
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.rodriquez@gitia.org*

RÉSUMÉ. L'environnement, en tant qu'espace partagé entre agents, est un élément essentiel des systèmes multiagents (SMA). Selon les systèmes, il réponds à différents points de vue. 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. Chacun de ces points de vue 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 une réutilisabilité et modularité limitée de l'utilisation des modèles d'environnement. Une alternative 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, i.e. contraintes par des règles du SMA, imposent 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

1. Introduction

L'environnement est une abstraction de premier ordre pour la conception de systèmes multiagents (SMA) (Weyns *et al.*, 2007). Il a notamment été montré qu'il est possible de prendre avantage de l'environnement afin 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 permet un partage des interactions. À chaque modèle d'environnement correspond une dynamique endogène et un modèle d'interaction régit 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.

Ainsi, l'environnement est un système complexe dans lequel les agents vivent et interagissent. Différents points de vue peuvent être adoptés sur l'environnement (Weyns *et al.*, 2007; Odell *et al.*, 2009) : environnement physique, communicationnel, social. Les environnements physiques se réfèrent à la classe de systèmes dans laquelle les agents, ainsi que les objets, disposent d'une position explicite et produisent des actions elles aussi localisées. Les environnements communicationnel supportent les interactions par échanges de messages. Les environnements sociaux constituent une spécialisation des environnements précédents dans lesquels les structures sociales régissant la communauté des agents sont spécifiées et gérées. Chacun de ces points de vue peut être déclinés en dimensions selon la dynamique et les règles choisies. Ainsi, un environnement physique peut être décliné en une dimension auditive et/ou visuelle selon qu'il supporte un modèle de propagation du son et/ou de la lumière. Il est restrictif de considérer qu'un agent puisse interagir selon une seule de ces dimensions. Par exemple, modéliser un système aussi complexe qu'une population urbaine nécessite de prendre en compte la ville et sa structure, et également les réseaux sociaux pouvant influencer les comportements des individus. Il faut donc considérer l'environnement comme le résultat de la combinaison de ses dimensions.

Une dimension définit les structures et les règles internes qui lui sont nécessaires pour répondre à ses objectifs. Dans la plupart des travaux, l'environnement est soit mono-dimensionnel, soit le résultat ad hoc de l'intégration ou juxtaposition de ses différentes dimensions. Dans ce dernier cas, c'est l'agent qui, par combinaison des informations véhiculées dans les différentes dimensions où il interagit, supporte leur connexion. Ainsi, un agent peut interagir dans une dimension (par exemple physique) suite à une interaction dans une autre (réception d'un message). Seulement, si dans chaque dimension des règles régissent l'interaction, alors l'agent pourra agir que selon les interactions le concernant dans au moins une des dimensions. Dans l'exemple précédent, si le message ne lui est pas destiné il ne pourra pas agir même si l'émetteur est suffisamment proche pour que la réception soit possible selon les règles de l'environnement physique.

En introduisant le concept de **dimension**, nous adoptons une vision systémique de l'environnement dans laquelle chaque dimension représente un sous-système en interaction avec les autres. Les interactions entre les dimensions sont alors explicitées par des règles dont la gestion est assurée par l'environnement. Ces règles inter-dimensions rendent possible la gestion de ses conséquences pour une même interaction dans les différentes dimensions, et de modifier ainsi le comportement global du SMA. Par exemple, l'envoi d'un message entre deux agents aura des conséquences dans la dimension sociale suite à la prise en compte de son contenu par le récepteur mais aussi dans la dimension physique par une augmentation du fond sonore.

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 inter-dimension par l'ajout ou le retrait de ses 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, notre approche favorise la modularité et réutilisabilité des différentes dimensions de l'environnement.

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. 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.

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

2. État de l'art

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

De nombreux travaux dont ceux résultats du groupe de travail E4MAS² (“*Environment for multiagent systems*”) ont montré que la modélisation d'un système 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. La notion de corps (Platon **et al.**, 2007; Saunier **et al.**, 2015) peut être utilisée pour faire l'interface entre l'agent et l'environnement. 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 est en charge de la maintenance des ressources et des 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, comme par exemple (Claes **et al.**, 2011), où l'environnement modélise le trafic routier, les agents étant les véhicules et l'infrastructure. Cette vision fonctionnelle de l'environnement est également présente dans (Ricci **et al.**, 2011), où l'environnement est un ensemble d'artéfacts correspondants aux services propres à un environnement. Si l'environnement est considéré comme un ensemble de services mis à disposition ou régulant les interactions des agents, alors

