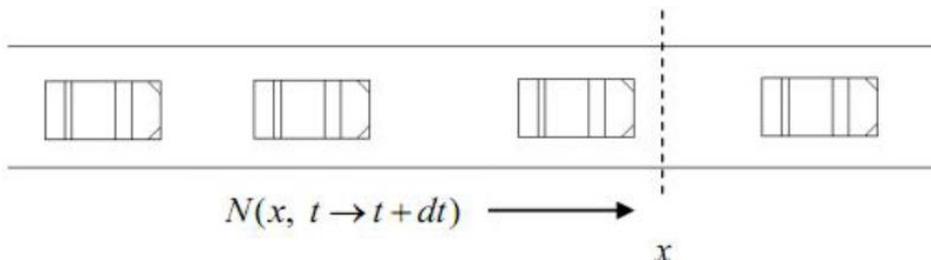


Figure I-2 Illustration de la variable macroscopique « le débit » [1].

La concentration (appelée également la densité spatiale instantanée) correspond au nombre de véhicules par unité de longueur  $\Delta x$  se trouvant sur une section voisine du point d'abscisse  $x$  à un instant  $t$ ;

$$K(x, t) = N(x \rightarrow x + \Delta x, t) / \Delta x \quad \text{Équation I-4}$$

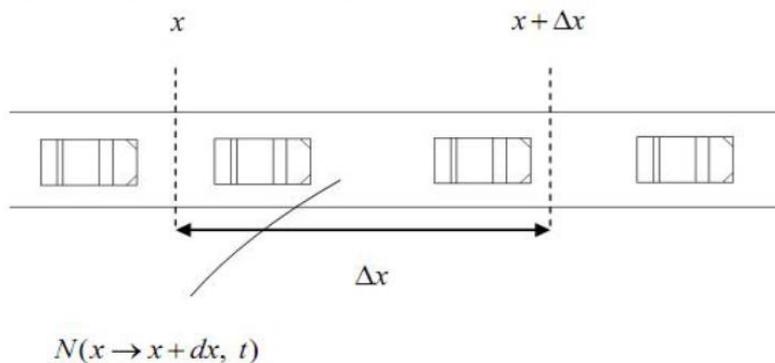


Figure I-3 Illustration de la variable macroscopique « la densité » [1].

La vitesse de flot, notée  $V(x, t)$  correspondant à la vitesse moyenne spatiale des véhicules situés dans la section  $[x, (x + \Delta x)]$  à un instant  $t$ . Cette vitesse de flot est donnée par le rapport entre le débit instantané et la concentration comme la figure suivante le montre (voir Figure I-4).

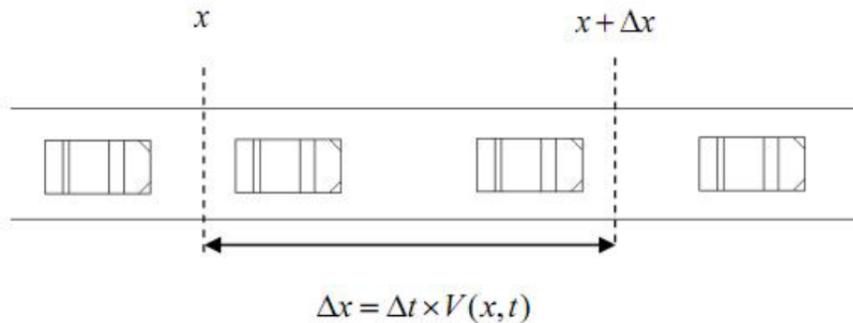


Figure I-4 Illustration de la vitesse de flot [1].

le modèle macroscopique du trafic routier le plus connu est celui de (LWR) [6, 7]. C'est un modèle d'écoulement qui décrit le trafic par une équation de conservation des véhicules et une équation de comportement (le diagramme fondamental)

$$Q(x, t) = N(x, t \rightarrow t + dt) / dt \quad \text{Équation I-5}$$

### I.2.3. Modèles mésoscopiques

L'idée derrière cette approche mésoscopique est d'effectuer des calculs de quantités agrégées à un niveau macroscopique ne fournit pas une description suffisamment détaillée des développements du trafic, mais que le suivi et la gestion des véhicules individuellement prend trop de temps Calcul. Pour surmonter ce problème, ces modèles proposent un compromis en composant l'approche macroscopique et l'approche microscopique : ils représentent le trafic par faisceaux de véhicules regroupés selon certaines caractéristiques. Différentes classes de conducteurs et différents types de véhicules peuvent par exemple être introduits[8].

Comme pour les modèles macroscopiques, les interactions entre les véhicules ne sont prises en compte qu'à travers des équations globales, tout comme les relations entre les véhicules et l'environnement [9].

## I.3. Modélisation microscopique de la tâche de conduite

Nous avons vu déjà une description rapide sur les modèles d'écoulement existants, classiquement divisés en trois catégories : les modèles microscopiques, les modèles macroscopique et les modèles mésoscopiques, Les modèles microscopiques permettent non seulement de gérer la poursuite d'un véhicule sur un autre, mais aussi de décrire, d'analyser et d'évaluer plusieurs caractéristiques du trafic comme la congestion. Ces modèles se divisent, selon le style de conduite, en deux classe Le changement de voie (contrôle latéral) et Le suivi de véhicules (contrôle longitudinal)

### I.3.1. Modèle de changement de voie

Les principes de caractéristiques du système voiture à voiture autonome peuvent être mises en évidence sur un exemple pratique. Un exemple de la situation qui présente des risques potentiels est un changement de voie (contrôle latéral). Imaginez une situation représentée sur la Figure I-5.

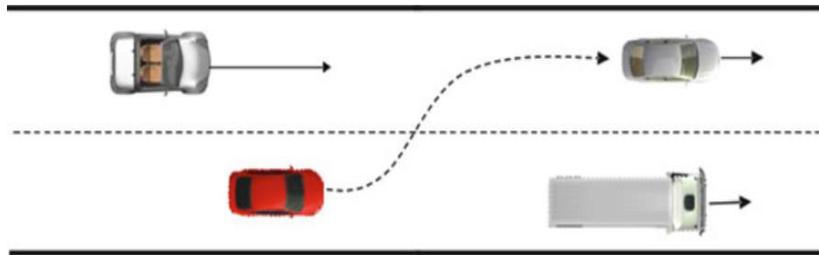


Figure I-5 Exemple de la situation dangereuse de changement de voie

La voiture rouge approche un camion se déplaçant lentement et doit le dépasser en toute sécurité. La manœuvre est simple, mais la situation est également influencée par le comportement des autres voitures de la voie de gauche.

La manœuvre de changement de voie est considérée comme l'un des plus difficiles tâches de conduite et une attention particulière est nécessaire. Les accidents résultant d'un changement de voie non conforme constituent près de 8% de tous les accidents de voiture [10].

Une sorte commune de la situation dangereuse peut se produire si le conducteur sous-estime la vitesse d'un véhicule qui approche ou donne du tout. Il a été démontré que la perception des conducteurs des distances entre véhicules est souvent précaire, en particulier à des vitesses élevées [11].

Les systèmes d'assistant au changement de voie (LCA) peuvent aider les conducteurs humains à éviter les accidents. Un LCA surveille les voies adjacentes et maintient le conducteur informé de la présence des autres véhicules à proximité. En outre, en cas de conflit ressources (soit une contradictoires partie d'une voie est visée par deux véhicules), une coopération de plusieurs LCA peut améliorer considérablement la gestion des ressources, le renforcement d'une collaboration bénéficiaire entre les conducteurs [12].

### I.3.2. Modèle de suivi de véhicules

L'objectif de la modélisation microscopique des flux de trafic urbain est d'établir une description mathématique du comportement du conducteur humain en fonction de l'état de son véhicule, puis de produire un environnement de simulation sans collision. Pour cela, deux processus doivent être envisagés ; contrôle longitudinal modélisé par un modèle de suivi des véhicules et contrôle latéral modélisé par un modèle de changement de voie.

Après le processus de contrôle longitudinal, le contrôle de la vitesse d'un véhicule à un moment donné n'est effectué que si la vitesse actuelle est différente

d'une vitesse souhaitée. Cette vitesse souhaitée est définie en fonction de nombreux attributs tels que l'état du trafic routier et le code de la route. De plus, la modélisation du contrôle longitudinal et la définition de la vitesse souhaitée doivent produire un modèle de suivi du véhicule capable d'établir un contrôle longitudinal sans collision. [13]

Sur la base des propriétés du trafic routier, plusieurs modèles de suivi de véhicules ont été proposés pour définir la vitesse souhaitée et établir un contrôle longitudinal sans collision. D'autre part, d'autres modèles sont basés sur la modélisation du comportement du conducteur humain, y compris l'imperfection du processus de perception et de prise de décision. Dans ce qui suit, nous présentons quelques modèles de suivi de véhicule proposés dans la littérature.[13]

### I.3.2.1. Modélisation basée sur les propriétés du la mobilité urbaine

Dans certains modèles classiques de suivi des véhicules, la modélisation du contrôle longitudinal est basée uniquement sur les propriétés du trafic. D'autre part, d'autres modèles décrivent le comportement de contrôle longitudinal et contrôlent la vitesse d'un véhicule pour maintenir une certaine vitesse optimale. En conséquence, les modèles conventionnels de suivi des véhicules sont divisés en deux classes; modèles à la pointe du progrès et modèles à vitesse optimale.[14]

Le premier type de modèle de suivi de véhicule, appelé modèle de suivi du leader, est basé sur l'hypothèse que le mouvement d'un véhicule  $i$  est régi uniquement par le mouvement de son véhicule précédent  $i + 1$ . Sur la base de ce type de modèle, en particulier dans un état stationnaire de simulation, les vitesses de tous les véhicules devraient être identiques sinon les véhicules entreraient en collision [14].

Un véhicule  $i$  applique une accélération définie par (l'Équation I-6) à un instant  $t$  pour atteindre une vitesse désirée  $v_{des}$  identique à  $v_{i+1}$  la vitesse de son prédécesseur. De plus, il convient de noter que l'espacement entre les deux véhicules devrait augmenter linéairement avec la vitesse afin de maintenir une conduite sans collision.

$$\frac{dv_i}{dt} = \frac{v_{i+1}(t) - v_i(t)}{\tau} \quad \text{Équation I-6}$$

Pour :

- $dv_i/dt$  : l'accélération appliquée par le véhicule  $i$  à un instant  $t$  pour correspondre à la vitesse du véhicule précédent  $i + 1$  à  $t + \Delta t$ ,
- $v_i(t)$  : la vitesse actuelle du véhicule  $i$ ,
- $v_{i+1}(t)$  : la vitesse actuelle du véhicule de tête  $i + 1$ .
- $t$  : une échelle de temps utilisé pour déstabiliser l'état de système.

À base de (l'Équation I-6), plusieurs modèles de suivi de véhicules ont été proposés pour établir un contrôle longitudinal sans collision. Par conséquent, de tels

modèles considèrent que  $v_i$  la vitesse du véhicule contrôlé à  $t + 1$  seconde sera identique à  $v_{i+1}$  la vitesse du véhicule de tête [15].

### I.3.2.2. Modélisation basée sur le comportement de conducteur humain

Une modélisation précise du contrôle longitudinal est cruciale pour produire un simulateur fiable qui permet non seulement le développement et le test de nouveaux systèmes, mais également la compréhension de la dynamique du trafic routier. De plus, la prise en compte des facteurs humains lors de la modélisation est essentielle pour obtenir une description plus réaliste du comportement de suivi des véhicules, en particulier dans les situations complexes [15].

Le modèle original de (weidemann) [16] a été largement utilisé dans la littérature et, le plus important, il a été implémenté et utilisé dans SUMO, un simulateur microscopique de trafic urbain [17], permettant le développement et la simulation de nombreux systèmes proposés dans la littérature. De plus, une version modifiée du modèle de (weidemann) [16] a été implémentée et utilisée dans VISSIM, un simulateur de trafic urbain [18].

Dans le modèle de (Krauß) [13], l'auteur a proposé un modèle déterministe du suivi de véhicules (voir Équation I-7). Ce modèle est basé sur une vitesse de sécurité  $v_{safe}$  permettant au véhicule de s'adapter au freinage de son leader et de maintenir un contrôle longitudinal sans collision. Et pour aller plus loin, l'auteur a proposé de multiplier l'accélération du véhicule par un nombre aléatoire  $e$  afin de modéliser l'imperfection du conducteur pendant le contrôle de vitesse.

$$v_{des}(t + \Delta t) = \max\{0, \text{rand}[v_{des} - ea, v_{des}]\} \quad \text{Équation I-7}$$

Pour :

- $v_{des}(t + \Delta t)$  : la vitesse désirée définie tout en tenant compte de l'imperfection du conducteur humain,
- $e$  : la capacité (ou bien le désir) du conducteur à respecter  $v_{des}$ ,
- $a$  : l'accélération maximale du véhicule,
- $v_{des}$  : une vitesse désirée.

En conséquence, semblable au comportement du conducteur humain, le modèle proposé par l'auteur [13] a montré un comportement stochastique et imparfait avec des réactions intelligentes et réalistes. De plus, ce modèle fournit un contrôle longitudinal réaliste et sans collision. Dans ce qui suit, nous décrivons en détail le comportement des modèles les plus utilisés dans la littérature tout en soulignant leurs principales caractéristiques et, plus important encore, l'un de leurs principaux inconvénients.

### I.3.3. Modélisation du contrôle longitudinal

Sur la base principalement des propriétés du trafic routier, de nombreux modèles de suivi des véhicules ont été proposés et, surtout, certains sont mis en

œuvre dans plusieurs simulateurs (par exemple SUMO, NS2, etc.). Parmi ces modèles, nous mentionnons les modèles proposés dans [16, 19]. Ces modèles ont fourni un contrôle longitudinal sans collision et, par conséquent, ont permis le développement et la simulation de plusieurs systèmes dans différents simulateurs de trafic urbain.

Dans [13], l'auteur a proposé un modèle de suivi de véhicules dans lequel la vitesse finale d'un véhicule est définie par deux règles : (i) la vitesse à un instant donné est limitée par une vitesse maximale ( $y \leq V_{max}$ ), (ii) dans un environnement urbain où plusieurs véhicules coexistent, le véhicule respecte et maintient toujours une vitesse de sécurité ( $y \leq V_{safe}$ ) qui doit établir un contrôle longitudinal sans collision.

Le modèle de Krauß [13] a été proposé pour atteindre une vitesse finale  $y$  à un instant  $t + \Delta t$  où  $y$  est le minimum de : (i) une vitesse maximale que le véhicule peut atteindre et ne peut pas dépasser après l'application d'une accélération  $a$ , et (ii) une vitesse de sécurité  $v_{safe}$  qui permet au véhicule de s'arrêter en cas d'urgence par l'application d'une décélération maximale  $b$  (voir Equation I-9, Equation I-8 ) [20] [21].

$$v(t + \Delta t) = \min\{v_{safe}, v(t) + a\Delta t, v_{max}\} \quad \text{Equation I-9}$$

$$v_{safe} = v_t(t) + \frac{g(t) - v_t(t)t_r}{\frac{v_t(t) + v_f(t)}{2b} + t_r} \quad \text{Equation I-10}$$

Pour :

- $v(t + \Delta t)$  : la vitesse de véhicule à  $t + \Delta t$ ,
- $v(t) = v_i(t)$  : la vitesse du véhicule à  $t$ ,
- $v_{safe}$  : une vitesse de sécurité,
- $v_{maxe}$  : la vitesse maximale du véhicule,
- $a > 0$  : l'accélération maximale qu'un véhicule peut appliquer,
- $b > 0$  : l'décélération maximale,
- $t_r$  : le temps de réaction du conducteur humain,
- $v_i(t)$  : la vitesse de leader (le véhicule de tête),
- $g(t)$  : la distance qui sépare le véhicule contrôlé et le leader.

En outre, il convient de noter que les accélérations positives et négatives du modèle Krauß [13] sont limitées  $-b \leq \frac{dv}{dt} \leq a$ .

