

Environnements multi-agents pour la conception d'applications dédiées au transport

FLAVIEN BALBO, FABIEN BADEIG ET MAHDI ZARGAYOUNA

Résumé : Cet article présente une approche reposant sur la modélisation d'un environnement multi-agent pour la conception d'applications dédiées au domaine du transport. Fondée sur le principe PbC (Property-based Coordination), cette approche a pour objectif la résolution de trois difficultés récurrentes dans la conception de ces applications : le problème de la gestion des connaissances, la dimension spatio-temporelle et la dynamique de l'environnement réel. Pour mettre en évidence les avantages de notre approche, deux applications différentes sont présentées, un système de transport à la demande et un simulateur de gestion de crises. Pour ces deux systèmes fondés sur PbC, nous présentons comment les trois difficultés récurrentes ont été résolues.

Mots clés : multi-agent, coordination basée sur les propriétés (PbC), gestion des connaissances, transport à la demande, gestion de crises

1. INTRODUCTION

Le domaine du transport a inspiré de très nombreux travaux en intelligence artificielle [9] et dans la communauté multi-agent [4, 12]. L'approche multi-agent traite de systèmes constitués de nombreux composants distribués physiquement et/ou logiquement. Ces composants, appelés agents, interagissent et possèdent un certain niveau d'autonomie. Ils sont capables de percevoir leur environnement et de réagir à ses changements en fonction de leurs buts. Fondé sur une approche montante de la conception de systèmes, le paradigme multi-agent simplifie la compréhension d'une réalité complexe par la réification des composants du système à gérer. Le paradigme multi-agent est d'autant mieux adapté qu'il facilite une approche par analogie dans le domaine du transport dont un des objectifs est la coordination d'entités distribuées. C'est pourquoi l'approche multi-agent est souvent choisie pour modéliser, résoudre et/ou simuler des problèmes de transport.

Lors de nos précédentes recherches dans la conception d'applications dans le domaine du trans-

port [2, 16], nous avons identifié trois difficultés. La première concerne la gestion de connaissances, qui peuvent être incomplètes, appartenir au domaine de l'expertise ou nécessiter des mécanismes de coordination pour prendre en compte des sources de connaissances distribuées et hétérogènes [10]. La seconde difficulté concerne la prise en compte de la dimension spatio-temporelle des problèmes. Par exemple, il est difficile de donner la bonne information et/ou service au bon moment au bon endroit [11, 16]. La dernière difficulté concerne la prise en compte de la dynamique de l'environnement réel qui a des conséquences sur la qualité de l'information et rend plus difficiles et coûteuses les communications directes. La dynamique des systèmes provient également de l'apparition et disparition de ses composants (par exemple, gestion de véhicules autoguidés [15]).

Dans le présent article, nous présentons un principe de coordination appelé PbC (Property-based Coordination) qui promeut l'utilisation par les agents de descriptions observables des com-

posants réels et abstraits d'un Système Multi-Agent (SMA) à des fins de coordination. Dans notre approche, l'environnement multi-agent contient les descriptions et supporte leur traitement. Nous décrivons deux applications qui sont fondées sur la modélisation d'un environnement conforme au principe PbC et présentons les solutions apportées aux difficultés présentées précédemment.

Le plan de l'article est le suivant : la section 2 présente le rôle de l'environnement dans la conception d'applications pour le transport et comment ce concept peut supporter le principe PbC. La section 3 détaille un système de résolution du problème de transport à la demande. La section 4 décrit un simulateur pour la gestion de crises en transport. Enfin, nous portons une conclusion générale et proposons des perspectives en section 5.

2. ENVIRONNEMENT COMME SUPPORT DE COORDINATION

2.1 RÔLE DE L'ENVIRONNEMENT

L'environnement dans un SMA a des responsabilités [14] qui peuvent être mises à profit dans la conception d'applications transports. L'environnement est un espace partagé pour les agents, ressources et services et sa première responsabilité est de structurer le SMA. La modélisation de l'environnement est une solution pour donner un référentiel spatio-temporel dans une application transport. Par exemple, les services fondés sur la localisation exploitent l'information relative à la localisation des utilisateurs mobiles afin d'adapter leur traitement. Dans les solutions multi-agents, cette tâche peut être réalisée par l'environnement : soit l'environnement centralise l'information sur les positions des utilisateurs et recherche les services/informations possibles [11], soit il est distribué et constitue un support homogène pour la conception de services [7]. Son rôle d'intermédiaire privilégié lui permet de supporter les modèles de coordination découplée dans le temps et dans l'espace qui sont appropriés pour prendre en compte un environnement réel dynamique. L'environnement du SMA devient la référence commune qui permet aux agents d'adapter leur comportement selon la dynamique de l'environnement réel.

