



HAL
open science

Spécification DEVS de modèles de trafic à l'échelle mésoscopique et microscopique

Alexis Pestelle, Florian Condette, Valentin Vendi, Eric Ramat

► **To cite this version:**

Alexis Pestelle, Florian Condette, Valentin Vendi, Eric Ramat. Spécification DEVS de modèles de trafic à l'échelle mésoscopique et microscopique. Journées Francophones de la Modélisation et de la Simulation (JFMS), Mar 2022, Cargèse, France. hal-03627049

HAL Id: hal-03627049

<https://hal.science/hal-03627049>

Submitted on 1 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Spécification DEVS de modèles de trafic à l'échelle mésoscopique et microscopique

A. Pestelle¹ F. Condette¹
V. Vendi¹ E. Ramat¹
LISIC - Université du Littoral Côte d'Opale
Calais - France

{alexis.pestelle,florian.condette,valentin.vendi,eric.ramat}@univ-littoral.fr

20 février 2022

Résumé :

La modélisation du trafic routier est un sujet d'intenses recherches de part les défis liés à la révolution des transports en cours et à venir : nouvelle organisation, nouveaux modes de transports, réduction des émissions polluantes, ... Cet article propose une nouvelle approche de modélisation à l'aide du formalisme Parallel-DEVS. À partir des modèles disponibles à l'échelle mésoscopique et microscopique, nous avons identifié les processus sous-jacents. Ces processus ont été, la plupart du temps, exprimés sous forme de contraintes elle-mêmes exprimées par des relations mathématiques sur les variables caractéristiques (position, vitesse et accélération). Le travail a donc consisté à identifier ce qui peut être reformulé sous forme d'automates à états finis, d'événements discrets et d'expressions permettant de calculer les dates des événements clés. De plus, les valeurs des variables caractéristiques sont aussi issues des mêmes relations mais leur calcul est réalisé aux dates des événements. Ce travail a abouti à la production d'un ensemble de modèles atomiques que l'on peut coupler pour produire des modèles multi-échelles.

Mots-clés :

Modélisation microscopique, modélisation mésoscopique, modélisation du trafic routier

Abstract:

Road traffic modeling is a subject of intense research due to the challenges related to the current and future transport revolution : new organization, new modes of transport, reduction of particle emissions, ... This article proposes a new modeling approach using the Parallel-DEVS formalism. From available models at the mesoscopic and microscopic scale, we have identified the underlying processes. These processes were, most of the time, expressed in the form of constraints themselves expressed by mathematical relations on the characteristic variables (position, speed and acceleration). The work consists of identifying what can be reformulated in the form of finite state automata, discrete events and expressions allowing to calculate the dates of events. In addition, the values of the characteristic variables are also derived from the same relationships but their calculation is carried out on the dates of the events. This work resulted in the production of a set of atomic models that can be coupled to produce multi-scale models.

Keywords:

Microscopic and mesoscopic (kinetic) scale modeling, traffic flow

1 Introduction

La modélisation et la simulation de systèmes complexes et plus précisément dans le contexte du trafic routier, implique la représentation à diverses échelles des processus mis en jeu (déplacement des véhicules, gestion des bouillons, circulation au sein des carrefours, ...). De plus, de manière générale, la taille et la complexité des systèmes à représenter (par exemple, à l'échelle d'une agglomération) peut conduire à des modèles complexes que ce soit en terme de concepts à prendre en compte (comportements des véhicules, éléments d'aménagement, ...) ou en temps de calcul. L'utilisation du formalisme DEVS et plus particulièrement dans nos travaux, du formalisme Parallel-DEVS [1] permet à la fois de prendre en charge des représentations multiples (à diverses échelles) à l'aide de la notion de couplage. Un travail d'abstraction, de généralisation et d'atomisation (au sens DEVS) doit être mené. D'autre part, DEVS permet d'optimiser les calculs en exprimant la dynamique des processus à l'aide d'événements discrets. Le concept d'événements discrets est surtout efficace dans le cas de trafics fluides [2], car il ne nécessite de mettre à jour les véhicules que lorsqu'un événement se produit. Afin d'étendre le domaine d'intérêt des événements discrets, il

est important de se pencher sur les notions d'événement et d'état et de définir la meilleure représentation de ces derniers. Pour cela, on peut s'appuyer sur l'abondante littérature sur la modélisation du trafic. Depuis les années 30, avec les premiers articles sur la représentation mésoscopique et le diagramme fondamental [3] jusqu'aux développements les plus récents à l'échelle microscopique en prenant en compte des phénomènes plus fins [4].

Dans nos travaux, nous n'allons pas réinventer les modèles existants mais nous allons montrer comment certains modèles de la littérature peuvent être efficacement ré-écrits en Parallel-DEVS et surtout comment nous allons pouvoir coupler ces différents modèles pour répondre à la question du multi-échelles. Le couplage de modèle sera vu à la fois à travers le besoin de représentation avec des niveaux de précision différents selon les éléments du système routier et le besoin d'intégrer des modèles en interaction avec le système comme, par exemple, des modèles d'émission de polluants.

Dans un premier temps, nous allons présenter les deux grandes familles de modèles : le mésoscopique à travers le diagramme fondamental et le microscopique avec le modèle de poursuite. Puis, on montrera comment Parallel-DEVS peut prendre en charge chacun de ces niveaux et leur couplage. Finalement, on conclura à la fois sur les perspectives d'améliorations des modèles, l'intégration des modèles au sein d'algorithmes de calibration, le couplage avec d'autres modèles et l'utilisation des modèles pour la prise de décision ou le test de scénarios d'aménagement.

2 État de l'art

Dans cette partie, nous abordons l'état de l'art des différents modèles existants pour les deux échelles étudiées. Le diagramme fondamental est utilisé comme point de référence pour l'échelle mésoscopique. Celui-ci permet de définir des composantes majeures de cette échelle comme la densité, le débit et la vi-

tesse du flux. La représentation à l'échelle microscopique se focalise principalement sur les modèles de poursuite. Ils décrivent comment un véhicule, appelé le suiveur, adapte son comportement au comportement du véhicule qui le précède, le leader. Les différentes catégories de modèles de poursuite seront présentées et le fonctionnement de celui faisant l'objet de la réécriture en DEVS sera détaillé.

La simulation multi agents (SMA) propose déjà des modélisations des deux échelles[5][6]. Celle-ci se base sur une approche orientée entités[7] (agents) et est basée sur des règles. L'avantage principal de ce type de modélisation est d'exprimer des règles complexes basées sur le comportement et la perception de l'environnement. Le formalisme DEVS propose une approche par composants via des modèles atomiques pouvant être couplés. Ils interagissent entre eux via des événements discrets envoyés ou reçus sur une échelle de temps continue clairement explicite dans le formalisme. Les principaux avantages de ce formalisme sont sa modularité de par sa décomposition en modèles atomiques ainsi que la synchronisation des processus physiques. Même si l'objectif de la modélisation et de la simulation reste le même, l'approche diffère, la première via les SMA s'intéressera aux entités ainsi qu'aux interactions (avec le modèle IODA par exemple[8]) tandis que l'autre approche traitera des processus et des dynamiques physiques spatio-temporelles. Cette dernière sera retenue pour nos travaux.