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

le modèle d'environnement est ajusté selon le modèle d'interaction du SMA. Si celui-ci intègre différentes dimensions, chacune est assimilée à un service. Par exemple la plateforme de Weyns *et al.* (2007) contient un service de perception et un autre communicationnel. Dans la plateforme de Ricci *et al.* (2011), des artefacts dédiés aux différents modèles d'interaction peuvent être utilisés. Selon cette approche, c'est à l'agent, utilisateur des services, de faire le lien entre les informations issues des différents services. Il n'est donc pas possible de faire exécuter des règles d'interaction en dehors des agents et utilisant des informations issues des différents modules sans le développement d'un nouveau service dédié. Par exemple, la gestion de communication situées demandera le développement d'un nouveau service exploitant les données issues des services gérant respectivement la localisation et la communication des agents. Dans la section suivante nous discutons les différents modèles d'environnements.

2.2. Environnements physiques, communicationnel et sociaux

L'environnement est un espace partagé où les agents interagissent par leurs actions ou communications, éventuellement selon un modèle social. Trois points de vue sur l'environnement sont ainsi obtenus (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 systèmes où les agents interagissent physiquement ou par la communication selon des règles sociales.

Pour chaque point de vue, le modèle d'environnement assure la gestion des interactions. 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) permet de prendre 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 communicationnel, 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 multiparties. Ces filtres sont alors gérés par l'environnement. Les environnements sociaux sont caractérisés par l'explicitation des relations sociales entre agents, ces dernières influençant l'exécution des actions et/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 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. La proxémie est une étude sociologique portant sur le rapprochement de personnes lors d'échanges en fonction de leurs affinités (Hall, 1990). Elle s'intéresse à l'utilisation de l'espace lors d'interactions sociales. 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. Problématique et cas d'utilisation

3.1. Problématique et focus du papier

Nous avons souligné dans la section précédente que la plupart des travaux considère l'environnement comme un bloc dont la modélisation réponds 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 composition des dimensions qui le composent. Chacune des dimensions conserve 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 définition de leur articulation. Ainsi, dans chacune des dimensions, le résultat de l'interaction d'un agent dépend de règles qui sont propres à un modèle particulier et l'articulation entre les dimensions doit être définie.

Nous nous intéressons, dans cet article, à trois classes d'articulation :

1. Une **interaction dont la perception dans une dimension est contrainte/favorisée par une autre**. Par exemple, la réception du message dans l'environnement social est contrainte par des règles physiques comme la proximité.

2. Un **même événement génère des interactions différentes selon les dimensions**. Par exemple, le contenu du message modifie le comportement des membres d'une communauté et la forme du message modifie leur environnement physique.

3. Une **interaction dans une dimension qui génère d'autres interactions dans d'autres dimensions**. Par exemple, le message est issu de l'environnement social et modifie l'environnement physique.

Nous pensons que ces trois classes couvrent 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 peut aisément supporter de nouvelles classes d'articulation en fournissant les outils nécessaires à leurs intégration.

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 complexe car ouvert, dynamique et incertain, il faut pouvoir déployer des systèmes capables de prendre en compte simultanément des interactions issues des trois points de vue de l'environnement que nous avons présentés. Nous prendrons l'exemple d'un conducteur de véhicule communicant (communication inter-véhiculaire et avec l'infrastructure type Vanet) 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. Une simulation de trafic reproduisant un comportement similaire doit pouvoir gérer toutes ces interactions.

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. Il s'agit d'un ensemble d'arcs équipés de capteurs pour connaître la densité du trafic, et d'intersections où des feux sont équipés pour relayer des messages. 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) et la présence de véhicules prioritaires (information issue de l'environnement social). Dans cet environnement, nous traitons le cas d'un véhicule prioritaire qui veut la priorité et qui envoie régulièrement un message à la communauté d'utilisateurs pour demander qu'on lui cède le passage

et informe par l'usage d'une sirène les véhicules à portée. 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. Ce scénario illustre l'utilisation conjointe des trois classes 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 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. Il sera pris en compte par les véhicules de secours pour planifier leur intervention.

4. Modèle de l'environnement multidimensionnel

Dans cette section, nous proposons un modèle de l'environnement combinant plusieurs dimensions : physique, communicationnel et sociale. 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 respectant "localement" la définition : préserver l'intégrité de l'environnement (l'évitement des inter-pénétrations et l'application de la gravité sont des exemples de préservation de l'intégrité de l'environnement), gérer les interactions avec les agents.

