
Environnement multidimensionnel pour contextualiser les interactions des agents

Application à la simulation du trafic routier urbain

**Stéphane Galland¹, Flavien Balbo², Gauthier Picard²,
Olivier Boissier², Nicolas Gaud¹, Sebastian Rodriguez³**

1. Université Bourgogne Franche-Comté, UTBM, LE2I UMR CNRS 6306
13 rue Ernest Thierry-Mieg 90010 Belfort, France
{stephane.galland,nicolas.gaud}@utbm.fr

2. Laboratoire Hubert Curien UMR CNRS 5516, Institut Henri Fayol
MINES Saint-Etienne 158 Cours Fauriel 42000 Saint-Étienne, France
{flavien.balbo,gauthier.picard,olivier.boissier}@mines-stetienne.fr

3. Grupo de Investigación en Tecnologías Informáticas Avanzadas
Facultad Regional Tucumán, Universidad Tecnológica Nacional
Rivadavia 1050, San Miguel de Tucumán, CPA T4001JJD, Argentine
sebastian.rodriguez@gitia.org

RÉSUMÉ. L'environnement, en tant qu'espace partagé entre agents, est un élément essentiel des systèmes multiagents (SMA). Selon les systèmes, il existe différentes classes de modèles d'environnement. Il est qualifié de physique ou communicationnel selon que les agents interagissent par des actions situées ou des échanges de messages, mais également de social si un modèle social régit les interactions. Chacune de ces classes est déclinable en dimensions dotées chacune de son propre modèle. Pour un système complexe dans lequel différentes dimensions doivent être combinées, il n'existe que des solutions ad hoc pour obtenir un modèle d'environnement global. La conséquence est la réutilisabilité et la modularité limitée des modèles d'environnement. Une alternative usuelle est de considérer l'environnement comme la juxtaposition de ses dimensions et faire de l'agent le lieu de jonction et de combinaison des informations véhiculées dans ces dimensions. Ce choix de conception accentue la complexité des agents et rend dépendante de ces derniers la gestion des interactions entre dimensions. Enfin, la mise en œuvre d'interactions contextuelles, c-à-d. contraintes par des règles du SMA, impose une gestion des interactions indépendante des agents. Dans cet article, nous proposons un modèle unifié assurant la combinaison de dimensions pour la mise en œuvre d'interactions contextualisées entre agents. Ce modèle est développé avec le langage multiagent SARL. Nous illustrons cette proposition par une application de simulation de trafic routier dans la ville de Belfort.

ABSTRACT. The environment, as a space shared between agents is essential to multi-agents systems (MAS). Depending on the systems, it responds to different points of view. It is described as physical or communicative as that agents interact through situated actions or exchanging messages, or social if a social model governs the interactions. Each of these views could be declined into a dimension with its own model. For a complex system in which different dimensions should be combined, there are only ad-hoc solutions for a global environment model. The consequence is a limited reusability and modularity of the use of environment models. An alternative is to consider the environment as the juxtaposition of its dimensions and make the agent the place for combining the information conveyed in these dimensions. This choice of design increases the agent complexity, and makes them dependent upon the mechanisms for managing the interactions between the dimensions. Finally, the implementation of contextual interactions, i.e. constrained by the rules of the MAS, requires a management of the interactions that is independent of the agents. In this paper, a model is proposed for ensuring the combination of the environment dimensions for the modeling and implementation of contextualized interactions between agents. This model is developed with the SARL agent-oriented programming language. This proposal is illustrated with a road traffic simulation application to the city of Belfort.

MOTS-CLÉS: environnement support à l'interaction, environnement physique, environnement communicationnel, langage de programmation, trafic routier.

KEYWORDS: environment as interaction support, physic environment, communication environment, programming language, road traffic.

DOI:10.3166/RIA.30.81-108 © 2016 Lavoisier

1. Introduction

L'environnement, en tant qu'espace partagé entre agents, est un élément essentiel des systèmes multiagents (SMA). Il a notamment été montré qu'il est possible de prendre avantage d'une modélisation explicite de l'environnement afin de supporter, ou d'améliorer, les possibilités d'interaction entre agents (Ricci *et al.*, 2011; Platon *et al.*, 2007; Saunier *et al.*, 2014). Par exemple, la propriété d'observabilité de l'environnement est utilisée dans les SMA réactifs pour supporter les modèles d'interaction indirecte, mais également dans les SMA cognitifs pour supporter des modèles d'interaction comme l'écoute flottante (Platon *et al.*, 2007) ou les communications multipartites (Saunier *et al.*, 2014). À chaque modèle d'environnement correspondent une dynamique endogène et un modèle d'interaction qui sont régis par des règles qui doivent être indépendantes des agents. Par exemple, si l'interaction entre deux agents a lieu selon la distance ou la présence d'obstacles entre eux, ce n'est pas aux agents d'appliquer la règle de mise en œuvre. Même si l'interaction est initiée par l'agent, elle doit s'exécuter ou non selon le type d'environnement dans lequel elle prend place.

Il existe trois classes de modèles d'environnement (Weyns *et al.*, 2007; Odell *et al.*, 2009). La première concerne les modèles physiques où les agents et les objets ont une position, et interagissent par des actions elles aussi localisées. La seconde concerne les modèles communicationnels où les agents interagissent par échanges de messages.

Enfin, la troisième concerne les modèles sociaux qui sont une spécialisation des modèles précédents, car des structures sociales régissant les interactions entre les agents sont spécifiées et gérées. Chacune de ces classes contient des modèles spécifiques, que nous appelons dimension, selon la dynamique et les règles choisies pour modéliser les interactions. Par exemple, un modèle d'environnement physique peut être décliné en une dimension sonore ou visuelle selon qu'il supporte un modèle de propagation du son ou de la lumière. Afin de s'approcher au plus près de la réalité des interactions dans un environnement complexe, il faut prendre en compte l'ensemble des dimensions pouvant avoir une influence. Par exemple dans un poste de commandement, l'envoi d'un message entre deux agents aura des conséquences dans la dimension sociale à la suite de la prise en compte de son contenu par le récepteur, mais aussi dans la dimension physique par une augmentation du fond sonore. Enfin, un troisième agent voyant dans la dimension visuelle les agents échanger pourra adapter son comportement s'il dépend de l'un des protagonistes du dialogue.

La modélisation d'un environnement complexe, comme celui de notre exemple, est le résultat ad hoc de l'intégration, ou de la juxtaposition, de ses différentes dimensions. Par exemple, les communications situées proposées dans (Weyns *et al.*, 2004) sont une combinaison ad hoc de modèles physiques et communicationnels. Il y a juxtaposition des dimensions lorsque l'agent doit dissocier ses interactions selon la dimension, comme c'est par exemple le cas avec l'utilisation d'artefacts (Ricci *et al.*, 2011). Dans ce dernier cas, c'est l'agent qui réalise la mise en relation des informations véhiculées dans les différentes dimensions où il interagit. Dans notre exemple, un des deux agents impliqués dans le dialogue pourrait se déplacer à la suite de l'échange. Seulement, si dans chaque dimension des règles régissent l'interaction, alors l'agent ne pourra agir que selon les interactions le concernant dans au moins une des dimensions. Dans notre exemple, le troisième agent ne pourrait donc pas adapter son comportement car il n'aurait pas connaissance de l'échange.

Dans cet article, nous adoptons une vision systémique de l'environnement dans laquelle chaque dimension représente un sous-système en interaction avec les autres. Notre objectif est de formaliser la gestion des conséquences d'une même interaction dans les différentes dimensions, et de modifier ainsi le comportement global du SMA. Pour cela, nous explicitons les interactions entre les dimensions par des règles dont la gestion est assurée par l'environnement. La nature discrète ou continue d'un modèle de l'environnement, ou la dimensionnalité de son espace (1D, 2D, 3D) ne sont pas discutées dans cet article. En effet, nous nous focaliserons sur les relations inter-dimensions possibles, et en considérant que chacune d'entre elles peut être définie par sa propre nature et sa propre dimensionnalité.

Les principaux avantages de cette approche concernent les phases de conception et de développement d'un SMA. Lors de la phase de conception, l'identification claire des rôles de l'environnement et des agents simplifie leur modélisation respective et permet l'utilisation d'interactions contextuelles. Dans l'exemple précédent, il devient possible de contraindre la réception d'un message à la valeur du fond sonore. L'explicitation des règles d'interaction entre les dimensions participe à la simplification

de la modélisation et autorise une gestion dynamique de l'interaction interdimension par l'ajout ou le retrait de ces règles. Dans l'exemple, la conséquence sonore d'un message peut être retirée et remplacée par une contrainte sur la distance s'il existe une dimension physique où cette valeur est gérée. Lors de la phase de conception, nous pensons que notre approche favorise la modularité des modèles des différentes dimensions de l'environnement et la réutilisabilité de modèles (ou de code) en permettant la définition d'abstractions communes aux différentes dimensions et en spécialisant ces abstractions aux cas d'applications. Par exemple, si nous proposons la définition de capacités d'interaction avec une dimension physique offerte aux agents cette capacité définit les mécanismes généraux d'interaction. Dans notre exemple de simulation du trafic routier, elle peut alors être raffinée pour créer une capacité dédiée à des conducteurs de véhicules ou à des piétons, chacun utilisant des mécanismes d'interaction différents.