Le modèle de Krauß [13] réussit toujours à atteindre l'état souhaité à  $t + \Delta t$  dû au fait que ce modèle applique directement la vitesse  $v_j$  sur la vitesse du véhicule

contrôlé (voir [[17], [13]]. De ce fait, [13] considère que les actions du véhicule, notamment l'accélération maximale  $a$  et la décélération maximale  $b$ , sont des actions déterministes et précises.

Dans [22], un modèle de suivi de véhicules, appelé Intelligent Driver Mode (IDM), et basé sur une fonction d'accélération continue et déterministe a été proposé dans le but de fournir un contrôle longitudinal sans collision (voir Equation I-11, )

$$\frac{dv}{dt} = a \left[ 1 - \left( \frac{v}{v_0} \right)^\delta - \left( \frac{s^*(v, \Delta v)}{s} \right)^2 \right] \quad \text{Equation I-11}$$

$$s^*(v, \Delta v) = s_0 + \max \left\{ 0, vT + \frac{v\Delta v}{2\sqrt{ab}} \right\} \quad \text{Equation I-12}$$

Pour :

- $v$  : La vitesse du véhicule contrôlé,
- $V_0$  : la vitesse maximale du véhicule,
- $\Delta v$  : la vitesse relative qui représente la différence entre la vitesse du véhicule contrôlé et la vitesse du prédécesseur (le leader),
- $S$  : la distance qui sépare le véhicule contrôlé et le leader,
- $S^*$  : l'écart minimum recherché ( $s \geq s^*$ ) (ex.  $2m$ ),
- $a$  : l'accélération maximale du véhicule (ex.  $a \in [1,2] m/s^2$ ),
- $b$  : la décélération maximale du véhicule (ex.  $b \in [1,4] m/s^2$ ).

Le modèle de IDM applique une action  $a = a \left[ 1 - \frac{v}{v_0} \delta \right]$  sur la vitesse d'un véhicule pour accélérer, par contre il peut appliquer une décélération  $b = -a * (s^*/s)^2$  pour freiner et empêcher toute collision imminente. De plus, ce modèle considère que ses actions, en particulier l'accélération et la décélération, sont déterministes et capables d'atteindre la vitesse désirée avec précision [22].

Dans [19], un modèle de suivi de véhicules a été proposé et utilisé pour développer et tester un système de communication inter-véhicules. Ce modèle est un modèle à base d'équations chargé de décrire le comportement de contrôle longitudinal d'un véhicule (voir Équation I-13).

$$\begin{aligned} v_{t+1} &= \max\{0, \min\{v_{free}, \tilde{v}_{t+1} + \xi_t, v_t + a\tau, v_{s,t}\}\} \\ v_{s,t} &= \min\{v_{safe}, \frac{g}{\tau} + v_i\} \\ \tilde{v}_{t+1} &= \max\{0, \min\{v_{free}, v_{s,t}, v_{c,t}\}\} \\ v_{c,t} &= \begin{cases} v_t + \Delta_t & \text{if } g_t \leq G_t \\ v_t + a_n \tau & \text{if } g_t > G_t \end{cases} \\ \Delta_t &= \max\{-b_t \tau, \min\{a_t \tau, v_{i,t} - v_t\}\} \end{aligned} \quad \text{Équation I-13}$$

Où :

- $v_{t+1}$  est la vitesse du véhicule à  $t + 1$ ,  $v_t$  est la vitesse à  $t$ .
- $v_{free}$  est la vitesse maximale que le véhicule peut atteindre à l'étape suivant  $t + \tau$
- $t$  correspond à un temps discret  $t = 0,1,2,\dots$  et  $\tau$  est un pas de temps.
- $v_{s,t}$  Est la vitesse de sécurité.

D'après [23], les auteurs ont proposé un modèle de suivi de véhicules, appelé full vitesse and accélération différence model (FVADM), qui décrit et modélise le comportement d'un conducteur pour prévenir les collisions pendant le freinage d'urgence.

À cette fin, de nombreux modèles de suivi de véhicules ont été proposés pour contrôler la vitesse d'un véhicule alors qu'ils agissent directement sur la vitesse pour correspondre à une vitesse désirée (ex. Modèle de : [19], [23], [24]). À cet effet, nous considérons que chaque de ces modèles applique une action déterministe  $a$  à un instant  $t$  pour atteindre une vitesse  $v_{des}$  à l'étape suivante  $t + \Delta t$  de la simulation (voir Équation I-14).

$$a = \frac{v_{des}(t + \Delta t) - v(t)}{\Delta t} \quad \text{Équation I-14}$$

Pour :

- $a$  : une accélération déterministe qu'un véhicule applique pour atteindre  $v_{des}$
- $v$  : la vitesse du véhicule à  $t + \Delta t$ ,
- $v_{des}$  : la vitesse que le modèle souhaite d'atteindre à  $t + \Delta t$  par l'exécution d'une action déterministe  $a$  à un instant  $t$ ,

Pourtant, d'autres modèles agissent sur la vitesse par l'application d'une valeur d'accélération définie en utilisant une fonction déterministe (ex. modèle de IDM[22]). En conséquence, ce type de modèles est conçu pour contrôler la vitesse sous des conditions idéales et aussi pour agir dans un environnement déterministe.

#### I.4. Communication v2x

Les systèmes de communication sans fil évoluent, la perspective des véhicules communiquant entre eux (V2V) et avec les unités en bord de route (V2I) devient une réalité. Ces systèmes de communication sans fil automobiles, qui sont abordés pour les communications des véhicules avec toute autre infrastructure (V2X), sont connus sous le nom de systèmes de transport intelligents (ITS).

### I.4.1. Les réseaux véhiculaires

Les réseaux véhiculaires représentent aujourd'hui un intérêt certain pour l'industrie automobile, les opérateurs des réseaux, les organisations et même les particuliers. Ils représentent une des briques de base sur lesquelles vont se fonder les systèmes de transport intelligents. Nous apercevons ces dernières années que ces réseaux VANET ont reçu un grand intérêt de la majorité des projets de coopérations, les unités industrielles et les organisations de recherches. Ceci reflète une importante évolution dans le domaine des réseaux VANET.

### I.4.2. Les réseaux VANETs

#### I.4.2.1. Définition d'un Réseaux VANETs

Un réseau VANET (véhiculaire Ad Hoc) est un réseau de communication entre véhicules intelligents équipés de calculateurs, de périphériques réseau et de différents types de capteurs. Les VANET font parti de la famille des réseaux MANET (mobile Ad Hoc Network) qui fonctionnent dans des réseaux à liaison point à point sans infrastructure, c'est-à-dire que tout nœud constituant le réseau est un point d'accès. [25]

#### I.4.2.2. Composant d'un réseaux VANETs

Un réseau VANET est constitué principalement de trois entités [26]:

- TA (Trusted Authority) : Dites CA (autorité de confiance). C'est une source d'authenticité de l'information. Elle assure la gestion et l'enregistrement de toutes les entités sur le réseau (RSU et OBU).
- RSUs (Road Side Unit) : Ces entités sont les subordonnés des TA. Elles sont installées au bord des routes.
- OBU (On-Board Unit) : Ce sont des unités embarquées dans les véhicules intelligents, elles regroupent un ensemble de composants matériels et logiciels de hautes technologies (GPS, radar, caméras, différents capteurs et autres). Ce sont des émetteurs-récepteurs qui assurent la connexion du véhicule au réseau.

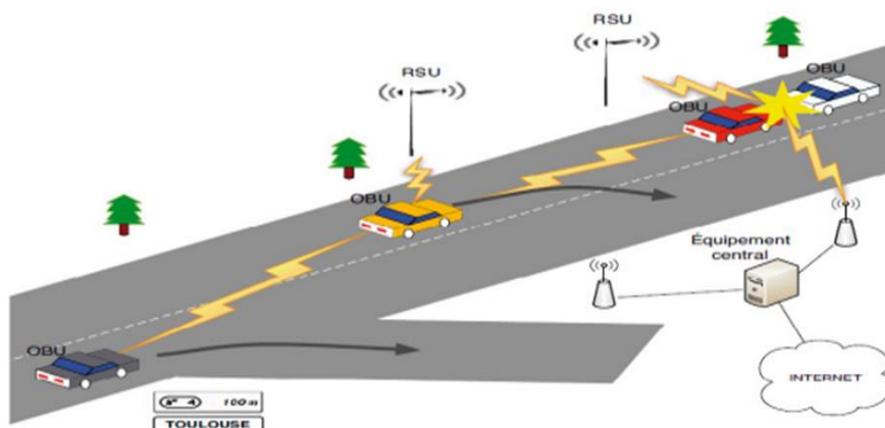


Figure I-6 Exemple de réseaux véhiculaire [26]

### I.4.3. Architectures des réseaux véhiculaires

Les architectures dans un réseau véhiculaire peuvent être déployées suivant trois catégories :

#### I.4.3.1. Communications de Véhicule à Véhicule (V2V)

Dans ce mode, chaque véhicule est équipé pour communiquer directement avec un autre véhicule [27] de sorte que aucune infrastructure et installation n'est nécessaire, Ce qui rend la communication moins coûteuse et plus flexible

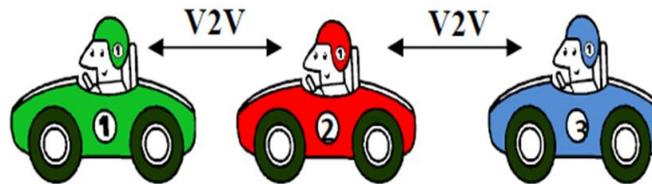


Figure I-7 communication véhicule à véhicule

#### I.4.3.2. Communications de Véhicule à Infrastructure (V2I)

Cette façon de communication permet une meilleure utilisation des ressources partagées et multiplie les services fournis par les RSUs installés sur le bord de la route, mais le problème de ce mode est coûteux. [27]



Figure I-8 Communication véhicule à Infrastructure

#### I.4.3.3. Communication hybride

Ce mode de communication est produit lors de la combinaison le mode de communication de véhicule à véhicule avec le mode de communication de véhicule avec utilisation d'infrastructures. [27]

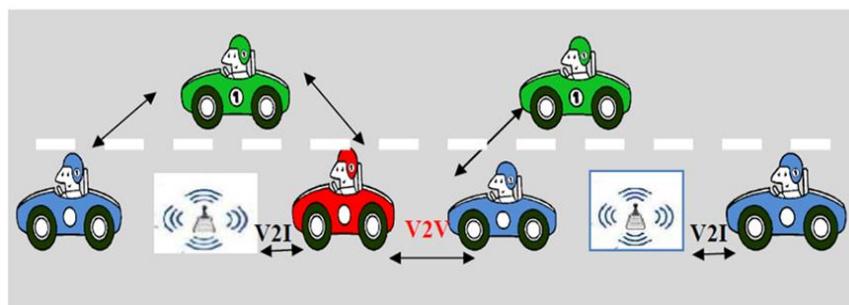


Figure I-9 communication hybride

#### **I.4.4. Les Applications des réseaux VANETs**

Les principales applications des réseaux VANET peuvent être classées en trois catégories[28].

##### **I.4.4.1. Applications de gestion du trafic routier**

Les applications de gestion de trafic sont axées sur l'amélioration des conditions de circulation dans le but de réduire les embouteillages et les risques d'accidents. Elles fournissent aux conducteurs un support technique leur permettant d'adapter leur parcours à la situation du trafic routier. Ces applications visent à équilibrer la circulation des véhicules sur les routes pour une utilisation efficace de la capacité des routes et des carrefours et à réduire par conséquent les pertes humaines, la durée des voyages et la consommation d'énergie...etc.

##### **I.4.4.2. Applications de confort**

Cette catégorie comprend toutes les applications qui contribuent au confort du conducteur et ne font pas partie de la gestion du trafic ou de la sécurité routière. Ces applications sont donc présentées comme des services fournis au conducteur. Parmi ces applications figurent les panneaux publicitaires locaux : commerciaux comme le propose le restaurant, la présence de stations-service à proximité, ou culturels comme informations touristiques sur la localisation du véhicule.

##### **I.4.4.3. Applications de sécurité du trafic routier**

Ces applications se basent en général sur une diffusion, périodique ou non, de messages informatifs permettant aux conducteurs d'avoir une connaissance de l'état de la route et des véhicules voisins. Ils visent à améliorer la sécurité des passagers sur les routes en avisant les véhicules de toute situation dangereuse. Des exemples répandus de services dans cette catégorie d'applications sont, l'avertissement des collisions, les avertissements sur les conditions de la route, l'assistance dépassement et changement de voie, etc.

#### **I.4.5. Simulateurs**

Il existe plusieurs simulateurs de réseaux ad hoc capables d'effectuer des simulations répondant aux critères des VANET. Nous présentons dans cette partie quelques simulateurs qui peuvent y répondre :

##### **I.4.5.1. La Simulation dans un Réseau VANET**

Un simulateur de réseau VANET réaliste se compose des éléments indispensables : un simulateur de réseau regroupant les bases réseaux d'une simulation VANET et un simulateur de mobilité. L'un des problèmes que l'on rencontre alors est l'interconnexion entre les simulateurs. Il existe cependant des simulateurs intégrés qui regroupent les deux simulateurs au sein d'une même entité pour répondre au mieux à ce problème.

Dans ce qui suit nous monteront les principes de ces deux éléments nécessaires à la simulation des VANETs, à savoir les simulateurs de réseau Ad Hoc et les simulateurs de mobilité [29].

#### **I.4.5.2. Les Simulateurs de Réseaux Ad Hoc**

Il n'y a pas de principe de fonctionnement standard pour les simulateurs de réseau Ad Hoc ou les simulateurs de mobilité. Il existe plusieurs stratégies et architectures qui permettent de simuler un réseau Ad Hoc.

##### ***Le simulateur NS-2***

NS-2 est un outil logiciel de simulation de réseau informatique. Il est essentiellement développé avec les idées de conception basée sur les objets, de réutilisation de code et de modularité. C'est aujourd'hui une référence standard dans ce domaine, plusieurs laboratoires de recherche recommandent son utilisation pour tester de nouveaux protocoles. Le simulateur NS-2 actuel est particulièrement bien adapté aux réseaux à commutation de paquets et aux simulations à grande échelle (le test de mise à l'échelle). Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage unicast ou multicast, des protocoles de transport, des protocoles de session, des protocoles de réservation, des services intégrés, des protocoles d'application comme FTP. [29]

#### **I.4.5.3. Les Simulateurs de mobilité et de trafic routier**

##### ***Le simulateur SUMO***

SUMO (Simulation of Urban Mobility) est un logiciel de simulations de trafic routier open source sous licence ONU public (OPL), dont le développement a commencé en 2002. L'objectif des développeurs est de mettre à la disposition du monde académique un outil leur permettant de modéliser le réseau routier aussi bien en milieu urbain.

La Figure I-10 montre le progiciel SUMO qui contient une suite d'applications qui aident à préparer et à exécuter la simulation d'un scénario de trafic routier. Il est très portable et nécessite uniquement l'installation des bibliothèques C++. Dans SUMO, chaque véhicule a son propre chemin, le comportement du véhicule est vivant comme le changement de voie. Les routes dans SUMO sont présentées sous forme de plusieurs voies. Il y a des intersections à base de règles de circulation et d'autres à priorité[29].