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.

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 une dyschromatopsie partielle, la fonction Percept_d peut être formulée pour ne percevoir que les objets proches dans la dimension physique, et n'étant pas de couleur rouge.

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

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.

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

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.

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

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 $\lambda_e \in \Lambda_e$ est définie par :

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

Une règle retourne un tuple constitué par l'ensemble d'actions $\psi_{d'}$ dans les autres dimensions $d' \neq d$, et par l'action à appliqué dans la dimension d .

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 applicables qu'après avoir été modifiée. Par exemple, la réception du message dans l'environnement social 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 : $\psi'_d = \psi_d/|\psi_d|$.

4.2. Dynamique de l'environnement

Dans cette section, la dynamique constituant le modèle d'environnement multidimensionnel décrit dans les sections précédentes sont présentées. Cette dynamique est décrite par la fonction *Cycle* permettant de déterminer l'état suivant σ'_e de ce système à partir de l'état courant $\sigma_e \in \Sigma$, tel que $\sigma_e = \{\sigma_d | d \in D_e\}$:

$$\begin{aligned} \text{Cycle} : E \times \Sigma &\rightarrow \Sigma \\ \text{Cycle}(e, \sigma_e) &= \text{Apply}(\sigma_e, \text{Multi}(\sigma_e, \text{Exec}(e, \sigma_e) \cup \{\text{Process}_d(\sigma_d) | d \in D_e\})) \end{aligned} \quad (7)$$

La fonction *Apply* détermine le prochain état σ'_e de l'environnement à partir des actions demandées par les agents ou les processus endogènes à l'environnement :

$$\begin{aligned} \text{Apply} : \Sigma \times [D \rightarrow \mathcal{P}(\Psi_{dom})] &\rightarrow \Sigma \\ \text{Apply}(\sigma, i) &= \sigma' \end{aligned} \quad (8)$$

Cette fonction est inspirée de la fonction Π proposée dans le modèle influence-réaction (Michel, 2007). La notation Ψ_{dom} dans l'expression $D \rightarrow \mathcal{P}(\Psi_{dom})$ correspond à l'ensemble des actions de la dimension donnée comme membre du domaine de cette expression.

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

$$\begin{aligned} \text{Multi} : \Sigma \times [D \rightarrow \mathcal{P}(\Psi_{dom})] &\rightarrow [D \rightarrow \mathcal{P}(\Psi_{dom})] \\ \text{Multi}(\sigma, I) &= \bigcup_{d \in \text{dom } I, \lambda_e \in L_e, a \in I(d)} r_1 \cup [d \mapsto r_2] \text{ avec } \langle r_1, r_2 \rangle = \lambda_e(d, a) \end{aligned} \quad (9)$$

Le premier membre de l'équation permet de générer des actions dans d'autres dimensions que la dimension d'origine d . Le second membre de l'équation permet d'appliquer l'action calculée par la règle λ_e dans la dimension d .

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

$$\begin{aligned} \text{Exec} : E \times \Sigma &\rightarrow [D \rightarrow \mathcal{P}(\Psi_{dom})] \\ \text{Exec}(e, \sigma_e) &= \bigcup_{a \in A, d \in D_e} \text{Act}_d(f(d), a) \\ &\text{avec } f = \text{Decide}_a\left(\bigcup_{m \in D_e} \text{Percept}_m(\sigma_m, a)\right) \end{aligned} \quad (10)$$

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 $\text{Decide}_a : \Sigma_d \rightarrow [D \rightarrow \mathcal{P}(\Psi_d)]$. Dans cet article, les actions pouvant amener à un changement d'état de l'agent a ne sont pas traitées.

5. Implantation en SARL

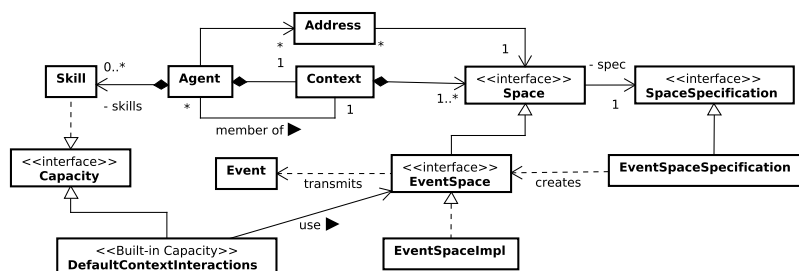


Figure 1. Concepts principaux de SARL

L'objectif de cette section est de présenter l'implantation avec le langage SARL³ du modèle présenté dans la section précédente. Nous illustrons également la capacité du langage SARL de supporter notre modèle en réduisant l'écart entre les concepts abstraits et les concepts d'implantation.