Dans cet article, nous proposons un modèle permettant la combinaison des dimensions pour la mise en œuvre d'interactions contextualisées entre agents. Ce modèle est développé avec le langage multiagent SARL¹. Ce langage s'appuie directement sur les concepts multiagents, et donne ainsi l'opportunité de prendre en compte les différentes dimensions de l'environnement au sein d'un même langage.

La suite de l'article est organisée de la manière suivante. En section 2, nous présentons un état de l'art des différentes dimensions de l'environnement dans un contexte multiagent et comment l'interaction est prise en compte dans ces dimensions. En section 3 nous discutons la problématique de l'usage d'un environnement multidimensionnel et nous introduisons notre exemple illustratif de gestion du trafic routier. En section 4, nous présentons la modélisation des dimensions de l'environnement et la manière dont on peut les relier. La mise en œuvre de ce modèle dans le langage SARL est présentée en section 5. La modélisation de l'environnement combiné, et son application dans notre exemple de simulation de trafic sont détaillées dans la section 6. Nous concluons enfin et donnons les perspectives envisagées pour ce travail en section 7.

2. État de l'art

Nous proposons dans cette section une présentation générale de ce que l'on considère comme propre à l'environnement dans une modélisation multiagent puis discutons les différentes classes de modèles d'environnement.

2.1. *Qu'est-ce que l'environnement ?*

De nombreux travaux, dont ceux résultants du groupe de travail E4MAS² ("Environment for multiagent systems"), ont montré que la modélisation d'un système

1. <http://www.sarl.io>

2. <https://distrinet.cs.kuleuven.be/events/e4mas/>

multiagent nécessite en plus de la modélisation des agents, la modélisation d'un environnement. Pour E4MAS, "The environment is a first-class abstraction that provides the surrounding conditions for agents to exist and that mediates both the interaction among agents and the access to resources." Selon cette définition, l'environnement est à la fois un support d'exécution (Rodriguez *et al.*, 2014) et d'interaction (Saunier *et al.*, 2014). Dans cet article, nous considérons uniquement cette seconde caractéristique. Lorsqu'un agent déplace un objet ou envoie un message, il le fait par interaction avec son environnement qui a la responsabilité de sa mise en œuvre qu'il s'agisse de sa validation ou de la prise en compte de ses conséquences sur l'évolution de l'environnement. Afin de pouvoir prendre en compte l'état de l'agent dans la gestion des interactions par l'environnement, la notion de corps a été proposée dans (Platon *et al.*, 2007; Saunier *et al.*, 2015). Le corps correspond à une réification de l'état de l'agent qui est perceptible depuis l'environnement.

Dans (Weyns *et al.*, 2007), les auteurs proposent que les responsabilités suivantes soient assurées par l'environnement :

- **Structuration** : l'environnement permet de structurer le SMA aussi bien physiquement que socialement, et prend en charge les communications.
- **Gestion du cycle de vie** : l'environnement maintient les ressources et les agents, ainsi que de leur dynamique d'exécution.
- **Observabilité** : l'environnement fournit les structures de représentation, ressources et "corps" des agents.
- **Accessibilité** : l'environnement gère l'accessibilité aux structures et aux ressources.
- **Régulation** : l'environnement contrôle les accès aux structures et aux ressources, et maintient les lois d'accès.
- **Ontologie** : l'environnement fournit une ontologie commune qui peut être consultée par les agents.

Dans (Weyns *et al.*, 2007), les auteurs présentent une architecture de référence proposant l'ensemble de ces fonctionnalités. Il s'agit ensuite d'implémenter leur mise en œuvre selon le modèle spécifique de SMA. Dans (Claes *et al.*, 2011) l'environnement modélise le trafic routier et régit les interactions indirectes à base de phéromones entre les agents véhicules. Dans (Ricci *et al.*, 2011), l'environnement est un ensemble d'artefacts correspondant aux services propres à un environnement. Cette approche de l'environnement comme un ensemble de fonctionnalités correspond à l'approche par juxtaposition des dimensions de conception d'un environnement complexe. Ainsi, il existe dans (Weyns *et al.*, 2007) des modules distincts pour la communication, la perception et l'interaction (au sens de la gestion des actions dans l'environnement). Dans (Ricci *et al.*, 2011), il y aura autant d'artefacts que de modèles d'interaction. Selon cette approche, c'est à l'agent, utilisateur des services, de faire le lien entre les informations issues des différents services.

Il s'agit d'une implémentation par des modules ou des artefacts de ce que nous appelons dimension. Cependant, les relations entre les différents modèles d'interaction

ne sont pas formalisées. De plus, il n'est pas possible de faire exécuter des règles d'interaction, en dehors des agents, et utilisant des informations issues des différents supports de l'interaction, sans le développement d'un nouveau service dédié (module, artefact ou autre). Par exemple, la gestion de communications situées demandera le développement d'un nouveau service exploitant les données issues des services gérant respectivement la localisation et la communication des agents. Notre objectif est de conserver cette vision modulaire des dimensions de l'environnement et de formaliser leur relation afin d'éviter le développement ad hoc de services dédiés.

2.2. Environnements physiques, communicationnels et sociaux

L'environnement est un espace partagé où les agents interagissent par leurs actions ou communications, éventuellement selon un modèle social. Trois classes de modèles d'environnement sont ainsi obtenues (Weyns *et al.*, 2007), à savoir :

- L'environnement **physique** où les agents, ainsi que les objets, disposent d'une position explicite et produisent des actions elles aussi localisées (Michel, 2007). Cet environnement possède un ensemble de règles physiques afin de valider et de calculer les conséquences des actions des agents.
- L'environnement **communicationnel** permet l'échange de messages selon différentes modalités entre les agents.
- L'environnement **social** concerne la classe de modèle où les agents interagissent physiquement ou par la communication selon des règles sociales.

Pour les environnements physiques, des définitions (semi-)formelles ont été proposées par Gouaïch, Michel (2005); Michel (2007). Ces travaux mettent notamment en avant deux caractéristiques principales de l'environnement : sa dynamique endogène et la gestion de la concurrence entre les actions des agents. Par exemple, le modèle influence-réaction (Michel, 2007) prend en compte la simultanéité des actions dans un environnement en considérant les interactions initiées par les agents comme incertaines, en détectant puis résolvant tout conflit entre deux interactions. Dans (Mathieu *et al.*, 2014), les auteurs considèrent quatre patrons de conception d'un environnement physique, et formalisent sa représentation pour trois d'entre eux (par exemple, le déplacement d'un agent est présenté), tandis que le quatrième est une extension aux modèles sociaux avec la prise en compte d'accointances. Pour les environnements communicationnels, les travaux de Platon *et al.* (2007); Saunier *et al.* (2014); Zargayouna, Balbo (2013) modélisent l'environnement comme un régulateur des communications selon les contraintes individuelles et collectives des agents. Dans (Saunier *et al.*, 2014; Zargayouna, Balbo, 2013), l'environnement est un espace partagé dans lequel les agents déposent ou retirent les filtres décrivant le contexte de leurs interactions, afin, notamment, de gérer les communications multipartites. Dans notre exemple, le troisième agent pourrait déposer dans l'environnement un filtre décrivant son intérêt pour les messages émis par l'un des deux interlocuteurs, et ainsi les recevoir également. Les environnements sociaux sont caractérisés par l'explicitation des relations sociales entre agents, ces dernières influençant l'exécution des actions ou

des communications. Certaines approches organisationnelles, comme (Gouaïch, Michel, 2005; Piunti *et al.*, 2009), proposent d'intégrer l'environnement dans un modèle organisationnel.

Dans chacun de ces modèles, l'environnement a sa propre dynamique et fournit l'observabilité, l'accessibilité et la structuration du SMA selon des règles qui restreignent les interactions. Ainsi sa dynamique endogène et les règles d'interaction caractérisent chaque modèle d'environnement, qui est ainsi spécifique à un modèle de SMA et ne favorise pas la réutilisabilité et la modularité des modèles d'environnement.

Il n'existe à notre connaissance aucun modèle pour lier différentes dimensions sans utiliser l'agent comme intermédiaire. Pourtant, dans d'autres domaines, cette vision multidimensionnelle de l'environnement a été étudiée. Ainsi, dans le domaine de la sociologie, des chercheurs ont étudié la relation entre l'espace physique et l'espace de travail collaboratif. Par exemple, la proxémie désigne *l'ensemble des observations et théories que l'Homme fait de l'espace en tant que produit culturel spécifique* (Hall, 1990). Hall (1990) a divisé l'espace en zones sociales en utilisant la distance physique entre deux individus comme critère de séparation. Ces zones donnent une indication sur la manière avec laquelle un individu souhaite communiquer avec les autres individus dans son environnement de travail.