La seconde responsabilité de l'environnement est de maintenir sa propre dynamique, qui peut prendre en compte la gestion de la dynamique de l'environnement réel et ainsi garantir la cohérence du SMA. Dans [15], l'environnement assure la propagation de champs qui sont utilisés par les agents afin d'organiser l'affectation de tâches pour des véhicules guidés automatiquement. De plus, l'environnement peut assurer des services qui ne sont pas au niveau de l'agent ou pour simplifier la conception des agents. Dans un système de gestion

de feux de circulation [3], l'environnement qui a un point de vue global, donne des récompenses et des punitions à des agents égoïstes selon leurs décisions locales.

Grâce à son contrôle de l'espace partagé, sa troisième responsabilité est de définir les règles du système multi-agent. Dans une simulation de réseau d'autobus [8], le rôle de l'environnement est de contraindre les perceptions et interactions des agents. Pour les applications transports pour lesquelles il n'existe pas de connaissances suffisantes, la conception du SMA est simplifiée par une séparation claire entre le rôle des agents et leur organisation.

Parce que les agents sont utilisateurs des services de l'environnement et afin de créer une connaissance commune, la dernière responsabilité de l'environnement est de laisser observable et accessible sa propre structure.

2.2 PRINCIPE DE LA COORDINATION FONDÉE SUR LES PROPRIÉTÉS (PbC)

Le principe de la coordination fondée sur les propriétés peut s'appuyer sur l'observabilité de l'environnement : *Le principe de la coordination fondée sur les propriétés est de concevoir les composants du système multi-agent comme des composants observables symboliques (SC) et de gérer leur traitement à des fins de coordination.*

Il existe deux catégories de SC enregistrés dans l'environnement. La première concerne les descriptions symboliques (SD) d'un composant réel du SMA : agent, message ou objet. Ces descriptions sont symboliques, car les composants réels sont représentés par un ensemble de paires propriété-valeur. Par exemple, un agent a son propre processus et ses propres connaissances et une SD est enregistrée pour lui dans l'environnement. Seuls les descriptions sont observables et un contrôle peut être réalisé par l'environnement. Par exemple, dans la Figure 1, l'environnement contient les SD de deux agents, deux messages et trois objets.

La seconde catégorie de SC concerne les composants abstraits du SMA et plus particulièrement les composants de coordination. L'objectif est de gérer comment les agents lient le contenu de l'environnement à leur comportement. Chaque lien représente le besoin d'un agent ou d'un ensemble d'agents et est réifié par un objet de sélection (SO) (quatre SO Figure 1) qui est un ensemble de contraintes sur les propriétés des SD. Par exemple, si les SD de messages et de véhicules sont enregistrées dans l'environnement, les besoins suivants pourraient être représentés : « Je souhaite envoyer un message au véhicule le plus rapide » ou « quand je suis près d'un véhicule, je dois ralentir ».

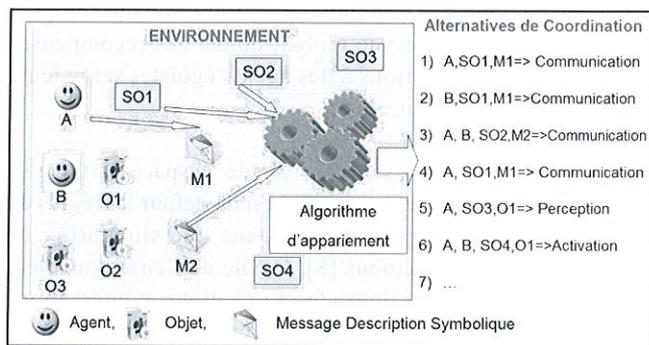


Figure 1 : Principe de la coordination fondée sur les propriétés

La gestion dynamique du traitement des *SC* permet aux agents d'adapter leurs interactions avec l'environnement selon leur besoin. Ainsi, les agents ajoutent (retirent) les *SC* dans l'environnement, cette dynamique assurant l'autonomie des agents. Le processus commun est un algorithme qui apparie les *SO* avec les *SD*. Quand un *SO* est déclenché, cela active une action qui est réalisée par l'agent (réception d'un message, activation de l'agent en cas de simulation, perception d'un objet).

La Figure 1 donne quelques exemples de la combinaison de descriptions. Grâce au type d'action qui est associé à chaque *SO*, différents modèles de coordination peuvent être supportés simultanément. Par exemple, si la coordination est fondée sur un modèle de communication, alors les messages ont un *SD*, les *SO* réifient les intérêts des agents pour ces messages et le processus d'appariement mis en œuvre par l'environnement donne pour chaque agent le message qui le concerne (Figure 1, alternatives 1 à 4).

L'état de l'environnement, qui est composé des *SD*, constitue le contexte des agents et leur permet d'adapter leur comportement selon cette information. La réification des besoins des agents par des *SO* est le point de départ pour le traitement contextuel du comportement des agents. Les agents conditionnent leur comportement selon l'état du contexte incluant leur propre état. Dans la Figure 1, l'alternative 6 est un exemple d'activation contextuelle qui prend en compte l'état des agents A et B et de l'objet O1.

