

# Un méta-modèle pour les applications basées sur les agents mobiles

Tahar Gherbi  
Université Bretagne-Sud  
Vannes, France  
tahar.gherbi@univ-ubs.fr

Isabelle Borne  
Université Bretagne-Sud  
Vannes, France  
isabelle.borne@univ-ubs.fr

Djamel Meslati  
Université d'Annaba  
Annaba, Algérie  
meslati\_djamel@yahoo.com

## Abstract

Mobile agents are a crucial part in most systems based on agents. Many works have focused on the modeling of multi-agent systems (MAS) without addressing the issue of mobility, others were interested in modeling the mobility of agents outside of any MAS modeling methodology, but rare are the works that have been interested, for the purpose of code generation, in modeling MAS with support of mobility.

This paper proposes a meta-model for the analysis and the design of SMA with support of mobility. We extend the meta-model proposed in MDAD (Model Driven Agent Development) [JG07] to take into account the specific aspects of agents' mobility: the agent execution site and the migration operation. The resulting meta-model will be used in our future work which consist to propose an MDE (Model Driven Engineering) approach for developing mobile agents applications.

**Keywords:** Multi-agent systems, Mobile Agent, Model-driven engineering

## Résumé

Les agents mobiles représentent une partie cruciale dans la plupart des systèmes basés sur les agents. Beaucoup de travaux se sont intéressés à la modélisation des systèmes multi-agents (SMA) sans traiter la question de la mobilité; d'autres se sont intéressés à la modélisation de la mobilité des agents en dehors de toute méthodologie de modélisation de SMA; mais rares sont les travaux qui se sont intéressés, dans un but de génération de code, à la modélisation des SMA avec la prise en compte de la mobilité.

Ce papier propose un méta-modèle pour l'analyse et la conception de SMA supportant la mobilité. Nous étendons le méta-modèle proposé dans MDAD (Model Driven Agent Development) [JG07] pour prendre en compte les aspects spécifiques à la mobilité des agents: le site d'exécution de l'agent et l'opération de migration. Le méta-modèle résultant sera utilisé dans notre prochain travail consistant à proposer une démarche IDM (Ingénierie Des Modèles) pour développer des applications d'agents mobiles.

**Mots clés:** Systèmes multi-agents, agent mobile, ingénierie des modèles

## 1 Introduction

La technologie des agents mobiles est un paradigme de programmation des applications réparties, susceptible de compléter ou de se substituer à d'autres paradigmes plus classiques tels le passage de messages, l'appel de procédure à distance, l'invocation d'objet à distance et l'évaluation à distance. Elle est d'un grand intérêt pour la mise en œuvre des applications dont les performances varient en fonction de la disponibilité et de la qualité des services et des ressources, du volume des données déplacées; ou tournant sur des réseaux ad hoc [Cro05]. Le développement des applications d'agents mobiles consiste habituellement à mettre en œuvre une plateforme d'exécution supportant la mobilité et à implémenter ensuite des applications,

souvent sans considérer en amont le concept de mobilité dans les phases d'analyse et de conception. La mobilité est traitée dans la phase d'implémentation. En faisant remonter le concept de mobilité vers les phases d'analyse et de conception, les analystes et concepteurs d'applications d'agents mobiles peuvent traiter ce concept en amont; ce qui leur permettra de mieux concevoir ce type d'applications.

Ce papier propose un méta-modèle pour la conception de SMA avec la prise en compte de la mobilité des agents dans la phase de conception. La section 2 présente notre méta-modèle. La section 3 illustre son usage en modélisant un exemple d'application simplifiée. La section 4 discute les travaux rencontrés dans la littérature, qui traitent la modélisation de la mobilité des agents. La section 5 conclut le papier et présente les perspectives de ce travail.