3. Positionnement

3.1. Problématique

Nous avons souligné dans la section précédente que la plupart des travaux considèrent l'environnement comme un bloc dont la modélisation répond à des fonctionnalités attendues et mettant en œuvre un modèle d'interaction particulier. **Nous proposons une approche différente qui considère l'environnement comme le résultat de la juxtaposition coordonnée de ses dimensions.** Chacune d'elles conserve en effet sa dynamique endogène et ses règles propres à un modèle d'interaction particulier. La problématique de modélisation d'un environnement pour un SMA n'est donc plus la définition d'un nouveau modèle d'interaction intégrant les dimensions prises en compte, mais la coordination des dimensions existantes.

Dans cet article, nous considérons trois types de coordination :

1. Une **interaction dont la perception dans une dimension est contrainte ou favorisée par une autre.** Dans notre exemple de poste de commandement, la réception du message dans l'environnement social est contrainte par des règles physiques comme le niveau sonore.

2. Une **même interaction a des modalités différentes selon les dimensions.** Dans l'exemple, le contenu du message modifie le comportement du récepteur et la forme du message modifie leur environnement physique (niveau sonore).

3. Une **interaction dans une dimension qui génère d'autres interactions dans d'autres dimensions.** Dans l'exemple, le troisième agent peut profiter du dialogue

pour s'informer des informations disponibles sur un tableau utilisé pour les interactions indirectes.

Nous pensons couvrir une large partie, si ce n'est la totalité, des relations pouvant exister entre les différentes dimensions de l'environnement. Toutefois, le modèle que nous proposons dans la suite de cet article n'est pas dépendant de cette énumération.

3.2. Exemple illustratif

Le domaine du transport a connu ces dernières années une forte mutation par l'introduction massive des nouvelles technologies de l'information. L'objectif est d'améliorer par une meilleure information la gestion de l'offre de transport. Dans un environnement ouvert, dynamique et incertain, il faut pouvoir déployer des systèmes capables de prendre en compte simultanément des interactions issues des trois classes de modèle d'environnement. Nous prendrons l'exemple d'un conducteur de véhicule communicant (communication intervéhiculaire et avec l'infrastructure) qui interagit physiquement par le déplacement de son véhicule, l'émission de son par l'avertisseur (ou une sirène) ou de signaux par ses optiques (ou un gyrophare). Il interagit par la communication en demandant la priorité au feu de signalisation d'un carrefour ou en annonçant des événements à son réseau social. Enfin, il interagit socialement en respectant ou non le Code de la Route. Toutes ces dimensions et ces sollicitations du conducteur existent déjà dans un système de trafic réel. Il s'agit d'un environnement complexe que nous proposons d'utiliser comme exemple illustratif.

Afin de montrer la faisabilité de notre proposition, nous proposons un simulateur de trafic dont l'environnement est le résultat de la combinaison de deux dimensions physiques et d'une dimension sociale. La première dimension physique assure le déplacement des véhicules ainsi que la régulation du trafic. Il s'agit d'un ensemble d'arcs équipés de capteurs qui permettent le calcul d'un indice de densité, et d'intersections où des feux sont équipés pour communiquer avec les véhicules. La seconde dimension physique concerne la propagation du son.

La dimension sociale ne concerne que les véhicules membres d'une communauté et équipés pour recevoir des messages. Cette dimension sera utilisée pour permettre à des véhicules prioritaires d'informer les membres de la communauté de leur passage. Similairement aux travaux proposés par Bhourri *et al.* (2012), un plan dynamique de feux régule les axes qui doivent être favorisés (durée du vert) selon l'importance du trafic (information issue de l'environnement physique, indice de densité) et la présence de véhicules prioritaires (information issue de l'environnement social).

Dans cet environnement, nous traitons le cas d'un véhicule de secours qui veut la priorité et qui envoie régulièrement un message à la communauté d'utilisateurs pour demander qu'on lui cède le passage. De plus, il informe par l'usage d'une sirène les véhicules dans le voisinage. Ce message contient un indice de priorité qui indique l'importance accordée par le véhicule à sa demande. À proximité d'un carrefour, ce message est reçu et relayé par le feu de signalisation à l'ensemble de la communauté

de conducteurs. Ce scénario illustre l'utilisation conjointe des trois types d'articulation :

1. Un feu ne prend en compte que les demandes issues des véhicules prioritaires (statut social de l'émetteur).
2. Le contenu du message modifie le comportement des membres de la communauté et la forme du message permet une augmentation de l'indice de densité de l'axe routier concerné et, par conséquent, une interaction indirecte avec les autres véhicules.
3. Une collision dans l'environnement physique provoque l'envoi d'un message d'alerte dans l'environnement social. À la réception de ce message, un véhicule de secours part pour se rendre par le plus court chemin sur le lieu de l'accident.

4. Modèle de l'environnement multidimensionnel

Dans cette section, nous proposons un modèle de l'environnement combinant plusieurs dimensions, qu'elles soient physiques, communicationnelles ou sociales. La définition du système multiagent est également raffinée afin d'intégrer cette perspective multidimensionnelle de l'environnement.

Nous considérons que chaque dimension de l'environnement est un sous-système préservant l'intégrité de l'état de l'environnement (l'évitement des interpénétrations et l'application de la gravité sont des exemples de préservation de l'intégrité de l'environnement dans sa dimension physique), et de gérer les interactions avec les agents dans le contexte de cette dimension.

Chaque dimension d est définie par :

$$d \in D: \langle \Sigma_d, \Lambda_d, \Psi_d, \text{Percept}_d, \text{Act}_d, \text{Process}_d, \sigma_d \rangle \quad (1)$$

où :

$\Sigma_d = \mathcal{P}(I \rightarrow VI)$ est l'ensemble des états possibles pour la dimension d . I est l'ensemble des informations stockées dans la dimension. VI est l'ensemble des valeurs pouvant être prises par ces informations. $\mathcal{P}(E)$ représente l'ensemble des parties de l'ensemble E . Il est aussi parfois noté 2^E .

Λ_d définit les lois régissant la dimension d . Ces lois permettent de contraindre et diriger les calculs de l'état suivant de la dimension de l'environnement à partir de son état courant $\sigma_d \in \Sigma_d$ et d'un ensemble d'actions Ψ_d pouvant être appliquées par les agents ou par l'environnement lui-même.

Pour un agent $a \in A$, la fonction Percept_d définit les mécanismes pour extraire un ensemble d'informations de la dimension d . Cette définition doit permettre de déterminer les éléments p qui sont "perçus" par l'agent. Par exemple, afin de modéliser le daltonisme, la fonction Percept_d peut être formulée pour ne percevoir que les ob-

jets proches dans la dimension physique (perception localisée), et en supprimant toute information relative à la couleur rouge qui ne peut être perçue pas un daltonien.

$$\begin{aligned} \text{Percept}_d: \Sigma_d \times A &\rightarrow \Sigma_d \\ \text{Percept}_d(\sigma_d, a) &= \{p | p \in \sigma_d \wedge p \text{ est perçue par } a\} \end{aligned} \quad (2)$$

La fonction Act_d définit les mécanismes permettant à un agent $a \in A$ d'agir dans la dimension d . Cette fonction a pour principal objectif de restreindre les capacités d'action de a dans la dimension d . Par exemple, une personne tétraplégique ne peut pas ou peu réaliser de mouvements dans la dimension physique de l'environnement.

$$\begin{aligned} \text{Act}_d: \mathcal{P}(\Psi_d) \times A &\rightarrow \mathcal{P}(\Psi_d) \\ \text{Act}_d(\psi_a, a) &= \{p | p \in \psi_a \wedge p \text{ est possible pour } a \text{ dans } d\} \end{aligned} \quad (3)$$

La fonction Process_d définit les processus endogènes à la dimension d . Ces processus modélisent la dynamique interne de la dimension de l'environnement. Ils génèrent des actions $\psi_d \in \Psi_d$ à partir de l'état courant $\sigma_d \in \Sigma_d$ de la dimension.

$$\begin{aligned} \text{Process}_d: \Sigma_d &\rightarrow \mathcal{P}(\Psi_d) \\ \text{Process}_d(\sigma_d) &= \{\psi_d\} \end{aligned} \quad (4)$$

Prenons l'exemple d'une partie de football en considérant les joueurs comme des agents et le ballon comme un objet. Après avoir été frappé par un joueur, le ballon doit suivre une trajectoire influencée par les propriétés (ex. gravité) et les dynamiques (ex. vent, pluie) de la dimension de l'environnement. Le mouvement du ballon en dehors du contrôle d'un agent est ainsi décrit par la fonction Process_d .

4.1. Environnement et dimensions

Dans cet article, nous considérons que l'environnement est un système composé par un ensemble de dimensions $D_e \in \mathcal{P}(D)$ et de règles représentées par $L_e \in \mathcal{P}(\Lambda_e)$, où Λ_e est l'ensemble des règles pouvant régir les relations entre les dimensions de l'environnement. L'environnement e est défini par :

$$e \in E: \langle D_e, L_e \rangle \quad (5)$$

Chaque règle $\lambda_e \in \Lambda_e$ prend comme paramètres la dimension qui lui est associée, ainsi que l'action dans cette dimension devant être traitée par λ_e . Chaque règle détermine une réaction particulière de l'environnement multidimensionnel à l'action donnée. Cette réaction est représentée par un tuple constitué par l'ensemble d'actions $\psi_{d'}$ dans les autres dimensions $d' \neq d$, et par l'action à appliquer dans la dimension d . Ainsi, le premier membre du tuple génère des actions dans une autre dimension à partir d'une action produite dans la dimension d'origine d . Par exemple, lorsqu'un mouvement dans la dimension physique provoque une collision, la règle peut être exprimée afin d'envoyer un message d'alerte dans la dimension sociale de l'environnement.

