

MODELISATION UML DES AGENTS DANS UN ENVIRONNEMENT DE SIMULATION 3D

Madeleine El-Zaher¹, David Trimoulet¹, Ewen Auffret¹, Soufiane Zerouali², M'hammed Sahnoun², D. Baudry², C. Viazzi¹

¹ LINEACT, Laboratoire du CESI, 16 rue Magellan, 31670 Labège.

² LINEACT, Laboratoire du CESI, 1 rue G. Marconi, 76130 Mont Saint Aignan.

***Abstract :** La simulation est devenue une étape obligatoire dans la conception des systèmes complexes, avant le passage aux étapes d'expérimentation et de production. Les simulateurs d'aujourd'hui intègrent l'ensemble des caractéristiques physiques ainsi que des modèles 3D des objets simulés. D'un point de vue informatique, les systèmes basés sur des agents sont de plus en plus utilisés. Ces systèmes, connus par leurs caractéristiques d'auto-organisation et de robustesse, sont actuellement utilisés dans la modélisation de divers systèmes complexes. Le but de ce papier est de présenter l'architecture d'un simulateur qui alliera un environnement graphique 3D avec un système basé agent. Les diagrammes UML représentant l'architecture de ce simulateur seront explicités dans ce papier. Le simulateur proposé sera utilisé pour la simulation d'un groupe de robots interagissant sur la chaîne de production dans une usine.*

***Mots-clés:** Agents, Modélisation, UML, Simulation.*

1. INTRODUCTION

Depuis quelques années, la simulation est devenue une étape obligatoire dans la conception des systèmes complexes, avant le passage aux étapes d'expérimentation puis de production. La simulation requiert un outil de simulation, qui permettra de représenter le comportement du système. Dans le but de produire les comportements-système les plus proches de la réalité, les simulateurs d'aujourd'hui intègrent souvent des modèles 3D des objets constituant le système simulé. Ils présentent également les caractéristiques physiques et les différents états/évolutions des objets. Les exemples des simulateurs sont nombreux : voitures autonomes[1], drones¹, industrie [2], etc.

D'un point de vue informatique, un changement de paradigme a eu lieu : on a évolué des solutions classiques dans lesquelles seule l'efficacité de calcul était importante vers des méthodes visant à donner des propriétés aux systèmes telles que la résilience, la capacité d'adaptation et/ou la sécurité des systèmes considérés. Dans cet esprit, les Systèmes Multi-Agents (SMA) ont été conçus. Selon Ferber[3], un SMA est un système composé d'un ensemble des agents situés dans un environnement et interagissant selon certaines relations. Un agent est une entité autonome qui peut représenter un processus, un robot, un être humain etc. Le comportement d'un agent est spécifié en fonction de ses perceptions et ses interactions. Les SMA montrent quelques caractéristiques telles que l'auto-organisation, la robustesse, et la simplicité d'utilisation. Par conséquent, les SMA ont été adoptés dans la modélisation des nombreux systèmes complexes pour résoudre un problème de localisation d'objets[4], pour modéliser un système de conduite en convoi[5], ou par exemple pour la simulation d'un trafic routier[6]. Pour simuler les comportements des agents dans leurs environnements, plusieurs plateformes Multi-agents ont été proposées dans la littérature [7], [8]. Le but

¹<http://www.realflight.com/drone/>

de ces plateformes est de donner une représentation visuelle des comportements des agents, leurs interactions et leurs évolutions dans le système.

L'industrie actuelle cherche des moyens pour optimiser sa production en s'adaptant le plus possible aux envies du client, on entend donc beaucoup parler de l'usine du futur. Le but d'une telle usine est d'amener plus d'agilité et de flexibilité à l'ensemble de la chaîne de production en augmentant son niveau d'automatisation. Dans ce cadre, nous cherchons à développer un système multi-robots afin d'introduire de l'automatisation à la chaîne de production. Le rôle des robots est de transporter, en coordonnant ensemble, des produits en confection d'une machine vers une autre dans la chaîne de production. Pour étudier ce système, nous cherchons à développer un simulateur permettant de simuler le comportement des robots. Le système multi-robots est modélisé comme un système multi-agent. Le simulateur alliera donc un environnement graphique 3D avec les SMA. Le papier discutera donc l'architecture du système proposé en illustrant des diagrammes UML capables de représenter graphiquement les deux parties du logiciel.