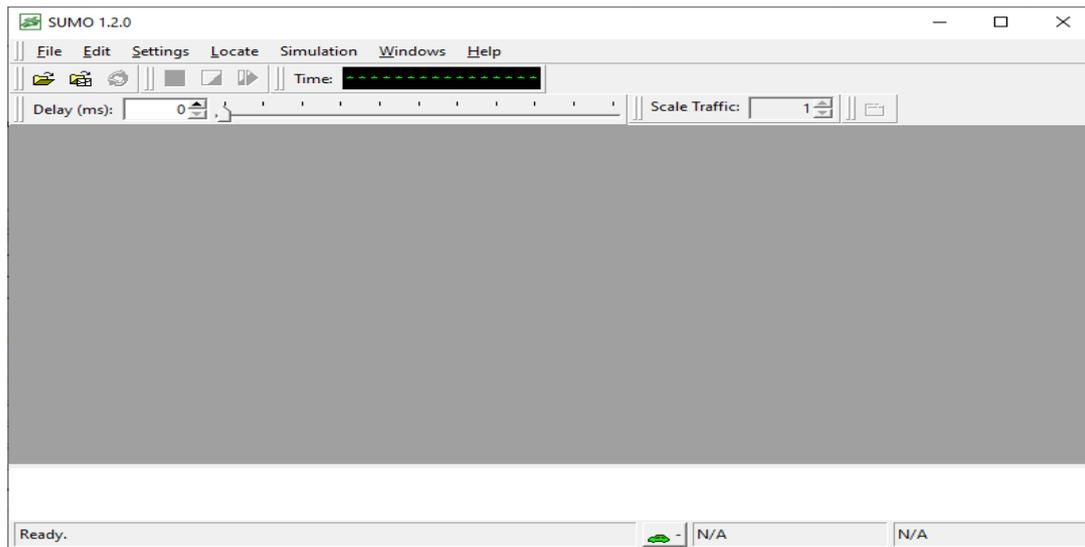


Figure I-10 L'interface graphique du SUMO

## **I.5. Conclusion**

Dans ce chapitre, nous avons présenté les approches de modélisation (approche microscopique, approche macroscopique, approche mésoscopiques), ensuite nous avons présenté les réseaux véhiculaires VANETs qui ne sont qu'une particularité des réseaux MANETs. Nous avons signalé également leur composants, leur application. Ainsi la communication dans les réseaux véhiculaires, Nous concluons par une vue d'ensemble des simulateurs utilisés pour un réseau VANET (sumo, NS2). Le chapitre suivant sera consacré pour les système multi agents.

# Chapitre 2

## Les systèmes multi-agents

### II.1. Introduction

Dans ce chapitre, nous commencerons par présenter un aperçu sur les agents (Définition du concept d'agent, et leur caractéristique) de manière globale, on suit nous parlent sur les systèmes multi-agents (SMA) et leur domaines d'application, et à la fin on site quelque exemple sur les système multi agent.

### II.2. Introduction aux systèmes multi-agents

#### II.2.1. Définition du concept d'agent

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-agents, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents [30]

Pour [31] un agent est une "entité computationnelle", comme un programme informatique ou un robot, qui peut être vue comme percevant et agissant de façon autonome sur son environnement

#### II.2.2. Caractéristiques d'un agent

Un agent est caractérisé par :

- Un agent est une entité autonome mais fortement dépendante des autres.
- Un agent communique à l'aide de messages, ou par partage d'informations.
- Un agent est autonome, il peut refuser de faire ce qu'on lui dit de faire.[30]

#### II.2.3. Système multi-agents

Un système multi-agents (SMA) est constitué d'un ensemble de processus informatiques se déroulant en parallèle, donc de plusieurs agents vivant au même moment, partageant des ressources communes et communiquant entre eux.

Le point clé des SMA réside dans la formalisation de la coordination entre les agents. Partant de la définition que donne [35] d'un agent logiciel, on peut définir

cette entité comme un système informatique situé dans un environnement, capable de mener de manière autonome des actions sur cet environnement en vue d'accomplir ses objectifs, possédant en plus les propriétés de:

- **Réactivité** : il perçoit des stimuli provenant de son environnement et réagit en fonction de ceux-ci. Pro activité : il est mû par un certain nombre d'objectifs qui guident ses actions.
- **Sociabilité** : il communique avec d'autres agents ou des humains et, peut se trouver engagé dans des transactions sociales (Négocier ou coopérer pour résoudre un problème) afin de remplir ses objectifs.

On parle ainsi d'intelligence artificielle distribuée.

Les SMA peuvent être vus comme la rencontre de divers domaines :

- L'intelligence artificielle pour les aspects prise de décision de l'agent.
- L'intelligence artificielle distribuée pour la distribution de l'exécution.
- Les systèmes distribués pour les interactions entre agents.

Les SMA peuvent être partagés en deux types d'architectures [32], en fonction du type de contrôle adopté :

- SMA à contrôle centralisé ou base de tableau noir Composé de trois éléments :
  - Les connaissances représentées par les agents.
  - Le tableau noir : qui est une zone de travail commune, dévolue à la transition d'informations entre les différents agents. Chacun peut venir le consulter à sa guise, y prélever et y déposer des objets qu'il peut également modifier. Le tableau structure la modélisation du domaine d'application comme l'espace des hypothèses et des solutions.
  - Le mécanisme de contrôle : concerne les contraintes sur les relations entre les conversations des protocoles qui régissent le système, et au quels l'agent peut participer simultanément ou successivement.
- Le SMA à contrôle centralisé possède en outre les propriétés suivantes :
  - Pas de communication directe entre les agents.
  - Interaction via le partage d'un même espace de travail qui est le blackboard. Mal adapté aux SMA large échelle.
  - SMA a contrôle distribuer

- Une distribution totale des connaissances et du contrôle.
- Communication entre agents par envoi de messages.

#### II.2.4. Domaines d'application des SMA

On distingue généralement 3 types d'utilisation des systèmes multi-agents[33]

- ✓ La simulation ou la modélisation de phénomènes complexes :

Où les SMA sont utilisés pour simuler les interactions existantes entre les agents autonomes. Le but est de déterminer l'évolution de ce système afin de prédire l'organisation finale. Ce qui compte, c'est le comportement global et non le comportement individuel. L'autonomie permet ici de simuler le comportement exact d'une entité. La première simulation utilisant les SMA, et qui d'ailleurs fut la source d'inspiration de ceux-ci est le système MANTA (simulation d'une fourmilière).

- ✓ La résolution de problèmes et prise de décision :

L'intelligence artificielle distribuée est née pour résoudre les problèmes de complexité des gros programmes de l'intelligence artificielle : l'exécution est alors distribuée, mais le contrôle reste centralisé. Contrairement aux SMA, où chaque agent possède un contrôle total sur son comportement. Pour résoudre un problème complexe, il est plus simple de concevoir des programmes relativement petits (les agents) en interaction, qu'un seul gros programme monolithique. L'autonomie permet au système de s'adapter dynamiquement aux changements imprévus qui interviennent dans l'environnement.

**Exemple** : minimisation d'impact pour des aménagements

- ✓ La conception de programmes

Intégrer un système d'information composé d'un ensemble d'agents organisés pour faciliter la compréhension et la prise de décision, individuellement ou collectivement. Contrairement à un objet, un agent peut prendre des initiatives, refuser d'obéir à une demande, se déplacer ...etc.

**Exemple** : systèmes d'aide à la négociation de projets

#### II.2.5. Quelques exemples de SMA

Les systèmes multi-agents associés à l'intelligence artificielle représentent actuellement un grand domaine d'application et de recherche. Plusieurs systèmes ont été développés, nous présenterons ici quelques-uns tels que :

- Le système MANTA [33]: ce système illustre parfaitement l'intérêt de la modélisation multi agents de type réactif. Il modélise la constitution d'une fourmilière mature à partir d'une ou plusieurs reines, étudie la capacité

d'adaptation d'une telle colonie, le mécanisme de polythéisme (division du travail), et la spécialisation des ouvrières. Cette simulation avait vérifié le fait qu'une société d'agents peut bien survivre et s'organiser en se passant de tout système de contrôle centralisé et d'une quelconque organisation hiérarchique.

- Le comportement de meute : les agents réactifs se montrent capables d'évoluer parfaitement en groupe tout en s'évitant mutuellement, constituant par là une meute aux comportements très souples. Le premier à s'être intéressé à ce comportement est Craig Reynolds (1987) [34], il a créé des créatures appelées "Boïdés", des agents réactifs capables d'interagir pour réaliser un comportement semblable à un vol d'oiseaux migrateurs, chacun des Boïdés se contentant d'appliquer un ensemble de règles comportementales :
  - Maintenir une distance minimale par rapport aux objets de l'environnement y compris les autres Boïdés.
  - Adapter sa vitesse à la moyenne de celle de ses voisins.
  - Aller vers le centre de gravité de l'ensemble des Boïdés voisins. Contrairement aux oiseaux, les Boïdés évoluent sans leader et sans contrôle global. Ils contournent tous ensemble un obstacle tout en restant naturellement groupés.
- Applications temps réel : Les agents ont été bien évidemment appliqués au domaine des systèmes temps réel ; ce dernier maintien des systèmes à contrainte souple. On voit de plus en plus des systèmes temps réel dit Hard utilisant des agents.
- Applications agents pour la Recherche d'Information : Une grande partie des applications de système multi agents est dans le domaine de recherche d'information. Parmi ces nombreuses applications dans ce domaine, on peut trouver "NetSA" [35]: une architecture de système multi-agents pour la recherche d'information dans des sources hétérogènes et réparties.

### II.3. Conclusion

Tout long de ce chapitre nous avons décrit en général les système multi agents, leurs concepts et leur fonctionnement, les domaines d'application du SMA, et à la fin on site quelque exemple sur de (SMA). Le chapitre suivant sera consacré pour une approche multi-agent pour la simulation de la mobilité.

# Chapitre 3

## Une approche multi-agent pour la simulation de la mobilité

### III.1. Introduction

Ce chapitre est consacré à Une approche multi-agent pour la simulation de la mobilité urbaine. Nous discutons, dans la première partie sur le simulateur sumo (Quel type de simulation de trafic est SUMO, exemples de l'utilisation, notion de base...), Ainsi que nous donnons une vue générale sur ses fonctionnalités. La deuxième partie Simulation Microscopique de la Mobilité Urbaine détaille les étapes de réalisation de notre simulateur à savoir les besoins fonctionnels et le différents diagrammes (séquences),

### III.2. SUMO : vers une simulation microscopique à base d'agents

SUMO est l'acronyme de "Simulation of Urban MObility "et est un projet open source concerné avec le développement et l'utilisation d'une simulation de trafic. Le projet fait partie de notre travail scientifique concerné avec vérification de différents modèles microscopiques du trafic, et leur comparaison. De plus, le la communauté scientifique de la circulation implique souvent des idées où chacun d'entre eux a besoin d'une simulation de trafic pour être validé. Au fil du temps, beaucoup de personnes plus ou moins sophistiquées des simulations ont été développées pour faire ce travail. Ils restent la plupart du temps inconnu.[36]

Cette approche est non seulement très inefficace comme une simulation de trafic a beaucoup de choses à ce qui concerne ; De plus, les résultats ne sont souvent ni reproductibles ni reproductibles. Moins difficile à comparer. Quand une plateforme commune est fournie, de tels problèmes ne devraient pas se produire. Dans cette publication, nous aimerions vous présenter notre paquet au public dans l'espoir de gagner encore plus intérêt. [37]

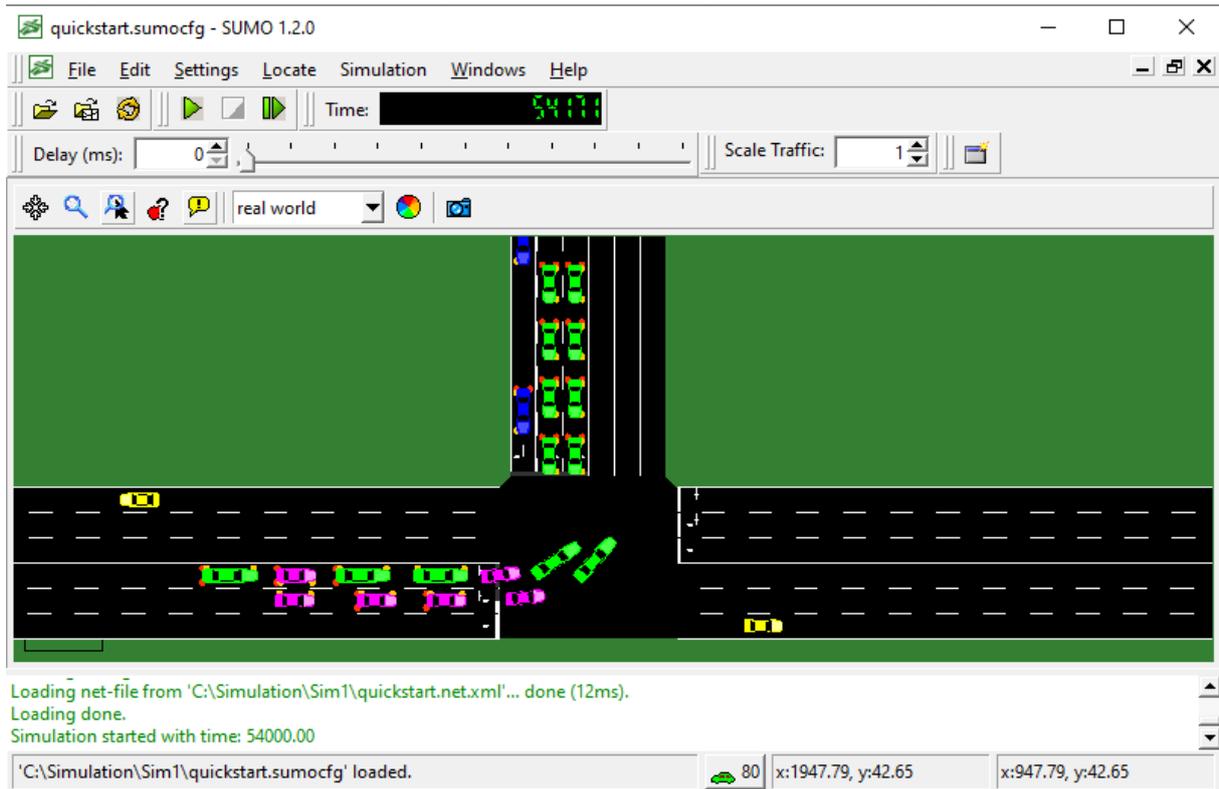


Figure III-1 Sumo

### III.2.1. Quel type de simulation de trafic est SUMO ?

SUMO est principalement une simulation microscopique de trafic routier continu dans l'espace. Il prend en charge le trafic terrestre multimodal et intermodal. SUMO modélise les véhicules individuels et leurs interactions à l'aide de modèles de comportement en matière de suivi de voiture, de changement de voie et d'intersection. Il utilise également des modèles piétonniers pour simuler le mouvement des personnes et leurs interactions avec les véhicules.

Pour permettre une simulation efficace de très grands scénarios, il est également possible d'exécuter SUMO en tant que simulation mésoscopique.[38]

### III.2.2. Exemples d'utilisation :

Depuis 2001, le paquet SUMO est utilisé dans le cadre de plusieurs projets de recherche nationaux et internationaux. Les applications incluses : Évaluation des feux de circulation, Choix d'itinéraire et re-routage, Évaluation des méthodes de surveillance du trafic, Simulation de communications véhiculaires, Prévisions de trafic.[38]

Dans le tableau suivant nous avons décrit quelque exemple de package :

### III.2.3. Applications incluses :

Tableau III-1 Le paquet comprend

Nom de l'application	Description courte
<b>SUMO</b>	La simulation microscopique sans visualisation; application en ligne de commande
<b>SUMO-GUI</b>	La simulation microscopique avec une interface utilisateur graphique
<b>SUMO-CONSOLE</b>	La simulation microscopique mode console
<b>NETCONVERT</b>	importateur et générateur de réseau; lit les réseaux routiers de différents formats et les convertit au format SUMO
<b>NETEDIT</b>	Un éditeur de réseau graphique.
<b>NETGENERATE</b>	Génère des réseaux abstraits pour la simulation SUMO

Plusieurs parties ont étendu le paquet SUMO au cours de leur travail et ont soumis leur code. S'ils ne font pas partie de la publication, ces contributions ne sont généralement pas testées fréquemment et peuvent être obsolètes. Les contributions suivantes sont incluses

Tableau III-2 suite des application

Nom de l'application	Description courte
<b>TraCI4Matlab</b>	Une interface Matlab pour la connexion et l'extension d'informations via TraCI Andres Acosta
<b>TraaS</b>	Une interface SOAP (web service) pour la connexion et l'extension d'informations via TraCI de [Mario Krumnow], également une bibliothèque client Java TraCI.
<b>LiSuM</b>	LiSuM est un middleware qui associe LISA + à SUMO pour exécuter un logiciel de contrôle du trafic réel dans SUMO.