Certaines règles impliquent que l'action ψ_d dans la dimension d n'est applicable qu'après avoir été modifiée. Par exemple, la réception du message dans la dimension sociale peut être contrainte par la proximité dans la dimension physique entre l'émetteur et le récepteur du message. Afin de refléter cette capacité, la fonction λ_e détermine l'action ψ'_d qui doit être réellement considérée et appliquée dans la dimension d . Cette action est généralement la résultante de la transformation de ψ_d . Par exemple, considérons que ψ_d est un mouvement, représenté par un vecteur dans \mathbb{R}^k . La règle peut alors imposer que tous les mouvements soient unitaires : $\psi'_d = \psi_d / |\psi_d|$.

Chaque règle $\lambda_e \in \Lambda_e$ est par conséquent définie par :

$$\begin{aligned} \lambda_e : D \times \Psi_d &\rightarrow [D \rightarrow \mathcal{P}(\Psi_{\alpha \in D \setminus \{d\}})] \times \Psi_d \\ \lambda_e(d, \psi_d) &= \langle d' \mapsto \psi_{d'}, \psi'_d \rangle \end{aligned} \quad (6)$$

4.2. Dynamique de l'environnement

Dans cette section, la dynamique constituant le modèle d'environnement multi-dimensionnel décrit dans les sections précédentes est présentée. Cette dynamique est basée sur les fonctions d'exécution des agents (*Exec*), d'exécution des dynamiques endogènes à l'environnement (*Process_d*, décrite précédemment), d'application des règles de l'environnement (*Multi*), et enfin de changement effectif de l'état de l'environnement (*Apply*).

La fonction *Exec* permet d'exécuter les comportements associés aux agents :

$$\begin{aligned} Exec : E \times \Sigma &\rightarrow [D \rightarrow \mathcal{P}(\Psi_{dom})] \\ Exec(e, \sigma_e) &= \bigcup_{a \in A, d \in D_e} Act_d(f(d), a) \\ &\text{avec } f = Decide_a \left(\bigcup_{m \in D_e} Percept_m(\sigma_m, a) \right) \end{aligned} \quad (7)$$

Chaque agent a perçoit son environnement dans les différentes dimensions accessibles. Puis, il détermine les prochaines actions à réaliser dans l'environnement en appliquant la fonction $Decide_a : \Sigma_d \rightarrow [D \rightarrow \mathcal{P}(\Psi_d)]$. Dans cet article, les actions pouvant amener à un changement d'état de l'agent a ne sont pas traitées.

La fonction *Multi* décrit le comportement dynamique de l'environnement multi-dimensionnel. Elle permet d'appliquer les règles décrites dans l'ensemble L_e :

$$\begin{aligned} Multi : \Sigma \times [D \rightarrow \mathcal{P}(\Psi_{dom})] &\rightarrow [D \rightarrow \mathcal{P}(\Psi_{dom})] \\ Multi(\sigma, I) &= \bigcup_{d \in \text{dom } I, \lambda_e \in L_e, a \in I(d)} r_1 \cup [d \mapsto r_2] \text{ avec } \langle r_1, r_2 \rangle = \lambda_e(d, a) \end{aligned} \quad (8)$$

L'expression r_1 permet de générer des actions dans d'autres dimensions que la dimension d'origine d . L'expression $d \mapsto r_2$ permet d'appliquer l'action calculée par la règle λ_e dans la dimension d . r_1 et r_2 sont les deux membres du tuple retourné par la règle λ_e . Chacune de ces deux expressions associe des ensembles d'actions à des dimensions dans lesquelles ils devront s'appliquer.

SARL est la création d'un langage ouvert et facilement extensible. Un tel langage doit fournir un ensemble réduit de concepts essentiels à la mise en œuvre d'un système multiagent. Le métamodèle de SARL est basé sur quatre concepts principaux : Agent, Capacité, Espace et Adresse (illustrés par la figure 1).

Un agent (**Agent**) est une entité autonome exhibant un comportement. Ce comportement peut nécessiter l'utilisation de capacités (**Capacity**) de l'agent. La réalisation concrète d'une capacité est assurée par une compétence (**Skill**) possédée par l'agent. La relation entre les concepts de capacité et de compétence est du même ordre que la relation entre les concepts d'interface et de classe d'implantation dans la programmation orientée-objet : un **skill** est une implantation possible d'une **capacity**. Un agent doit pouvoir interagir avec d'autres agents ou avec son environnement. Il existe de nombreuses modélisations de l'interaction (Peña *et al.*, 2012) : messages, événements, influence-réaction, etc. Pour permettre la modélisation de ces interactions, le langage SARL propose le concept d'espace (**Space**) décrivant un espace d'interaction, dans lequel chaque agent possède une adresse. Dans le langage SARL, la définition d'un espace concret est proposée par défaut. Il s'agit d'un espace permettant de propager des événements, nommé **EventSpace** (et son implantation **EventSpaceImpl**). De nouveaux espaces d'interaction peuvent être définis en créant une spécification (**SpaceSpecification**) décrivant les attributs, les fonctions et les propriétés de ces espaces.

L'espace, au sens SARL, est l'abstraction permettant de définir un espace d'interaction entre des agents, ou entre des agents et un environnement. Nous utilisons ce concept pour définir les interactions entre un agent et l'environnement dans le cadre des dimensions physique et sociale de ce dernier. Dans la suite de cet article, nous utilisons l'espace SARL comme vecteur d'interaction entre un agent et une dimension d'environnement. Ainsi, il existera autant d'espace SARL que de dimensions de l'environnement. Pour chacun d'entre eux, nous définirons une capacité permettant à un agent de mettre en œuvre son interaction avec la dimension associée.

SARL est intégré dans l'environnement de développement Eclipse, et la nouvelle version de la plate-forme JANUS⁴ supporte pleinement ces concepts.

5.1. Dimension physique de l'environnement

Une dimension physique de l'environnement, $p \in D$, contient un ensemble d'objets, incluant les corps des agents. Afin de permettre à l'agent d'interagir avec une dimension physique de l'environnement, il est nécessaire de définir une capacité dédiée. Le script 1 décrit les fonctionnalités accessibles à un agent pour tous les types d'environnements physiques (**AbstractPhysicEnvironmentCapacity**). Cette capacité doit être adaptée au cadre d'application. La définition de **RoadEnvironmentCapacity** illustre un exemple de spécialisation qui est utilisée

4. <http://www.janusproject.io>

dans la section 6. Nous proposons cette distinction afin de faciliter la définition d'autres capacités dédiées à des dimensions qui utiliseraient des objets et des types géométriques différents. L'introduction de ces deux niveaux d'abstraction n'est pas spécifique à la dimension physique de l'environnement. Ce principe de généricité peut être également utilisé pour les autres dimensions de l'environnement, afin de faciliter la réutilisation et la modularisation des modèles.

```

    capacity AbstractPhysicEnvironmentCapacity {
2      def getLinearSpeed : double
4      def setPhysicalPerceptionAlterator(filter : PhysicalPerceptionAlterator)
4      def setInterestFilter(filter : InterestFilter)
6      def influence(inf : Influence)
6      def killBody
8      def getPosition : Object
8      def getOrientation : Object
    }
10   capacity RoadEnvironmentCapacity extends AbstractPhysicEnvironmentCapacity {
12     def getPosition : Pair<Segment, double>
12     def getOrientation : Direction
    }

```

Script 1 – Capacités d'interaction avec l'environnement physique (en SARL)

En accord avec le modèle présenté dans la section précédente, les mécanismes permettant aux agents d'agir dans la dimension physique p de l'environnement sont définis par Act_p . Ainsi, un agent peut agir dans l'environnement physique, par l'intermédiaire de son corps, en utilisant le modèle d'interaction influence-réaction (Michel, 2007). Une influence peut décrire un mouvement (*MotionInfluence*) ou l'exécution d'une action sur un objet (*ActionInfluence*).