Dans la section 2, on présentera un état de l'art sur les différentes modélisations UML utilisées pour les systèmes basés agents. La section 3 présentera le diagramme UML que nous proposons pour représenter un agent dans le simulateur. La section 4 donnera l'architecture globale du simulateur. La section 5 présentera une implémentation possible de ce simulateur : usine du futur. Finalement, la section 6 conclura ce papier en évoquant les futurs travaux.

2. ETAT DE L'ART

Un SMA est souvent défini comme un ensemble des agents qui interagissent entre eux dans un environnement commun. Ferber[3] a défini deux types d'agents: les agents réactifs et les agents cognitifs. Les agents réactifs sont plutôt pulsionnels. En général, ils répondent à des stimuli de l'environnement. Leurs comportements sont guidés par l'état local de l'environnement dans lequel ils se trouvent. Les agents cognitifs sont capables de raisonner. Ils sont considérés comme des agents rationnels et leurs actions suivent un principe de rationalité par rapport aux buts qui les dirigent. Les agents cognitifs peuvent aussi négocier entre eux pour effectuer une tâche donnée. Plusieurs types d'algorithmes de négociation entre agents ont été proposés dans la littérature : négociations basées sur les théories de jeux[9], sur des heuristiques[10], [11], sur l'argumentation [12], etc.

Une étape indispensable dans le développement d'un logiciel est sa modélisation d'une façon graphique. L'UML (Unified Modeling Language) est l'un des langages de modélisation graphique les plus utilisés. Il permet de donner une description globale et générale du logiciel à développer[13]. Plus précisément, l'UML modélise des systèmes basés sur des concepts orientés objets. Il est basé sur des pictogrammes conçus pour fournir une méthode normalisée pour la visualisation de la conception d'un système plus ou moins complexe. Dans ce papier, nous nous intéressons à l'UML pour modéliser le simulateur basé sur un système orienté agent.

Des travaux de recherche se sont intéressés à la modélisation des systèmes orientés agents. Notamment, Bauer et al.[14] et Parunak et al. [15] proposent l'utilisation de l'AUML, une extension de l'UML pour des systèmes basés agents, permettant de spécifier le rôle de l'agent ainsi que les protocoles d'interaction entre les agents. Les travaux présentés par Da Silva et al.[16] proposent l'utilisation de MAS-ML, qui étend les méta-classes utilisés en UML pour proposer des diagrammes de rôle et d'organisation. Les travaux cités proposent toujours une extension de l'UML pour modéliser le protocole global des SMA et les interactions entre les agents. Cependant, elles ne proposent pas de solution quant à la modélisation d'un agent et ses différents composants logiciels.

Le but de ce papier est de proposer donc une modélisation UML des agents en tant que composant d'un logiciel. Cette modélisation sera ensuite intégrée dans l'architecture globale

simulateur. Le simulateur à développer comprend une représentation 3D de l'environnement, basée sur le moteur de jeu UNITY 3D².

3. MODELISATION UML D'UN AGENT LOGICIEL

Selon Hayes-Roth[17], un agent intelligent possède trois fonctionnalités qu'il exécute d'une façon continue: 1- la perception des conditions dynamiques dans l'environnement, 2- le raisonnement pour interpréter les perceptions, résoudre les problèmes et déterminer les actions et 3- l'action pour affecter les conditions de l'environnement.

Pour modéliser un agent cognitif, nous proposons un diagramme qui développe les trois fonctionnalités citées, et qui détaille les différents éléments qui y interviennent. LaFigure 1 montre la modélisation proposée pour un agent cognitif :

- la perception est assurée par des capteurs qui reçoivent et normalisent les données, un *listener* qui filtre les informations reçues, et un déclencheur qui informe le raisonneur ;
- le raisonnement est assuré par le raisonneur qui produit une liste d'actions que l'agent peut effectuer pour répondre à l'évènement en question. Le décideur choisit l'ensemble des actions à effectuer en prenant en considération différentes contraintes (urgences, sécurités, ...). Finalement, le planificateur ordonnance l'ordre d'exécution des actions choisies. Ces trois composants s'appuient sur les connaissances de bases de l'agent pour effectuer leurs tâches ;
- l'action est assurée par un exécuteur et l'ensemble des actuateurs associés à l'agent.