2.1 Le diagramme fondamental

Le trafic routier présente une forte ressemblance avec l'écoulement d'un fluide[9]. La première ressemblance est dans la définition du réseau en lui même. On y trouve des sections routières (équivalent à des tuyaux), des carrefours (jonctions), des bifurcations ... La définition de la demande fait également partie de ces ressemblances. Le flux entrant peut-être qualifié de demande et le réseau en lui même d'offre. Lorsque

cette demande s'écoule sur le réseau, et que l'offre ne suffit plus à écouler la demande, on peut constater des débordements (équivalent à des congestions) qui peuvent être locaux voir globaux.

Le diagramme fondamental[3] permet de décrire le flux de véhicules lors de la modélisation du trafic routier. Il décrit trois composantes majeures :

- Le débit : Q où $Q = \frac{N}{\Delta t}$ avec N qui représente le nombre de véhicules passés pendant une période Δt .
- La concentration : K où $K = \frac{N}{\Delta x}$ avec N qui représente le nombre de véhicules présents sur une section pour un instant donné Δx .
- La vitesse du flux : V où $V = \frac{Q}{K}$.

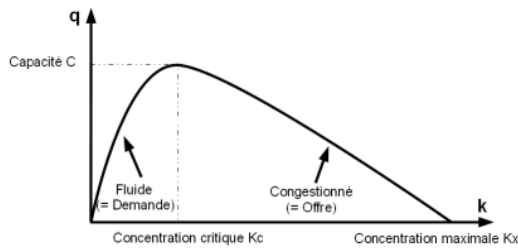


FIGURE 1 – Diagramme fondamental[10].

2.2 Modèles de poursuite

Les modèles de poursuite décrivent comment un véhicule (le suiveur) suit un autre véhicule (le leader) dans une circulation ininterrompue. Ces modèles ont pour objectif de représenter comment un conducteur réagit aux changements de dynamiques du véhicule devant lui.

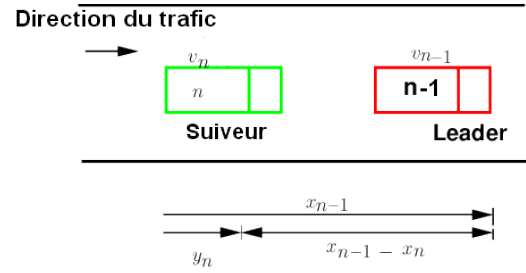


FIGURE 2 – Notations pour les modèles de poursuite.

La réaction du suiveur est fonction de sa perception de la distance qui le sépare du leader ainsi que de la vitesse relative entre les deux véhicules. La forme générale et paramétrique de l'accélération du suiveur à un instant t pour la plupart des modèles est alors :

$$a_n(t + \tau) = l_n \frac{[v_{n-1}(t) - v_n(t)]^k}{[x_{n-1}(t) - x_n(t)]^m} \quad (1)$$

où v_{n-1} , v_n et x_{n-1} , x_n sont respectivement la vitesse et la position du leader et du suiveur, τ le temps de réaction du conducteur et l_n , k et m des paramètres à calibrer. Les trois paramètres permettent d'ajuster le modèle à des données réelles. Ils jouent sur le niveau d'importance de la vitesse relative (k) ou de la distance relative (m) dans le rapport et l_n dans l'expression de l'accélération.

Olstam [11] catégorise les modèles de poursuite en trois catégories selon la logique utilisée :

- la réponse aux stimuli : les conducteurs modifient leur comportement en fonction de certains stimuli, tels que l'espace entre leur véhicule et le leader ou la vitesse relative avec le leader. Parmi ces modèles, on trouve celui de General Motors (GM), qui a servi de base aux autres : *Gazis-Herman-Rotery* (GHR [12]), *Optimal Velocity Model* (OVM [13]) et *Intelligent Driver Model* (IDM [14]).
- la distance de sécurité ou évitement de collision : les conducteurs cherchent

toujours à tenir une distance de sécurité telle qu'aucune collision avec le véhicule leader ne soit possible, même en cas d'évènement imprévu. Les bases de cette famille ont été posées par Pipes [15] et le modèle le plus influent est celui de Gipps [16].

- psychophysique : ces modèles utilisent des seuils de perception. Les conducteurs réagissent aux changements d'espace et de vitesse relative uniquement lorsque les valeurs seuils sont atteintes. Le modèle de Wiedemann [17] permet en outre de modéliser chaque conducteur avec ses propres capacités par le biais de lois normales.

Dans notre contexte de modélisation en Parallel-DEVS, nous avons posé comme hypothèse que la vitesse doit être linéaire par morceau. Le modèle de référence choisi est celui de Gipps, car il est le seul à ne pas remettre en cause l'accélération entre chaque instant. Ce modèle se base sur trois principes :

- il doit pouvoir reproduire le comportement d'un trafic réel
- les paramètres doivent correspondre à des caractéristiques observables et quantifiables d'un conducteur afin de pouvoir assigner des valeurs sans avoir besoin de calibrations complexes
- il doit se comporter correctement quand la fréquence des calculs de la vitesse et de la position est égale au temps de réaction

Pour cela, un ensemble de contraintes est appliqué au véhicule suiveur : il ne peut pas dépasser la vitesse maximum souhaitée par son conducteur et l'accélération doit augmenter avec la vitesse puis descendre à 0 quand le véhicule approche de sa vitesse prévue. Dans le cas d'un trafic fluide, l'auteur propose d'appliquer un modèle issu d'un ajustement d'une série d'observations réelles :

$$v_n(t + \tau) \leq v_n(t) + 2.5a_n\tau(1 - v_n/V_n) (0.025 + v_n(t)/V_n)^{1/2} \quad (2)$$

avec a_n et V_n respectivement l'accélération et la vitesse maximale souhaitée par le conducteur. Le modèle IDM propose un autre modèle mais qui définit l'accélération comme une différentielle :

$$dv_n(t)/dt = a_n(1 - (v_n(t)/V_n)^\delta) \quad (3)$$

Si le véhicule n-1 commence à freiner aussi fort que désirable, il s'arrêtera au point :

$$x_{n-1}^* = x_{n-1}(t) - v_{n-1}(t)^2/2b_{n-1} \quad (4)$$

Le véhicule suiveur ne pourra réagir qu'à l'instant $t + \tau$ et s'arrêtera au point :

$$x_n^* = x_n(t) + [v_n(t) + v_n(t + \tau)]\tau/2 - v_n(t + \tau)^2/2b_n \quad (5)$$

avec b_n le freinage le plus sévère possible par le conducteur.

Pour respecter les distances de sécurité, le suiveur doit s'assurer que la différence entre le point d'arrêt du leader et la taille effective du véhicule s_{n-1} , qui comprend la taille physique ainsi qu'une marge de sécurité, doit être plus grande que la différence entre le point d'arrêt du suiveur. A cela, Gipps ajoute une marge de sécurité θ et la capacité de freinage du leader b_{n-1} est remplacée par une estimation \hat{b} par le suiveur de celle-ci. La contrainte sur le freinage donne alors :

$$v_n(t + \tau) \leq b_n\tau + \sqrt{b^2\tau^2 - b_n[2[x_{n-1}(t) - s_{n-1} - x_n(t)] - v_n(t)\tau - v_{n-1}(t)^2/\hat{b}]} \quad (6)$$

La forme finale de la vitesse à l'instant $t + \tau$ est :

$$v_n(t + \Delta t) = \min\{v_n(t) + 2.5a_n^d \tau (1 - v_n(t)/v_n^d) \\ (0.025 + v_n(t)/v_n^d)^{1/2}, \\ b_n \tau + \sqrt{b^2 \tau^2 - b_n [2[x_{n-1}(t) \\ - s_{n-1} - x_n(t)] - v_n(t)\tau - v_{n-1}(t)^2 / \hat{b}]}\} \quad (7)$$

Le premier argument correspondant à l'état fluide (*free-flow*) du trafic où le véhicule n'est pas affecté par le leader et le second argument à l'état de congestion.