## 2 Méta-modèle pour les SMA avec support de la mobilité

Vu l'absence de consensus sur la définition d'agent, sur un méta-modèle de SMA et sur une méthodologie de développement de SMA, nous adoptons le méta-modèle proposé dans MDAD (Model Driven Agent Development) [JG07], que nous enrichissons et étendons pour supporter les aspects liés à la mobilité des agents, à savoir: le site d'exécution de l'agent et l'opération de migration[BKKW03], [KWT04], [BMC09]. L'opération de migration peut consister à transférer l'agent lui-même ou alors à cloner l'agent et à transférer son clone. MDAD est une approche d'IDM pour développer des SMA organisationnels conçus selon la décomposition AEIO (Agent, Environnement, Interaction, Organisation) de l'approche Voyelles [Dem95] et adoptant le méta-modèle organisationnel AGR (Agent, Groupe, Rôle) de l'approche Aalaadin [FG98].

La figure 1 présente l'ensemble de notre méta-modèle pour lequel nous allons expliciter les principales extensions que nous avons réalisées pour modéliser la mobilité des agents.

Nous avons commencé par compléter le méta-modèle original avec des stéréotypes permettant de distinguer les actions d'appartenance à un groupe: "JoinGroup" et "LeaveGroup". Nous avons également ajouté deux associations entre les classes stéréotypées "SendMessage" et "ReceiveMessage" pour permettre à l'agent de savoir, à la réception d'un message, s'il s'agit d'une réponse à une requête qu'il a envoyé précédemment à l'émetteur (présence de l'attribut `itsPreviousSend`) ou s'il s'agit d'une nouvelle requête de l'agent émetteur (absence de l'attribut `itsPreviousSend`); et par ailleurs de savoir, à l'émission d'un message, s'il est en train de répondre à une requête reçue précédemment de l'agent destinataire (présence de l'attribut `itsPreviousReceive`) ou s'il est en train d'envoyer une nouvelle requête vers l'agent destinataire (absence de l'attribut `itsPreviousReceive`). Cette information conservée grâce aux deux associations ajoutées, sera vitale pour la génération automatique de code qui n'est pas abordée ici. Notons également que nous avons utilisé des diagrammes d'états/transitions pour décrire les comportements des agents et des rôles, ceci pour faciliter par la suite la génération automatique du code. Finalement, le but que réalise l'agent ou le rôle est représenté implicitement à travers la tâche que remplit son comportement.

Tous les agents sont susceptibles de migrer. Pour supporter la mobilité nous avons ajouté les stéréotypes "Jump" (pour modéliser le déplacement physique de l'agent vers un autre site), "Clone" (pour modéliser le clonage de l'agent, dans le cas d'une migration par clonage), "Site" (pour modéliser le site où veut migrer l'agent), "Migration" (pour modéliser le traitement à faire, quand un agent demande sa migration, afin d'assurer une migration saine) et "AfterMigration" (pour modéliser le traitement à déclencher automatiquement, après chaque migration d'un agent, afin d'assurer sa bonne intégration dans le SMA). Le traitement déclenché par l'action stéréotypée "Migration" emploie forcément l'action stéréotypée "Jump" et éventuellement (selon la valeur taguée `{clone=true/false}` que nous avons rajouté) l'action

stéréotypée "Clone". De plus, le concepteur du SMA peut indiquer les classes du domaine susceptibles d'être transférées avec l'agent, grâce à la valeur taguée {transferable=true/false} rajoutée au niveau des classes stéréotypées "DomainConcept".