SARL est un langage orienté-agent généraliste (Rodriguez *et al.*, 2014). Il vise à fournir les abstractions fondamentales pour mettre en œuvre la concurrence, la distribution, l'interaction, la décentralisation, la réactivité, l'autonomie et la reconfiguration dynamique d'agents. Ces fonctionnalités de haut niveau sont désormais considérées comme les principales exigences pour une mise en œuvre simple et pratique d'applications logicielles orientées-agent. La principale perspective qui a guidé la création de 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. 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.

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

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.

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 de l'environnement physique, il est nécessaire de définir une capacité dédiée. Le script 1 décrit les fonctionnalités accessibles à un agent pour tous les environnements physiques (`AbstractPhysicEnvironmentCapacity`), et plus particulièrement pour les environnements constitués de réseaux routiers (`RoadEnvironmentCapacity`). Nous proposons cette distinction afin de faciliter la définition d'autres capacités dédiées à des dimensions qui utiliseraient des objets et des représentations différents.

```

capacity AbstractPhysicEnvironmentCapacity {
2   def getLinearSpeed : double
   def setPhysicalPerceptionAlterator(filter : PhysicalPerceptionAlterator)
4   def setInterestFilter(filter : InterestFilter)
   def influence(inf : Influence)
6   def killBody
}
8 capacity RoadEnvironmentCapacity extends AbstractPhysicEnvironmentCapacity {
   def getPosition : Pair<Segment,double>
10  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 via son corps en utilisant le modèle d'interaction influence-réaction. Une influence peut décrire un mouvement (`MotionInfluence`) ou l'exécution d'une action sur un objet (`ActionInfluence`).

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

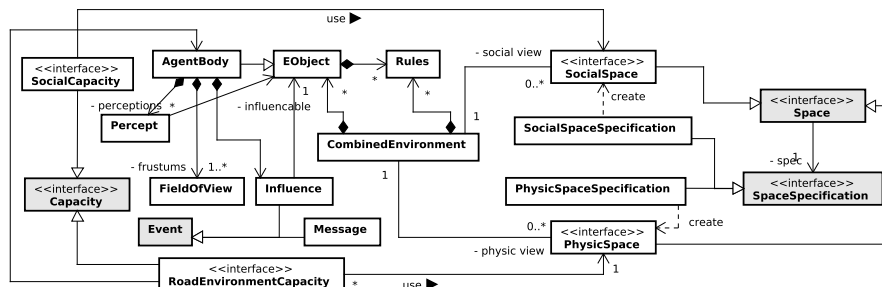


Figure 2. Concepts relatifs au modèle d'environnement (en blanc)

```

4  event MotionInfluence extends Influence {
5      var linearSpline : double
6      var linearShift : double
7      var path : Segment[]
8  }
9
10 event ActionInfluence extends Influence {
11     var actionName : String
12     var actionParameters : Object[]
13 }
14 event Perception {
15     var objects : Percept[]
16 }

```

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`), et/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.

```

2  public interface PhysicSpace extends Space {
3      public PhysicBodyFactory getBodyFactory();
4      public void putInEnvironment(AgentBody body, Agent perceptionListener);
5      public void influence(AgentBody body, influences : Influence[]);
6      public void destroyBody(AgentBody body);
7  }
8  public class PhysicSpaceImpl extends EventSpaceImpl implements PhysicSpace {
9      private CombinedEnvironment env;

```

```

10   public void influence(AgentBody body, influences : Influence[]) {
11       for(Influence i : influences) emit(i, new Scope(env));
12   }
13   ...
14 }

```

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 choisi de l’environnement, par exemple Jaak⁵. Nous proposons que chaque influence émise par un agent soit envoyée uniquement à 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 correspond à la fonction `emit` fournit par cette capacité.

```

2   event Message {
3       var destination : Scope<Address>
4   }
5   capacity SocialCommunicationCapacity {
6       def emit(e : Message, scope : Scope<Address> = null)
7   }

```

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

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

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

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** fournit par le langage SARL.

