



HAL
open science

Intégration d'un agent mobile dans un Système multi-agent hiérarchique

Maghri Dorra, Amara Nejla

► **To cite this version:**

Maghri Dorra, Amara Nejla. Intégration d'un agent mobile dans un Système multi-agent hiérarchique. MajecSTIC 2005: Manifestation des Jeunes Chercheurs francophones dans les domaines des STIC, IRISA – IETR – LTSI, Nov 2005, Rennes, France. pp.426-429. inria-00000831

HAL Id: inria-00000831

<https://inria.hal.science/inria-00000831>

Submitted on 22 Nov 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Intégration d'un agent mobile dans un Système multi-agent hiérarchique

Maghri D¹, Amara N²

¹Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75776, Paris Cedex 16

²LIPN, Université Paris 13, Institut Galilée, 99Avenue J.-B. Clément, 93430 Villetaneuse

¹maghri.dorra@lipn.univ-paris13.fr, ²na@lipn.univ-paris13.fr

Résumé : Dans cet article, nous présentons une démarche pour l'intégration d'un agent mobile à base de composants dans un Système Multi-Agents (SMA) à structure hiérarchique. La solution proposée permet une intégration « ouverte » qui lève la contrainte de structure commune entre l'agent et les agents du SMA qui va l'accueillir. Pour ce faire, nous proposons, en premier lieu, la mise en place d'un nouveau composant qui reflète d'un côté l'agent et la structure particulière du SMA et en second lieu, des primitives que l'agent devra exécuter après la migration.

Mots-clés: agent mobile, système multi-agents hiérarchique, intégration.

1 INTRODUCTION

De nos jours, les agents mobiles acquièrent de plus en plus d'importance vu leur capacité à migrer d'une manière autonome à travers un réseau dans le but d'avoir accès aux ressources et services distants.

Dans la littérature, plusieurs architectures d'agents mobiles ont été proposées. Dans ce travail, nous nous intéressons aux agents mobiles GAMA (Generic Adaptive Mobile Agents) proposés par [Amara 04].

Ces agents ont l'avantage d'avoir une architecture générique à base de composants leur permettant de couvrir plusieurs domaines d'application et d'être adaptables.

Dans ce travail, nous traitons le problème de l'intégration de cet agent GAMA dans un système multi-agents à structure hiérarchique.

L'article se présente comme suit. La section suivante présente le cadre du travail. Elle inclut la définition de l'agent mobile, l'agent GAMA et de la structure hiérarchique. La section d'après présente la méthode proposée. Elle inclut les primitives dégagées et un exemple illustratif. Et nous finissons par une section discussion et conclusion qui appuient le bien-fondé de la solution proposée.

2 CADRE DU TRAVAIL

Ce travail met en place une démarche qui va permettre l'intégration d'un agent mobile à base de composants dans une structure multi-agent hiérarchique.

Un agent mobile est agent [Ferber, 95] capable de migrer d'une manière autonome d'une machine à une autre à travers un réseau. C'est un programme qui représente l'utilisateur et qui se charge d'effectuer certains traitements pour lui. Il se déplace d'un noeud à un autre du réseau et une fois son traitement terminé, il revient à la machine d'origine pour retourner les résultats trouvés. L'utilisation des agents mobiles est très avantageuse et son apport principal est la réduction considérable de la communication dans le réseau.

L'agent mobile utilisé dans cette solution est à base de composants, il s'agit de l'agent GAMA pour Agent Mobile Générique et adaptable, agent proposé dans le cadre de travail de thèse de [Amara, 05]. Cet agent comme son nom l'indique se forme de plusieurs composants (communication, action, etc.) regroupés autour d'un contrôleur qui représente un médiateur entre les différents composants.

Grâce à cette architecture particulière, l'agent proposé par [Amara, 05] acquiert d'une part la caractéristique de généricité et d'autre part celle d'adaptation. En effet, la généricité découle du fait que cet agent peut être utilisé dans n'importe quel contexte. Quant à l'adaptation, elle peut être mise en œuvre grâce au remplacement des composants inadaptes par d'autres plus adéquats.

Dans le cadre de ce travail, l'agent GAMA s'introduit dans un Système Multi-Agents (SMA) hiérarchique lors de sa migration. Un SMA hiérarchique est un SMA [Ferber, 95] dans lequel les relations entre les agents sont organisées selon une hiérarchie. Ce type de système possède un Agent racine à partir duquel se prolonge une arborescence d'agents constituée selon un critère bien déterminé.