### III.2.4. Critères de conception du logiciel :

Deux objectifs de conception majeurs sont abordés : le logiciel doit être rapide et portable. Pour cette raison, les toutes premières versions ont été développées pour être exécutées à partir de la ligne de commande uniquement. Aucune interface graphique n'a été fournie au départ et tous les paramètres ont dû être insérés manuellement. Cela devrait augmenter la vitesse d'exécution en laissant de côté la visualisation lente. En outre, en raison de ces objectifs, le logiciel a été divisé en plusieurs parties. Chacun d'entre eux a un but précis et doit être exécuté individuellement.

C'est quelque chose qui rend SUMO différent des autres packages de simulation où, par exemple, l'affectation d'utilisateur dynamique est effectuée dans la simulation elle-même, et non via une application externe comme ici.

Cette division permet une extension plus facile de chacune des applications du package, car chacune est plus petite qu'une application monolithique qui fait tout. En outre, il permet l'utilisation de structures de données plus rapides, chacune étant adaptée à l'objectif actuel, au lieu d'utiliser des structures compliquées et chargées en ballast.

Néanmoins, cela rend l'utilisation de SUMO un peu inconfortable par rapport à d'autres logiciels de simulation. Comme il reste encore beaucoup à faire, nous ne pensons pas pour le moment à une refonte vers une approche intégrée. [39]

### III.2.5. Quelques Contributeurs et participants

Tableau III-3 quelques participants

nom	Sujets / Contribution
Christian Rössel	Noyau de micro simulation initiale ; mise en œuvre initiale des détecteurs
Peter Wagner	Modèles, organisation, guide spirituel
Daniel Krajzewicz	tout
Julia Ringel	Modèles, organisation, guide spirituel
Eric Nicolay	Tout
Michael Behrisch	tout
Yun-Pang Wang	Assignation d'utilisateur

### III.2.6. Notions de base :

#### III.2.6.1. XML exemple :

Les éléments XML et les attributs sont montrés comme ceci. Leurs valeurs, si variables, <LIKE THIS>. Des exemples complets de fichiers XML sont présentés comme suit :

```
<myType>  
  <myElem myAttr1="0" myAttr2="0.0"/>  
  <myElem myAttr1="1" myAttr2="-500.0"/>  
</myType>
```

#### III.2.6.2. Types de données référencés :

<BOOL> : une valeur booléenne, utilisez "t" ou "true" et "f" ou "false" pour le codage

<INT> : une valeur entière, peut être négative

<UINT> : un entier non signé, doit être >= 0

<FLOAT> : un nombre à virgule flottante

<TIME> : heure, exprimée en secondes ; les fractions sont autorisées, par exemple "12.1"

<STRING> : n'importe quelle chaîne, mais utilisez uniquement des caractères ASCII

<ID> : une chaîne qui ne doit pas contenir les caractères suivants : '#'

### III.2.6.3. Types de fichiers référencés

<NETWORK\_FILE> : un fichier réseau SUMO construit par NETGENERATE ou NETCONVERT

<ROUTES\_FILE> : un fichier de routes SUMO tel que construit par DUAROUTER ou JTRROUTER ou manuellement

<TYPE\_FILE> : un fichier de type SUMO Edge construit à la main ou téléchargé

<OSM\_FILE> : un fichier OpenStreetMap tel qu'exporté depuis OpenStreetMap+

### III.2.6.4. Description de l'utilisation

Pour un bref aperçu, voir Simulation / Définition\_de\_base. Les détails du modèle et la documentation sur les fonctionnalités sont disponibles dans les rubriques Simulation.

### III.2.7. Les options

Vous pouvez utiliser un fichier de définition de schéma XML pour configurer une configuration SUMO : sumoConfiguration.xsd.

#### III.2.7.1. Table de « Configuration »

Tableau III-4 SUMO configuration [40]

Option	Description
<b>-c &lt;FILE&gt;</b> <b>--configuration-file &lt;FILE&gt;</b>	Charge la configuration nommée au démarrage
<b>-C &lt;FILE&gt;</b> <b>--save-configuration &lt;FILE&gt;</b>	Enregistre la configuration actuelle dans FILE
<b>--save-template &lt;FILE&gt;</b>	Enregistre un modèle de configuration (vide) dans FILE
<b>--save-schema &lt;FILE&gt;</b>	Enregistre le schéma de configuration dans FILE
<b>--save-commented &lt;BOOL&gt;</b>	Ajoute des commentaires au modèle, à la configuration ou au schéma enregistré. défaut: false

### III.2.7.2. Table de « Input »

Tableau III-5 Input[40]

Option	Description
<b>-n &lt;FILE&gt;</b> <b>--net-file &lt;FILE&gt;</b>	Charger la description du réseau routier à partir du fichier
<b>-r &lt;FILE&gt;</b> <b>--route-files &lt;FILE&gt;</b>	Charger les descriptions d'itinéraires à partir de FICHER (s)
<b>-a &lt;FILE&gt;</b> <b>--additional-files &lt;FILE&gt;</b>	Charger d'autres descriptions à partir de FICHER (s)
<b>-w &lt;FILE&gt;</b> <b>--weight-files &lt;FILE&gt;</b>	Charger d'autres descriptions à partir de FICHER (s)
<b>-x &lt;STRING&gt;</b> <b>--weight-attribute &lt;STRING&gt;</b>	Nom de l'attribut xml qui donne le poids de l'arête; défaut: traveltime
<b>--load-state &lt;FILE&gt;</b>	Charge un état de réseau à partir de FILE
<b>--load-state.offset &lt;TIME&gt;</b>	Décale tous les temps chargés d'un état sauvegardé du décalage donné; défaut: 0
<b>--load-state.remove-vehicles &lt;STRING&gt;</b>	Supprime les véhicules avec les identifiants donnés de l'état chargé

### III.2.7.1. Table de « Output »

Tableau III-6 Output[40]

Option	Description
<b>--write-license &lt;BOOL&gt;</b>	Inclure les informations de licence dans chaque fichier de sortie; défaut: false
<b>--output-prefix &lt;STRING&gt;</b>	Préfixe appliqué à tous les fichiers de sortie. La chaîne spéciale 'TIME' est remplacée par l'heure actuelle.
<b>--precision &lt;INT&gt;</b>	Définit le nombre de chiffres après la virgule pour la sortie en virgule flottante; défaut: 2
<b>--precision.geo &lt;INT&gt;</b>	Définit le nombre de chiffres après la virgule pour les sorties lon, lat; défaut: 6
<b>-H &lt;BOOL&gt;</b> <b>--human-readable-time &lt;BOOL&gt;</b>	Ecrivez les valeurs horaires en heure: minute: seconde ou jour: heure: minute: seconde plutôt que seconde; défaut: false
<b>--netstate-dump &lt;FILE&gt;</b>	Sauvegarder les états complets du réseau dans FILE
<b>--netstate-dump.empty-edges &lt;BOOL&gt;</b>	Écrivez également les bords complètement vides lors du déchargement; défaut: false
<b>--netstate-dump.precision &lt;INT&gt;</b>	Ecrire les positions et les vitesses avec la précision donnée (défaut 2); défaut: 2
<b>--emission-output &lt;FILE&gt;</b>	Enregistrer les valeurs d'émission de chaque véhicule
<b>--emission-output.precision &lt;INT&gt;</b>	Ecrire les valeurs d'émission avec la précision donnée (par défaut 2); défaut: 2

### III.2.7.2. Table de « Time »

Tableau III-7 Time[40]

Option	Description
<b>-b &lt;TIME&gt;</b> <b>--begin &lt;TIME&gt;</b>	Définit le temps de début en secondes; La simulation commence à ce moment; défaut: 0
<b>-e &lt;TIME&gt;</b> <b>--end &lt;TIME&gt;</b>	Définit l'heure de fin en secondes; La simulation se termine à ce moment; défaut: -1
<b>--step-length &lt;TIME&gt;</b>	Définit la durée du pas en secondes; défaut: 1

### III.2.7.3. Table de « Processing »

Tableau III-8 processing[40]

Option	Description
<b>--no-internal-links &lt;BOOL&gt;</b>	Désactiver (jonction) les liens internes; défaut: false
<b>--threads &lt;INT&gt;</b>	Définit le nombre de threads pour la simulation parallèle. défaut: 1
<b>--lateral-resolution &lt;FLOAT&gt;</b>	Définit la résolution en m lors du traitement du positionnement latéral dans une voie (-1 étant tous les véhicules conduisant au centre de leur voie; par défaut: -1
<b>-s &lt;HEURE&gt;</b> <b>--route-steps &lt;TIME&gt;</b>	Charge les itinéraires pour le nombre de secondes à venir; défaut: 200

### III.2.7.4. Table de « Traci server »

Tableau III-9 Traci Server[40]

Option	Description
<b>--remote-port &lt;INT&gt;</b>	Active TraCI Server s'il est défini; défaut: 0
<b>--num-clients &lt;INT&gt;</b>	Nombre attendu de clients en connexion; défaut: 1

### III.2.7.5. Table da « Gui only »

Tableau III-10 Gui only[40]

Option	Description
<b>-g &lt;FILE&gt;</b> <b>--gui-settings-file &lt;FILE&gt;</b>	Charger les paramètres de visualisation depuis le fichier
<b>-Q &lt;BOOL&gt;</b> <b>--quit-on-end &lt;BOOL&gt;</b>	Quitte l'interface graphique lorsque la simulation s'arrête. défaut: false
<b>-G &lt;BOOL&gt;</b> <b>--game &lt;BOOL&gt;</b>	Démarrez l'interface graphique en mode jeu. défaut: false
<b>--game.mode &lt;STRING&gt;</b>	Sélectionnez le type de jeu ('tls', 'drt'); défaut: tls
<b>-S &lt;BOOL&gt;</b> <b>--start &lt;BOOL&gt;</b>	Lancer la simulation après le chargement ; défaut : false
<b>-B &lt;STRING&gt;</b> <b>--breakpoints &lt;STRING&gt;</b>	Utilisez TIME [] comme moments où la simulation doit s'arrêter
<b>--edgedata-files &lt;FILE&gt;</b>	Poids de la rive / de la voie pour la visualisation à partir de FILE
<b>-D &lt;BOOL&gt;</b> <b>--demo &lt;BOOL&gt;</b>	Redémarrez la simulation après avoir terminé (mode démo); défaut: false
<b>-T &lt;BOOL&gt;</b> <b>--disable-textures &lt;BOOL&gt;</b>	Ne chargez pas d'images d'arrière-plan; défaut: false
<b>--registry-viewport &lt;BOOL&gt;</b>	Charger la fenêtre d'affichage actuelle à partir du registre; défaut: false
<b>--window-size &lt;STRING&gt;</b>	Créer une fenêtre initiale avec la taille x, y donnée
<b>--window-pos &lt;STRING&gt;</b>	Créer la fenêtre initiale à la position x, y donnée
<b>--tracker-interval &lt;FLOAT&gt;</b>	La période d'agrégation pour les fenêtres de suivi de valeur; défaut: 1
<b>--osg-view &lt;BOOL&gt;</b>	Commencez avec une vue OpenSceneGraph au lieu de la vue 2D normale; défaut: false
<b>--gui-testing &lt;BOOL&gt;</b>	Activer le délai d'attente pour la reconnaissance d'écran; défaut: false
<b>--gui-testing-debug &lt;BOOL&gt;</b>	Activer les messages de sortie lors du test de l'interface graphique; défaut: false

#### III.2.7.1. Table de « Batterie »

Tableau III-11 batterie[40]

Option	Description
<b>--device.battery.probability &lt;FLOAT&gt;</b>	La probabilité qu'un véhicule dispose d'un dispositif 'batterie'; défaut: -1
<b>--device.battery.explicit &lt;STRING&gt;</b>	Assigne un périphérique "batterie" à des véhicules nommés
<b>--device.battery.deterministic &lt;BOOL&gt;</b>	Les dispositifs de type "batterie" sont définis comme déterministes à l'aide d'une fraction de 1000; défaut: false

### III.2.8. Traci

#### III.2.8.1. Introduction à TraCI :

TraCI est le terme court pour "Interface de contrôle du trafic". Donnant accès à une simulation de circulation routière en cours d'exécution, il permet de récupérer les valeurs d'objets simulés et de manipuler leur comportement "en ligne".

#### III.2.8.2. Utiliser TraCI :

Démarrage de SUMO : TraCI utilise une architecture client / serveur basée sur TCP pour fournir un accès à SUMO. Ainsi, SUMO agit en tant que serveur démarré avec des options de ligne de commande supplémentaires : `--remote-port <INT>` ou `<INT>` est le port que SUMO écoutera pour les connexions entrantes.

Lorsqu'il est lancé avec l'option `--remote-port <INT>`, SUMO prépare uniquement la simulation et attend que toutes les applications externes se connectent et prennent le contrôle. Veuillez noter que l'option `--end <TIME>` est ignorée lorsque SUMO s'exécute en tant que serveur TraCI. SUMO s'exécute jusqu'à ce que le client demande la fin de la simulation.[41]

Lors de l'utilisation de SUMO-GUI en tant que serveur, la simulation doit être lancée à l'aide du bouton de lecture ou en définissant l'option `--start` avant que les commandes TraCI ne soient traitées.

#### III.2.8.3. Plusieurs clients :

Le nombre de clients pouvant se connecter peut-être indiqué en option supplémentaire `--num-clients <INT>`, où 1 correspond à la valeur par défaut. Veuillez noter que dans les scénarios multi-clients, vous devez spécifier explicitement l'ordre d'exécution des clients à l'aide de la commande `SetOrder`.

Chaque client doit spécifier une valeur entière unique (mais arbitraire) et les commandes du client seront traitées dans l'ordre, de la valeur la plus basse à la valeur la plus élevée dans chaque étape de la simulation.

Les clients sont automatiquement synchronisés après chaque étape de la simulation. Cela signifie que la simulation n'avance pas à l'étape suivante tant que tous les clients n'ont pas appelé la commande 'simulationStep'. De plus, la commande `simulationStep` ne restitue le contrôle au client qu'une fois la simulation avancée.[41]

#### III.2.8.4. Utiliser SUMO comme une bibliothèque

TraCI est normalement utilisé pour coupler plusieurs processus : un processus serveur SUMO et un ou plusieurs processus clients TraCI. Libsumo peut également être utilisé pour incorporer SUMO en tant que bibliothèque dans le processus client. Cela permet d'utiliser les mêmes signatures de méthode que dans les bibliothèques clientes, tout en évitant la surcharge de la communication par socket. Libsumo prend en charge la génération de bibliothèques client à l'aide de SWIG et peut donc être utilisé avec un

grand nombre de langages de programmation. Les liaisons C ++, Java et Python sont incluses lors du téléchargement d'un sumo-build.[41]

#### III.2.8.5. Exemple de description

Le scénario consiste en une seule route qui est traversée par un sentier. Le passage à niveau est contrôlé par un feu de signalisation et doit changer lorsque les piétons le déclenchent via un bouton-poussoir. Pour modéliser le comportement en appuyant sur un bouton, on suppose qu'un piéton qui arrive à un feu rouge active le bouton s'il n'est pas encore activé. Pour éviter toute interférence indue avec la circulation automobile, une durée minimale de vert de 15 secondes est prévue pour la phase verte du véhicule.