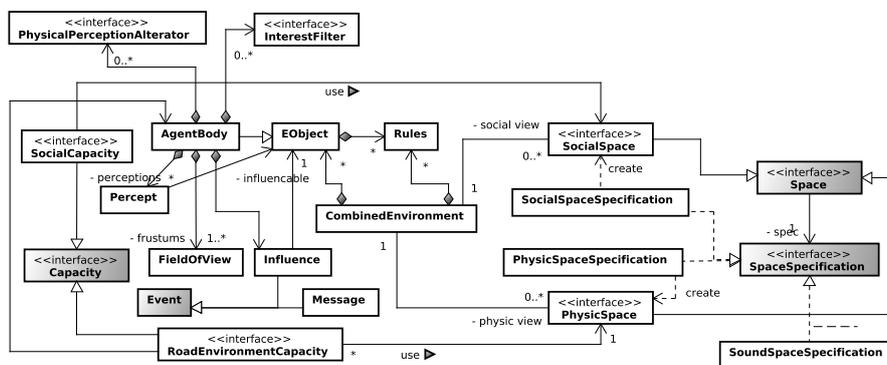


Figure 2. Concepts relatifs au méta-modèle d'environnement (en blanc) et concepts hérités du méta-modèle SARL (en gris)

Nous considérerons que les influences sont des événements ayant un impact sur un objet de l'environnement (voir figure 2). Ces deux types d'influences sont décrits par des événements reçus par l'environnement (script 2). La fonction Act_p est mise en œuvre par ce script et par l'implantation de l'AgentBody.

```

event Influence {
2   var object : EObject
}
4 event MotionInfluence extends Influence {
   var frontAcceleration : double
6   var sideAcceleration : double
   var path : Segment[]
8 }
event ActionInfluence extends Influence {
10  var actionName : String
   var actionParameters : Object[]
12 }
event Perception {
14  var objects : Percept[]
}

```

Script 2 – Événements liés à l’environnement physique (en SARL)

Le mécanisme de perception d’un agent dans l’environnement physique est défini par $Percept_p$. Il est mis en œuvre par le corps de l’agent (*AgentBody*) qui contient entre autres une description géométrique du champ de perception *FieldOfView*. L’agent peut affiner le calcul de ses perceptions en fournissant une description permettant d’altérer les propriétés physiques de ses capteurs (*PhysicalPerceptionAlterator*), ou un filtre décrivant les objets qui l’intéressent (*InterestFilter*). Le premier ne tiendra compte que des altérations liées à la physique des capteurs, le second que de la sémantique des objets perçus. Une fois que l’ensemble des objets se trouvant dans le champ de perception de l’agent est déterminé, il est fourni à ce dernier par un événement de type *Perception*.

Pour chaque capacité décrite, une compétence concrète doit être définie. Cette compétence permettra de créer l’espace d’interaction (script 3), et d’invoquer les fonctions associées à cet espace. Un espace dédié à l’environnement physique doit permettre de créer un corps d’agent, de déposer ce corps dans l’environnement, et de notifier l’environnement de l’existence d’influences émises par un corps.

```

public interface PhysicSpace extends Space {
2   public PhysicBodyFactory getBodyFactory();
   public void putInEnvironment(AgentBody body, Agent perceptionListener);
4   public void influence(AgentBody body, influences : Influence[]);
   public void destroyBody(AgentBody body);
6 }
public class PhysicSpaceImpl extends EventSpaceImpl implements PhysicSpace {
8   private CombinedEnvironment env;
   public void influence(AgentBody body, influences : Influence[]) {
10     for(Influence i : influences) emit(i, new Scope(env));
   }
12   ...
}

```

Script 3 – Spécification de l’espace physique (en Java)

L’implantation de l’espace lié à l’environnement physique (présenté dans le script 3) est dépendante du modèle concret de l’environnement, par exemple *Jaak*⁵. Nous proposons que chaque influence émise par un agent soit envoyée uniquement à

5. Code source : <https://github.com/gallandarakhneorg/jaak>

l'entité environnement, qui la traitera dans son comportement propre. Pour cela, nous utilisons le mécanisme d'envoi de messages et d'événements proposé par le langage SARL (un message est considéré comme un événement du point de vue de son récepteur). Nous restreignons l'ensemble des récepteurs de l'événement par la création d'une portée (Scope).

L'environnement physique considéré dans cet article est un environnement simulé afin de faciliter les expérimentations. Toutefois, il est possible de connecter les agents au monde réel en leur fournissant un outil logiciel représentant leurs corps dans cet environnement physique. Ce corps peut alors proposer des capteurs et des effecteurs dans un univers virtuel ou dans le monde réel. Par exemple, l'agent capable de conduire un véhicule autonome proposé par Gechter *et al.* (2012) a été utilisé à la fois dans un simulateur et dans un véhicule réel : seules les implantations des capteurs et des effecteurs associés au corps de l'agent (AgentBody dans la figure 2) ont été modifiées.

5.2. Dimension sociale de l'environnement

Une dimension sociale, $c \in D$, permet aux véhicules de communiquer dans un contexte social. SARL fournit les outils pour définir ce type particulier d'espace, appelé SocialSpace, qui fournit un support aux interactions directes entre les agents. Dans ce type d'espace, les agents communiquent en échangeant des messages à l'aide de la capacité décrite dans le script 4. Ainsi, l'implantation de la fonction Act_c , définie dans la section 4, correspond à la fonction `emit` qui est fournie par cette capacité.

Considérons le modèle d'interaction proposé par défaut par le langage SARL (interface EventSpace et son implantation EventSpaceImpl). Il utilise les événements comme support de l'interaction : un message devient un événement pour le ou les agents devant le recevoir. Dès lors, l'environnement social peut être considéré comme une spécialisation d'un espace d'échange d'événements, et assurant le lien vers l'entité environnement. L'implantation de la fonction $Percept_c$ est ainsi basée sur l'opération `on Event` fournie par le langage SARL.

```

event Message {
2   var destination : Scope<Address>
}
4 capacity SocialCommunicationCapacity {
   def emit(e : Message, scope : Scope<Address> = null)
6 }

```

Script 4 – Capacités d'interaction entre les agents (en SARL)

```

public interface SocialSpace extends Space {
2   public void emit(Message e, Scope<?> scope);
4   public Address register(Agent agent);
6   public void unregister(Address agentAddress);
8 }
10 public class SocialSpaceImpl extends EventSpaceImpl implements SocialSpace {
12   private CombinedEnvironment env;

```

```

14 public void emit(Message e, Scope<?> scope) {
    e.destination = scope;
16     super.emit(e, new Scope(env));
    }
18 }

```

Script 5 – Spécification de l'espace social (en Java)

Le script 5 fournit la définition d'un espace social (`SocialSpace`) et d'une implantation possible (`SocialSpaceImpl`) basée sur les outils de SARL. La section 5.3 précise le traitement de ces événements par l'environnement.

5.3. Intégration des dimensions de l'environnement

Dans le modèle présenté dans la section 4.1, l'environnement e est constitué par un ensemble de dimensions, et un ensemble Λ_e de règles régissant les relations entre ces dimensions. Les dimensions présentées dans la section précédente sont supportées et combinées au sein d'une même entité logicielle (`CombinedEnvironment`). La figure 2 illustre les relations entre cette entité et les vues associées aux dimensions considérées dans cet article.

Le comportement général de l'environnement est une combinaison des comportements associés à ses dimensions : (i) calculer les réactions correspondant à la modification de l'état de l'environnement physique à partir d'influences émises par les agents ou par l'environnement lui-même ; (ii) calculer les perceptions dans une dimension physique ; (iii) propager ou router les messages dans une dimension sociale. Ces différents points ont été implantés par quatre fonctions invoquées à chaque pas de temps dans le simulateur (événement `SimulationStep` dans le script 6).

```

behavior CombinedEnvironment {
2   var rules : List<Pair<(Behavior, Event, Object) => boolean, (Behavior,
    Event, Object) => boolean>>
    on Initialize {
4     every(500) [ wake(new SimulationStep) ]
    }
6   on SimulationStep {
    computeEndogenousInfluences
8     computePhysicReactionsFromInfluences
    computePhysicPerceptions
10    deliverMessages
    }
12  on Influence {
    if (multi(occurrence, occurrence.object)) {
14    saveInfluence(occurrence)
    }
16  }
    on Message {
18    for(participant : this.socialSpace.participants) {
        if (occurrence.scope.matches(participant)) {
20          if (multi(occurrence, participant)) {
              saveMessage(occurrence)
22          }
        }
24    }
    }
26  def multi(e : Event, o : Object) : boolean {

```

```

28     for(pair : rules) {
        if (pair.first(this, e, o)
            && !pair.second(this, e, o)) {
30         return false
        }
32     }
        return true
34     }
}

```

Script 6 – Comportement de l’environnement (en SARL)

Afin de permettre les trois cas d’utilisation décrits dans la section 2.1, un ensemble de règles Λ_e est stocké dans l’environnement e . L’expression concrète du prédicat $\langle d', \phi'_d \rangle$ d’une règle λ_e peut dépendre de l’état de l’environnement dans une ou plusieurs de ses dimensions, ou des états d’objets particuliers (objets physiques ou messages). Nous proposons d’utiliser une clôture syntaxique⁶ pour permettre la définition de λ_e . Dans SARL, la notation pour déclarer une clôture est $(P) \Rightarrow R$, où P est la liste des types des paramètres formels, et R est le type des valeurs retournées. Dans le script 6, cet ensemble de règles est représenté par l’attribut `rules`. Cet attribut est initialisé avec les règles propres au cas d’application, comme cela est illustré dans la section suivante.