Pour être conforme au principe PbC, l'environnement doit fournir les services pour la gestion des *SC*, et un algorithme d'appariement. Il n'y a aucune hypothèse quant à la modélisation des *SC* et l'algorithme d'appariement. Les deux applications suivantes sont fondées sur un environnement qui est conforme au principe PbC.

3. COORDINATION PAR L'ENVIRONNEMENT POUR LE TRANSPORT À LA DEMANDE

Nous proposons un système de transport à la demande (STAD) modélisé par un SMA dans

lequel l'activité des agents est coordonnée à travers l'environnement conformément au principe PbC (plus de détails dans [17]).

3.1 SYSTÈMES DE TRANSPORT À LA DEMANDE

Un STAD est un système conçu pour répondre à des clients qui désirent être transportés d'un point du réseau à un autre. Les clients contraignent leurs demandes par une fenêtre de temps pour l'arrivée et une autre pour le départ. Les critères d'évaluation de l'efficacité d'un STAD sont, d'abord, la minimisation du nombre de véhicules utilisés pour servir tous les utilisateurs et ensuite la minimisation de la distance totale parcourue par les véhicules.

Un STAD est confronté aux trois difficultés présentées dans l'introduction. Premièrement, la difficulté concernant la connaissance est due au fait que les utilisateurs sont inconnus au démarrage du système. Dans notre système, l'utilisation de l'environnement permettra la mise en commun de toutes les informations dès qu'elles deviennent disponibles. Deuxièmement, la dimension spatio-temporelle du problème est importante, puisque la distribution spatio-temporelle des utilisateurs ainsi que la largeur moyenne de leur fenêtre temporelle conditionne le comportement des véhicules et leur coordination, influençant ainsi grandement la qualité des solutions fournies par le système. Dans notre application, l'environnement structure les composants du SMA temporellement et spatialement de telle manière que l'interaction entre les agents est dirigée selon leur perception de cette information. La dernière difficulté concerne la dynamique de l'environnement. Un SMA modélisant un STAD est un SMA ouvert puisque les agents (clients et véhicules) rejoignent et quittent le système librement. Dans un tel environnement dynamique, la limitation des communications est très importante, car la diffusion de toutes les informations serait très coûteuse.

Nous proposons un modèle de coordination conforme à PbC, dans lequel la communication est découplée dans le temps et l'espace palliant le risque de perte d'information dans un environnement dynamique. Cela permet aux agents d'interagir sans qu'ils s'exécutent au même endroit ni au même instant.

3.2 DESCRIPTION DU SYSTÈME

Nous avons conçu un modèle distribué pour un STAD, dans lequel il existe deux catégories d'agents : les agents véhicules (*AV*) et les agents clients (*AC*). Les *AV* et *AC* sont générés dynamiquement : un nouveau *AC* est associé à chaque nouveau client qui se connecte au système et un nouveau *AV* est associé à chaque nouveau véhicule créé (pour prendre en charge un client qui ne peut pas être pris par les *AV* existants). Les *SD* du système concernent les *AV* et *AC* (Figure 2). Un *AV* est

MULTIAGENTS & TRANSPORTS

décrit symboliquement par sa position courante et le nombre de places restantes. Un *AC* est décrit par les nœuds de départ et d'arrivée, ses fenêtres temporelles, le véhicule *veh*, et les clients successeurs et prédécesseurs (propriété *succ* et *pred*) dans l'itinéraire de *veh*. Un *AC* qui n'est pas associé à un véhicule a ses propriétés mises à *inconnu*.

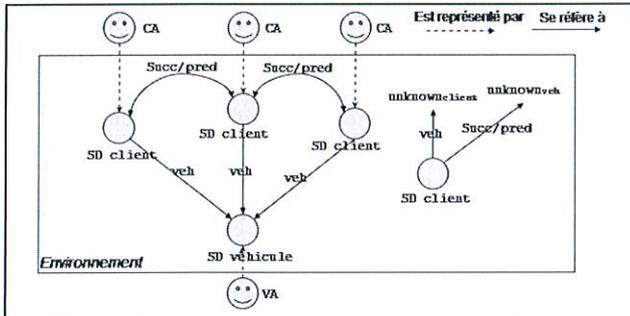


Figure 2 : Description Symbolique

Dans notre application, les *SO* sont définis de manière à ce que les *AV* ne perçoivent que les clients qu'ils peuvent insérer dans leur itinéraire. Le *SD* d'un nouveau *AC* (avec les valeurs de *veh*, *succ* et *pred* à inconnu) est perçu si : (i) il y a deux nœuds dans l'itinéraire d'un véhicule entre lesquels le nœud de départ peut être inséré sans violer aucune des fenêtres temporelles des clients sur cet itinéraire, (ii) s'il reste des places disponibles dans le véhicule au moment de l'insertion, (iii) s'il y a deux nœuds dans l'itinéraire d'un véhicule entre lesquels le nœud d'arrivée peut être inséré sans violer aucune des fenêtres temporelles des clients sur cet itinéraire.