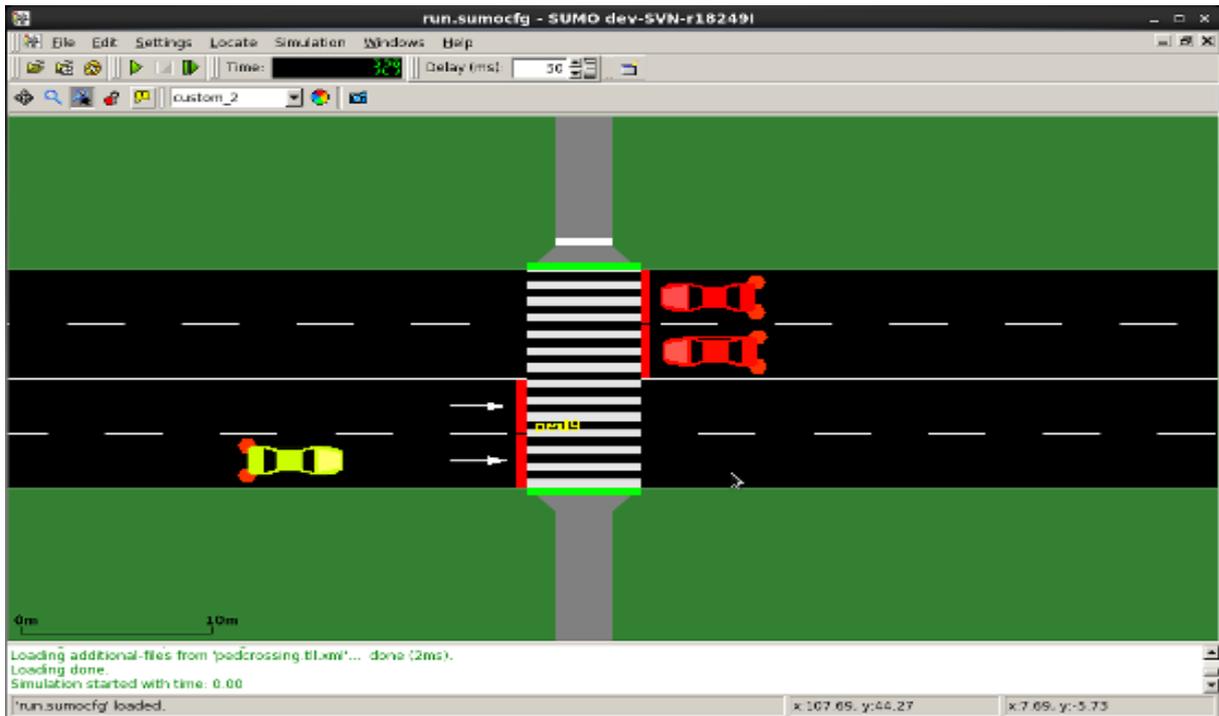


Figure III-2 Exemple

### III.3. SiMMU : Simulation Microscopique de la Mobilité Urbaine

Dans ce cas, nous utilisons un système multi-agents. Il s'agit de programmer des agents dans un environnement défini. Ces agents ont des interactions entre eux et avec leur environnement en fonction de règles simples définies par le modélisateur. Nous avons modélisé le système à partir d'un certain nombre de données. Les agents évoluent dans un environnement. Nous avons décrit les rôles de chaque agent, plus précisément ses propriétés et son comportement, plus sa relation avec les autres du simulateur conçu.

Pour atteindre notre objectif, plus précisément la modélisation et la simulation de la mobilité urbaine et de la communication V2X, nous utilisons :

- SUMO comme l'environnement urbain de simulation dans lequel nous visualisons le réseau des routes ainsi que le trafic urbain,

- Java/JADE pour : (i) contrôler la simulation, (ii) permettre aux chercheurs d'implémenter leurs propres systèmes/modèles, (iii) fournir des outils de suivi de l'état de simulation et des systèmes implémentés, (iv) sauvegarder et visualiser les résultats des simulations.

Après effectuer les modifications nécessaires sur SUMO, vous devez implémenter un système multi-agents constitué des agents suivants (voir le tableau ci-dessous) :

Tableau III-12 Composants du systèmes proposé

Agents	Rôles
<b>SimControlAgent</b>	<p>C'est l'agent chargé du contrôle de la simulation, plus précisément il est chargé de (à l'aide d'une interface graphique) :</p> <ul style="list-style-type: none"> <li>• Charger le scenario de simulation (sélectionner et importer le lien du fichier *.cfg contenant la configuration SUMO de la simulation),</li> <li>• Charger la configuration des agents (un fichier XML),</li> <li>• Lancer la simulation (en mode graphique ou console),</li> <li>• Mettre la simulation en pause, ou arrêter et terminer la simulation.</li> <li>• Suivre l'état d'un véhicule, et enregistre le résultat (le changement de son état pendant la simulation).</li> <li>• Perception (vitesse, distance, poids, longueur, etc.)</li> <li>• Décision (vitesse, direction)</li> <li>• Action (accélération, décélération)</li> <li>• Communication</li> </ul>
<b>DriverAgent</b>	<p>Cet agent est chargé d'implémenter le système conçu par un chercheur pour accomplir une tâche donnée (ex. contrôler la vitesse, communiquer certaines informations avec d'autres agents et notamment avec d'autres véhicules, etc.).</p>
<b>HAgent</b>	<p>Cet agent permet à un acteur humain de contrôle manuellement la vitesse d'un véhicule et d'interagir également avec l'environnement de simulation.</p>

Un diagramme de séquence décrivant les interactions entre les agents pour accomplir certaines tâches,

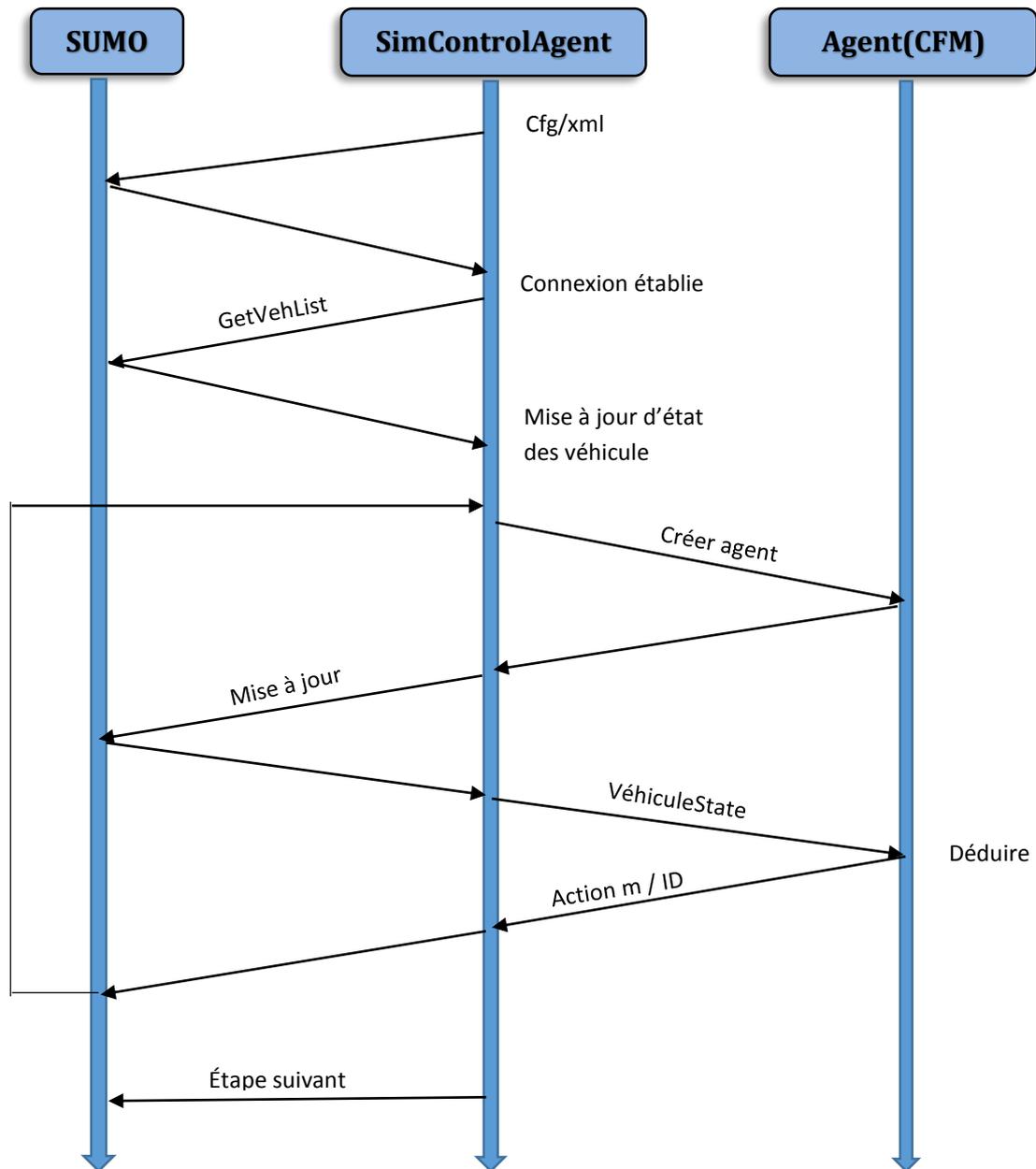


Figure III-3 Diagramme de séquence décrivant les interactions entre les agents

Dans ce cas nous avons décrit quelques instructions invoquées par SimControlAgent pour communiquer avec SUMO (création d'une connexion, passage de paramètres au TraSS, etc).

L'agent SimControlAgent extends Agent est caractérisé par les attributs suivants :

- SumoTraciConnection conn; (une connexion vers SUMO)
- SimControlHMI HMI : un objet d'une classe SimControlHMI extends JFrame.
- String sumo\_bin\_gui=lance la simulation en mode graphique « E:/Simulation/sumo-1.2.0/bin/sumo-gui.exe" »
- String sumo\_bin\_console= lancer la simulation on mode console"E:/Simulation/sumo-1.2.0/bin/sumo.exe"
- String sim\_config\_file (lien vers le fichier cfg – la configuration SUMO de la simulation)
- String agent\_config\_file (lien vers le fichier xml – la configuration des agents)
- Updatevehiculeslist : mise à jour la liste des véhicules
- Getnlistlength : récupérer la liste des véhicules
- GetSpeed () : récupérer la vitesse d'un véhicule a instance donné
- Conn.isClosed() : fermer la connexion

#### **III.4. Conclusion**

Dans ce chapitre, nous avons présenté Une approche multi-agent pour la simulation de la mobilité urbaine, on premier lieu nous avons discuté sur le simulateur SUMO et sa fonctionnalité, après on a détaillé les étapes de réalisation de notre simulateur à savoir les besoins fonctionnels. Dans le chapitre suivants nous avons présenté l'Expérimentation et les résultats de la simulation.

# Chapitre 4

## Expérimentation et résultats

### IV.1. Introduction

Dans ce chapitre nous allons présenter l'environnement matériels et logiciels de simulation et la configuration des paramètres d'entrée, ainsi que nous décrirons la simulation et les étapes de son déroulement. Enfin l'interprétation des résultats des simulations.

### IV.2. Environnement de travail

Nous allons détailler les outils utilisés dans la réalisation de notre simulation :

#### V.2.1. Environnement matériel

La simulation a été réalisée sur un ordinateur ASUS dont la configuration est :

- Système D'exploitation : Windows
- Mémoire RAM : 6 GO
- Processeur : Intel Core i5-5200U
- Carte graphique : Intel Nvidia Geforce GT 940M

#### IV.2.1. Environnement logiciel

##### IV.2.1.1. Eclipse

Éclipse est un environnement de développement intègre (IDE) libre extensible, universel et multi langage, permettant potentiellement de créer des projets de développement mettant en oeuvre n'importe quel langage de programmation. La spécification d'éclipse IDE vient du fait de son architecture totalement développée autour de la notion de plugin (extension), toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plugin.[42]

##### IV.2.1.2. Java

C'est un langage de programmation orienté objet développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh). Java donne aussi la possibilité de développer des programmes pour téléphones portables. Enfin, ce langage peut être

utiliser sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp).[43]

#### IV.2.1.3. Jade

JADE est une plateforme de programmation multi-agent implémentée en Java. Les agents qui tournent sous JADE communiquent via le langage Agent Communication Language ou ACL.[44]

#### IV.2.1.4. SUMO (Simulation of Urban MObility)

Le logiciel ouvert SUMO est le simulateur de trafic gratuit le plus communément utilisé dans les récents travaux de recherche sur les réseaux véhiculaires. C'est un simulateur hautement portable capable de fournir des schémas de mobilité précis. Il prend en compte plusieurs types de véhicules, les feux de signalisation, les intersections avec priorité (priorité à droite par exemple), le changement de voies, les connexions de voie à voie, etc. SUMO utilise le modèle de suivi de véhicules de Krauss pour simuler le comportement des conducteurs et Random Way point pour le mouvement sur les routes [37].

### IV.3. Création des scénarios avec SUMO

Les fichiers .XML générés à partir d'une interface graphique java qui sont : nod.xml, edg.xml et fows.xml, contiennent les données du réseau routier, ces informations sont utilisées pour créer un scénario avec SUMO.

- Node file : C'est un fichier composé d'un ensemble de nœud du réseau routier, ces derniers sont définis par les coordonnées x et y et un identificateur (id)

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <nodes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3  xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/nodes_file.xsd">
4  <node id="91" x="-1000.0" y="1000.0" />
5  <node id="92" x="-1000.0" y="0.0" />
6  <node id="93" x="3000.0" y="0.0" />
7  <node id="94" x="+3000.0" y="+1000.0" />
8  <node id="911" x="-500.0" y="+1000.0" />
9  <node id="912" x="-500.0" y="0.0" />
10 <node id="913" x="+2500.0" y="0.0" />
11 <node id="914" x="2500.0" y="+1000.0" />
12 <node id="1" x="0.0" y="+1000.0" />
13 <node id="2" x="0.0" y="0.0" />
14 <node id="3" x="+1000.0" y="0.0" />
15 <node id="4" x="+2000.0" y="0.0" />
16 <node id="5" x="+2000.0" y="+1000.0" />
17 <node id="6" x="+1000.0" y="+1000.0" />
18 </nodes>

```

Figure IV-1 Nod.XML

- Edge file : C'est un fichier qui relie les nœuds par des routes. Il est spécifié par un identifiant unique (id), un nœud source, nœud destination, priorité, nombre de voies et la vitesse maximal autorisée

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <edges xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/edges_file.xsd">
4   <edge id="D1" from="911" to="91" type="a"/>
5   <edge id="D2" from="91" to="911" type="b"/>
6   <edge id="D3" from="912" to="92" type="a"/>
7   <edge id="D4" from="92" to="912" type="b"/>
8   <edge id="D5" from="913" to="93" type="a"/>
9   <edge id="D6" from="93" to="913" type="b"/>
10  <edge id="D7" from="914" to="94" type="a"/>
11  <edge id="D8" from="94" to="914" type="b"/>
12  <edge id="L1" from="1" to="911" type="a"/>
13  <edge id="L2" from="911" to="1" type="b"/>
14  <edge id="L3" from="2" to="912" type="a"/>
15  <edge id="L4" from="912" to="2" type="b"/>
16  <edge id="L5" from="4" to="913" type="a"/>
17  <edge id="L6" from="913" to="4" type="b"/>
18  <edge id="L7" from="5" to="914" type="a"/>
19  <edge id="L8" from="914" to="5" type="b"/>
20  <edge id="L9" from="5" to="6" type="a"/>
21  <edge id="L10" from="6" to="5" type="a"/>
22  <edge id="L11" from="6" to="1" type="a"/>
23  <edge id="L12" from="1" to="6" type="a"/>
24  <edge id="L13" from="3" to="2" type="a"/>
25  <edge id="L14" from="2" to="3" type="a"/>
26  <edge id="L15" from="6" to="3" type="c"/>
27  <edge id="L16" from="3" to="6" type="c"/>
28  <edge id="L17" from="4" to="3" type="a"/>
29  <edge id="L18" from="3" to="4" type="a"/>
30 </edges>