La mise en œuvre de l’ensemble des règles au sein de l’environnement e est définie par la fonction $Multi(\sigma, I)$, également définie dans le script 6. Cette dernière utilise une liste de règles de la plus prioritaire à la moins prioritaire. Dans le script 6, nous définissons l’ensemble des règles (variable `rules`) comme une liste de paires (p, f) . p est un prédicat prenant la forme d’une clôture avec comme paramètres un `Behavior`⁷, un événement dans une des dimensions de l’environnement, un objet concerné par l’événement, et retournant une valeur booléenne indiquant si le prédicat (la règle) est activable ou non. f est une clôture prenant les mêmes paramètres que p , et réalisant la tâche associée à la règle et retournant une valeur booléenne indiquant si la propagation de l’influence ou du message dans sa dimension d’origine est autorisée.

Cet ensemble de règles est utilisé lorsqu’une influence est émise dans une dimension physique, ou un message dans une dimension sociale. Dans le script 6, ces deux cas sont distingués au sein de deux fonctions (`on Influence` et `on Message`) afin de fournir aux prédicats l’objet sur lequel ils pourront s’appliquer : l’objet physique “influencé” ou le message. Ces deux fonctions appliquent les règles connues par l’environnement en utilisant la fonction `multi`. Si aucune règle n’annule l’interaction, alors l’environnement la mémorise pour une utilisation ultérieure dans son comportement global (`SimulationStep`).

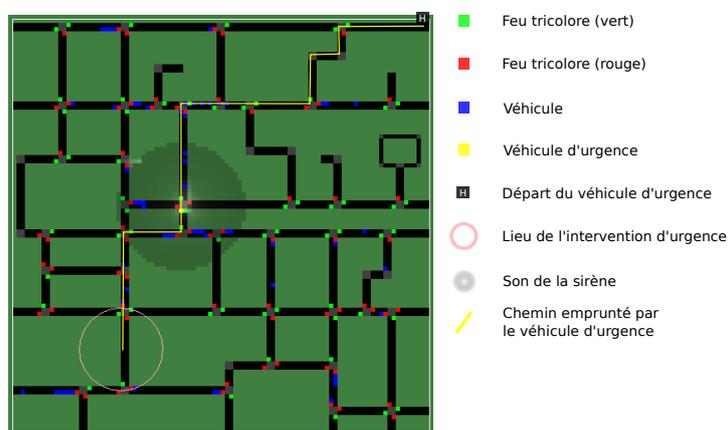
6. Une clôture est une fonction qui capture des références à des variables libres dans l’environnement lexical de la fonction dans laquelle elle est définie. Cette fonctionnalité permet de définir simplement des fonctions à l’intérieur d’autres fonctions. Ces clôtures peuvent alors être passées en paramètres de fonctions, par exemple.

7. `Behavior` : abstraction pour tout algorithme exhibant un comportement dynamique, y compris les agents.

Le principe d'intégration des dimensions présenté dans cette section n'est pas restreint à une dimension physique et une dimension sociale. Il est possible d'intégrer plusieurs dimensions physiques et plusieurs dimensions sociales en modifiant légèrement le code présenté dans le script 6. Pour supporter une dimension particulière, il faut ajouter les fonctions (blocs **on**) définissant les réactions aux événements participant à la description de l'interaction entre un agent et la nouvelle dimension.

6. Modélisation du cas d'application de trafic routier

L'objectif de cette section est de déployer le modèle présenté dans la section précédente sur un cas d'application de trafic, c'est-à-dire adapter le modèle général à la simulation de trafic routier⁸.



L'objectif principal de ces expérimentations est d'illustrer l'usage de différentes dimensions de l'environnement dans le modèle de simulation. Ainsi, les trois cas d'expérimentations listés dans le tableau 1 sont considérés et exécutés en parallèle. Chaque colonne représente un type d'entité ou d'information pouvant être perçue par les conducteurs simulés. Les deux premières colonnes décrivent les contraintes de déplacement constituées par les routes, d'une part, et d'autre part les feux tricolores. La troisième colonne représente l'information sonore dans la dimension physique de l'environnement, dans laquelle le son de la sirène se propage. La dernière colonne correspond à la position sociale des véhicules qui sont membres de la communauté.

Le premier cas d'expérimentation se restreint à une perception monodimensionnelle de l'environnement physique. Il s'agit du cas classiquement considéré dans les simulations de trafic. Étant constitué que d'une seule dimension physique, il servira de base de référence dans notre discussion. Le second cas d'usage permet aux

8. Code source : <https://github.com/gallandarakhneorg/jaak/tree/trafficsim/io.sarl.jaak.demos>.

Tableau 1. Types d'éléments perçus par les conducteurs

N° expéri- mentation	Routes	Feux trico- lores	Sirène	Position sociale
1	✓	✓		
2	✓	✓	✓	
3	✓	✓	✓	✓

agents de percevoir la sirène du véhicule d'urgence en plus des informations limitant leurs déplacements. Enfin, le troisième cas d'application vient illustrer l'impact d'une dimension sociale sur les résultats de la simulation, en reproduisant la propagation d'informations sur le réseau de communication intervéhicules.

Le niveau d'information augmente de l'expérimentation une à l'expérimentation trois. Nous pensons que cela permettra aux agents conducteurs d'ajuster plus précisément leur comportement. Les deux derniers cas d'expérimentation peuvent être mis en œuvre à l'aide du modèle multidimensionnel proposé dans cet article. Les expérimentations viendront étayer cette hypothèse. Pour cela, nous utilisons le premier cas d'expérimentation comme cas de référence (cas utilisant un modèle monodimensionnel). Les deux autres cas seront comparés à celui-ci afin d'illustrer l'impact d'un modèle multidimensionnel pour la simulation de trafic. Il faut toutefois remarquer que nous ne voulons pas réaliser une étude exhaustive et détaillée d'un système de trafic. Nous utilisons des modèles reconnus de comportements de conducteurs ("Intelligent Driver Model", etc.) et de modèles d'environnement (grille inspirée des modèles NetLogo). Ainsi, nous pouvons nous focaliser sur la modélisation des relations entre les dimensions de l'environnement considérées dans les cas d'application.

Ces expérimentations illustrent également la facilité de conception des différents dimensions afin de valider différentes hypothèses du modèle de simulation de système de transport. En effet, l'activation d'une dimension particulière nécessite peu de lignes de code : instanciation de la dimension, et les règles régissant ses relations avec les autres dimensions.

6.1. Composants de la simulation

Dans notre cadre d'application, nous avons choisi de modéliser l'environnement par une grille régulière afin de faciliter la mise en œuvre sans nuire à la preuve de concept. Pour cela, nous utilisons la bibliothèque Jaak⁹ qui étend le langage SARL, et la plate-forme JANUS, en fournissant les outils nécessaires à la modélisation de l'environnement et à la mise en œuvre d'un simulateur orienté-agent.

Dans notre cadre d'application, nous considérerons les trois dimensions de l'environnement suivantes :

9. <https://github.com/gallandarakhneorg/jaak>

1. La **dimension physique de déplacement** dans laquelle les agents peuvent se déplacer. Elle est constituée d'un ensemble de routes (carrés noirs ou gris sur la figure 3). Cette dimension contient également les feux tricolores (carrés rouges ou verts sur la figure 3).

2. La **dimension sonore** permet de déposer une source sonore dans l'environnement et de calculer sa propagation. Dans ce cadre d'application, nous utilisons une propagation circulaire simplifiée (cercle sombre sur la figure 3).

3. La **dimension sociale** représentant l'espace d'interaction du réseau social auquel participent les agents. Les différents véhicules possèdent un statut dans cette communauté. Ce statut est utilisé par les feux tricolores pour donner la priorité aux voies sur lesquelles se trouve le véhicule d'urgence. De plus, le véhicule d'urgence informe les autres membres de la communauté de son départ en intervention en diffusant son itinéraire.

Les agents participant à la simulation sont des conducteurs de véhicules (carrés bleus sur la figure 3). Chaque véhicule est ajouté à la simulation à un point d'entrée du réseau routier choisi aléatoirement. Chaque conducteur a pour objectif d'atteindre le plus rapidement possible le point de sortie du réseau à l'opposé de son point d'entrée.

```

agent StandardDriver {
2   uses DrivingCapacity
   var path : Path
4   on Initialize {
       setSkill(DrivingCapacity, IDM_Dstart_DrivingSkill)
6   }
   on Perception {
8     var stopVehicleInStandardCondition = isVehicleStop(occurrence)
       var siren = occurrence.body.getFirstPerceptionAtCurrentPosition(Siren)
10    var stopVehicleForEmergencyVehicle = isStopWhenEmergencyVehicle(siren)
       if (!stopVehicleForEmergencyVehicle && !stopVehicleInStandardCondition){
12      var motion : Vector2i = null
         path = updatePathWithDstart(path, occurrence)
14      if (!path.empty) {
           motion = followPathWithIDM(path, occurrence)
16      }
         if (motion != null && motion.lengthSquared > 0) {
18           move(motion, true)
             this.previousOrientation = direction
20         }
       }
22   }
}

```

Script 7 – Comportement d'un conducteur (en SARL)

Le comportement des conducteurs, décrit par la script 7, est une adaptation des algorithmes "Intelligent Driver Model" (IDM) pour déterminer l'accélération du véhicule, et D* pour planifier dynamiquement le plus court chemin à suivre. Dans un premier temps, un agent conducteur détermine l'objet dans la dimension de déplacement pouvant provoquer un arrêt du véhicule : feu rouge, véhicule prédécesseur, véhicule prioritaire par la règle de priorité à droite. Cet obstacle est utilisé comme paramètre de l'algorithme IDM afin de déterminer si l'accélération du véhicule doit être nulle (ligne 8). Lorsqu'un agent conducteur perçoit la sirène du véhicule d'urgence dans la dimension sonore, il peut décider de s'arrêter (ligne 10). Cette fonction prend

également en compte le chemin du véhicule d'urgence publié dans la dimension sociale pour adapter la trajectoire suivie par le véhicule afin d'éviter l'ambulance. Dans notre cas d'application, ce recalcul est réalisé en augmentant les coûts associés aux cellules traversées par l'ambulance. Lorsque l'agent conducteur décide de ne pas s'arrêter, il suit le chemin qu'il a précédemment calculé (lignes 14 à 16).