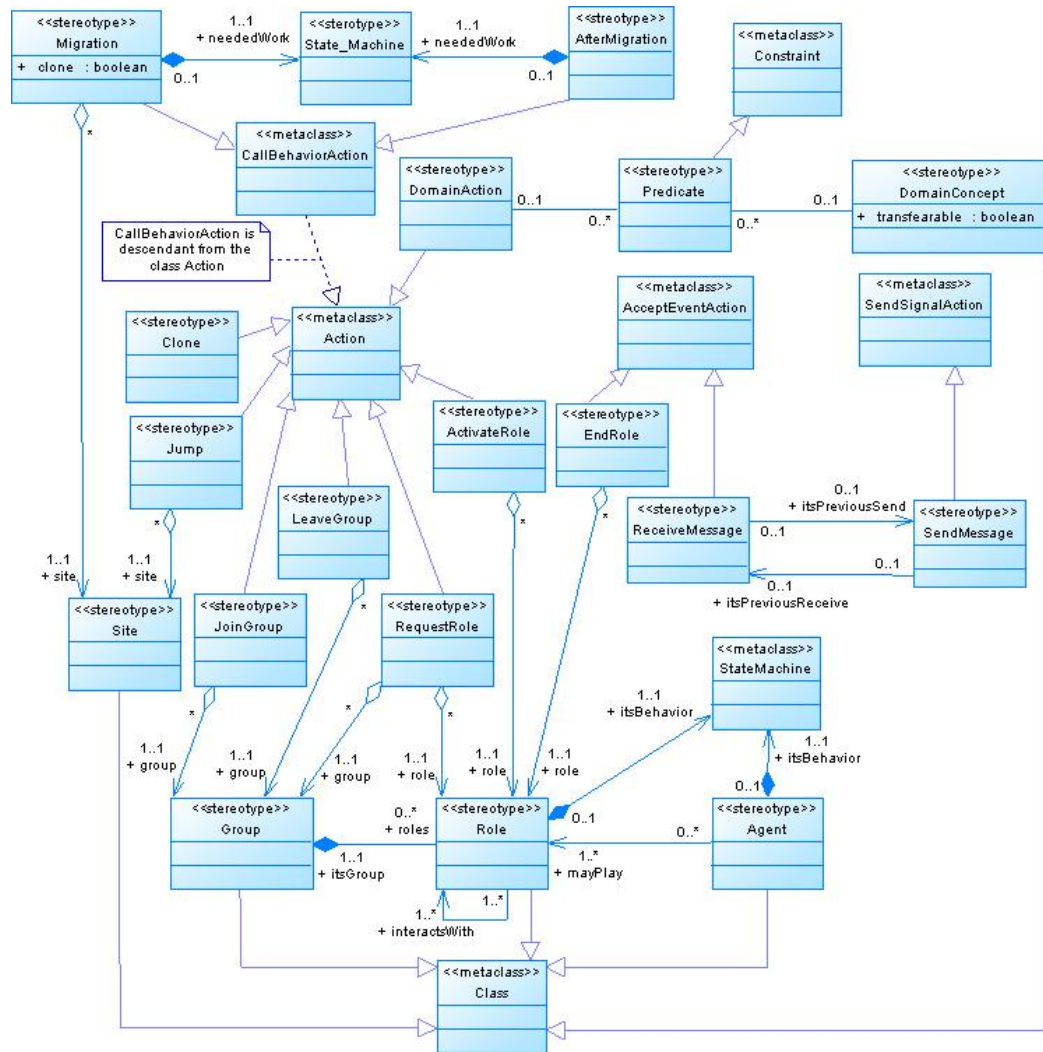


Figure 1: Méta-modèle pour les SMA avec support de la mobilité

### 3 Exemple d'application

Supposons une application répartie tournant dans des conditions encourageant la mobilité (voir le 1er paragraphe de l'introduction). Pour illustrer l'utilisation de notre méta-modèle, nous avons choisi un exemple d'application simplifiée de collecte de prix faisant intervenir trois sites. Sur chaque site s'exécute un agent fixe (fournisseur ou *Supplier*) qui délivre le prix d'un produit (grâce au rôle *PriceDeliver*). Les trois fournisseurs fournissent un seul produit: ordinateur par