```

Figure IV-2 Edge.XML

- Net file : C'est un fichier généré à partir de la commande NETCONVERT qui prend en argument les fichiers edg.xml et nod.xml. La commande de NETCONVERT permet de lire ces fichiers et de les convertir dans le format SUMO.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/netconvertConfiguration.xsd">
5
6   <input>
7     <node-files value="quickstart.nod.xml"/>
8     <edge-files value="quickstart.edg.xml"/>
9     <connection-files value="quickstart.con.xml"/>
10    <type-files value="quickstart.typ.xml"/>
11  </input>
12
13  <output>
14    <output-file value="quickstart.net.xml"/>
15  </output>
16
17  <processing>
18    <no-turnarounds value="true"/>
19  </processing>
20
21  <report>
22    <verbose value="true"/>
23  </report>
24
25 </configuration>

```

Figure IV-3 net.cfg

- Route file : un fichier de routes SUMO tel que construit par la commande DUAROUTER qui permet de calculer les itinéraires les plus rapides à travers le réseau.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <routes>
3   <vType accel="1.5" decel="4.5" id="carAuto" length="5" maxSpeed="36.1" vClass="passenger" carFollowModel="SCFM" roadSlip="0.4"/>
4   <vType accel="3.0" decel="4.5" id="CarA" length="5.0" minGap="2.5" maxSpeed="36.1" sigma="0.5" />
5   <vType accel="2.0" decel="4.5" id="CarB" length="5.0" minGap="2.5" maxSpeed="36.1" sigma="0.5" />
6   <vType accel="1.0" decel="4.5" id="CarC" length="6.0" minGap="2.5" maxSpeed="20.0" sigma="0.5" />
7   <vType accel="0.8" decel="4.5" id="CarD" length="7.0" minGap="2.5" maxSpeed="15.0" sigma="0.5" />
8
9   <route id="route01" edges="D2 L2 L12 L10 L7 D7"/>
10  <route id="route02" edges="D2 L2 L12 L15 L18 L5 D5"/>
11  <route id="route03" edges="D2 L2 L12 L15 L13 L3 D3"/>
12  <route id="route04" edges="D4 L4 L14 L18 L5 D5"/>
13  <route id="route05" edges="D4 L4 L14 L16 L10 L7 D7"/>
14  <route id="route06" edges="D4 L4 L14 L16 L11 L1 D1"/>
15  <route id="route07" edges="D6 L6 L17 L13 L3 D3"/>
16  <route id="route08" edges="D6 L6 L17 L16 L11 L1 D1"/>
17  <route id="route09" edges="D6 L6 L17 L16 L10 L7 D7"/>
18  <route id="route10" edges="D8 L8 L9 L11 L1 D1"/>
19  <route id="route11" edges="D8 L8 L9 L15 L13 L3 D3"/>
20  <route id="route12" edges="D8 L8 L9 L15 L18 L5 D5"/>
21
22  <flow begin="54000" departPos="free" id="veh00" period="02" number="10" route="route01" type="CarA" color="1,0,0"/>
23  <flow begin="54000" departPos="free" id="veh01" period="05" number="10" route="route02" type="CarB" color="0,1,0"/>
24  <flow begin="54000" departPos="free" id="veh02" period="05" number="5" route="route03" type="CarC" color="0,0,1"/>

```

Figure IV-4 route file

- Type file : un fichier de type SUMO Edge construit à la main ou téléchargé

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <types>
3   <type id="a" priority="3" numLanes="3" speed="13.889"/>
4   <type id="b" priority="3" numLanes="2" speed="13.889"/>
5   <type id="c" priority="2" numLanes="3" speed="13.889"/>
6 </types>

```

Figure IV-5 type file

- Sumocfg file : C'est le fichier résultant des deux fichiers Route file et Net file, Sumocfg est donc le fichier de configuration de SUMO

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/sumoConfiguration.xsd">
5
6   <input>
7     <net-file value="quickstart.net.xml"/>
8     <route-files value="quickstart.rou.xml"/>
9   </input>
10
11  <time>
12    <begin value="54000"/>
13    <end value="54900"/>
14  </time>
15
16  <processing>
17    <time-to-teleport value="-1"/>
18  </processing>
19
20 </configuration>
21

```

Figure IV-6 sumo.cfg

#### IV.4. Présentation de simulateur

Nous allons désormais consacrer cette partie du chapitre à la présentation de simulateur :

##### IV.4.1. Communication vers SUMO

Pour établir la connexion avec notre application et le simulateur sumo nous avons utilisé un TraCI(sumo) et TraCI4J (java), la communication entre les deux simulateurs peut se faire avec des messages (requête/ réponse).

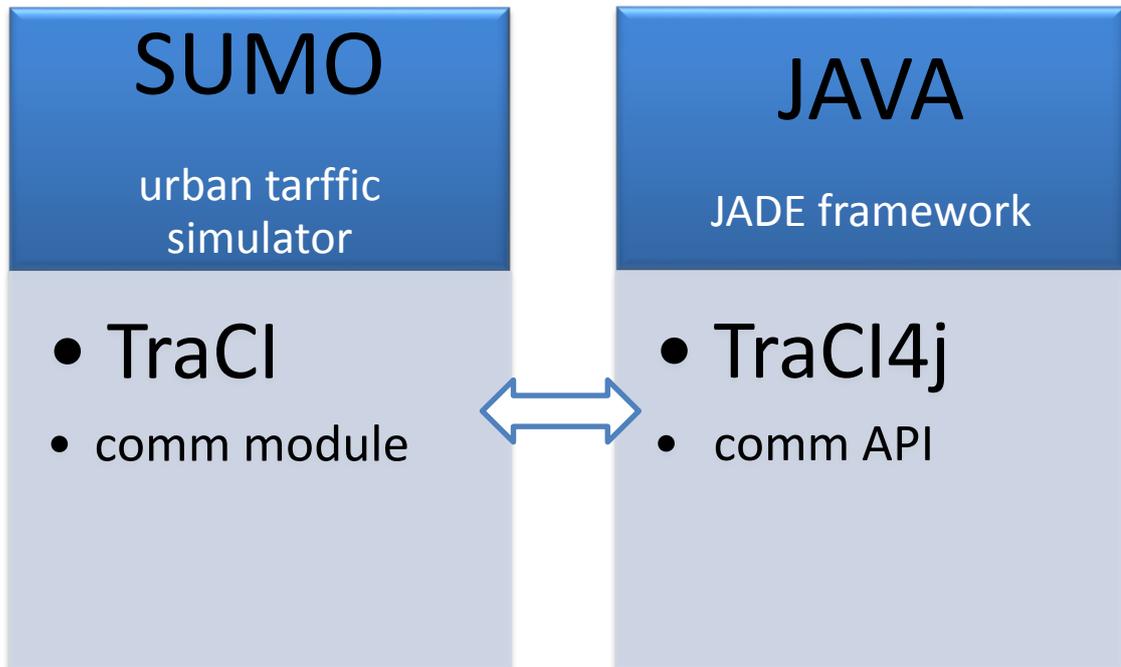


Figure IV-7 la communication d'un SMA avec SUMO

##### IV.4.2. Interface graphique pour la simulation

- Nous allons concevoir une interface graphique pour l'agent SimControlAgent comme suit (voir la figure ci-dessous) :



Figure IV-8 interface graphique

Les boutons comportent les différents paramètres :

- 1ier bouton : charge le scenario de simulation (\*.cfg)
- 2ième bouton : charge la configuration des agents (\*.xml)
- 3ième bouton : lance la simulation en mode graphique
- 4ième bouton : lance la simulation en mode console
- 5ième bouton : pause la simulation
- 6ième bouton : arrête la simulation
- 7ième bouton : relance le dernier scenario sélectionné,
- 8ième bouton : sauvegarde le résultat de simulation.

#### IV.4.3. Lancer la simulation avec SUMO

- Charger le scenario de simulation (nom.cfg) : Charger le scenario de simulation (sélectionner et importer le lien du fichier \*.cfg contenant la configuration de SUMO), puis Charger la configuration des agents (un fichier XML).

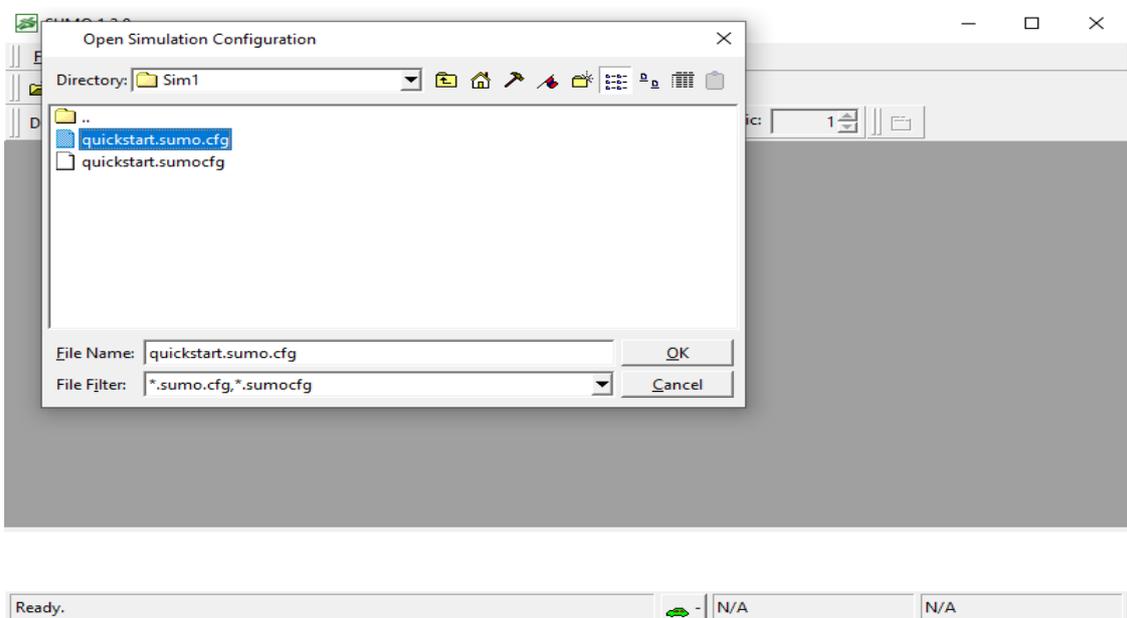


Figure IV-9 charger un fichier cfg

- Lancer la simulation en cliquant sur le bouton Start

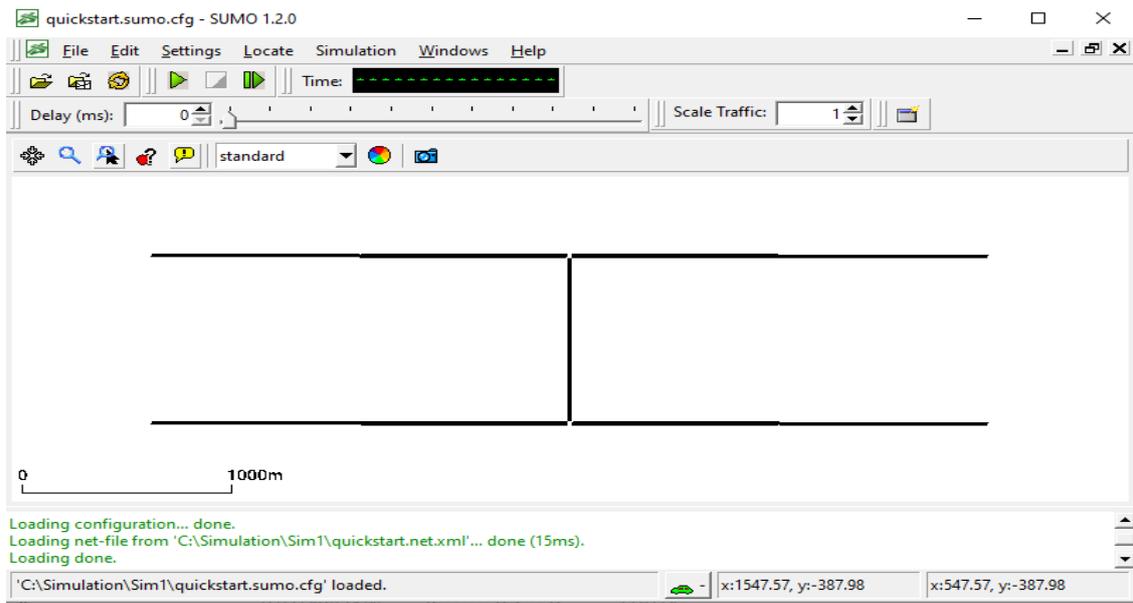


Figure IV-10 lancer la simulation

- Création d'un véhicule : ce bouton permet de localiser la position de la voiture et choisir la voiture par sélectionné leur id.

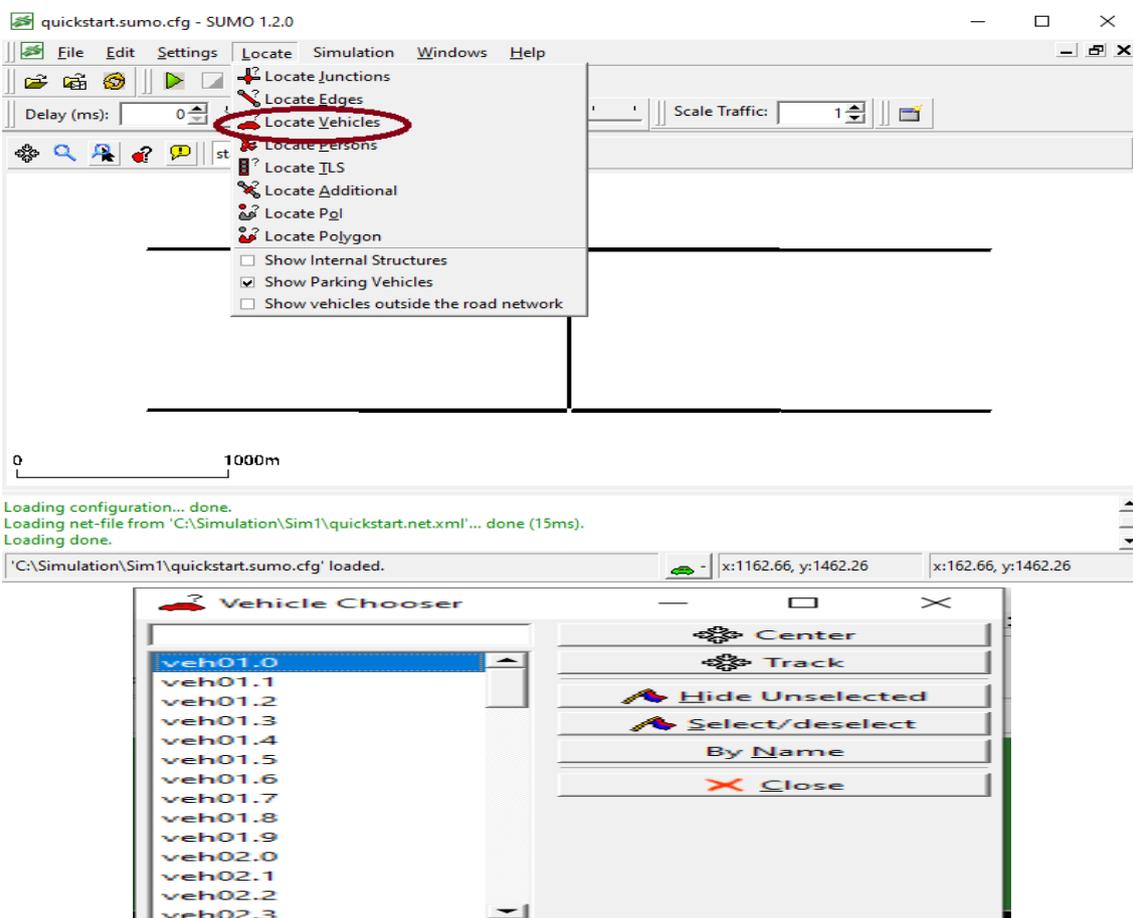


Figure IV-11 locate véhicule

- Voir la position de véhicule sur la route : Suivre l'état d'un véhicule, et enregistrer le résultat (le changement de son état pendant la simulation).