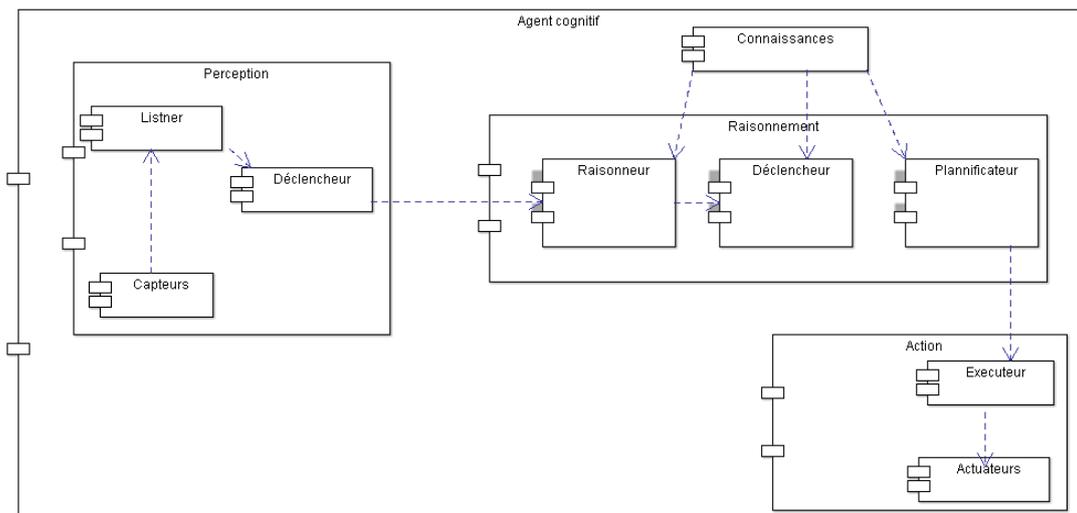


Figure 1: modélisation UML d'un agent cognitif.

Un agent réactif répond à un stimulus de l'environnement. En général, un agent réactif n'a pas de connaissances, son raisonnement se limite à exécuter une action définie et associée à un stimulus donné. LaFigure 2montre le diagramme associé à un agent réactif. Dans ce diagramme, seulement les fonctionnalités de perception et d'action sont présentes. La fonction de raisonnement est omise.

²<https://unity3d.com>

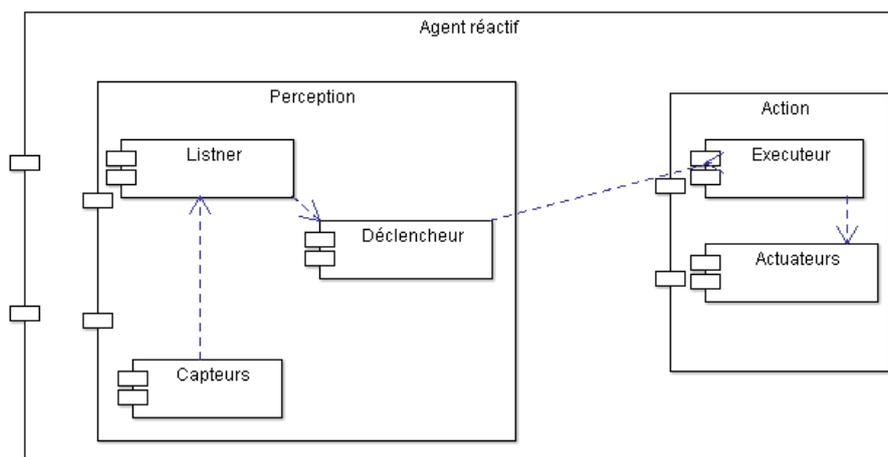


Figure 2: modélisation UML d'un agent réactif.

4. ARCHITECTURE GLOBALE DU SIMULATEUR

Le simulateur proposé offre une représentation 3D de l'environnement et des objets qui y interviennent. Un agent du SMA sera associé à une entité physique dans le simulateur représentée avec ses caractéristiques. Chaque agent est donc associé à un « corps-physique » représenté dans l'environnement. L'ensemble des caractéristiques associées au corps-physique diffèrent selon l'application. Dans le cas de l'usine du futur, les caractéristiques associées au corps physique sont : la position, les dimensions, la charge maximale, la vitesse maximale, l'accélération maximale, etc. Le corps physique est équipé de l'ensemble des capteurs et actuateurs associés à l'agent. Un agent perçoit et réagit dans l'environnement à travers son corps physique. Un agent ne peut pas changer directement l'état de l'environnement.

L'environnement est constitué de trois entités principales : les entités percevables, les entités influençables et les entités influenceurs. Les entités percevables sont les objets qui peuvent être perçus par l'agent. Les entités influençables sont les objets sur lesquels un agent peut influencer en changeant leurs états. Les entités influenceurs sont les objets de l'environnement qui peuvent influencer les décisions d'un agent ou modifier son état.