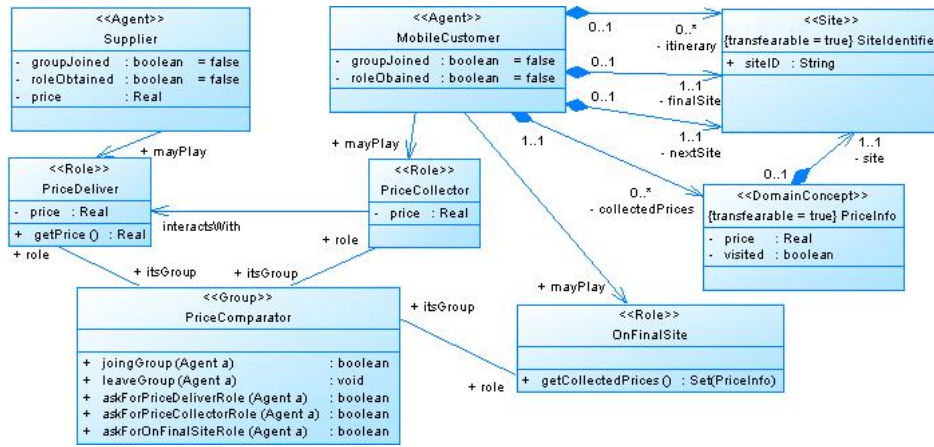


Figure 2: Digrammes de classes pour un exemple d'application de collecte de prix

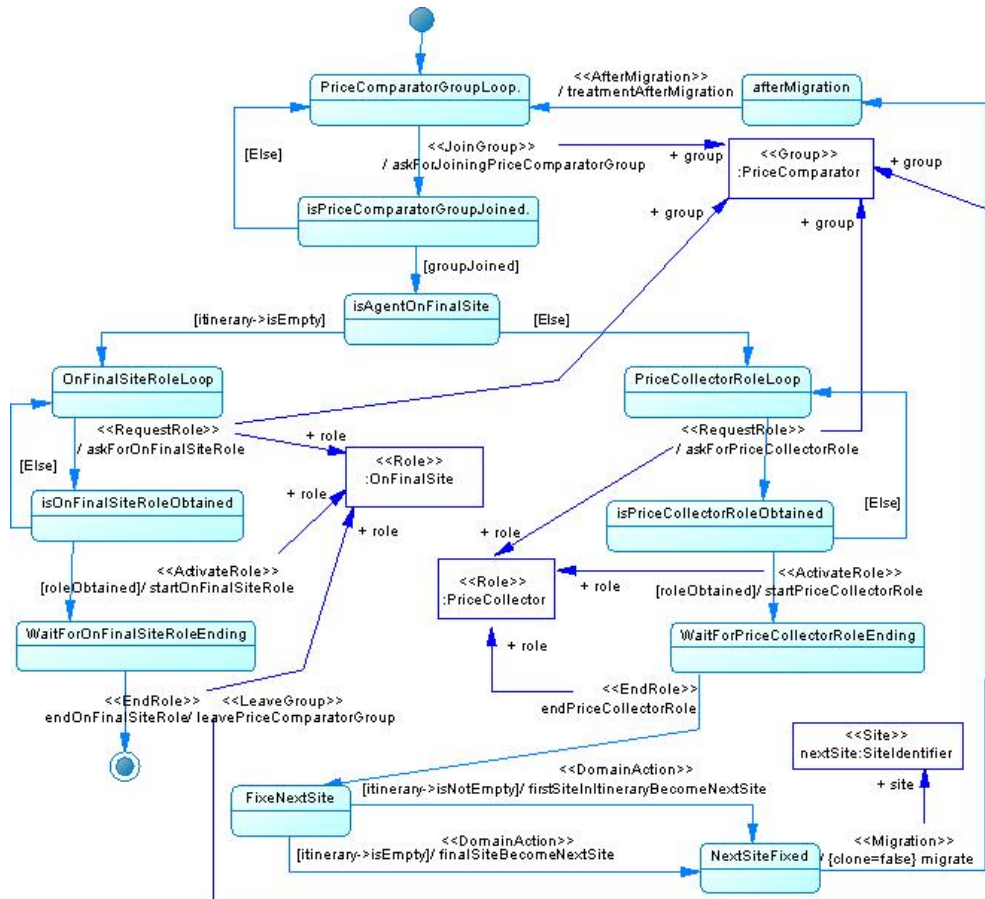


Figure 3: Comportement de l'agent MobileCustomer

exemple. Un agent mobile (client mobile ou Mobile Customer) est lancé et nous lui indiquons l'itinéraire des sites à visiter et le site final sur lequel il communiquera les résultats. L'agent fait le tour des trois fournisseurs, récupère de chacun le prix de vente de l'ordinateur (grâce au rôle *PriceCollector*), se déplace ensuite sur le site *FinalSite* où il peut communiquer les prix collectés (grâce au rôle *OnFinalSite*), après quoi son activité est terminée. La figure 2 montre le diagramme de classes modélisant cette application. Dans cette figure, les classes Agent et Role sont normalement reliées, via leurs attributs *itsBehavior*, à leurs comportements (qui ne sont pas présentés pour alléger le schéma). Par manque d'espace, nous présentons seulement le comportement de l'agent *MobileCustomer* (Figure 3). Dans cet exemple simple, l'action *migrate* supprime seulement l'agent du groupe et l'action *treatmentAfterMigration* ne fait rien.

## 4 Travaux existants

La modélisation de la mobilité des agents a été étudiée dans plusieurs travaux qui proposent des extensions des diagrammes du langage UML2.0.

Dans [Fip03], la mobilité est modélisée à travers les diagrammes d'activité (pour spécifier quand l'agent migre) et de déploiement (pour spécifier pourquoi l'agent migre et où il migre). Ces diagrammes font parties du langage AUML(<http://www.auml.org/>). Les diagrammes de déploiement décrivent statiquement la mobilité des agents, alors que les diagrammes d'activité décrivent leurs comportements.