3 METHODE

Le point clé de la solution proposée est la construction d'un nouveau composant appelé "intégration" qui présentera l'agent en tant qu'un ensemble de connaissances, de compétences, de processus [El Fallah, 05] et de fils. L'agent possède une adresse qui se base sur la structure hiérarchique du SMA cible et se compose du niveau et du rang d'un agent dans la hiérarchie. Cette solution permet de représenter l'agent d'une manière abstraite et indépendante de son architecture. Ce nouveau composant est facile à constituer vu qu'on peut avoir ces données à partir des différents composants de l'agent mobile. La base des connaissances et la base des compétences seront alimentées à partir du composant "Interface" [Amara 04] qui fournit entre autres les fonctions de l'agent et la base des processus le sera à partir du composant "Action" qui contient les différentes tâches de l'agents, quant à la base des fils et la variable niveau, elles seront vides.

Grâce à ce nouveau composant l'agent mobile pourra s'intégrer dans la structure hiérarchique et ne fera pas appel à la communication pour exécuter ses tâches.

Il faut aussi préciser que dans le cadre de ce travail, le système cible est organisé conformément au standard MASIF d'OMG [Milojicic, 98]. MASIF est une collection de définitions et d'interfaces permettant une interopérabilité pour les SMA mobiles. Ce système présente deux interfaces : le MAFFinder et le

MAFAgentSystem. Tout agent de la plateforme est enregistré dans le registre du MAFAgentSystem y compris les agents mobiles qui arrivent sur la plateforme, donc on peut savoir à un moment donné quels agents sont sur la plateforme. Le MAFFinder, quant à lui, connaît les compétences, les services offerts par les différents agents se trouvant dans le système et l'emplacement exacte de chaque agent sachant que le domaine de la plateforme est organisée en places.

Afin de mettre en œuvre son intégration, l'agent mobile doit exécuter les primitives suivantes.

3.1 Primitives

Pour effectuer cette intégration, on doit disposer de plusieurs primitives (Move, Detach, Depth, Width, Attach et go) qui sont spécifiées ci-après grâce à l'établissement de leurs descriptions, pré-conditions et post-conditions.

Soient X un agent dans une structure hiérarchique avec $X(K^X, C^X, A^X, S^X, L^X, R^X)$ et M un agent mobile avec $M(K^M, C^M, A^M, S^M, L^M, R^M)$
 K: Knowledge, C: capabilities, A: Action, S: Son, L: Level et R: row.

Primitive: Move (R,M)

Description: L'agent M vient se mettre sous la racine et communique avec elle afin d'avoir l'adresse de l'agent (adresse=(l,r) avec l: le niveau de l'agent destination et r: le rang de l'agent destination) dont il a besoin.

Pré-condition: L'agent M a la permission d'intégrer le système.

Post-condition: $R(K^R, C^R, A^R, S^R, 0, 0)$ avec $K^R=K^R$, $C^R=C^R$, $A^R=A^R$ et $S^R=S^R+M$.

Primitive: Detach (X,M)

Description: L'agent M se détache de l'agent X.

Pré-condition: L'agent M doit être le fils de l'agent X.

Post-condition: $X(K^X, C^X, A^X, S^X, L^X, R^X)$ avec $K^X=K^X$, $C^X=C^X$, $A^X=A^X$, $S^X=S^X-M$, $L^X=L^X$ et $R^M=R^M$.

Primitive: Depth (M)

Description: L'agent M se déplace en profondeur à travers la hiérarchie jusqu'à atteindre le niveau souhaité. Pour savoir s'il est arrivé au niveau souhaité, l'agent M compare le niveau atteint et celui souhaité.

Pré-condition: L'agent M doit avoir une adresse de destination: adresse=(l,r) avec $l \neq 0$ et $r \neq 0$.

Post-condition: $M(K^M, C^M, A^M, 0, L^M, R^M)$ avec $K^M=K^M$, $C^M=C^M$, $A^M=A^M$, $L^M=L^M+1$ et $R^M=R^M$.

Primitive: Width (M)

Description: Arrivé au bon niveau hiérarchique, l'agent se déplace en largeur pour atteindre le bon agent. Pour savoir s'il est arrivé à l'agent souhaité, l'agent M compare le rang atteint et celui souhaité.

Pré-condition: L'agent M doit être positionné sur le bon niveau: adresse=(l,r) avec $l = L^M$.

Post-condition: $M(K^M, C^M, A^M, 0, L^M, R^M)$ avec $K^M=K^M$, $C^M=C^M$, $A^M=A^M$, $L^M=L^M$, $R^M=R^M+1$.

Primitive: Attach (X,M)

Description: La primitive Attach permet à l'agent M de devenir le fils de l'agent X.

Pré-condition: L'agent M a atteint sa destination: adresse=(l,r) avec $l = L^M$ et $r = \text{agentX}$.