3 Spécifications Parallel-DEVS

Les travaux de cet article s'inscrivent dans un contexte d'utilisation du formalisme Parallel-DEVS et de reformulation des modèles de trafic. Les différents modèles présentés forment donc notre point de départ et nous nous intéressons à leur expression en DEVS. Cette transformation nous conduit à nous poser les questions suivantes :

- quels sont les composants du système au sens de DEVS ?
- quelles sont les interactions possibles entre les composants ?
- quelles sont les dynamiques internes et externes de chaque composant ?

Les deux premières questions sont relativement simples à la fois à l'échelle méso et micro. En revanche, la transformation des équations et des contraintes sous-jacentes est beaucoup plus délicate en particulier pour l'échelle micro. En effet, les équations représentent les variations des variables caractéristiques des processus : l'accélération, la vitesse et la position. Elles n'explicitent pas directement les changements d'états sous l'influence d'événements.

Dans cette partie, nous allons présenter les modèles Parallel-DEVS et les hypothèses retenues pour chacune des échelles. Au préalable, une présentation succincte de Parallel-DEVS est proposée ainsi que les éléments de la plate-

forme de simulation Artis*.

3.1 Le formalisme Parallel-DEVS

Le formalisme Parallel-DEVS (PDEVS) [18], est une extension du formalisme DEVS qui introduit le concept d'événements simultanés. Ce formalisme permet une plus grande souplesse au niveau de la gestion des priorités en les traitant non plus au niveau du modèle couplé mais au niveau du modèle atomique. PDEVS ajoute, pour cela, au modèle atomique une fonction de conflit $\delta_{con}(s, x)$. Cette fonction traite les cas d'événements internes et externes concomitants. De plus, contrairement à DEVS, ce formalisme ne traite plus seulement un événement d'entrée ou de sortie à la fois mais un ensemble d'événements, pouvant se produire au même moment. Ces ensembles d'événements portent le nom de "bags" et se note X^b et Y^b .

PDEVS définit un modèle atomique par la structure suivante :

$$M = \langle X, Y, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta \rangle$$

Où,

- X est l'ensemble des ports et des valeurs d'entrée.
- Y est l'ensemble des ports et des valeurs de sortie.
- S est l'ensemble des valeurs possibles des variables d'états du système.
- $\delta_{int} : S \rightarrow S$ est la fonction de transition interne dont le rôle est de faire passer le système d'un état à un autre de manière autonome. Elle spécifie les états futurs des états actifs.
- $\delta_{ext} : Q \times X^b \rightarrow S$ est la fonction de transition externe dont le rôle est de faire passer le système d'un état à un autre suite à l'apparition des événements externes où, $Q = \{(s, e) | s \in S, 0 \leq e \leq ta(s)\}$ est l'ensemble des états totaux du système et e représente le temps écoulé dans l'état s depuis la dernière transi-

tion. Le concept d'état total (s, e) permet de spécifier un état futur en fonction du temps écoulé dans l'état présent.

- $\delta_{con} : S \times X^b \rightarrow S$ est la fonction de conflit dont le rôle est de gérer le cas où plusieurs événements se réalisent à la même date que la fin de l'état courant.
- $\lambda : S \rightarrow Y^b$ est la fonction de sortie. Elle n'est activée que lorsque le temps écoulé dans un état donné est égal à sa durée de vie.
- $ta : S \rightarrow R_0^+ \cup +\infty$ est la fonction d'avancement du temps, son rôle est de donner la durée pendant laquelle le système sera dans un certain état.

Un modèle PDEVS se comporte donc comme suit : si aucun événement externe n'apparaît, le système reste dans l'état s pour la durée définie par $ta(s)$. Lorsque $e = ta(s)$, le système change d'état par appel de la fonction δ_{int} . Si un événement externe de valeur x apparaît, lorsque le système est dans l'état (s, e) , le système change d'état par appel de la fonction $\delta_{ext}(s, e, x)$. S'il apparaît quand $e = ta(s)$, le système change son état par appel de la fonction de conflit $\delta_{con}(s, e, x)$.

La formalisation d'un modèle couplé peut se résumer ainsi :

$$N = \langle X, Y, D, \{M_d\}, \{I_d\}, \{Z_{i,d}\} \rangle$$

Où,

- $\forall d \in D, M_d$ est un modèle PDEVS
- $\forall d \in D \cup \{N\}, I_d$ est l'ensemble d'influence de $d : I_d \subseteq D \cup \{N\}$
- $d \notin I_d, \forall d \in D \cup \{N\}, \forall i \in I_d, Z_{i,d}$ est une fonction, le transfert de sortie entre i et d :

$$Z_{i,d} : X \rightarrow X_d, \text{ si } i = N$$

$$Z_{i,d} : Y_i \rightarrow Y, \text{ si } d = N$$

$$Z_{i,d} : Y_i \rightarrow X_d, \text{ si } i \neq N \text{ et } d \neq N$$

L'ensemble d'influence de d est un ensemble de modèles qui interagissent avec d . $Z_{i,d}$ spécifie le type de relations qui existe entre les modèles i et d . En PDEVS, la fonction *Select* présente en

DEVS classique est remplacée par la fonction de conflit δ_{con} .

PDEVS est un formalisme opérationnel qui offre des algorithmes abstraits pour son exécution. Ils sont intégralement présentés dans les différents livres de Bernard P. Zeigler [19]. On utilise une implémentation C++ de ces algorithmes qui se nomme : Artis* (<https://gitlab.com/artis-star/artis-star>). Le socle d'Artis* repose sur les algorithmes de DSDE [20]. Le principe général est d'exprimer tous les éléments d'un modèle sous forme de code et d'utiliser au maximum les templates pour éviter le polymorphisme (voir figure 3).

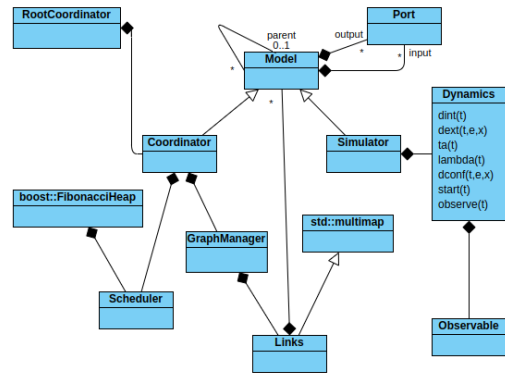


FIGURE 3 – Diagramme de classes d'Artis*.

De plus, l'échéancier, élément fondamental, est basé sur un tas de Fibonacci [21]. Mettre en oeuvre un modèle atomique consiste à développer une classe par héritage de la classe Dynamics qui regroupe l'ensemble des fonctions de la structure d'un modèle atomique. Cette classe spécifie aussi les ports d'entrée et de sortie ainsi que les éléments d'observation. Le processus d'observation étant distinct de la dynamique [22]. Les modèles couplés sont quant à eux définis par l'écriture d'une sous-classe de GraphManager. Cette classe définit les connexions internes et externes ainsi que les ports d'entrée et sortie. Un mécanisme d'expression des conditions expérimentales est disponible sous la forme de struct définit au

sein de chaque modèle atomique ou couplé. Dernier élément important : les événements. Par souci d'efficacité, les éléments transportés par les éléments sont pris en charge par une classe de sérialisation automatique.