Le véhicule d'urgence (représenté par un carré jaune sur la figure 3) apparaît dans la simulation lorsqu'une situation nécessitant son intervention apparaît. Afin de permettre une montée en charge du réseau routier, le véhicule d'urgence est créé à un temps de simulation fixé (200). Le comportement de l'agent conduisant le véhicule d'urgence est de rallier par le plus court chemin le lieu de l'intervention. Lors de son départ, le véhicule d'urgence émet une notification dans la dimension sociale afin d'indiquer son changement de statut. Durant son déplacement, le véhicule d'urgence émet un son de sirène dans la dimension sonore de l'environnement. Ce son peut être perçu par les autres agents afin d'adapter leurs comportements de conduite, comme cela est expliqué dans le paragraphe précédent.

6.1.1. Règles entre les dimensions de l'environnement

Dans cette section, les trois cas de figure associés à notre exemple illustratif (section 3.2) sont introduits dans la base de règles L_e de notre modèle. Le script 8 propose la définition de ces trois règles en utilisant la syntaxe du langage SARL. La notation [paramètres | instructions] permet de définir les clôtures associées à chaque règle, décrite avec la syntaxe $p \Rightarrow f$; où p est le prédicat permettant d'activer la règle; et f décrit la modification d'état impliqué par la règle.

```

2      /* Règle pour le cas 1 */
   environment.rules +=
         [ env,e,o | e instanceof PriorityRequest
4           && o instanceof TrafficLight ]
         => [ env,e,o | env.socialSpace.participant(e.source.ID)
6           instanceof PriorityVehicle ]

8      /* Règle pour le cas 2 */
   environment.rules +=
         [ env,e,o | e instanceof PriorityRequest ]
10         => [ env,e,o | var sp = env.physicSpace
         for(road : sp.getRoadsBetween(sp.getEObject(o.ID).position,
12           sp.getEObject(e.source.ID))) {
         road.priorityIndex = road.priorityIndex + e.priority
14         }
         return true ]

16     /* Règle pour le cas 3 */
   environment.rules +=
18         [ env,e,o | e instanceof PhysicCollision ]
20         => [ env,e,o | env.emit(new Alert(e.position))
         return false ]))

```

Script 8 – Règles inter-dimensions relatives au cas d'application de trafic routier (en SARL)

La première règle (ligne 2 du script 8) indique qu'un feu tricolore ne prend en compte que les demandes issues des véhicules prioritaires. La seconde règle (ligne 8) permet, à partir d'une demande de priorisation dans la dimension sociale, de modifier le comportement des membres de la communauté, et d'augmenter l'indice de priorité

de l'axe routier concerné. La dernière règle (ligne 17) implantée dans notre cas d'application permet de générer un message d'alerte dans la dimension sociale lorsqu'une collision entre véhicules est détectée dans la dimension de déplacement.

6.2. Résultats expérimentaux

Durant les expérimentations, les véhicules (2 500 entités) peuvent appartenir à une, ou plusieurs, des trois catégories suivantes. La première catégorie *A* est constituée par les véhicules percevant la sirène du véhicule d'urgence. La seconde catégorie *B* est composée par les véhicules pour lesquels les trajets initialement calculés sont modifiés à la suite de la détection du véhicule d'urgence dans la dimension sociale. Un véhicule peut appartenir simultanément à ces deux premières catégories. La troisième catégorie est constituée de tous les véhicules n'appartenant pas à l'une des deux catégories précédentes. Le tableau 2 illustre les durées et les longueurs des trajets des véhicules simulés. Les trois premières lignes de ce tableau mettent en lumière une diminution significative du nombre de véhicules percevant la sirène du véhicule d'urgence (catégorie *A*). Dans le cas d'expérimentation n°3, le nombre de véhicules percevant la sirène est en forte diminution de ~ 880 à ~ 630 . Ce résultat s'explique par le recalcul des trajets lorsque les agents perçoivent le départ du véhicule d'urgence dans la dimension sociale de l'environnement. Des entités ayant recalculé leurs chemins, et ayant également perçu la sirène apparaissent dans les résultats de l'expérimentation (~ 500 entités). Ce phénomène peut s'expliquer par les positions des véhicules, au moment de la perception de la sirène, qui ne permettent pas de trouver un autre chemin. Par exemple, lorsqu'un véhicule est engagé sur une rue le menant à un croisement traversé par le véhicule d'urgence, il lui est alors impossible d'éviter ce dernier, et cela même en recalculant un nouveau trajet. Les longueurs moyennes des trajets restent sensiblement identiques dans les trois cas d'expérimentation, et cela même dans le cas où les trajets sont recalculés. En effet, même si le nouveau calcul des trajets augmente en moyenne la longueur des trajets de $\sim 8\%$, une plus grande proportion d'agents dans ces cas d'expérimentation ne modifie pas leurs trajets, soit parce qu'ils peuvent prédire qu'ils ne rencontreront pas le véhicule d'urgence, soit parce qu'il est trop tard pour obtenir un trajet ne croissant pas le véhicule d'urgence.

Le tableau 3 illustre les mesures des temps de trajet du véhicule d'urgence pour les trois cas d'expérimentation. Dans le premier cas, les autres véhicules ne modifient pas leurs comportements en présence du véhicule d'urgence. Ainsi, ce dernier doit attendre pour progresser vers son objectif que les véhicules qui le précèdent avancent. Cette progression est facilitée par les changements d'état des feux tricolores qui donnent la priorité à la voie sur laquelle se déplace le véhicule d'urgence. L'amélioration mesurée par l'expérimentation n°2 s'explique par le ralentissement des autres véhicules lorsqu'ils entendent la sirène. Enfin, l'expérimentation n°3 montre que le véhicule d'urgence atteint pratiquement son temps de trajet optimal. En effet, il rencontre peu de véhicules sur son trajet. Soit, ces derniers utilisent un autre trajet. Soit, ils sont bloqués par les feux tricolores lorsque le véhicule d'urgence s'approche de ces derniers.

Tableau 2. Durées et longueurs des trajets des véhicules

	Exp. 1	Exp. 2	Exp. 3
Nombre de véhicules perturbés par le véhicule prioritaire (A)	873	933	636
Nombre de véhicules avec tentative de recalcul chemin (B)	-	-	927
Nombre de véhicules ayant obtenus un autre chemin ($C \subseteq B$)	-	-	360
Nombre de véhicules ayant gardés le même chemin ($B \setminus C$)	-	-	567
Nombre de véhicules ayant obtenus un autre chemin mais perturbés par le véhicule prioritaire ($A \cap C$)	-	-	236
Durées des trajets de A (en unités temporelles)			
Moyenne arithmétique	401	429	493
Écart-type	101	101	184
Durées des trajets de $B \setminus A$ (en unités temporelles)			
Moyenne arithmétique	-	-	372
Écart-type	-	-	176
Durées des trajets de C (en unités temporelles))			
Moyenne arithmétique	-	-	593
Écart-type	-	-	211
Longueurs des trajets de A (en nb. cellules)			
Moyenne arithmétique	161	162	182
Écart-type	29	26	48
Longueurs des trajets de $B \setminus A$ (en nb. cellules)			
Moyenne arithmétique	-	-	143
Écart-type	-	-	49
Longueurs des trajets de C (en nb. cellules)			
Moyenne arithmétique	-	-	205
Écart-type	-	-	55
Augmentations des longueurs des trajets de C (en nb. cellules)			
Moyenne arithmétique	-	-	+52
Écart-type	-	-	47

Tableau 3. Temps moyens des trajets du véhicule d'urgence (en unités temporelles)

	Exp. 1	Exp. 2	Exp. 3
Moyenne arithmétique	289,6	313,9	157,8
Écart-type	77,9	70,7	8,2

Concernant le premier objectif des expérimentations, les résultats montrent qu'il est possible d'influencer le comportement des agents par l'utilisation de différentes dimensions dans la conception de l'environnement. Ainsi, le niveau d'information augmentant de l'expérimentation une à trois, il permet aux agents conducteurs d'ajuster plus précisément leur comportement. L'évolution des vitesses des conducteurs et du véhicule prioritaire illustre cette évolution du comportement. Une étude approfondie de cet impact reste une perspective des travaux proposés dans ce document.