Grâce à PbC, un *AV* n'est mis en contact qu'avec les nouveaux *AC* qui peuvent être insérés. Cette mise en relation est réalisée sans connaissance mutuelle *a priori* tout en limitant les communications dans le système aux seuls agents qui peuvent aboutir à un accord (insertion dans l'itinéraire d'un véhicule). Il est important de noter que tout *AV* qui ne participe pas à l'insertion d'un *AC* reste candidat pour l'insertion d'autres clients.

Le protocole suivi est un mécanisme de négociation entre *AC* et *AV*. Chaque *AV* participant calcule un prix correspondant à l'insertion du nouvel *AC*, et le lui propose. Ce dernier choisit l'*AV* proposant le prix le plus bas. Ce protocole est une version distribuée des heuristiques d'insertion [13]. Ces heuristiques sont les plus rapides puisque une décision d'insertion n'est pas remise en cause ultérieurement. Cependant, ces heuristiques sont connues pour être myopes [5], car le prix calculé est généralement fonction de la distance supplémentaire qu'implique pour le véhicule l'insertion du nouveau client. Nous avons décidé de conserver ce protocole de négociation en proposant une nouvelle méthode de calcul du prix d'insertion.

3.3 CALCUL DU PRIX D'INSERTION

Plutôt que de prendre en compte la distance parcourue par les *AV* pour calculer le prix d'insertion

d'un client, nous avons choisi de prendre en compte leur disponibilité future. Dans ce contexte, l'utilisation d'un environnement commun conforme à PbC nous aide à concevoir une nouvelle méthode pour matérialiser la disponibilité des *AV*. Nous rappelons que l'environnement multi-agent pour un STAD peut être modélisé comme un réseau espace-temps dans lequel il n'y a pas seulement des nœuds mais des paires nœud-temps. Nous appelons champ de perception d'un *AV*, les nœuds espace-temps qu'il peut visiter, c'est-à-dire l'ensemble des nœuds qu'il peut visiter étant donné son itinéraire actuel. L'insertion d'un nouveau client dans l'itinéraire d'un *AV* implique qu'un certain nombre de nœuds ne pourront plus être visités, réduisant ainsi les possibilités de cet *AV* de participer à de nouvelles négociations pour l'insertion de clients. Par conséquent, plus le champ de perception d'un *AV* sera large plus il aura de chances d'être candidat pour l'insertion de futurs clients (n'ayant aucune information sur les clients, nous supposons par défaut qu'il y a une distribution uniforme des clients).

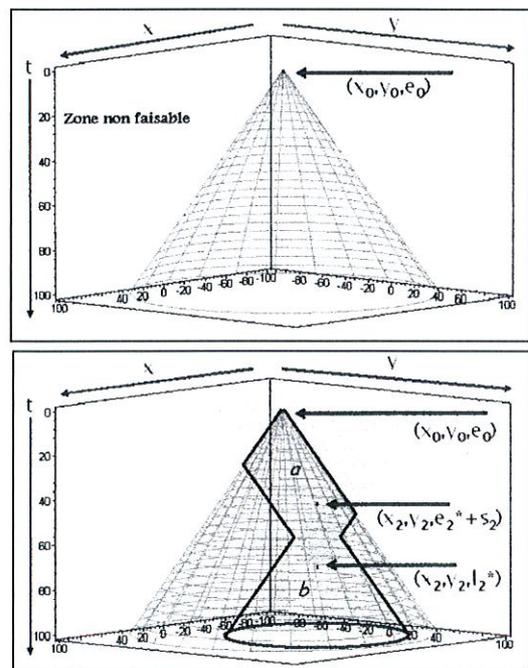


Figure 3 : Champs de perception

La notion de champ de perception est illustrée à la Figure 3 avec une version Euclidienne du problème où le réseau de transport est un plan. L'environnement du SMA est dans ce cas un cube représentant toutes les positions spatio-temporelles virtuelles possibles de tous les véhicules dans le réseau. Le cône sur la partie gauche représente le champ de perception d'un *AV* positionné en (x_0, y_0) au moment e_0 . Cet *AV* ne peut pas être candidat pour des clients dont le nœud d'arrivée ou de départ ainsi que le temps associé, qui est en dehors de ce cône. En effet, même si le véhicule partait à e_0 il ne pourrait les desservir étant donnée la métrique euclidienne. Sur la partie droite de la Figure 3, nous illustrons la conséquence sur le

► champ de perception d'un *AV* de l'insertion d'un nouveau client. La zone spatio-temporelle qui est en dehors de la zone en gras mais qui est dans le cône initial correspond à la perte de perception de l'*AV* résultat de l'insertion de ce nœud.