```

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

```

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 d’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 modélisés 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)) {
19         if (multi(occurrence, participant)) {
20             saveMessage(occurrence)
21         }
22     }
23 }
24 }
25 }
26 def multi(e : Event, o : Object) : boolean {
    for(pair : rules) {
27     if (pair.first(this, e, o)
28         && !pair.second(this, e, o)) {
29         return false
30     }
31 }
32 return true
33 }
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, et/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.

La mise en œuvre de l’ensemble des règles au sein de l’environnement e est définie par la fonction $Multi(\sigma, I)$. 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

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.

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. 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.

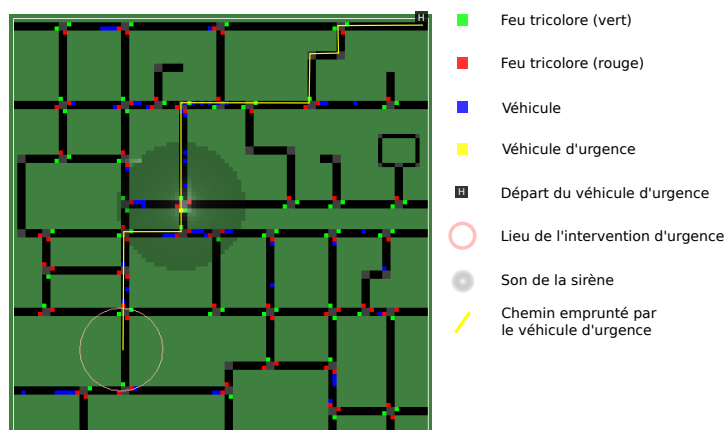


Figure 3. Visualisation d'une simulation

Le premier objectif de ces expérimentations est d'illustrer l'usage de différentes dimensions de l'environnement dans le modèle de simulation. Ainsi, trois cas d'expérimentations sont considérés et listés dans la table 1. Chaque colonne représente une dimension de l'environnement contenant des éléments pouvant être perçus par les conducteurs simulés. Les deux premières colonnes représentent la dimension de déplacement constituée par les routes, d'une part, et d'autre part les feux tricolores. La troisième colonne représente la dimension sonore de l'environnement, dans laquelle le son de la sirène se propage. La dernière colonne correspond à la dimension sociale associées aux véhicules.

Le second objectif est d'illustrer la facilité de conception des différentes 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.

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	Environ- nement social
1	✓	✓		
2	✓	✓	✓	
3	✓	✓	✓	✓

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 la langage SARL et la plate-forme JANUS en fournissant les outils nécessaires à la modélisation 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 :

1. La **dimension physique de déplacement** dans laquelle les agents peuvent de 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 et 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 priorité les voies sur lesquelles se trouvent le véhicule d'urgence. De plus, le véhicule d'urgence informe les autres membres de la communauté de son départ en intervention.

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
   on Initialize {
4     setSkill(DrivingCapacity, StandardDrivingSkill)
   }
6   on Perception {
     var stopVehicleInStandardCondition = isVehicleStop(occurrence)
8     var siren = occurrence.body.getFirstPerceptionAtCurrentPosition(typeof(
       Siren))

```

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

```

    var stopVehicleForEmergencyVehicle = isVehicleStopWhenEmergencyVehicle(
siren)
10  if (!stopVehicleForEmergencyVehicle && !stopVehicleInStandardCondition) {
12      var motion : Vector2i = null
14      if (!path.empty) {
16          motion = followPath(path, occurrence)
18      }
19      if (motion === null) {
20          motion = driveRandomly(occurrence)
21      }
22      if (motion != null && motion.lengthSquared > 0) {
23          move(motion, true)
24          this.previousOrientation = direction
25      }
26  }
27  }
28  }

```

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 7). Lorsqu’un agent conducteur perçoit la sirène du véhicule d’urgence dans la dimension sonore, il peut décider de s’arrêter (ligne 9). 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. Lorsque l’agent conducteur décide de ne pas s’arrêter, il peut suivre le chemin qu’il a précédemment calculé ou sélectionner aléatoirement une direction (lignes 12 à 17).

Le véhicule d’urgence (représenté par un carré jaune sur la figure 3) apparaît dans la simulation lorsqu’une situation nécessitant son intervention est nécessaire. 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. Dans notre cas d’application, les agents recalculeront les chemins afin d’éviter de croiser l’ambulance : les cellules traversées par l’ambulance voient leur coûts augmentés lors des calculs des plus courts chemins.