Concernant l'objectif secondaire, le langage SARL est adapté à l'implantation d'un modèle de simulation de trafic routier intégrant un modèle multidimensionnel de l'environnement. La base de règles L_e , ainsi que les concepts de *Capacity* et *Skill*, permettent de faciliter l'intégration de nouvelles dimensions. En effet, nous pensons que ces outils permettent d'adopter une approche modulaire de modélisation et de programmation, et respectant les principes de "Don't Repeat Yourself". La quantification de cette facilité d'implantation reste une perspective de ce travail.

7. Conclusion et perspectives

L'environnement est communément considéré comme l'une des parties essentielles d'une simulation multiagent. Selon les systèmes, il est qualifié de physique ou communicationnel selon que les agents interagissent par des actions situées ou des échanges de messages, mais également de social si un modèle social régit les interactions. Chacune de ces classes est déclinable en dimensions dotées chacune de son propre modèle. Pour un système complexe dans lequel différentes dimensions doivent être combinées, il n'existe que des solutions ad hoc pour obtenir un modèle d'environnement global.

Nous proposons un modèle permettant la combinaison des différentes dimensions de l'environnement. Afin de modéliser les relations entre ces dimensions, nous proposons de définir des règles. Elles permettent de modifier l'état de l'environnement lié à une dimension en fonction des éléments liés à l'autre dimension. Nous pensons que l'usage de règles fournit un outil assez général et adaptable pour différentes classes d'applications. Nous proposons de définir les dimensions à l'aide d'espaces d'interaction dans le langage SARL. Ces espaces peuvent être considérés comme des vues sur un environnement combinant les différentes dimensions. Nous présentons une application du modèle à la simulation d'un système de transport routier. Quelques définitions concrètes de règles entre dimensions de l'environnement sont également proposées.

Une première perspective de nos travaux concerne la mise en correspondance ou l'adaptation de notre modèle pour permettre d'autres approches de modélisation de l'environnement : artefacts (Ricci *et al.*, 2005), "smart objects" (Thalmann, Musse, 2007), holarchies (Rodriguez *et al.*, 2011), etc.

L'enrichissement des capacités comportementales des agents peut être réalisé par un modèle d'environnement informé (Farenc *et al.*, 1999). Un environnement informé fournit une information sémantique détaillée aux agents à l'aide de marqueurs sémant-

tiques ou d'ontologies (Thalmann, Musse, 2007; Béhé *et al.*, 2014). Les premiers travaux proposant un modèle d'univers virtuel ont été étendus dans le cadre de la simulation multiagent. L'utilisation d'ontologies (et des outils associés) peut être considérée pour faciliter l'expression des règles d'interaction entre les différentes dimensions de l'environnement.

La nature intrinsèquement hiérarchique de l'environnement n'est pas traitée dans cet article. Certains auteurs ont déjà considéré la décomposition structurelle de l'environnement physique (Musse, Thalmann, 2001), et sa relation avec son comportement dynamique, en utilisant une vision holonique de l'environnement (Galland, Gaud, 2015). Cette nature hiérarchique a été également modélisée dans la dimension sociale par des holarchies, des groupes agentifiés ou des agents récursifs. Notre modèle peut être adapté pour correspondre à un holon (agent) décomposé en sous-holons.

Enfin, en relation avec la seconde perspective de nos expérimentations, nous voulons faire évoluer et évaluer l'apport d'éléments syntaxiques du langage SARL pour faciliter la définition et la description des environnements et de leurs règles, à l'instar de travaux tels que NetLogo (Wilensky, Rand, 2015) ou GAML (Grignard *et al.*, 2013).

Bibliographie

- Béhé F., Galland S., Gaud N., Nicolle C., Koukam A. (2014, janvier). An ontology-based metamodel for multiagent-based simulations. *International Journal on Simulation Modelling, Practice, and Theory*, vol. 40, p. 64-85. Consulté sur <http://authors.elsevier.com/sd/article/S1569190X13001342>
- Bhourri N., Balbo F., Pinson S. (2012). An agent-based computational approach for urban traffic regulation. *Progress in AI*, vol. 1, n° 2, p. 139-147.
- Claes R., Holvoet T., Weyns D. (2011). A decentralized approach for anticipatory vehicle routing using delegate multiagent systems. *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, n° 2, p. 364-373.
- Farenc N., Boulic R., Thalmann D. (1999). An informed environment dedicated to the simulation of virtual humans in urban context. In *Proceedings of eurographics 99*, p. 309-318.
- Galland S., Gaud N. (2015, octobre). Organizational and holonic modelling of a simulated and synthetic spatial environment. *E4MAS 2014 - 10 years later, LNAI*, vol. 9068, n° 1, p. 1-23. Consulté sur <http://www.springer.com/us/book/9783319238494>
- Gechter F., Contet J.-M., Lamotte O., Galland S., Koukam A. (2012, mai). Virtual intelligent vehicle urban simulator: Application to vehicle platoon evaluation. *Simulation Modelling Practice and Theory (SIMPAT)*, vol. 24, p. 103-114.
- Gouaïch A., Michel F. (2005, may). Towards a unified view of the environment(s) within multi-agent systems. *Informatica*, vol. 29, n° 4, p. 423-432.
- Grignard A., Taillandier P., Gaudou B., Vo D. A., Huynh N. Q., Drogoul A. (2013). GAMA 1.6: Advancing the art of complex agent-based modeling and simulation. In *16th international conference on principles and practices in multi-agent systems (prima)*, vol. 8291, p. 242-258. Dunedin, New Zealand.

- Hall E. T. (1990). *The hidden dimension* (Reissue éd.). New York, Anchor.
- Mathieu P., Picault S., Secq Y. (2014). Les environnements: en avoir ou pas? formalisation du concept et patterns d'implémentation. In *Jfsma*, p. 55–64.
- Michel F. (2007, mai). The IRM4S model: the influence/reaction principle for multiagent based simulation. In *Sixth international joint conference on autonomous agents and multiagent systems (aamas07)*. ACM.
- Musse S., Thalmann D. (2001). A hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, n° 2, p. 152-164.
- Odell J., Parunak H., Fleisher M., Brueckner S. (2009). Modeling Agents and their Environment. In *Agent-oriented software engineering III*, vol. 2585. N.Y. (USA), Springer-Verlag.
- Peña J., Levy R., Hinchey M., Ruiz-Cortés A. (2012). Dealing with complexity in agent-oriented software engineering: The importance of interactions. In *Conquering complexity*, p. 191-214. Springer London.
- Piunti M., Ricci A., Boissier O., Hübner J. (2009). Embodying organisations in multi-agent work environments. In *Ieee/wic/acm Int. conf. on web intelligence and intelligent agent technology (wi-iat 2009)*. Milan, Italy..
- Platon E., Sabouret N., Honiden S. (2007). Tag interactions in multiagent systems: Environment support. In *Environment for multi-agent systems (e4mas)*, vol. 4389, p. 106-123. Springer.
- Ricci A., Piunti M., Viroli M. (2011). Environment programming in multi-agent systems: an artifact-based perspective. *Autonomous Agents and Multi-Agent Systems*, vol. 23, n° 2, p. 158–192.
- Ricci A., Viroli M., Omicini A. (2005, July). Programming MAS with artifacts. In *International workshop on programming multi-agent systems*. Springer Verlag.
- Rodriguez S., Gaud N., Galland S. (2014, Aug). SARL: A general-purpose agent-oriented programming language. In *Web intelligence (wi) and intelligent agent technologies (iat), 2014 ieee/wic/acm international joint conferences on*, vol. 3, p. 103-110.
- Rodriguez S., Hilaire V., Gaud N., Galland S., Koukam A. (2011, mars). Holonic multi-agent systems. In *Self-organizing software: From natural to artificial adaptation*, first éd., p. 238-263. Springer.
- Saunier J., Balbo F., Pinson S. (2014). A formal model of communication and context awareness in multiagent systems. *Journal of Logic, Language and Information*, p. 1-29. Consulté sur <http://dx.doi.org/10.1007/s10849-014-9198-8>
- Saunier J., Carrascosa C., Galland S., Kanmeugne P. s. (2015, octobre). Agent bodies: An interface between agent and environment. *E4MAS 2014 - 10 years later, LNAI*, vol. 9068, n° 1, p. 1-16. Consulté sur <http://www.springer.com/us/book/9783319238494>
- Thalmann D., Musse S. R. (2007). *Crowd simulation*. Springer.
- Weyns D., Omicini A., Odell J. (2007, février). Environment as a first-class abstraction in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, vol. 14, n° 1, p. 5-30. (Special Issue on Environments for Multi-agent Systems)

Weyns D., Steegmans E., Holvoet T. (2004). Protocol based communication for situated multi-agent systems. In *Proceedings of the third international joint conference on autonomous agents and multiagent systems-volume 1*, p. 118–125.

Wilensky U., Rand B. (2015). *Introduction to agent-based modeling: Modeling natural, social and engineered complex systems with NetLogo*. The MIT Press.

Zargayouna M., Balbo F. (2013). Langage de coordination multi-agent sécurisé. *Revue d'Intelligence Artificielle*, vol. 27, n° 3, p. 271-298.