Dans le processus de négociation, le prix qui est proposé aux *AC* est la différence entre l'ancien champ de perception (avant l'insertion) et le nouveau (après l'insertion). Le but principal du processus de négociation devient l'optimisation des champs de perception des agents, avec l'objectif de minimiser le nombre de véhicules utilisés pour répondre à la demande. Dans [17], des résultats préliminaires montrent que ce protocole se comporte mieux que les heuristiques traditionnelles d'insertion pour la minimisation du nombre de véhicules mais pas pour la minimisation de la distance totale parcourue. De plus, nos expérimentations ont montré la pertinence de l'utilisation de l'environnement pour limiter le coût des communications puisque le gain en termes de messages échangés est plus que proportionnel au nombre de clients connectés au système. Cela implique que plus il y a d'agents dans le SMA, plus nous économisons en termes de charge réseau en comparaison avec une solution fondée sur la diffusion des demandes des clients à tous les véhicules (solutions usuelles pour la mise en œuvre des heuristiques d'insertion).

4. ENVIRONNEMENT INTERACTIONNEL POUR LA GESTION DE CRISES

Cette section propose de mettre en œuvre une approche centrée environnement conforme au principe PbC pour résoudre les difficultés énoncées en introduction dans le cas de simulation de crise pour le transport.

4.1 LE CONTEXTE D'UNE CRISE

L'expertise concernant la gestion de crise est incomplète, car chaque situation de crise est complexe et unique. Une crise est un phénomène dynamique défini par une situation initiale dépendant du lieu et du moment, et par son impact sur la population et l'infrastructure. Pour répondre aux situations de crise liées au trafic routier, des Plans de Gestion de Trafic (PGT) sont développés et spécifient les mesures à prendre en cas de crise pour coordonner les différents services d'intervention (services d'urgence, les médias, les gestionnaires de trafic). Notre proposition est de concevoir un système d'aide à la décision pour l'évaluation et la validation de PGT.

Chaque service dispose de ses propres protocoles de communication et d'actions en vue d'atteindre ses buts locaux. Un service avec sa propre organisation peut être vu comme une source de connaissances et une organisation globale efficace doit être trouvée. Un service est composé de décideurs et d'intervenants. Les décideurs adaptent le processus de décision selon leur reconsti-

tution de la situation de crise et les intervenants mettent en œuvre les décisions prises. La sphère fonctionnelle fournit des informations sur la situation pour permettre aux décideurs de concevoir une représentation cohérente de la situation. Selon son service, son rôle et son but, les intervenants ont une perception sélective de leur environnement réel. D'un point de vue multi-agent, un agent est soit cognitif (décideur), soit réactif (intervenant) et les deux doivent être modélisés dans la même simulation, PbC [1] permettant de reproduire la perception partielle de l'environnement réel.

Une crise est un système multi-acteur dont la réponse opérationnelle est fondée sur une représentation cohérente et complète de la situation. Or les informations sont réparties dans l'espace, le temps et par intervenants. Il faut être capable de prendre en compte et de modéliser cette répartition temporelle et spatiale de l'information ainsi que la mise à disposition d'informations par les intervenants. Par définition, l'environnement est un espace partagé et il devient par conséquent le référentiel spatio-temporel avec une structure de contrôle. Pour la référence spatiale, les agents réactifs sont localisés dans un environnement spatial et l'environnement assure la validité des actions des agents. Pour la référence temporelle, le simulateur a besoin d'une horloge globale pour synchroniser les agents et l'environnement assure la gestion de ce temps global. Ainsi, l'information spatio-temporelle est partagée parmi les composants de la simulation et apparaît dans les SD des agents.

La dernière difficulté concerne la dynamique de l'environnement réel dans lequel la crise évolue et qui modifie le comportement des intervenants. Puisqu'il est utilisé afin de contraindre le comportement des agents, l'environnement peut gérer le processus de simulation qui n'est pas à leur niveau et prendre en compte la simulation de l'environnement réel. Ainsi, l'environnement applique le scénario qui doit être testé (voir prochaine section) modifiant si nécessaire le processus d'interaction. Par exemple, le concepteur peut spécifier une probabilité de rupture d'un canal de communication.

4.2 DESCRIPTION FONCTIONNELLE DU SIMULATEUR DE CRISES

L'objectif est d'évaluer la réponse organisationnelle des services impliqués dans un Plan de Gestion du Trafic (PGT) pour une situation donnée, où une autorité coordinatrice est en charge du déroulement du plan. Il faut maintenant définir les spécifications de la plate-forme de simulation en commençant par définir les entrées et les sorties du simulateur.