3.2 Échelle mésoscopique

Le modèle DEVS actuellement développé repose sur plusieurs hypothèses. Les véhicules se déplacent sur des tronçons à une vitesse constante (*free_speed* - m/s), les notions de décélération et d'accélération liées au comportement du véhicule ne sont pas nécessaires à cette échelle. Les véhicules circulent à la vitesse maximale autorisée sur le tronçon. Le flux entrant, exprimé en véhicules par seconde, est limité afin de représenter la capacité de prise en charge des véhicules par un tronçon. En effet, deux véhicules ne peuvent entrer en même temps sur une section routière mono voie et la distance entre les véhicules doit être garantie. De plus, un tronçon limite le nombre total de véhicules présents via la notion de *concentration* permettant de définir un nombre maximal de véhicules par mètre.

L'hypothèse forte du niveau mésoscopique est que l'on résume la dynamique des véhicules à deux phases : le déplacement à vitesse constante et la mise en file d'attente à la fin du tronçon (congestion). La mise en file d'attente est liée à la possibilité que les véhicules puissent passer ou non dans le tronçon aval. Afin de simplifier la dynamique globale, le modèle de tronçon, *Link*, est un modèle couplé (voir 4). Il se compose de deux sous-modèles atomiques : *LinkTravel* et *LinkQueue*, qui permettent de gérer les deux phases. Les tronçons sont connectés entre eux via des noeuds (*Node*).

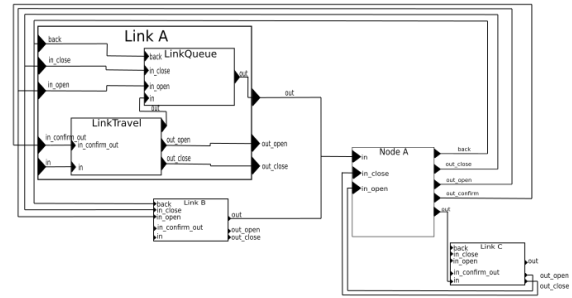


FIGURE 4 – Modèles atomiques à l'échelle mésoscopique.

L'utilisation de deux sous-modèles pour les tronçons est motivée par la volonté de concevoir des modèles simples. Les tronçons suivent deux logiques différentes et donc implique deux graphes d'états que l'on doit fusionner par un produit cartésien des états. La conséquence négative de ce choix est la génération d'un nombre d'événements plus importants.

L'état du modèle *LinkTravel* est défini par l'ensemble des véhicules en cours de déplacement avec leur date d'arrivée à la fin du tronçon. Le modèle sera réveillé soit par l'arrivée dans le tronçon d'un nouveau véhicule (δ_{ext}) soit par l'arrivée d'un véhicule à la fin du tronçon (δ_{int}). Dans le premier cas, un événement de sortie est généré pour signaler la fermeture du tronçon via un état transitoire à durée nulle et la fonction de sortie λ . Dans le second cas, un événement de sortie transportant le véhicule est généré afin de le transmettre au modèle *LinkQueue* qui va vérifier l'état de congestion.

Le modèle *LinkQueue* reçoit les véhicules arrivés à la fin de tronçon et selon l'état de la file d'attente, va soit envoyer le véhicule au modèle *Node* soit placer le véhicule dans la file d'attente. Ce processus est perturbé par l'ouverture et la fermeture de l'élément suivant. Les modèles *Link* sont soumis à des contraintes de capacité et de concentration donnant lieu à des fermetures temporaires du tronçon et à des réouvertures. Les ports de sortie *out_close* et *out_open* permettent d'envoyer des événements en amont afin de spécifier une fermeture. Les

ports d'entrée *in_close* et *in_open* assurent le même fonctionnement avec le tronçon aval. Le modèle *Node* possède également ces ports d'entrée et de sortie qui suivent une logique similaire. Les ports d'entrée *in* permettent l'accès aux véhicules dans les tronçons et les ports de sortie *out* de sortir du tronçon courant.

L'avancement du temps du modèle *LinkQueue* est principalement dépendant du mécanisme d'ouverture/fermeture. De plus, plusieurs *Link* peuvent converger sur un seul et même noeud. En cas de congestion des n tronçons amonts, il a été retenu qu'un seul des n véhicules se présentant à l'entrée du noeud pourra le franchir et donc poursuivre son trajet. L'autre véhicule sera replacé dans son *LinkQueue* d'origine via un évènement sur le port *back*. Le choix du véhicule autorisé respecte des propositions définies par la modélisation. En effet, en général, lors de la convergence de plusieurs tronçons, un ordre de priorité est défini.

Revenons sur l'onde de congestion afin de préciser les mécanismes mis en jeu. L'onde de congestion [10] est un phénomène de décélération qui se propage au sein du trafic routier. Elle se produit dès lors que les tronçons aval ne disposent pas d'une capacité suffisante ou que la concentration atteint sa limite. Elle peut se déplacer d'amont en aval, ce qui correspond à un ralentissement, et à l'inverse, un redémarrage si celle-ci se déplace d'aval en amont. Une vitesse négative induira un ralentissement tandis qu'une vitesse positive induira un redémarrage. Dans les modèles *LinkTravel* et *LinkQueue*, le paramètre *wave_speed* en mètres par seconde permet d'exprimer cette onde, il est utilisé lors du calcul du temps que mettra un véhicule à traverser *LinkTravel*. Le modèle *Node* signale au modèle *LinkTravel* la sortie effective d'un véhicule du tronçon. Cela permet de calculer le retard dt_{reel} à appliquer au prochain véhicule entrant dn_{reel} dans le tronçon selon les formules suivantes :

- $dn = \lceil L * c \rceil$ où L est la longueur du tronçon et c la concentration max

- $dt = \lceil L/w \rceil$ où w est la vitesse de l'onde de congestion
- $dn_{reel} = N + dn$ où N est le nombre de véhicules dans le tronçon
- $dt_{reel} = t + dt$ où t est la date d'arrivée du nouveau véhicule

Le modèle développé s'est inspiré d'une implémentation Python réalisée par des membres du Cerema : la librairie Stream (<https://github.com/AureClai/stream-python>). Elle permet de faire des simulations de trafic à l'échelle mésoscopique via de l'évènementiel.

Afin de valider le modèle, le plan d'expériences suivant a été réalisé. A partir d'un réseau de tronçons, nous nous sommes intéressés à un tronçon particulier qui peut être sujet à congestion en fonction de la demande. Les données de la figure 5 sont issues de 150 simulations d'une durée de 3600 secondes (durée nécessaire pour atteindre un état stable). Le tronçon étudié possède une longueur de 296.17 mètres. Les véhicules sont autorisés à circuler à une vitesse de 30.55 mètres par seconde. La capacité du tronçon est de 1.2 véhicules par seconde et sa concentration maximale est de 0.12 véhicules par mètre. Ce tronçon est suivi par deux tronçons de capacité très faible. Nous faisons alors varier la fréquence d'arrivée des véhicules dans le tronçon. Cette fréquence varie d'un véhicule toutes les 0.2 seconde en moyenne (suivant une loi gaussienne) à toutes les 2 secondes. Le pas de variation est de 0.01 seconde par itération.