Dans [BKKW03], le diagramme d'activités est étendu pour représenter les concepts de "localisation", d'"objet mobile", de "localisation mobile", de l'action "move" (pour déplacer un agent) et de l'action "clone" (pour cloner un agent). Deux notations de modélisation sont présentées : une notation centrée sur l'agent qui exécute l'action "move" ou "clone" et une notation centrée sur la localisation (le site où est exécutée l'action "move" ou "clone"). Dans [KWT04] la mobilité est modélisée avec des diagrammes d'activités également, mais un nouveau stéréotype est introduit "host"+paramètre pour représenter la localisation actuelle de l'agent. La migration d'un agent d'un site "host1" vers un site "host2" s'effectue en utilisant l'action "Go" qui est l'action classique utilisée pour lancer la migration dans les systèmes d'agents mobiles.

Dans [KJ06], on étend les diagrammes de séquences pour élaborer quatre variantes de notation: un diagramme de mobilité stéréotypé, un diagramme de mobilité utilisant des partitions, un diagramme de mobilité représentant les états et un diagramme de mobilité avec des fragments délimités. Ces quatre variantes permettent de modéliser le cheminement des déplacements de l'agent. Les deux premiers diagrammes ne sont pas pratiques quand le nombre de sites devient important. Le troisième diagramme ne fait pas apparaître l'action de migration (vitale pour nous). Le dernier diagramme devient illisible et peu pratique, si on modélise des agents qui font des déplacements et qui reviennent après chaque déplacement à leurs sites d'origine.

Dans [BMC09], plusieurs extensions sont apportées aux diagrammes d'UML (cas d'utilisation, séquence, classes, états/transitions, activité et déploiement) pour permettre la modélisation des systèmes d'agents mobiles en prenant en compte les concepts de "migration", "clonage" et "localisation" de l'agent. Dans [HA11], le diagramme d'états/transitions d'UML est exploité pour modéliser le cheminement des déplacements de l'agent en utilisant deux tableaux: un tableau pour les déplacements de l'agent et un autre pour la création de nouveaux agents.