La première entrée du simulateur est le PGT qui définit les services concernés, leurs rôles et,

pour chaque mesure d'intervention, les conditions de mise en œuvre et les actions associées à cette mesure. Cette entrée est chargée par un fichier XML qui est une formalisation du PGT. La deuxième entrée du simulateur correspond au fichier d'initialisation qui détermine la situation initiale et les paramétrages de la simulation. Ce fichier définit le scénario qui spécifie la dynamique de la simulation indépendamment de la modélisation des agents. Il définit les modifications possibles des caractéristiques et des comportements des agents, l'évolution possible de la crise en parallèle de l'intervention des agents et la sensibilité des agents à la situation de crise en fonction de la nature de l'intervenant. Il précisera des probabilités de réussite des actions entreprises en fonction des catégories d'agents ou/et la participation des agents en fonction de critères d'intérêt. Il déterminera également l'ambiguïté possible de certains messages en fonction de l'émetteur, du récepteur et du contexte des agents. Pour résumer, ce fichier définit une dynamique de crise indépendamment de la dynamique du système générée par les agents et leurs interactions.

La plate-forme dispose d'une seule sortie qui est un historique de la simulation sur un certain nombre de données. Les critères d'évaluation définis par le fichier d'initialisation détermineront le fichier de sortie de la simulation.

Dans le cadre de notre simulateur, l'environnement n'est pas uniquement un référentiel spatial et temporel. Il est également un environnement d'exécution ce qui en fait l'organe central de notre plate-forme. Quatre modules sont identifiés dont trois s'articulent autour de l'environnement. Le premier module, appelé *plan de gestion du trafic*, gère la traduction du plan d'intervention pour l'autorité coordinatrice. Il constitue une source de données pour l'agent décisionnel représentant cette autorité. Ce module est initialisé par l'entrée correspondant au PGT. Ensuite, une simulation dispose d'un cycle de vie correspondant à un scénario d'exécution. Le deuxième module, appelé *cycle de vie*, définit les paramètres de simulation.

Il est directement relié à l'environnement pour initialiser les variables liées au processus d'exécution. Ce module est initialisé par le scénario qui est la deuxième entrée du simulateur. Le troisième module, qui est l'*observateur*, est directement connecté à l'environnement.

Les informations envoyées à ce module par l'environnement sont définies par le fichier d'initialisation qui détermine quelle variable doit être « observée ». Ainsi, ce module permet de récupérer certaines données liées à l'exécution d'un scénario pour une situation de crise donnée pour étudier des aspects spécifiques du déroulement du plan d'intervention. Le dernier module définit une *interface* entre la plate-forme de simulation et une source de données extérieure.

Ce module est essentiel, car il permet au simulateur de communiquer et de se synchroniser avec cette source. Pour notre problème, l'objectif est d'avoir une représentation pertinente de l'état du réseau et plus précisément de l'état du trafic. Pour ce faire, le simulateur s'interface avec un simulateur macroscopique de trafic. Le simulateur retenu est *Magister*, qui est développé par l'INRETS. *Magister* est une plate-forme multi-modèle intégrant plusieurs types de modélisation : LWR, LWR_{ACB}, METACOR, ARZ. Le transfert de données entre le module d'interfaçage et le simulateur *Magister* est assuré par le protocole TCP (Transmission Control Protocol).

L'environnement spatial doit alors se synchroniser avec le simulateur macroscopique de trafic par l'intermédiaire du module. L'objectif est d'évaluer les modifications du réseau (fermeture de voies, déviation, ...) engendrées par les mesures prises par la sphère décisionnelle et de pouvoir visualiser les conséquences de ces mesures sur l'ensemble du système (communication, organisation des acteurs, ...). La Figure 4 illustre l'architecture fonctionnelle de notre simulateur de gestion de crise en identifiant les quatre modules.

4.3 SPÉCIFICATIONS DU MODÈLE DE SIMULATION

Les interactions des agents sont dirigées par leur contexte qui nécessite habituellement un traitement local dans l'agent. Ce traitement complexifie les modifications du comportement des agents et le passage de la conception à l'implémentation du modèle. Notre objectif est de séparer le processus de conception du processus de simulation en déplaçant l'analyse du contexte des agents dans l'environnement. L'environnement contient les SO concernant la communication (traitement des messages) et l'activation (activité des agents) qui sont le lien entre le contexte d'un agent et son action. La gestion des interactions est alors unifiée simplifiant la prise en compte dans la même simulation d'agents cognitifs et réactifs. De plus, la

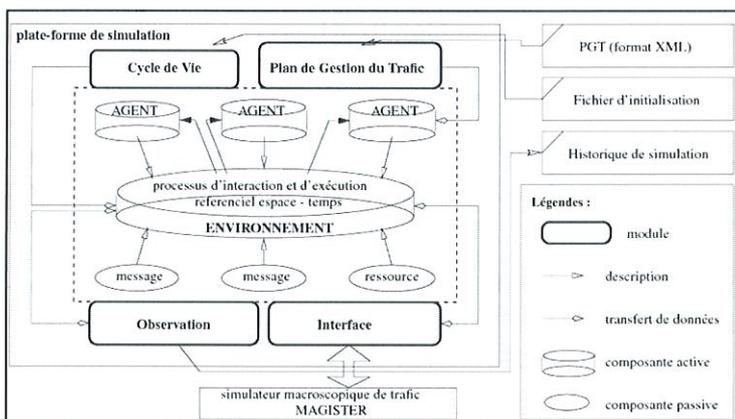


Figure 4 : Description fonctionnelle du simulateur de crise

► gestion des interactions par l'environnement fournit un nouveau levier de paramétrage de la simulation.