La

Figure 3 propose le diagramme de classe de l'environnement du simulateur. Deux types d'agents sont présents : cognitif et réactif. Une relation d'agrégation est proposée entre le Corps-physique et l'agent. Un agent fait donc partie du corps-physique. Par contre, si l'agent n'existera plus, le corps-physique continuera à exister.

Un environnement est composé des entités. Ces entités peuvent être mobiles. Les entités peuvent être classées en : percevable, influençable et influenceur.

Dans ce diagramme, les entités *Percevable*, *Influénçable* et *Influenceur* sont représentées par des Interfaces. L'interface *Percevable* est implémentée par toutes les entités qui peuvent être perçues par l'agent. L'interface *Influénçable* est implémentée par toutes les entités qui peuvent être influencées par l'agent. L'interface *Influenceur* est implémentée par toutes les entités qui peuvent influencer l'agent.

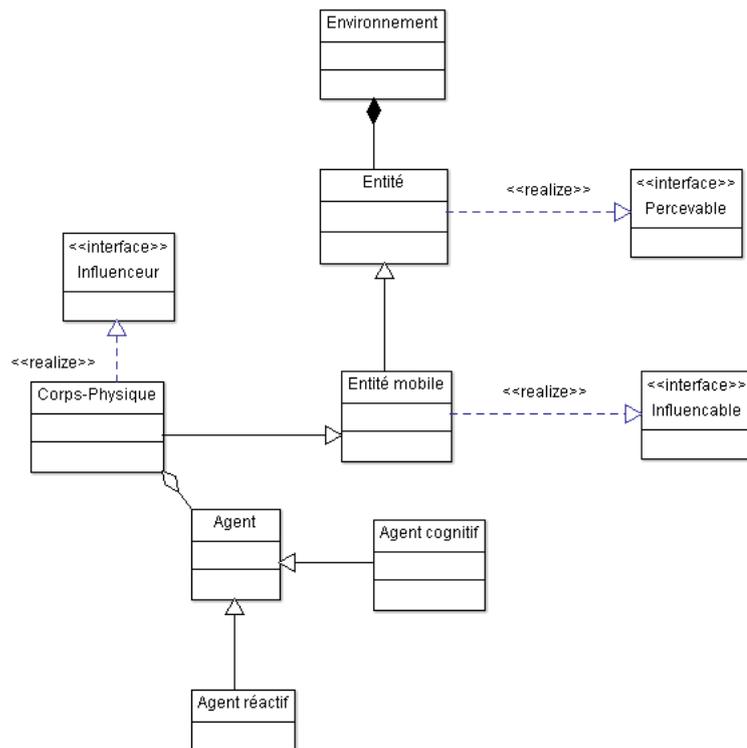


Figure 3: Diagramme de classe UML simplifié pour l'architecture globale du système.

5. APPLICATION A UN SIMULATEUR USINE DU FUTUR

Le simulateur développé est dédié à une application dans les usines de futur. L'idée est de pouvoir simuler le comportement d'un ensemble de robots évoluant dans une usine. Ces robots ont pour rôle principal de transporter les produits en confection d'une machine à l'autre.

L'environnement est donc constitué de l'ensemble des machines dans la chaîne de production, l'ensemble des produits en cours de confections et l'ensemble des robots de transports.

L'ensemble des robots est modélisé en utilisant le paradigme des SMA. Chaque robot est considéré comme un agent capable de percevoir son environnement et de réagir en fonction de ses perceptions. Cinq fonctionnalités sont associées à un agent :

- il perçoit les produits et détermine ceux à transporter ;
- il négocie avec les autres robots pour décider qui fera le transport. Pour la négociation, des algorithmes de négociations entre agents seront utilisés ;
- il se déplace dans l'environnement ;
- il évite les obstacles ;
- il transporte les produits.

Les agents considérés pour représenter les robots sont des agents cognitifs. Ils possèdent des capacités de négociation et de raisonnement. Ils possèdent aussi des connaissances : la carte de l'environnement dans lequel ils évolueront.

Les capteurs intégrés au robot doivent lui permettre : d'observer l'environnement (obstacles fixes et mobiles), de communiquer avec les robots et avec les produits en confection. Les actuateurs associés doivent lui permettre de se déplacer, d'envoyer des messages et de porter des objets.

Le « corps-physique » du robot implémente donc les attributs suivants : position, poids, charge maximale, vitesse maximale, accélération maximale et dimensions.

L'interface *Percevable* est implémentée par les produits, les robots, ainsi que les objets fixes de l'environnement. L'interface *Influencable* est implémentée par les produits et les robots. L'interface *Influenceur* est implémentée par les obstacles de l'environnement et les robots.