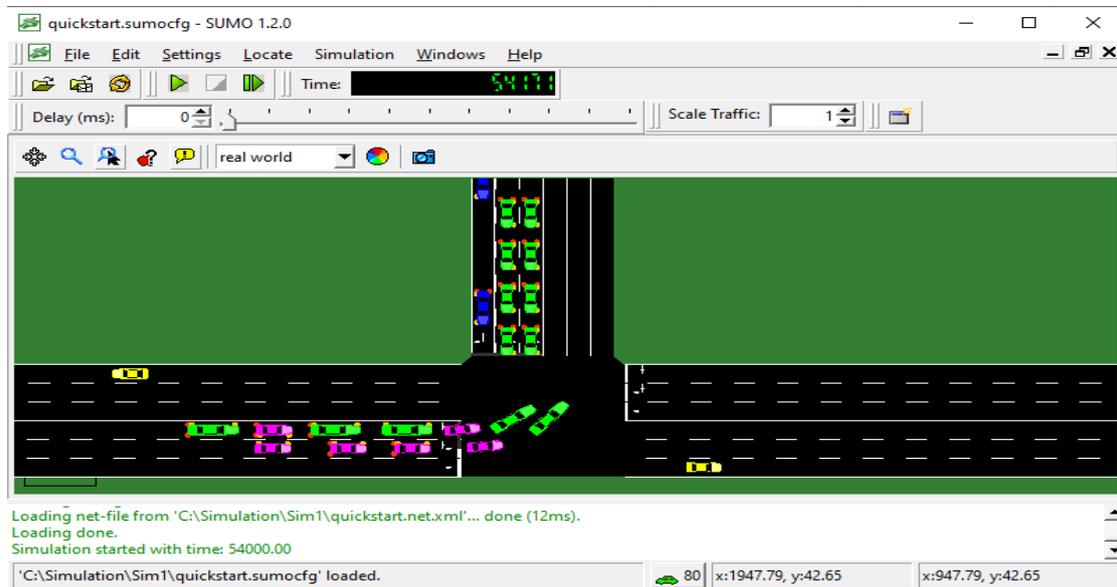


Figure IV-12 position véhicule

## IV.5. Exemple des modèles de suivi de véhicules

Nous avons implémenté trois modèles de suivi de véhicules dans notre projet java qui sont :

- Modèle de Kerner [19] : (voir l'équation suivant)

$$V_{n+1} = \max(0, v_{free}, v'_{n+1} + \delta_n, v_n + \alpha\tau, v_{s,n}),$$

$$X_{n+1} = X_n + V_{n+1}\tau, \quad \text{Équation IV-1}$$

Nous avons implémenté le code suivant :

```

5 public class Kerner extends MSCF{
6     private double tau = 1.0;
7
8     protected void setup() {
9         Object[] args = getArguments();
10        for (int i = 0 ; i < args.length ; i++) {
11            String arg = getarg(args[i].toString());
12            switch (arg) {
13                case "tau":
14                    this.setTau(getDoubleParam(args[i].toString()));
15                    break;
16                default:
17                    System.out.println("Parameter '"+arg+"' does not correspond to any of the KraussOrig1 CFM's arguments.");
18            }
19        }
20        addBehaviour(new TickerBehaviour(this, 50) {
21            @Override
22            protected void onTick() {
23                if (receiveVehicleState(myAgent)){
24                    setAction(defineNextAction(followSpeed()));
25                    sendAction(myAgent);
26                }
27            }
28        });
29
30        public double getTau() {
31            return tau;
32        }
33        public void setTau(double tau) {
34            if (tau>=0) {
35                this.tau = tau;
36            }
37        }
38        @Override
39        public double followSpeed() {
40            if (getLeadSpeed()<0.){
41                return getMaxSpeed();
42            }else{
43                if (getGap() == 0.){
44                    return 0.;
45                }else{
46                    double G = MAX2(0., (SPPED2DIST(getSpeed()) + 0 / getAccel() * getSpeed() * (getSpeed() - getLeadSpeed())));
47                    double vcond = getGap() > G ? getSpeed() + ACCEL2SPEED(getAccel()) : getSpeed() + MAX2(ACCEL2SPEED(getDecel()),
48                        MIN2(ACCEL2SPEED(getAccel()), getLeadSpeed() - getSpeed()));
49                    double va = MAX2(0., MIN3(maxNextSpeed(), vsafe(), vcond));
50                    return (MAX2(0., MIN4(maxNextSpeed(), va, getSpeed() + ACCEL2SPEED(getAccel()), vsafe())));
51                }
52            }
53
54            public double defineNextAction(double followSpeed) {
55                double nextAction = followSpeed - getSpeed();
56                if (nextAction>0) {
57                    return MIN2(nextAction,this.getAccel());
58                }else {
59                    return MAX2(nextAction,-this.getDecel());
60                }
61            }
62
63            public double vsafe() {
64                if (getLeadSpeed()<0.){
65                    return getMaxSpeed();
66                }else{
67                    if (getGap() == 0.){
68                        return 0.;
69                    }else{
70                        return super.getLeadSpeed()+((getGap()-getLeadSpeed()*getTau())/
71                            |((getLeadSpeed()+getSpeed())/(2*Math.abs(getLeadMaxDecel()))+getTau())|);
72                    }
73                }
74            }
75        }
76    }
77 }

```

Figure IV-13 Modèle de Kerner[19]

- Modèle de Krauß [13]: (voir l'équation suivant)

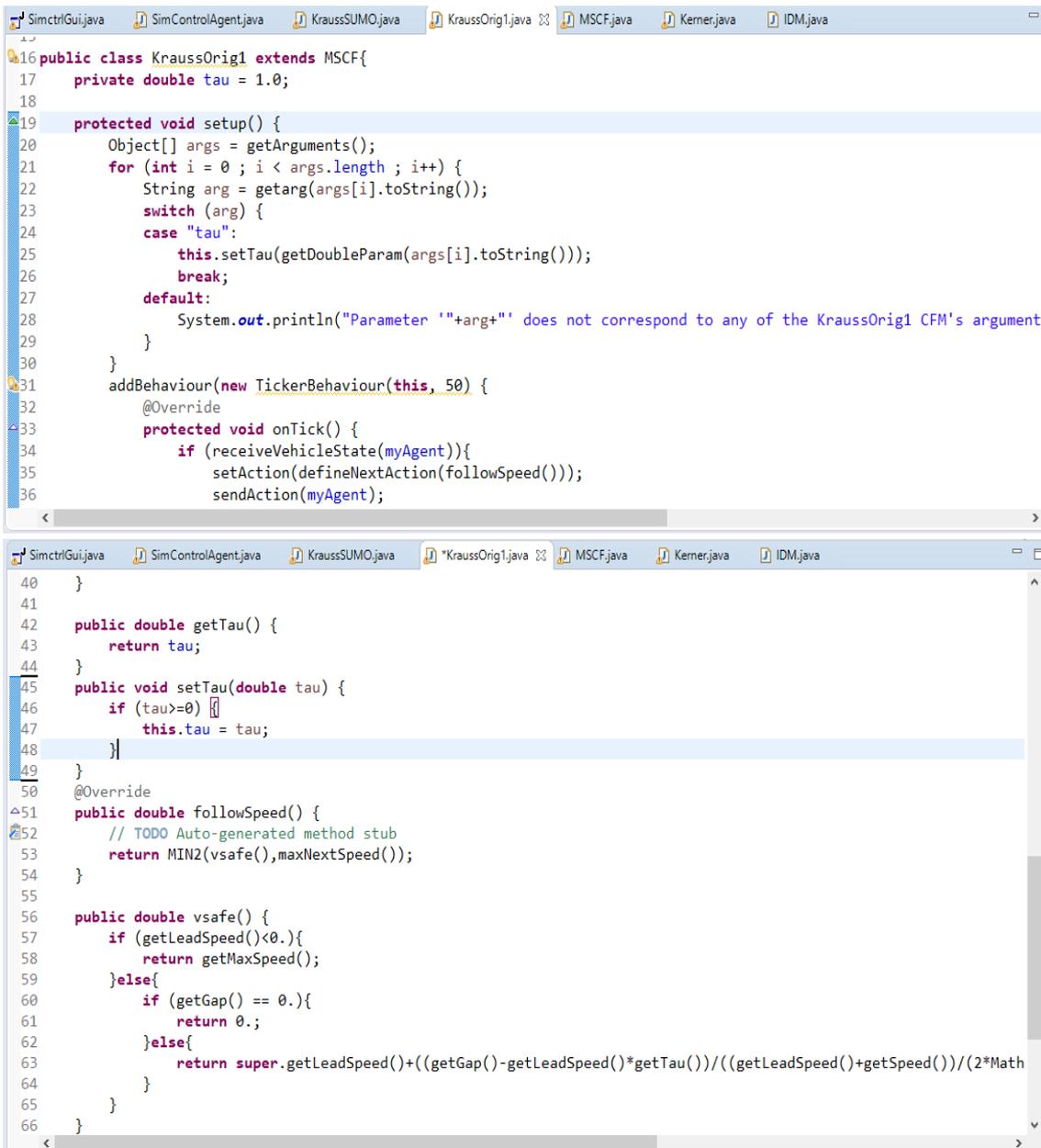
$$v_{safe}(t) = v_i(t) + g(t) - \frac{g_{des}(t)}{\tau b + \tau}$$

$$v_{des}(t) = \min[v_{max}, v(t) + a(v)\Delta t, v_{safe}(t)], \quad \text{Équation IV-2}$$

$$v(t + \Delta t) = \max[0, v_{des}(t) - n],$$

$$x(t + \Delta t) = x(t) + v\Delta t,$$

Nous avons implémenté le code suivant :



```

16 public class KraussOrig1 extends MSCF{
17     private double tau = 1.0;
18
19     protected void setup() {
20         Object[] args = getArguments();
21         for (int i = 0 ; i < args.length ; i++) {
22             String arg = getarg(args[i].toString());
23             switch (arg) {
24                 case "tau":
25                     this.setTau(getDoubleParam(args[i].toString()));
26                     break;
27                 default:
28                     System.out.println("Parameter '"+arg+"' does not correspond to any of the KraussOrig1 CFM's argument
29             }
30         }
31         addBehaviour(new TickerBehaviour(this, 50) {
32             @Override
33             protected void onTick() {
34                 if (receiveVehicleState(myAgent)){
35                     setAction(defineNextAction(followSpeed()));
36                 }
37             }
38         });
39     }
40 }
41
42 public double getTau() {
43     return tau;
44 }
45 public void setTau(double tau) {
46     if (tau>=0) {
47         this.tau = tau;
48     }
49 }
50 @Override
51 public double followSpeed() {
52     // TODO Auto-generated method stub
53     return MIN2(vsafe(),maxNextSpeed());
54 }
55
56 public double vsafe() {
57     if (getLeadSpeed()<0.){
58         return getMaxSpeed();
59     }else{
60         if (getGap() == 0.){
61             return 0.;
62         }else{
63             return super.getLeadSpeed()+((getGap()-getLeadSpeed()*getTau())/((getLeadSpeed()+getSpeed()))/(2*Math
64         }
65     }
66 }

```

```

67
68 public double defineNextAction(double followSpeed) {
69     double nextAction = followSpeed - getSpeed();
70     if (nextAction>0) {
71         return MIN2(nextAction,this.getAccel());
72     }else {
73         return MAX2(nextAction,-this.getDecel());
74     }
75 }
76 }
77

```

Figure IV-14 Modèle de Krauß [13]

- Modèle de KraussSUMO [17] (voir l'équation suivant) :

$$v_{des}(t + 1) = \min[v_{safe}, v(t) + a\Delta t, v_{max}]$$

$$v_{safe}(t + 1) = -\tau b + \sqrt{(\tau b^2) + v_{leader}^2 + 2bg_{leader}}$$

Where:

$v_{des}$  : vitesse souhaitée pour le temps  $t + 1$ ,

$v_{safe}$  : une vitesse de sécurité pour  $t + 1$ ,

$a$  : valeur de l'accélération maximale ( $a > 0$ ),

$b$  : valeur de décélération maximale ( $-b < 0$ ),

$v_{max}$  : la vitesse maximale du véhicule,

$\Delta t$  : time step,

$\tau$  : temps de réaction du conducteur,

$V_{leader}$  : la vitesse de leader (le véhicule de tête),

$g_{leader}$  : la distance qui sépare le véhicule contrôlé et le leader.

Nous avons implémenté le code suivant :

```

11 public class KraussSUMO extends KraussOrig1{
12
13     public double vsafe() {
14         if (getLeadSpeed() < 0.){
15             return getMaxSpeed();
16         }else{
17             if (getGap() == 0.){
18                 return 0.;
19             }else{
20                 return -(super.getTau()+super.getLeadMaxDecel()+Math.sqrt(Math.pow(super.getTau()
21 +super.getLeadMaxDecel(), 2)+Math.pow(super.getLeadSpeed(), 2)
22 +2*super.getLeadMaxDecel()*super.getGap()));
23             }
24         }
25     }
26 }
27

```

Figure IV-15 Le modèle utilisé par défaut dans SUMO [17]

### IV.6. Comparaison des résultats entre les trois modèles

Pour comparer entre les trois modèles de suivi de véhicules nous avons Les différents paramètres sont illustrés dans la figure ci-dessous. Nous présentons également une comparaison détaillée du comportement de chacun des trois modèles, plus précisément le changement de : la vitesse du véhicule contrôlé (voir Figure IV-17), la vitesse du véhicule prédécesseur (voir Figure IV-18), la distance entre les deux véhicules (voir Figure IV-19) et l'accélération appliquée par le véhicule contrôlé (voir Figure IV-20).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		KraussOrig1					KraussSUMO					Kerner			
2	Step	Speed	LeadSpeed	Gap	Action		Speed	LeadSpeed	Gap	Action		Speed	LeadSpeed	Gap	Action
3	54051	0	7,57	19,84	0		0	7,57	19,84	0		0	7,57	19,84	0
4	54052	1,18	8,76	27,42	1,5		1	8,76	27,6	1,5		1,1	8,76	27,5	1,5
5	54053	2,37	10,6	35,65	1,5		2	10,6	36,2	1,5		2,2	10,6	35,9	1,5
6	54054	3,56	12,59	44,68	1,5		3	12,59	45,79	1,5		3,29	12,59	45,2	1,5
7	54055	4,74	13,82	53,76	1,5		4	13,82	55,61	1,5		4,39	13,82	54,62	1,5
8	54056	5,93	14,83	62,66	1,5		5,01	14,83	65,44	1,5		5,49	14,83	63,97	1,5
9	54057	7,12	16,69	72,24	1,5		6,02	16,69	76,12	1,5		6,59	16,69	74,07	1,5
10	54058	8,31	18,23	82,16	1,5		7,03	18,23	87,31	1,5		7,7	18,23	84,6	1,5
11	54059	9,21	19,63	92,58	1,5		8,04	19,63	98,9	1,5		8,81	19,63	95,42	1,5
12	54060	10,11	21,26	103,73	1,5		9,05	21,26	111,11	1,5		9,92	21,26	106,76	1,5
13	54061	11,01	22,78	115,5	1,5		10,06	22,78	123,82	1,5		11,03	22,78	118,51	1,5
14	54062	11,91	23,99	127,58	1,5		11,07	23,99	136,74	1,5		12,14	23,99	130,37	1,5
15	54063	12,81	25,91	140,68	1,5		12,09	25,91	150,56	1,5		13,25	25,91	143,02	1,5
16	54064	13,71	27,85	154,83	1,5		13,11	27,85	165,3	1,5		14,36	27,85	156,52	1,5
17	54065	14,61	29,47	169,69	1,5		14,13	29,47	180,63	1,5		15,48	29,47	170,51	1,5
18	54066	15,52	30,75	184,93	1,5		15,15	30,75	196,23	1,5		16,6	30,75	184,67	1,5
19	54067	16,43	-1	0	1,5		16,17	-1	0	1,5		17,71	-1	0	1,5
20	54068	17,33	-1	0	1,5		17,19	-1	0	1,5		18,83	-1	0	1,5
21	54069	18,24	-1	0	1,5		18,22	-1	0	1,5		19,95	-1	0	1,5
22	54070	19,15	-1	0	1,5		19,25	-1	0	1,5		21,07	-1	0	1,5
23	54071	20,06	-1	0	1,5		20,28	-1	0	1,5		22,2	-1	0	1,5