6.1.1. Règles entre les dimensions de l’environnement

Dans cette section, les trois cas de figure associée à notre exemple illustratif (section 3.2) sont introduit dans la base de règles inter-dimensionnelles L_e de notre modèle. Le script 8 propose la définition de ces trois règles en utilisant la syntaxe du

language 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  /* Regle pour le cas 1 */
   environnement.rules +=
4     [ env,e,o | e instanceof PriorityRequest
           && o instanceof TrafficLight ]
   => [ env,e,o | env.socialSpace.participant(e.source.ID)
6         instanceof PriorityVehicle ]
   /* Regle pour le cas 2 */
8  environnement.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  /* Regle pour le cas 3 */
   environnement.rules +=
18  [ env,e,o | e instanceof PhysicCollision ]
   => [ env,e,o | env.emit(new Alert(e.position))
20         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) implanté 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, la population de véhicules (2500 entités) est divisée en trois groupes. Le premier groupe est constitué par les véhicules percevant la sirène du véhicule d'urgence. Le second groupe est composé 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 premiers groupes. Le troisième groupe, constitué de tous les véhicules n'appartenant pas à l'un des deux groupes précédents, est ignoré dans la suite de cette section. En effet, nous pensons qu'il n'offre pas de plus-value significative pour répondre aux objectifs de nos expérimentations. La table 2 illustre les durées et les longueurs des trajets des véhicules. Les trois premières lignes de cette table mettent en lumière une évolution positive significative du nombre de véhicules percevant la sirène du véhicule d'urgence (groupe 1). 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. Un sous-ensemble d'entités qui ont recalculé un chemin ont également perçu la sirène durant l'expérimentation (~ 500 entités). Ce phénomène s'explique par les positions courantes des véhicules 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. Il devient alors difficile pour un autre véhicule de déterminer un chemin ne croisant pas la trajectoire du véhicule d'urgence. Les longueurs moyennes des trajets restent sensiblement identiques dans les trois cas d'expérimentation. Toutefois, le recalcul des trajets augmente en la longueur des trajets de ~ 8%. Cette augmentation est compensée par une plus grande proportion d'agents ne modifiant pas leur trajet ou n'obtenant pas un trajet modifié.

La table 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 ne rencontre pratiquement plus 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.

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 augmente de l'expérimentation une à la trois et 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 le second objectif, 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* de SARL, 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

Dans cet article, nous proposons un modèle permettant la combinaison de différentes dimensions, incluant les dimensions physique et sociale, de l'environnement

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 obtenu un autre chemin ($C \subseteq B$)	-	-	360
Nombre de véhicules ayant gardés le même chemin ($B \setminus C$)	-	-	567
Nombre de véhicules ayant obtenu un autre chemin mais perturbés par le véhicule prioritaire ($A \cap C$)	-	-	236
Durées des trajets de A			
Moyenne arithmétique	401.7698	429.9282	493.6714
Écart-type	101.5345	101.6514	184.8414
Durées des trajets de $B \setminus A$			
Moyenne arithmétique	-	-	372.3214
Écart-type	-	-	176.2145
Durées des trajets de C			
Moyenne arithmétique	-	-	593.1705
Écart-type	-	-	211.1705
Longueurs des trajets de A (en nb. cellules)			
Moyenne arithmétique	161.5006	162.1511	182.1903
Écart-type	29.6449	26.9169	48.3560
Longueurs des trajets de $B \setminus A$ (en nb. cellules)			
Moyenne arithmétique	-	-	143.9181
Écart-type	-	-	49.5870
Longueurs des trajets de C (en nb. cellules)			
Moyenne arithmétique	-	-	205.4334
Écart-type	-	-	55.6448
Augmentations des longueurs des trajets de C (en nb. cellules)			
Moyenne arithmétique	-	-	+52.9945
Écart-type	-	-	47.1474

Tableau 3. Temps moyens des trajets du véhicule d'urgence

	Exp. 1	Exp. 2	Exp. 3
Moyenne arithmétique	289.6	313.85	157.8334
Écart-type	77.8655	70.6897	8.1530

afin d'enrichir les capacités de modélisation de l'environnement et de fournir aux agents des outils de définition de comportements plus complexes. 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, “*smart objects*”, holarchies, etc.

L'enrichissement des capacités comportementales des agents peut être réalisé par un modèle d'environnement informé. Un environnement informé fournit une information sémantique détaillée aux agents à l'aide de marqueurs sémantiques ou d'ontologies. 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, et sa relation avec son comportement dynamique, en utilisant une vision holonique de l'environnement. 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 ou GAML.

Bibliographie

- Bhouri 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.

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