Deux réseaux ont été utilisés pour réaliser et calibrer nos modèles. Un premier réseau a été utilisé afin d'intégrer les différentes dynamiques inhérentes à l'échelle mésoscopique. Ce premier réseau est composé de 3 générateurs de véhicules, 12 modèles *Link* et 10 modèles *Node*, 4 noeuds sont en sortie de réseau et sont utilisés comme point de comptage des véhicules en sortie de réseau. Le second réseau, utilisé pour la calibration de nos modèles, se base sur des données empiriques de la jonction d'auto-

route A115-A15, deux types de voies sont distinguées dans ce réseau, Les voies d'autoroute et les voies d'insertion. Il se compose de 13 générateurs de véhicules, 69 modèles *Link*, 44 modèles *Node*. Parmi les 44 *Node*, 13 sont utilisés comme point de comptage des véhicules en sortie de réseau.

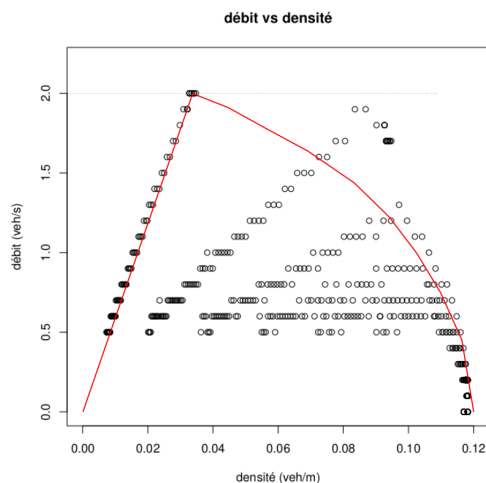


FIGURE 5 – Comparaison entre le diagramme fondamental et la simulation DEVS.

Le diagramme fondamental (figure 1) décrit un débit croissant tant que la capacité du réseau permet de prendre en charge la demande de véhicules. Lorsque la capacité ne suffit pas à écouler le trafic, on peut changer de régime et on passe en congestion. Dans un premier temps, il s'agit d'un ralentissement, avec un débit qui décroît, puis une congestion totale avec un débit quasi nul.

Les données issues du tronçon étudié montrent bien une croissance similaire tant que la capacité du réseau n'est pas atteinte. Dès que la section routière comporte plus de véhicules et que cette capacité n'est plus suffisante pour satisfaire la demande, on constate une réduction du débit. Ce ralentissement se poursuit tant que la concentration croît pour atteindre une congestion totale du réseau avec un débit de 0 véhicules par seconde. Le modèle développé est donc bien conforme à la dynamique du diagramme fondamental.

Ce modèle a aussi été l'objet d'une calibration grâce à des données réelles (vitesse moyenne des véhicules à des points de comptage, temps de parcours moyen entre deux points, ...) et l'utilisation de méthodes type méta-heuristiques [23]. La conclusion de ce travail est que les solutions proposées sont cohérentes.

Afin de prendre en charge des infrastructures et des dynamiques plus complexes, nous allons poursuivre ce travail de modélisation et compléter le modèle. La notion de sections multivoies est une première piste d'amélioration. Actuellement, un tronçon peut posséder plusieurs voies mais sans prise en compte sur la dynamique des véhicules, en particulier, le changement de voie et ses conséquences. Il faut donc apporter une réponse sans dévoyer la logique du niveau mésoscopique. Autres améliorations : les voies réservées (co-voiturage, par exemple) ou l'onde de congestion différenciée. Actuellement, la prise en compte de l'onde de congestion est la même pour tous les types de véhicules. Or un véhicule léger ne se comporte pas comme un poids lourd.

3.3 Échelle microscopique

Plusieurs hypothèses et quelques simplifications sont proposées pour cette adaptation du modèle de Gipps au formalisme P-DEVS. Tout d'abord, l'accélération et la décélération sont constantes par morceau. La vitesse en état libre (*free-flow*) est équivalente à la vitesse maximum autorisée d'un tronçon. Le véhicule cherchera ainsi par défaut à atteindre la vitesse maximum du tronçon. Le suiveur est capable de réagir aux changements du rapport entre la vitesse relative et la distance inter-véhicules après un temps de réaction ($\tau = 1s$) et s'adapte pour garantir une distance de sécurité minimale avec le leader. Ce même suiveur pourra être le leader d'un autre véhicule, formant des paires de leader/suiveur à l'intérieur du tronçon.

Dans cette modélisation P-DEVS, les modèles atomiques principaux sont les tronçons (*Link*) et les noeuds (*Node*). Un tronçon est à voie unique et est représenté par sa longueur et la vitesse maximale autorisée dessus. Il prend en charge l'ensemble des véhicules présents dans un tronçon. La fonction δ_{ext} intègre les véhicules arrivants dans le tronçon dans la liste des véhicules. La fonction δ_{int} va traiter les changements d'états des véhicules et l'activation de l'envoi des véhicules qui sortent du tronçon. A ce niveau, la fonction δ_{int} est activée par la plus petite date de changement d'état des véhicules. L'état d'un véhicule est défini par :

$$S = \langle ls, tn \rangle$$

Où,

- ls est l'ensemble des états successifs noté s du véhicule avec $s = (p, r, b, is, fs, a)$
- $p \in \{STOP, RESTART, IN_ACCELERATION, RUNNING, IN_DECELERATION\}$
- $r \in \{false, true\}$, $r = true$ si le véhicule peut sortir lorsqu'il sera arrivé en fin de tronçon
- b , date de début de l'état
- is , vitesse au début de l'état
- fs , vitesse théorique en fin d'état
- a , accélération appliquée tout au long de l'état
- tn , date de fin de l'état

Par défaut, le modèle se comporte selon le graphe d'états 6.

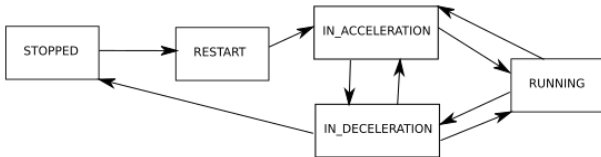


FIGURE 6 – États d'un véhicule à l'échelle microscopique.

À la fin d'état d'un des véhicules pris en charge par le modèle *Link* (modèle d'un tronçon), la

fonction δ_{int} est réalisée pour déterminer le nouvel état et définir via une variable σ la durée de l'état. La fonction $ta(S)$ se limite donc à σ . La vitesse et la position du véhicule considéré à la fin de la phase doivent aussi être connues, en considérant ce qui l'entoure. Pour un véhicule n'ayant pas de véhicule devant lui et étant en accélération, il faut par exemple déterminer si il peut effectuer son accélération jusqu'à atteindre la vitesse maximale du tronçon. Si tel est le cas, il passera en état libre (*RUNNING*) à la fin de la phase accélération (*IN_ACCELERATION*), mais si la distance ne peut pas être garantie (par exemple lorsqu'une file d'attente s'est formée), il devra adopter une vitesse inférieure, voire décélérer immédiatement.

Le passage en événements discrets implique de ne calculer l'état, l'accélération, la vitesse et la position d'un véhicule que lors d'un événement. Durant ce laps de temps entre événements, une phase, le véhicule restera dans le même état. Néanmoins, la vitesse v et la position x ne sont pas des constantes mais suivent les lois du mouvement d'un objet dans un espace 1D et dépendent de la variation de l'accélération à chaque changement d'état :

- $v(t) = v(t_0) + \sum_{k=0}^{k=N} a_k(t_k^f - t_k^i)$ où $v(t_0)$ est la vitesse initiale, a_k est l'accélération lors de la k -ème phase, t_k^f et t_k^i sont les dates de fin et de début de la k -ème phase et $t_N^f = t$ avec t comme date avant la fin de la phase courante N
- $x(t) = x(t_0) + \sum_{k=0}^{k=N} (v(t_k^i) + \frac{1}{2}a_k(t_k^f - t_k^i))(t_k^f - t_k^i)$ où $v(t_k^i)$ est la vitesse en début de k -ème phase

De plus, un mécanisme d'ouverture / fermeture est introduit. Lorsque la somme de la taille effective S_n des véhicules dépasse un certain seuil fixé, un événement est envoyé sur le port *CLOSE* au modèle précédant le tronçon pour signifier que celui-ci est plein, en attendant que des véhicules en sortent.