La différence entre ces travaux est que les propositions faites dans [KJ06] et [HA11] sont adéquates pour la modélisation du cheminement des déplacements de l'agent; par contre, les propositions faites dans : [BMC09], [KWT04], [BKKW03] et [Fip03] sont de bonnes sources d'inspiration pour modéliser le comportement interne d'un agent mobile. Le point commun est que ces travaux modélisent tous le concept de la mobilité de manière générale en dehors de

toute méthodologie de développement de SMA; par conséquent, ils ne traitent pas la question du nettoyage à faire au niveau du SMA quand un agent quitte un site donné et l'effort à faire pour intégrer correctement l'agent dans le SMA sur le site de destination.

Le méta-modèle que nous proposons répond à cette question et supporte la mobilité grâce à une notation intégrée dans un méta-modèle de conception de SMA. Cette notation ressemble à celle de [BKKW03]. La différence est que nous associons comme information à l'action stéréotypée "Jump" uniquement le site de destination. Aussi, nous nous intéressons à modéliser seulement la mobilité des agents et non la mobilité éventuelle des unités physiques qui les hébergent (concept de localisation mobile).

## 5 Conclusion et perspectives

Dans ce papier nous avons proposé un méta-modèle qui permet de modéliser des SMA organisationnels tout en supportant le concept de mobilité dans la phase de conception. Une premier exemple a permis de montrer son utilisation et devra être étendu pour illustrer tous les aspects d'une modélisation d'un système à agents mobiles. Notre objectif est d'offrir au concepteur d'application à agents mobiles une méthodologie reposant sur l'ingénierie des modèles en lui permettant de gérer en amont la mobilité. Les prochaines étapes de notre travail sont de formaliser une démarche méthodologique et de réaliser la génération de code sur des plates-formes supportant la mobilité.

## References

- [BKKW03] H. Baumeister, N. Koch, Kosiuczenko, and M. Wirsing. Extending activity diagrams to model mobile systems. In *Proceedings of NODe '02 Revised Papers from the International Conference NetObjectDays on Objects, Components, Architectures, Services, and Applications for a Networked World*, 2003.
- [BMC09] M. R. Bahri, R. Mokhtari, and A. Chaoui. Towards an extension of uml2.0 to model mobile agent-based systems. *International Journal of Computer Science and Network Security*, 9(10), 2009.
- [Cro05] C. Cubat Dit Cros. *Agents mobiles coopérants pour les environnements dynamiques*. PhD thesis, Département d'informatique, INP de Toulouse, France, 2005.
- [Dem95] Y. Demazeau. From cognitive interactions to collective behaviour in agent-based systems. *1st European Conference on Cognitive Science*, 1995.
- [FG98] J. Ferber and O. Gutknecht. Alaadin : a meta-model for the analysis and design of organizations in mutli-agent systems. *ICMAS'98*, 1998.
- [Fip03] Fipa modeling tc. fipa modeling area: Deployment and mobility. 2003.
- [HAS11] F. A. Hanandeh and M. Y. Al-Shannag. Extending uml state diagrams to model agentmobility. *Proc. of Int. Conf. on Advances in Information and Communication Technologies*, 2011.
- [JG07] T. Jarraya and Z. Guessoum. Towards a model driven process for multi-agent system. *Multi-agent systems and applications V. 5th international CEEMAS*, 2007.
- [KJ06] M. Kusek and G. Jezic. Modeling agent mobility with uml sequence diagram. *AOSE*, 2006.
- [KWT04] M. Kang, L. Wang, and K. Taguchi. Modelling mobile agent applications in uml2.0 activity diagram. 2004.