L'environnement est la référence spatio-temporelle pour les agents simulés et chacun d'eux a sa propre horloge interne. Le temps de l'environnement est discret et est le temps global de la simulation. À un cycle donné, le simulateur doit assurer que les agents qui sont prêts à agir sont activés et qu'ils ne le seront qu'une fois par cycle. Ce contrôle est assuré par l'environnement grâce à la comparaison entre le temps global et le temps interne de chaque agent pour chaque *SO* d'activation. Par conséquent, le temps interne des agents doit être observable et correspond à une propriété de la *SD* de chaque agent. Afin d'assurer une activation par défaut, un *SO default* est ajouté systématiquement dans l'environnement et son déclenchement dépend de la comparaison entre le temps global et celui des agents qui est mis à jour à chaque activation. Cette modélisation offre deux avantages : (1) un agent peut choisir de ne pas être activé pendant plusieurs cycles permettant un gain en temps d'exécution ; (2) un même agent ne peut pas être activé plus d'une fois par cycle.

L'activité d'un agent est modélisée par un automate de comportement. Chaque état de l'automate est une référence à un comportement qui est une suite cohérente d'actions. La transition d'un état à un autre dépend du contexte. Pour concevoir l'automate de comportement de chaque agent, le concepteur doit identifier les contextes d'activation pour chaque comportement puis construire une bibliothèque de comportements. Les agents choisissent dynamiquement les *SO* qu'ils ajoutent et retirent de l'environnement. Chaque *SO* correspond à une activation dans un contexte particulier. Ainsi, le même comportement peut être activé par différents *SO* et donc dans différents contextes. Réciproquement, la réaction d'un agent à un contexte donné peut être modifiée en changeant le comportement associé à un *SO*. Cette approche simplifie la mise en œuvre de différents scénarios. La Figure 5 décrit les relations entre les comportements des agents et leur *SO* d'activation.

5. CONCLUSION

Dans cet article, nous avons présenté un nouveau principe de coordination qui, lorsqu'il est mis en

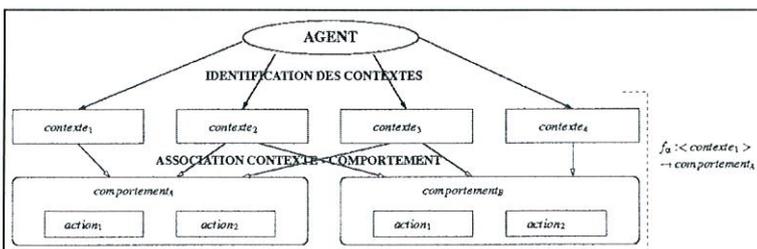


Figure 5 : Relation entre le contexte local d'un agent et son comportement

œuvre par la modélisation d'un environnement multi-agent explicite, permet d'apporter une réponse aux difficultés présentées dans l'introduction. Dans les applications dédiées au transport décrites dans cet article, l'utilisation d'un environnement multi-agent conforme au principe PbC a deux avantages. Premièrement, il structure la coordination et l'interaction des agents en prenant en compte lors de la modélisation le contexte d'action des agents dans un environnement dynamique. Deuxièmement, il permet la définition d'un nouvel objectif : l'optimisation du champ de perception de l'environnement qui s'est montrée être pertinente dans le contexte des STAD et doit être étendue à d'autres catégories d'applications transports.

6. RÉFÉRENCES

- [1] F. Badeig, F. Balbo *et al.* : *Contextual activation for agent-based simulation* ; in ECMS07, pages 128-133, 2007.
- [2] F. Balbo et S. Pinson : *Dynamic modeling of a disturbance in a multiagent system for traffic regulation* ; International Journal of Decision Support Systems, 41 (1), pp.131-146, 2005.
- [3] A. L. C. Bazzan : *A distributed approach for coordination of traffic signal agents* ; Autonomous Agents and Multiagent Systems, vol. 10, n°1, pp.131-164, mars 2005.
- [4] P. Davidsson, L. Henesey, L. Ramstedt, J. Törnquist et F. Wernstedt : *An analysis of agent-based approaches to transport logistics* ; Transportation Research part C: emerging technologies (special issue), vol. 13, n°4, Elsevier Science Publishers B. V., pp. 255-271, 2005.
- [5] M. Diana : *The importance of information flows temporal attributes for the efficient scheduling of dynamic demand responsive transport services* ; Journal of Advanced Transportation, vol. 40, n°1, pp.23-46, 2006.
- [6] R. Haesevoets, B. V. Eylen *et al.* : *Context-driven dynamic organizations applied to coordinated monitoring of traffic jams* ; in D. Weyns, S. Brueckner et Y. Demazeau, editors, EEMMS07, pages 126-143, 2007.
- [7] J. I. Hong et J. A. Landay : *An architecture for privacy-sensitive ubiquitous computing* ; in MobiSys04, pages 177-189, juin 2004.
- [8] D. Meignan, O. Simonin *et al.* : *Adaptive traffic control with reinforcement learning* ; in ATT06, pages 50-56, 2006.
- [9] J. Niittymki et M. Pursula : *Artificial intelligence on transportation systems and science* ; in European Journal of Operational Research, vol. 131, n° 2, pp. 229-458, 2001.
- [10] S. Ossowski, J. Z. Hernandez *et al.* : *Decision support for traffic management based on organisational and communicative multiagent abstractions* ; in Transportation Research part C, volume 13, n°4, pages 272-298, Elsevier, 2005.