Figure IV-16 paramètre de chaque modèle

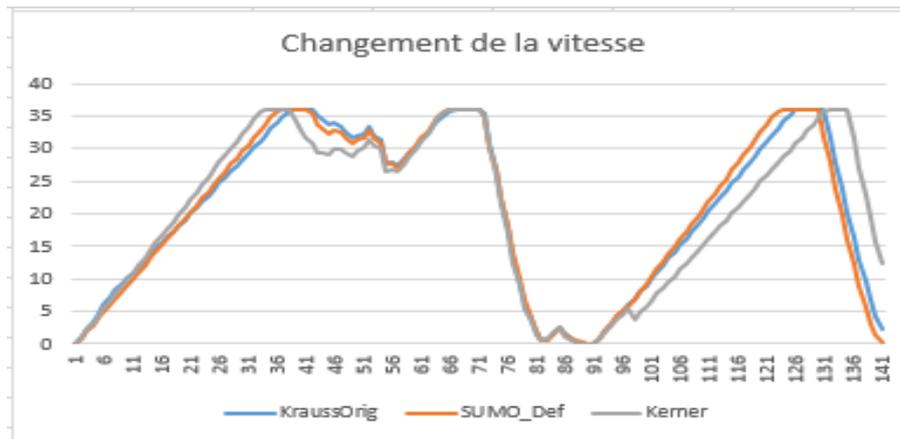


Figure IV-17 résultat de changement de vitesse

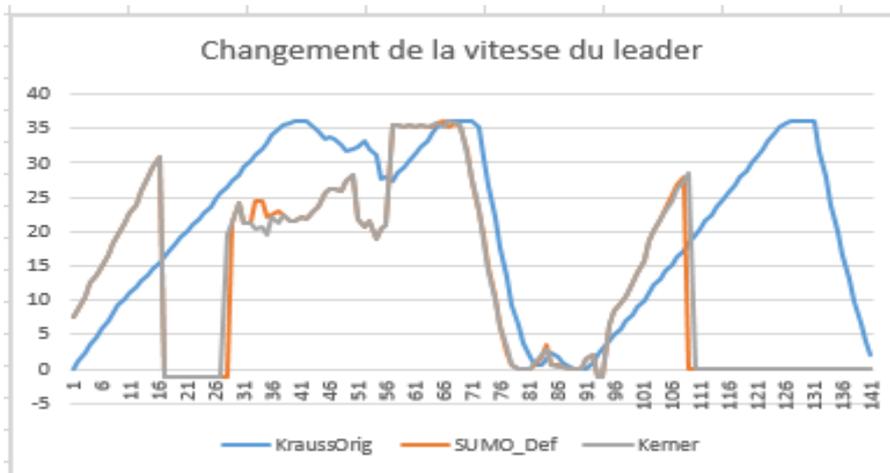


Figure IV-18 résultat de changement la vitesse de leader

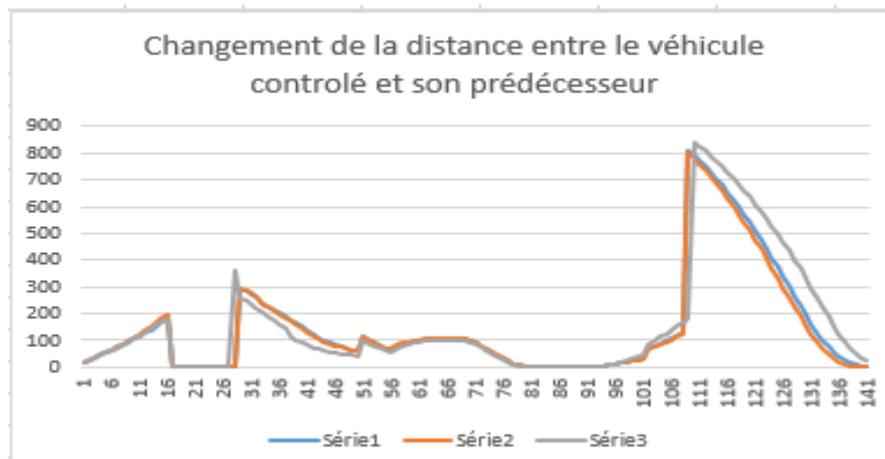


Figure IV-19 résultat de changement de la distance

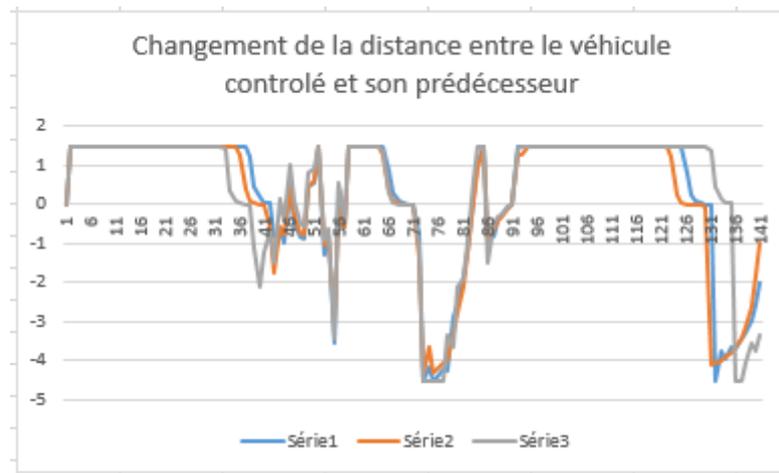


Figure IV-20 résultat de changement de l'accélération appliquée à tout instant.

### IV.7. Conclusion

Dans ce chapitre, nous avons présenté l'Expérimentation et résultats de notre simulateur qui permet de créer les différents scénarios de circulation et de consulter les informations relatives pendant la simulation et à la fin une comparaison entre les résultats de la simulation.

## Conclusion générale

Le développement technologique a fait un grand pas en avant dans l'évolution des réseaux véhiculaires, Cette évolution vise à rendre les réseaux plus sûrs, plus efficaces, plus fiables et plus respectueux de l'environnement du point de vue de l'industrie automobile et réseaux et services.

Les réseaux véhiculaires sont en effet une classe émergente de réseaux mobiles Ad Hoc, ils apportent une certaine contribution dans l'objectif d'un transport routier sécurisé et fiable et qu'en mettant en possession du véhicule la possibilité d'échanger des informations avec d'autres véhicules mais aussi des infrastructures, ainsi comme le confort de l'utilisateur avec l'offre de nombreux services d'information et d'aide à la décision, l'accompagnement et l'accès à Internet.

Pour cela nous avons présenté dans ce mémoire une étude générale sur les réseaux véhiculaires, et Nous avons aussi présenté quelque simulateur les plus utilisés dans les réseaux VANETs tels que le simulateur de réseau NS-2 et le simulateur de trafic routier SUMO. Nous avons jugé utile de faire notre simulation sur un scénario de réseaux routiers. L'objectif est d'obtenir des résultats d'évaluation cohérents avec sumo.

Le but de notre travail est de concevoir un simulateur permettant à la communauté des chercheurs d'implémenter et de tester leurs propres modèles et systèmes, incluant notamment : des modèles de suivi de véhicules, des modèles de changement de voies.

En effet, ce travail étant un essai, n'est donc pas un modèle unique et parfait, c'est pourquoi nous restons ouverts à tous les critiques et nous sommes prêts à recevoir toutes les suggestions et remarques tendant à améliorer davantage cette initiative.

## Bibliographie

### a. Bibliographie :

- [1] Costeseque, G. : 'Analyse et modélisation du trafic routier : Passage du microscopique au macroscopique', Mém. Maitrise À L'Université Lyon-2011, 2011
- [2] Buisson, C., Lebacque, J., and Lesort, J.: 'STRADA, a discretized macroscopic model of vehicular traffic flow in complex networks based on the Godunov scheme', in Editor (Ed.) ^ (Eds.): 'Book STRADA, (1996, edn.), pp.
- [3] Khalil, W. : 'Contribution à la modélisation graphique de système de systèmes', Lille 1, 2012
- [4] Herman, R., Montroll, E.W., Potts, R.B., and Rothery, R.W.: 'Traffic dynamics: analysis of stability in car following', Operations research, 1959, 7, (1), pp. 86-106
- [5] Gazis, D.C., Herman, R., and Rothery, R.W.: 'Nonlinear follow-the-leader models of traffic flow', Operations research, 1961, 9, (4), pp. 545-567
- [6] Richards, P.I.: 'Shock waves on the highway', Operations research, 1956, 4, (1), pp. 42-51
- [7] Badii, A.: 'Unification of Kinematic Wave Propagation Model of Traffic with Fundamental Diagram of Traffic Flow (June 2014)'
- [8] Jayakrishnan, R., Mahmassani, H.S., and Hu, T.-Y.: 'An evaluation tool for advanced traffic information and management systems in urban networks', Transportation Research Part C: Emerging Technologies, 1994, 2, (3), pp. 129-147
- [9] L.Benoit: 'La différenciation comportementale par les normes, et son application au trafic dans les simulateurs de conduite, ' Université des sciences et technologies de Lille, 2009
- [10] Schaefer, M., Vokřínek, J., Pinotti, D., and Tango, F.: 'Multi-agent traffic simulation for development and validation of autonomic car-to-car systems': 'Autonomic Road Transport Support Systems' (Springer, 2016), pp. 165-180
- [11] Roelofsen, M., Bie, J., Jin, L., and van Arem, B.: 'Assessment of safety levels and an innovative design for the lane change assistant', in Editor (Ed.) ^ (Eds.): 'Book Assessment of safety levels and an innovative design for the lane change assistant' (IEEE, 2010, edn.), pp. 83-88
- [12] Heesen, M., Baumann, M., Kelsch, J., Nause, D., Friedrich, M ' Enquête sur le comportement de conduite coopératif lors d'un changement de voie dans un environnement de simulation multi-pilote. Dans : Facteurs humains : Une vue d'une perspective intégrative. Facteurs humains et ergonomie Société en Europe Chapitre', 2012 toulouse
- [13] Krauß, S.: 'Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics', Dt. Zentrum für Luft-und Raumfahrt eV, Abt. Unternehmensorganisation und ..., 1998
- [14] Pipes, L.A.: 'An operational analysis of traffic dynamics', Journal of applied physics, 1953, 24, (3), pp. 274-281
- [15] Helbing, D.: 'Traffic and related self-driven many-particle systems', Reviews of modern physics, 2001, 73, (4), pp.
- [16] Wiedemann, R. : 'Simulation des strassenverkehrsflusses, vo-lume 8', Institut fiir Verkehrswesen der Universität Karlsruhe, 1974

- [17] Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L.: 'Recent development and applications of SUMO-Simulation of Urban MObility', *International Journal On Advances in Systems and Measurements*, 2012, 5, (3&4)
- [18] Fellendorf, M., and Vortisch, P.: 'Microscopic traffic flow simulator VISSIM': 'Fundamentals of traffic simulation' (Springer, 2010), pp. 63-93
- [19] Kerner, B., Klenov, S., and Brakemeier, A.: 'Testbed for wireless vehicle communication: A simulation approach based on three-phase traffic theory', in Editor (Ed.)^(Eds.): 'Book Testbed for wireless vehicle communication: A simulation approach based on three-phase traffic theory' (IEEE, 2008, edn.), pp. 180-185
- [20] Eissfeldt, N., and Wagner, P.: 'Effects of anticipatory driving in a traffic flow model', *The European Physical Journal B-Condensed Matter and Complex Systems*, 2003, 33, (1), pp. 121-129
- [21] Schreckenberg, M., and Wolf, D.E.: 'Traffic and Granular Flow'97: Gerhard-Mercato-Universität Duisburg, Germany, 6-8 October 1997' (Springer Verlag, 1998. 1998)
- [22] Treiber, M., Hennecke, A., and Helbing, D.: 'Congested traffic states in empirical observations and microscopic simulations', *Physical review E*, 2000, 62, (2), pp. 1805
- [23] Zhao, X.-m., and Gao, Z.-y.: 'A new car-following model: full velocity and acceleration difference model', *The European Physical Journal B-Condensed Matter and Complex Systems*, 2005, 47, (1), pp. 145-150
- [24] Tang, T., Shi, W., Shang, H., and Wang, Y.: 'A new car-following model with consideration of inter-vehicle communication', *Nonlinear dynamics*, 2014, 76, (4), pp. 2017-2023
- [25] Mazot, R., Meslem, W., Layouni, M., and Tran, A.: 'Communication inter véhiculaire', PhD thesis, Arles Avignon, 2013
- [26] Busanelli, S., Ferrari, G., and Veltri, L.: 'Short-lived key management for secure communications in VANETs', in Editor (Ed.)^(Eds.): 'Book Short-lived key management for secure communications in VANETs' (IEEE, 2011, edn.), pp. 613 618
- [27] BEKTACHE, D. : 'Application et Modélisation d'un protocole de communication pour la sécurité routière Thèse de Doctorat de l'Université BADJI MOKHTAR ANNABA -2014.', in Editor (Ed.)^(Eds.): 'Book (edn.), pp.
- [28] J. Petit, S.d.l.a.e.d.c.d.l.s.d.r.s., and fil véhiculaires. PhD thesis, l.u.d.T., 2011.
- [29] Ledy, J. : 'Stratégie d'adaptation de liens sur canaux radios dynamiques pour les communications entre véhicules-Optimisation de la Qualité de Service', 2012
- [30] Ferber, J. : 'Les systèmes multi-agents : un aperçu général', *Techniques et sciences informatiques*, 1997, 16, (8)
- [31] Bousquet, F., Mathevet, R., and Le Page, C. : 'Modélisation d'accompagnement et modèles multi-agents appliqués à la conservation de la biodiversité', *Sciences de la Conservation*. Paris, De Boeck, 2014, pp. 231-236
- [32] LEMLOUMA, T., and BOUDINA, A. : 'L'intelligence Artificielle Distribuée et les Systèmes Multi-Agents', 1993
- [33] Drogoul, A., Corbara, B., and Fresneau, D.: 'MANTA: New experimental results on the emergence of (artificial) ant societies', *Artificial Societies: the computer simulation of social life*, 1995, pp. 190-211
- [34] Reynolds, C.W.: 'Flocks, herds and schools: A distributed behavioral model' (ACM, 1987. 1987)

[35] Côté, M., and Troudi, N.: 'NetSA: Une architecture multiagent pour la recherche sur Internet', Expertise Informatique, 1998, 3, (3)

[36] Krajzewicz, D., Hertkorn, G., Rössel, C., and Wagner, P.: 'SUMO (Simulation of Urban MObility)-an open-source traffic simulation', in Editor (Ed.)^(Eds.): 'Book SUMO (Simulation of Urban MObility)-an open-source traffic simulation' (2002, edn.), pp. 183-187

[37] Brockfeld, E., Kühne, R.D., Skabardonis, A., and Wagner, P.: 'Toward benchmarking of microscopic traffic flow models', Transportation research record, 2003, 1852, (1), pp. 124-129

## **b. Webographie**

[38] <https://sumo.dlr.de/docs/FAQ.html>

[39] [https://sumo.dlr.de/docs/Sumo\\_at\\_a\\_Glance.html](https://sumo.dlr.de/docs/Sumo_at_a_Glance.html)

[40] <https://sumo.dlr.de/docs/SUMO.html>

[41] <https://sumo.dlr.de/docs/TraCI.html>

[42] <https://www.techno-science.net/definition/517.html>

[43] <https://www.java.com/fr/>

[44] [https://fr.wikipedia.org/wiki/JADE\\_\(langage\)](https://fr.wikipedia.org/wiki/JADE_(langage))