Un noeud représente une intersection entre plusieurs tronçons et fonctionne aussi selon un

mécanisme d'ouverture / fermeture. Au début de la simulation, il pourra être fermé pendant un certain laps de temps. Passé cette période, il alternera entre phase ouverte et phase fermée pendant des durées définies. Le temps nécessaire à un véhicule pour le traverser est aussi paramètre du modèle.

A ces deux modèles s'ajoutent les générateurs de véhicules (*Generator*) et les générateurs de perturbations (*Disruptor*). Le modèle global peut donc s'exprimer comme suit :

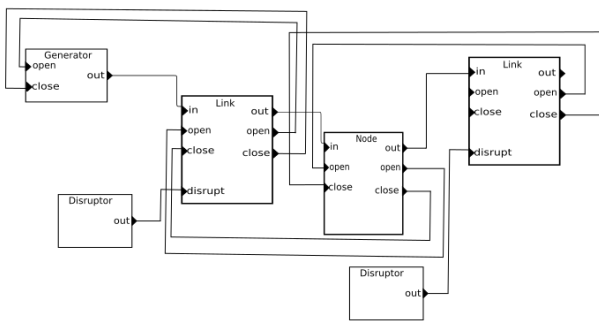


FIGURE 7 – Modèles atomiques à l'échelle microscopique.

La dynamique décrite jusqu'à maintenant considère qu'un suiveur n'est pas perturbé par le changement de comportement du leader. Avec le modèle de Gipps, un suiveur réagit aux changements et y réagira toujours. Selon Lazar [24], cela peut être considéré comme une faiblesse du modèle, car il ne prend pas en compte la perception du conducteur, qui ne pourra pas voir les changements si le leader est trop éloigné de lui. On propose donc d'introduire un seuil C , qui correspond simplement à la différence de vitesse sur la différence de position. Si ce seuil est trop faible (< 0.005), cela signifie que le suiveur ne doit pas être affecté par le changement de comportement du leader.

$$C = \frac{v_{n-1} - v_n}{x_{n-1} - x_n}$$

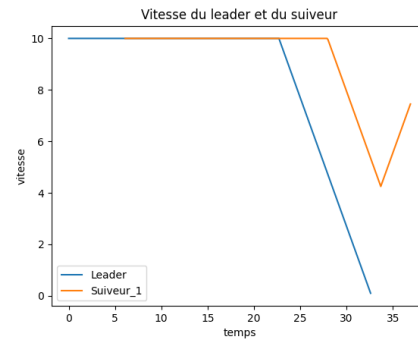


FIGURE 8 – Vitesse d'un véhicule leader et d'un suiveur avec arrêt du leader.

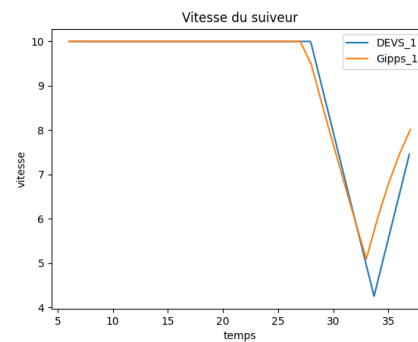


FIGURE 9 – Comparaison de la vitesse du véhicule suiveur entre DEVS et Gipps.

L'impact de ce seuil est particulièrement visible sur la figure 10 où le modèle de Gipps applique une forte décélération pour palier à la différence de vitesse avec le leader. Le modèle DEVS reste en état libre et est capable de garantir la distance de sécurité même après la décélération du leader. La décélération est due à la fermeture du nœud suivant. On peut observer le processus de redémarrage à $t = 120s$.

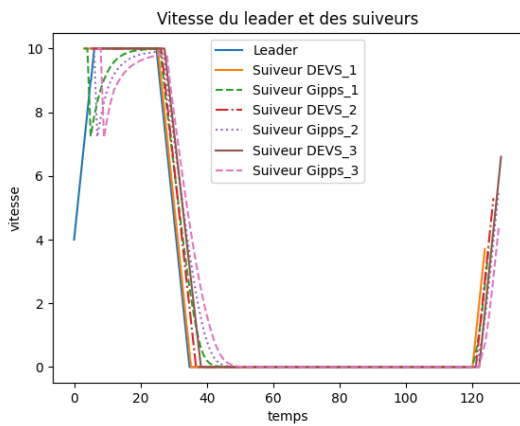


FIGURE 10 – Vitesse d’un véhicule leader et de 3 suiveurs en DEVS et en Gipps.

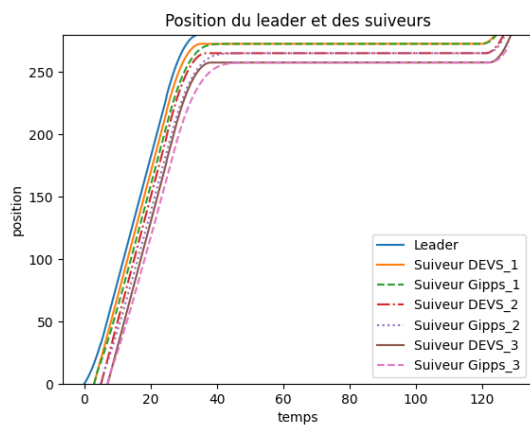


FIGURE 11 – Position d’un véhicule leader et de 3 suiveurs en DEVS et en Gipps.

Le leader (véhicule en tête du tronçon) peut être sujet à des perturbations via la fonction δ_{ext} : des modifications de son comportement. Ces perturbations prennent la forme de phases d’accélération ou de décélération puis une phase de vitesse constante et ensuite une phase de retour à la vitesse maximum du tronçon. Dans le cas d’une phase d’accélération, la vitesse du leader pourra être supérieure à celle autorisée par le tronçon, mais le suiveur ne cherchera pas à dépasser cette limite. Ces perturbations ont pour but de représenter des événements imprévus qui sont capables d’affecter le comportement du leader (traversée de piétons par exemple), mais pourraient par la suite être étendues à des perturbations ”spa-

tiales” sur la position des véhicules afin d’illustrer différentes infrastructures où la vitesse est limitée telles que les ponts et les tunnels.

Lors d’un cycle de décélération/accélération, le suiveur réagit par défaut aux changements de comportement du leader en adoptant une accélération garantissant les distances de sécurité. Le suiveur ne sachant pas quand le leader arrêtera sa décélération, le modèle calcule la date où sa vitesse sera nulle. Afin d’améliorer cette première approche, nous avons adapté la contrainte sur le freinage de Gipps en calculant l’instant T où le suiveur décélère.

$$T = t + K\tau \text{ où } K = \lceil \frac{-c0}{c1} \rceil$$

$$\text{et } c0 = 3b_n\tau v_n + 2b_n(S_n - (x_{n-1} - x_n)) + \frac{b_n}{b}v_{n-1}^2 - v_n^2$$

$$\text{et } c1 = b_n\tau(2v_n - \tau * \hat{b})$$

Dès que le leader stoppe sa décélération, le suiveur réagit et adapte une nouvelle fois son comportement. Les quatre graphes suivants montrent ces adaptations, en particulier la figure 15 où la décélération du suiveur est accrue après le passage en vitesse constante du leader et du temps de réaction. Le modèle Gipps propose lui une décélération non constante durant toute la phase. On pose $t0$ la date d’arrivée du suiveur et d la durée de la phase vitesse constante de la perturbation avec des phases accélération/décélération à 5s.