Unity 3D est utilisé pour la modélisation 3D de l'environnement et des objets (robots, machines, produits,...).

6. CONCLUSION

Dans ce papier, nous avons proposé l'architecture globale d'un simulateur, conçu pour être utilisé dans le domaine de l'industrie de futur. Le simulateur proposé est une combinaison entre un moteur physique capable de présenter les caractéristiques physiques des objets dans l'environnement, un modèle 3D dupliquant l'environnement réel de simulation et un moteur d'intelligence artificielle capable d'introduire une modélisation à base d'agent au simulateur.

Le diagramme UML représentant un agent dans ce simulateur est présenté dans ce papier ainsi que l'architecture globale du simulateur.

Les prochains travaux seront dédiés à l'implémentation de ce simulateur. Unity 3D sera utilisé pour le rendu 3D. Le simulateur sera utilisé pour simuler un ensemble des robots conçus pour être utilisés dans une usine du futur. Le rôle des robots étant de transporter les produits d'une machine à une autre dans la chaîne de production, tout en évitant les obstacles et en coordonnant leurs mouvements.

7. BIBLIOGRAPHIE

- [1] O. Lamotte, S. Galland, J.-M. Contet et F. Gechter, «Submicroscopic and physics simulation of autonomous and intelligent vehicles in virtual reality,» *Advances in System Simulation (SIMUL)*, pp. 28-33, 2010.
- [2] S. A. Harp, S. Brignone, B. F. Wollenberg et T. Samad, «SEPIA. A simulator for electric power industry agents,» *IEEE Control Systems*, vol. 20, n° 14, pp. 53-69, 2000.
- [3] J. Ferber, *Multi-agent systems: an introduction to distributed artificial intelligence*, Addison-Wesley Reading, 1999.
- [4] S. Moujahed, S. Olivier et K. Abderrafiaa, «Location problems optimization by a self-organizing multiagent approach,» *Multiagent and Grid Systems*, vol. 5, n° 11, pp. 59--74, 2009.
- [5] M. El-Zaher, F. Gechter, P. Gruer et M. Hajjar, « New Linear Platoon Model Based on Reactive Multi-agent Systems,» *{IEEE} 23rd International Conference on Tools with Artificial Intelligence*, pp. 898--899, 2011.
- [6] S. E. hadouaj et A. Drogoul, «A study of coordination within a road traffic environment,» *Intelligent Agent Technology*, pp. 491-495, 2004.
- [7] F. L. Bellifemine, G. Caire et D. Greenwood, «Developing multi-agent systems with JADE,» *John Wiley & Sons*, vol. 7, 2007.
- [8] O. Gutknecht et J. Ferber, «Madkit: a generic multi-agent platform,» *Proceedings of the fourth international conference on Autonomous agents*, pp. 78--79, 2000.
- [9] A. J. Trappey, C. V. Trappey et W. C. Ni, «A multi-agent collaborative maintenance platform applying game theory negotiation strategies,» *Journal of Intelligent Manufacturing*, pp. 613-623, 2013.
- [10] S. Costantini, G. De Gasperis, A. Provetto et P. Tsintza, «A heuristic approach to proposal-based negotiation: with applications in fashion supply chain management,» *Mathematical Problems in*

Engineering, 2013.

- [11] L. Zhang, T. N. Wong et R. Y. Fung, «A multi-agent system to support heuristic-based dynamic manufacturing rescheduling,» *Intelligent Decision Technologies*, vol. 7, n° 13, pp. 197-211, 2013.
- [12] L. Amgoud et H. Prade, «Using arguments for making decisions: A possibilistic logic approach,» *In Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 10-17, 2004.
- [13] G. Booch, *The unified modeling language user guide*, Pearson Education India, 2005.
- [14] B. Bauer, «UML class diagrams revisited in the context of agent-based systems,» *Agent-Oriented Software Engineering II*, Springer, pp. 101-118, 2001.
- [15] H. V. D. Parunak et J. J. Odell, «Representing social structures in UML,» *Agent-Oriented Software Engineering II*, Springer, pp. 1-16, 2001.
- [16] V. T. Da Silva, R. Choren et C. J. De Lucena, «A UML based approach for modeling and implementing multi-agent systems,» *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 2, pp. 914-921, 2004.
- [17] F. Hayes-Roth, *Towards a framework for distributed AI*, Sigart Newsletter, 1980.

Contact principal: Madeleine EL-ZAHER

Coordonnées : melzaher@cesi.fr