Post-condition: $X(K^X, C^X, A^X, S^X, L^X, \text{agentX})$ avec $K^X=K^X$, $C^X=C^X$, $A^X=A^X$, $S^X=S^X+M$, $L^X=L^X$ et $R^M=R^M$.

Primitive: Go (ADR,M)

Description: La primitive Go permet à l'agent M de poursuivre sa migration à travers le réseau.

Pré-condition: L'agent M vient se détacher du système multi-agent.

Post-condition: -

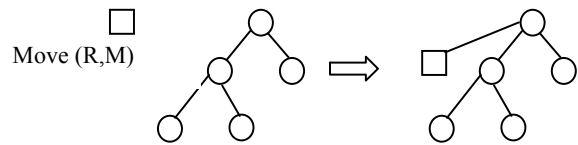
Il est à noter que bien que les primitives « Move » et « Attach » ont les mêmes pré et post conditions, elles restent toutefois différenciées parce que les traitements effectués au cours de chaque primitive est différent.

3.2 Exemple

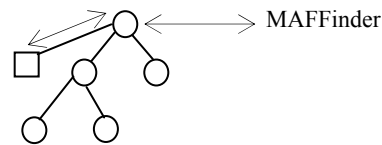
Afin de matérialiser l'utilisation des primitives énoncées ci-dessus, voici un exemple schématique du déroulement de la solution proposée.

Etape 0 : L'agent M demande au MAFAgentSystem la permission d'intégrer le système.

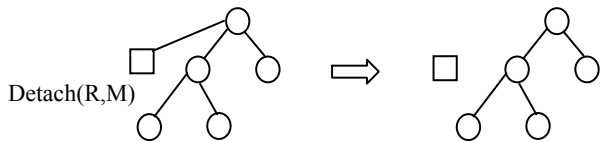
Etape 1 : L'agent M effectue un *Move (R,M)*.



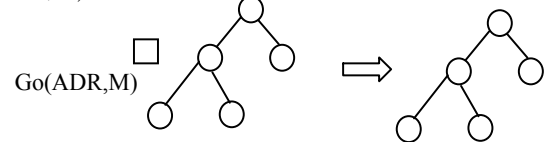
Etape 2 : L'agent M établit une communication avec la racine qui à son en établit une avec le MAFFinder.



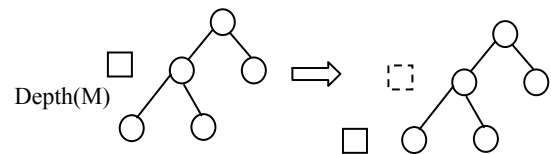
Etape 3 : L'agent M effectue un *Detach (R,M)*.



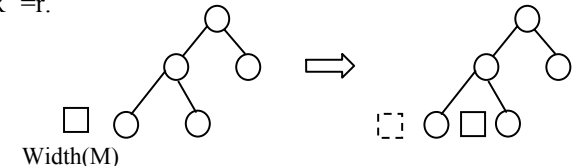
Etape 3' : Si adresse est nulle, l'agent M effectue un *Go (ADR,M)*.



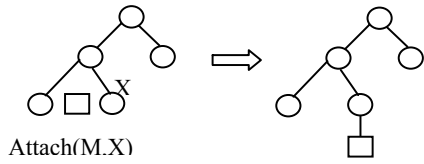
Etape 4 : Si adresse est égale à (l,r), l'agent M effectue des *Depth(M)* jusqu'à $L^M=l$.



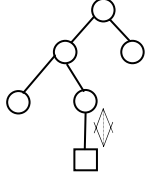
Etape 5 : L'agent M effectue des *Width (M)* jusqu'à $R^M=r$.



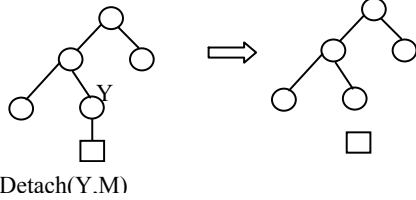
Etape 6 : L'agent M effectue un *Attach* (X,M).



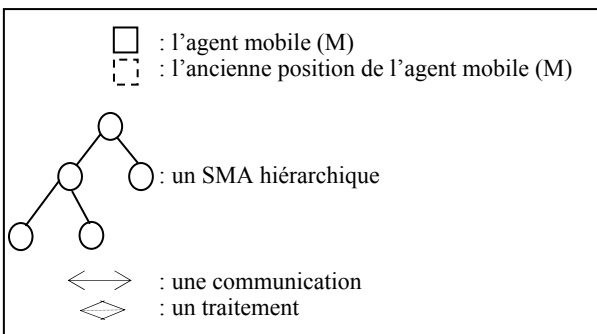
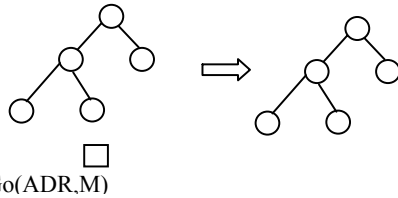
Etape 7 : L'agent M exécute son traitement.