Les expérimentations sont réalisées sur une base de tests unitaires. Les cas traités comportent un unique tronçon, un noeud en fin de tronçon et de 2 à N véhicules.

Le comportement du suiveur lors des phases de décélération s’approche de celui de Gipps, notamment dans le cas où le suiveur est relativement proche du leader et où la perturbation est courte comme la figure 12. Dans le cas où la distance est plus importante au moment de la perturbation, on observe justement un délai dû

au déclenchement de la contrainte provenant de Gipps, ce qui amène à une phase de décélération plus tardive (13).

En revanche, si la phase de vitesse constante est plus longue, la seconde phase de décélération du suiveur sera plus longue et donc s'écartera de la dynamique non linéaire de Gipps. Encore une fois, lors de la réception de l'événement de changement de comportement du leader, le suiveur adapte son comportement et adopte une décélération garantissant la distance de sécurité si le leader maintient sa vitesse.

Le troisième changement de comportement du leader (phase d'accélération) va mettre fin à la phase de décélération du suiveur dans les cas où la vitesse constante du leader est courte mais si cette phase est longue, le suiveur aura adopté une vitesse constante égale à la vitesse du leader (voir figures 14 et 15).

Les phases d'accélération du suiveur sont aussi purement linéaires en DEVS et plus progressives (non linéaires) en Gipps. Le suiveur en DEVS finira donc son accélération bien plus tôt. Il pourrait être intéressant de mettre en relation cette phase avec d'autres modèles de la littérature tel que l'IDM pour vérifier ou non cette propriété. L'intérêt de ces phases de décélération / accélération non constantes doit aussi être étudié. Une application en DEVS de celles-ci entraînera une augmentation de la complexité du modèle et aura potentiellement un impact sur l'efficacité.

Dans cet article, nous avons focalisé notre discours sur le tronçon et la dynamique du véhicule au sein de ce tronçon. D'autres aspects ont été pris en charge : l'arrivée d'un véhicule dans un tronçon et la sortie du véhicule en fonction de l'état de l'élément suivant. La dynamique lors de ces deux événements peut être complexe : par exemple, en fonction de l'état (vitesse, accélération et phase) du véhicule lors de son arrivée et du leader. De nombreux cas particuliers ont été traités.

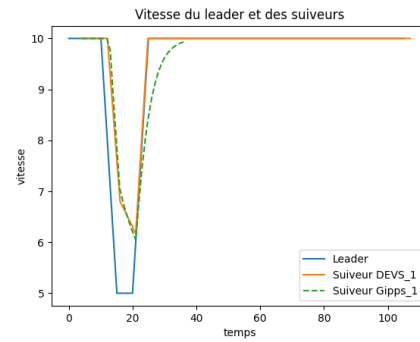


FIGURE 12 – $t_0 = 4s$, $d = 5s$.

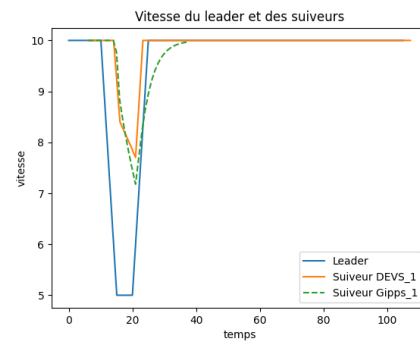


FIGURE 13 – $t_0 = 6s$, $d = 5s$.

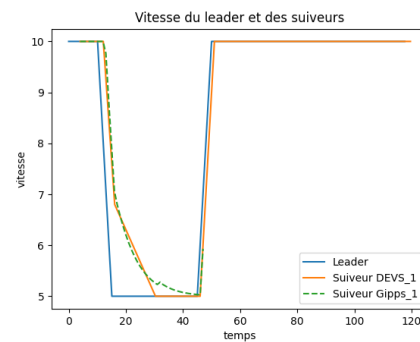


FIGURE 14 – $t_0 = 4s$, $d = 30s$.

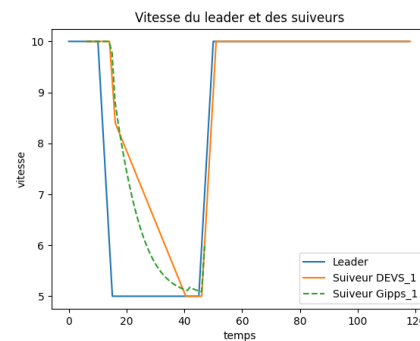


FIGURE 15 – $t_0 = 6s$, $d = 30s$.

L'étape suivante va consister à étendre la collection des modèles aux éléments de connexion entre les tronçons : les carrefours, les voies d'accès, les rond-points, ...

3.4 Couplage

Un modèle DEVS met en jeu deux éléments : son interface via les ports et les événements et sa dynamique. Il est possible de connecter deux modèles si la sémantique des ports et des événements est partagée. Dans notre cas et quel que soit le niveau, le modèle de tronçon est constitué de trois ports d'entrée et de trois ports de sortie (figure 16).

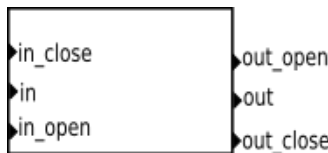


FIGURE 16 – Interface DEVS commune entre les deux échelles.

Les événements liés aux ports *in* et *out* transportent des véhicules. La définition des véhicules à l'échelle microscopique englobe celle de l'échelle mésoscopique. En mésoscopique, seul le trajet du véhicule (l'ensemble des tronçons à parcourir au sein du réseau) est à définir. En microscopique, le trajet est accompagné de la longueur du véhicule, de la vitesse max, de l'accélération/décélération max, du temps de réaction (si on désire l'individualiser) et de son état. L'état se compose de la vitesse, l'accélération et la phase courante (si le véhicule est en phase d'accélération, de décélération, à vitesse constante, à l'arrêt ou en cours de redémarrage). D'autre part, les événements d'ouverture et de fermeture d'élément du réseau sont simplement des signaux : des événements sans données.

Coupler deux modèles de tronçons à des échelles différentes consiste à compléter les informations d'un véhicule lorsque ce dernier sort d'un tronçon à l'échelle mésoscopique.

Cette transformation se passe au sein de noeuds spéciaux et consiste à affecter une vitesse, une accélération/décélération et une phase. En sortie de tronçon mésoscopique, un véhicule est soit à la vitesse max du tronçon soit en cours de redémarrage. Par défaut, cette information n'est pas utile pour la logique de l'échelle mésoscopique. Néanmoins, pour notre couplage, l'événement de sortie sur le port *out* est composé du véhicule et d'un booléen spécifiant son état (à vitesse max ou non). Afin de pouvoir compléter l'état du véhicule sortant, le modèle de noeud doit disposer sous forme d'un paramètre de la valeur de la vitesse max du tronçon.

Dans le cas d'une connexion micro vers méso, aucune transformation n'est nécessaire. Quel que soit l'état du véhicule, le véhicule est considéré comme étant à vitesse max à l'arrivée dans le tronçon.