- [11] A. Repenning et A. Ioannidou : *Mobility agents: guiding and tracking public transportation users* ; in AVI '06, New York, NY, États-Unis, 2006. ACM Press.
- [12] R. Schleiffer : *Intelligent agents in traffic and transportation* ; Transportation Research part C: emerging technologies, Elsevier Science Publishers B. V., volume 10, n° 5, p 34, octobre, 2002
- [13] M. M. Solomon : *Algorithms for the vehicle routing and scheduling with time window constraints* ; Operations Research, 15, pp. 254-265, 1987.
- [14] D. Weyns, H. V. D. Parunak et al. : *Environments for multiagent systems, state-of-the-art and research challenges* ; LNCS Series, 3374, pp. 2-52, 2005.
- [15] D. Weyns, N. Boucké et al. : *Gradient field-based task assignment in an agv transportation system* ; in AAMAS'06, pages 842-849, New York, NY, États-Unis, ACM, 2006
- [16] M. Zargayouna, F. Balbo et al. : *Agent information server: a middleware for traveler information* ; in ESAW'05, volume 3963 of LNAI, pages 3-16, Springer Verlag, 2005.
- [17] M. Zargayouna : *Modèle et langage de coordination pour les systèmes multi-agents ouverts. Application au problème du transport à la demande* ; Thèse soutenue à l'Université de Paris-Dauphine, Paris, 2007.

UML&FM'2008: FIRST INTERNATIONAL WORKSHOP ON UML&FORMAL METHODS

October 27, 2008, Kitakyushu-City, Japan

<http://www.artist-embedded.org/artist/UML-FM-08.html>

Workshop held in conjunction with ICFEM 2008, the 10th International Conference on Formal Engineering Methods (<http://www.icfem2008.org/>)

Many interest groups from a research perspective are in favour of the creation of this workshop. For more than a decade now, the two communities of UML and formal methods have been working together to produce a simultaneously practical (via UML) and rigorous (via formal methods) approach to software engineering.

UML is the de facto standard for modelling various aspects of software systems in both industry and academia, despite the inconvenience that its current specification is complex and its syntax imprecise. The fact that the UML semantics is too informal have led many researchers to formalise it with all kinds of existing formal languages, like OCL, Z, B, CSP, VDM, Petri Nets, UPPAAL, HOL, Coq, PVS, etc.

This first workshop will be open to various subjects, as the main objective is to encourage new initiatives of building bridges between informal, semi-formal, and formal notations. This workshop will present contributions from researchers and practitioners interested in all aspects of integrating UML and formal methods. The following principal topics will be discussed:

TOPICS

- Consistent specifications, model transformations (QVT technologies, transformation repositories).
- Transformations to make models more analysable so as to make them executable.
- Automation of traceability through transformations
- Refinement techniques: developing detailed design from a UML abstract specification
- Refinement of OCL specification as well formal reasoning on models for code generation
- Technologies for compositional verification of models
- Specification of a formal semantics for the UML giving an abstract syntax to UML diagrams
- Formal validation and verification of software
- Co-modelling methods formal/informal mapping techniques
- End-to-end methodologies or software process engineering, correct-by-construction design providing and supporting tools for safety-critical embedded systems design.

Workshop Format: This full-day workshop will consist of an introduction of the topic by the workshop organizers, presentations of accepted papers, and in depth discussion of previously identified subjects emerging from the submissions. A summary of the discussions will be made available.

Sponsors: OMG (<http://www.omg.org/>) and ARTIST (<http://www.artist-embedded.org/artist/>)
Organizers and Programme Steering committee:

Jean-Michel Bruel (*Liuppa, France*)
Agusti Canals (*CS, France*), General Chair

Sébastien Gérard (*CEA-LIST, France*)
Isabelle Perseil (*TELECOM ParisTech, France*)

IEEE CS Coordinator: Mike Hinchey (*NASA GSFC and Loyola College in Maryland, USA*)