Etape 8 : L'agent M procède à un *Detach* (Y,M).



Etape 9 : L'agent M effectue un *Go* (ADR,M).



4 TRAVAUX SIMILAIRES ET CONCLUSION

Dans ce travail, nous avons proposé une solution au problème de l'intégration d'un agent mobile à base de composants dans un système multi-agents à structure hiérarchique. Ce problème a été traité dans d'autres travaux selon d'autres contextes. En effet, le travail de [Shapiro, 86] s'applique aux objets pour mettre en place un agent Proxy qui représente une interface permettant aux agents mobiles d'utiliser les ressources du système. L'agent mobile n'a pas une visibilité sur la structure interne du système puisque tout passe par le Proxy. Cette approche n'est adaptée à notre problème car notre but est réaliser l'intégration des agents GAMA sans passer par un composant intermédiaire. Dans [El Fallah 05], les auteurs proposent un langage basé sur le calcul d'ambients et réalisent l'intégration d'un agent mobile dans SMA en fusionnant leurs connaissances, compétences et processus. Cette solution n'est pas adaptée aux agents GAMA qui n'ont pas forcément la même structure que ceux du système cible. Dans [Satoh,

00] les auteurs proposent une solution générique d'intégration d'agents mobiles dans un SMA. Nous nous sommes inspirés de leur travail pour réaliser l'intégration d'un agent à base de composant dans un système hiérarchique.

La pertinence de la création d'un nouveau composant réside dans le fait qu'un tel composant permet de représenter l'agent particulier GAMA d'une manière générique abstraction faite de son architecture. Cette solution permet, par conséquent, une intégration « ouverte » qui ne nécessite aucune parité de structure entre l'agent mobile et les agents du SMA.

L'étape suivante de ce travail est l'implémentation de cette solution et son application à un cas réel. Un exemple peut être une bibliothèque virtuelle déployée sur un site Web. Après avoir eu les permissions nécessaires, l'agent mobile vient se mettre sous la racine de la bibliothèque (Fig.1) : « Agent Bibliothécaire » et dérouler toutes les étapes de l'exemple présenté dans la section 2.3.

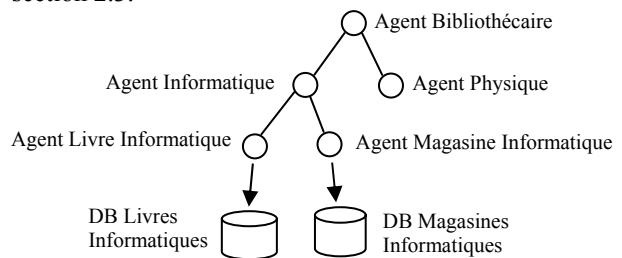


Fig. 1- Structure de la bibliothèque virtuelle

Bibliographie

- [Amara, 04] Amara N., "A Generic Component-based Methodology for Extensible Mobile Agents Design", IEEE International Conference on Advances in Intelligent Systems: Theory and Applications AISTA'04. Luxembourg, November 2004.
- [Amara, 05] Amara N., El Fallah A. : "Toward a Generic Architecture for Dynamically Adaptive Agents and MultiAgents Systems", AAMAS'05, Paris.
- [El Fallah, 05] El Fallah A., Suna A. : "Himalaya Framework: Hierarchical Intelligent Mobile Agents for Building Large-scale and Adaptive sYstems based on Ambients" In International Workshop on Massively Multi-Agent Systems, Lecture Notes in Artificial Intelligence, Vol 3346, pages 126-146, Springer Verlag.
- [Ferber, 95] Ferber J., "Les systèmes multi-agents vers une intelligence collective", pages 13-29, InterEditions, 1995.
- [Milojicic, 98] Milojicic D., Breugst M., Busse I., Campbell J., Covaci., Friedman B., Kosaka K., Lange D., Ono K., Oshima M., Tham C., Virdhagriswaran S., White J. : "MASIF: The OMG Mobile Agent System Interoperability Facility" In Proceedings of the second International Workshop on Mobile Agents, pages 50-67.
- [Satoh, 00] Satoh I.: "A Formalism for Hierarchical Mobile Agents" In International Symposium on Software Engineering for Parallel and Distributed Systems, Limerick, Ireland, 2000.
- [Shapiro, 86] Shapiro M.: "Structure and encapsulation in Distributed Systems: the Proxy Principle" In Proc. 6th Int. Conf. on Distributed Computing Systems (ICDCS), pages 198-204.