4 Conclusion et perspectives

La littérature sur la modélisation du trafic aux échelles mésoscopique et microscopique est foisonnante ([25, 26]) mais une approche purement à événements discrets unifiant les deux échelles n'existe pas. Le formalisme Parallel-DEVS a permis d'offrir un cadre formel et opérationnel de modélisation à événements discrets. La reformulation des propriétés et des dynamiques de certains modèles de la littérature à chacune des échelles nous a conduit à proposer une collection de modèles atomiques et couplés pour répondre au couplage de modèles de trafic. Le code est disponible à l'adresse suivante : <https://gitlab.com/artis-traffic>. La stratégie utilisée pour chaque niveau, nous a permis de coupler les deux niveaux tant en garantissant la logique temporelle que les points de vue. Un véhicule possède la même définition à chaque niveau mais sa dynamique diffère.

Le modèle ainsi développé a deux domaines d'application : l'amélioration de la qualité de l'air à l'échelle d'une agglomération par

identification des sources via des méthodes de calibration et la mise au point de protocoles réseaux ad-hoc à base de clusters de véhicules. Le couplage entre simulation et optimisation est au coeur des deux applications. L'utilisation d'un moteur DEVS permet de prendre en charge des systèmes de grande taille avec des temps de simulation raisonnables. De plus, DEVS permet d'avoir une grande modularité, que ce soit dans la description de réseaux ou encore dans des changements d'échelles.

Afin de couvrir les différentes échelles, il serait intéressant d'étudier le couplage avec des modèles macroscopiques. Ces derniers représentent les véhicules sous forme de flux. Il est donc nécessaire d'imaginer des modèles de couplage dont l'objectif est d'une part, d'individualiser un flux en véhicules et d'agréger les véhicules en flux. Cette opération a déjà été menée dans le cadre de l'exemple A15-A115 car les entrées du réseau étaient exprimées sous forme de flux.

Remerciements :

Nous tenons à remercier la société Néovya pour nous avoir fourni des données réelles afin de valider à la fois les concepts de modélisation et la méthodologie de calibration. De plus, ces travaux de recherche sont financés par la Région Hauts-de-France, le Pôle Métropolitain de la Côte d'Opale et l'Université du Littoral - Côte d'Opale.

Références

- [1] A. C. H. CHOW et B. P. ZEIGLER. "Parallel DEVS : a parallel, hierarchical, modular modeling formalism". In : *Proceedings of Winter Simulation Conference* (1994), p. 716-722.
- [2] M. MAHUT, M. FLORIAN et N. TREMBLAY. "Application of a simulation-based dynamic traffic assignment model". In : *Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems*. 2002, p. 439-444.
- [3] B. GREENSHIELDS, J. BIBBINS, W. CHANNING et H. MILLER. "A study of traffic capacity". In : *Highway research board proceedings*. T. 1935. National Research Council (USA), Highway Research Board. 1935.
- [4] F. KESSELS, J. LINT, C. VUIK et S. HOOGENDOORN. "Genealogy of traffic flow models". In : *EURO Journal on Transportation and Logistics* 4 (jan. 2014), p. 1-29.
- [5] M. P. BONHOMME A. et P. S. "Simuler le trafic routier à partir de données réelles". In : *Revue des Sciences et Technologies de l'Information-Série RIA : Revue d'Intelligence Artificielle* 30.3 (2016), p. 329-352.
- [6] P. S. DONIEC A. Mandiau R. et E. S. "A behavioral multi-agent model for road traffic simulation". In : *Engineering Applications of Artificial Intelligence* 21.8 (2008), p. 1443-1454.
- [7] R. B. CETIN N. Nagel K. et V. A. "Large-scale multi-agent transportation simulations". In : *Computer Physics Communications* 147.1-2 (2002), p. 559-564.
- [8] M. P. KUBERA Y. et P. S. "IODA : an interaction-oriented approach for multi-agent based simulations". In : *Autonomous Agents and Multi-Agent Systems* 23.3 (2011), p. 303-343.
- [9] E. BOURREL. "Modélisation dynamique de l'écoulement du trafic routier : du macroscopique au microscopique". In : *These de Doctorat, l'Institut National des Sciences Appliquées de Lyon, France* (2003).
- [10] CEREMA. *Théorie du trafic et régulation dynamique*. Bron : Cerema, 2018. Collection : Connaissances. ISBN : 978-2-37180-275-9.
- [11] J. OLSTAM et A. TAPANI. "Comparison of car-following models". In : 2004.

- [12] D. C. GAZIS, R. HERMAN et R. W. ROTHERY. “Nonlinear Follow-the-Leader Models of Traffic Flow”. In : *Operations Research* 9.4 (1961), p. 545-567.
- [13] M. BANDO, K. HASEBE, A. NAKAYAMA, A. SHIBATA et Y. SUGIYAMA. “Dynamical model of traffic congestion and numerical simulation”. In : *Phys. Rev. E* 51 (2 fév. 1995), p. 1035-1042.
- [14] M. TREIBER, A. HENNECKE et D. HELBING. “Congested Traffic States in Empirical Observations and Microscopic Simulations”. In : *Physical Review E* 62 (fév. 2000), p. 1805-1824.
- [15] L. A. PIPES. “An Operational Analysis of Traffic Dynamics”. In : *Journal of Applied Physics* 24.3 (1953), p. 274-281.
- [16] P. GIPPS. “A behavioural car-following model for computer simulation”. In : *Transportation Research Part B : Methodological* 15.2 (1981), p. 105-111.
- [17] R. WIEDEMANN. “SIMULATION DES STRASSENVERKEHRSFLUSSES.” In : 1974.
- [18] A. C.-H. CHOW. “Parallel DEVS : A Parallel, Hierarchical, Modular Modeling Formalism and Its Distributed Simulator”. In : *Trans. Soc. Comput. Simul. Int.* 13.2 (déc. 1996), p. 55-67.
- [19] B. ZEIGLER, A. MUZY et E. KOFMAN. *Zeigler, B. P., Muzy, A., Kofman, E. (2018). Theory of Modeling and Simulation : Discrete Event & Iterative System Computational Foundations. Academic Press. Déc. 2018.*
- [20] F. BARROS, B. ZEIGLER et P. FISHWICK. “Multimodels and dynamic structure models : an integration of DSDE/DEVS and OOPM”. In : *1998 Winter Simulation Conference. Proceedings (Cat. No.98CH36274)*. T. 1. 1998, 413-419 vol.1.
- [21] M. L. FREDMAN et R. E. TARJAN. “Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms”. In : *J. ACM* 34.3 (juill. 1987), p. 596-615.
- [22] G. QUESNEL, R. TREPOS et E. RAMAT. “Observations of Discrete Event Models”. In : *Simultech* (2012), p. 32-41.
- [23] V. VENDI, A. PESTELLE, S. TARI et E. RAMAT. “Optimisation multiobjective pour le calibrage d’un simulateur à événements discrets : modélisation du trafic routier”. In : *ROADEF* (2021).
- [24] H. LAZAR, K. RHOULAMI et D. RAHMANI. “A Review Analysis of Optimal Velocity Models”. In : *Periodica Polytechnica Transportation Engineering* 44.2 (2016), p. 123-131.
- [25] A. ROMANOWSKA et K. JAMROZ. “Comparison of Traffic Flow Models with Real Traffic Data Based on a Quantitative Assessment”. In : *Applied Sciences* 11.21 (2021), p. 9914.
- [26] H. U. AHMED, Y. HUANG et P. LU. “A Review of Car-Following Models and Modeling Tools for Human and Autonomous-Ready Driving Behaviors in Micro-Simulation”. In : *Smart Cities* 4.1 (2021), p. 314-335.