

---

# Gestion d'équipes et autonomie des agents

**Damien DEVIGNE, Philippe MATHIEU, Jean-Christophe ROU-  
TIER**

*LIFL - Bâtiment M3  
59655 Villeneuve d'Ascq Cédex  
{devigne, mathieu, routier}@lifl.fr*

---

*RÉSUMÉ. Dans cet article<sup>12</sup>, nous présentons une solution permettant de gérer des équipes d'agents hétérogènes situés dirigés par un chef. Celui-ci est chargé d'établir le plan d'équipe et de distribuer les tâches à effectuer aux agents en fonction de leurs spécificités. Dans une approche traditionnelle, le chef planifie tout. Or, il est important de laisser de l'autonomie de décision aux agents afin de leur permettre de s'adapter à la configuration de leur environnement qui n'est pas forcément connue par le chef lors de la planification. Notre solution consiste à créer des plans d'équipes abstraits où tout n'est pas résolu par le chef, ce qui laisse aux agents une réelle autonomie de décision. Nous proposons ensuite une technique d'allocation des tâches calculées aux agents. Notre approche permet à une équipe d'agents hétérogènes de réaliser des tâches qui requièrent les capacités de plusieurs agents. Ces agents agissent de manière autonome en fonction des buts donnés par le chef et de la configuration de l'environnement.*

*ABSTRACT. In this paper, we present a solution that permits to manage teams of heterogeneous situated agents with a leader. The latter establishes a team plan and distributes some tasks to the teammates according to their abilities. In a classical way, the leader plans all. Therefore, it's important to give some autonomy to the agents in order to allow them to adapt to the configuration of their environment that is not necessary known during the initial leader planning. Our solution consists in creating team abstract plans that are not completely solved, and distributing dynamically the tasks to the agents. This gives some real decision autonomy to the agents. Our approach permits to a team of heterogeneous agents to realize some tasks that require the use of several agents abilities. These agents behave in a autonomous way according to the goals given by the leader and the configuration of the environment.*

*MOTS-CLÉS : Agent, Cognitif, Interaction, Equipe, Autonomie*

*KEYWORDS: Agent, Cognitive, Interaction, Team, Autonomy*

---

1. Ce travail est en partie financé par le programme TAC du contrat plan Etat-Région Nord-Pas de Calais ainsi que le FEDER
2. Une version longue se trouve à <http://www.lifl.fr/evenements/publications/2005-02.ps>

1<sup>re</sup> soumission à *JFSMA'05*, le 15 septembre 2005.

## 1. Introduction

Pour réaliser certaines expériences scientifiques, il est parfois nécessaire de recourir à un substitut de la réalité grâce à la simulation. Certaines applications requièrent la simulation de comportements humains. Pour cela, on peut utiliser des agents réactifs ou des agents cognitifs qui utilisent la planification. Des applications nécessitent de plus des comportements d'équipe. Plusieurs approches sont possibles : par exemple l'émergence ou le plan merging.

Notre approche est basée sur des agents cognitifs hétérogènes. Ils possèdent un nombre limité de capacités, ce qui rend souvent indispensable l'utilisation de plusieurs agents pour résoudre un but de haut niveau. Pour cela, nous utilisons un chef qui esquisse les grandes lignes du plan d'équipe sans amputer l'autonomie de décision des agents. Une fois les actions "abstraites" distribuées par le chef, les agents sont chargés de poursuivre la planification. Notre modèle est composé de trois parties indépendantes : l'environnement, les agents et les interactions. Le moteur de comportement est générique et permet la réutilisation indépendante des différents composants d'une simulation à une autre.

## 2. Le modèle d'agents et d'interactions

Notre modèle [DEV 05] distingue deux catégories d'agents : ce sont les agents inanimés (objets présents dans la simulation) et les agents animés (acteurs dans la simulation). Un agent inanimé possède en particulier une propriété *peut-subir* qui donne la liste des interactions dont il peut potentiellement être la cible. Un agent animé est doté d'une propriété supplémentaire *peut-effectuer* qui détermine les actions qu'il peut déclencher. Il dispose aussi d'un moteur de comportement qui lui permet de déterminer la prochaine action qu'il va entreprendre, ainsi qu'une mémoire (imparfaite) qui sert de base à son raisonnement. Un halo de vision permet à l'agent de maintenir à jour sa mémoire. Comme classiquement en I.A., une difficulté majeure est la représentation des connaissances. Celles-ci incluent non seulement les informations relatives à la topologie de l'environnement, les connaissances relatives aux autres agents mais aussi les règles qui régissent le monde. Nous nommons "interaction" une règle qui décrit formellement une action qu'un agent animé (acteur) peut déclencher sur un autre agent (cible). Une interaction est composée de trois parties : la première (*conditions*) définit les conditions nécessaires à l'exécution de l'interaction. Ces conditions portent sur des propriétés des agents de la simulation. L'agent qui veut déclencher une interaction doit d'abord résoudre ces conditions. La seconde partie (*garde*) de l'interaction permet d'isoler une condition spécifique portant sur la distance relative de l'acteur à la cible. Cette condition particulière oblige l'acteur à se trouver proche de la cible pour interagir, et évite les exécutions à distance [DEV 04]. La troisième partie (*actions*) définit les conséquences de l'exécution de l'interaction, qui portent sur les propriétés des agents ou les agents eux-mêmes (création, destruction, ...). Le moteur de comportement de l'agent est un chaînage arrière basé sur les interactions présentes dans

ses *peut-effectuer* et sur le contenu de sa mémoire. Il permet de produire un plan sous forme d'arbre de buts pour résoudre le but donné.

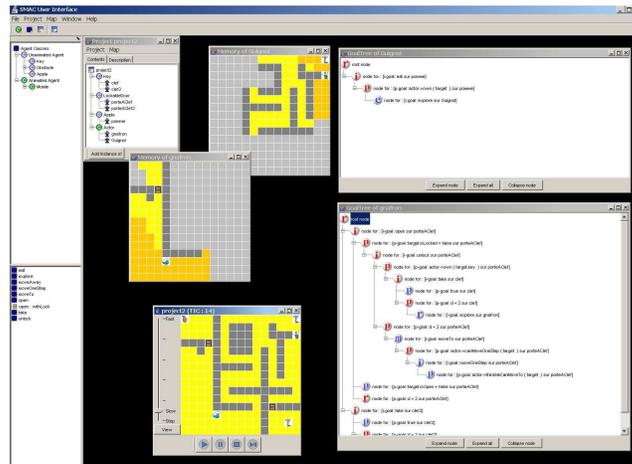
### 3. Planification et exécution dans une équipe

La résolution des certains problèmes nécessite l'utilisation de plusieurs agents, car ils n'ont pas les mêmes capacités. Pour permettre aux agents de collaborer, il faut mettre au points des stratégies d'équipes. Nous réalisons cela grâce à l'adjonction d'un chef dans l'équipe. Celui-ci possède un rôle de planification et de coordination : il établit le plan pour les agents, et distribue les tâches à effectuer aux membres de l'équipe en tenant compte de leurs capacités. Toutefois, l'établissement d'un plan d'équipe complet par le chef ne présente pas un grand intérêt. Nous proposons une méthode permettant au chef d'établir un plan abstrait et de distribuer des tâches non atomiques aux membres. Ces derniers sont chargés de poursuivre le travail de planification en fonction de leurs capacités pour mener à bien le but qu'ils ont reçu du chef. Notre méthode permet au chef d'arrêter la planification dès que le travail peut être continué par un agent. Le principe que nous adoptons est que le chef sait quelles sont les actions à accomplir pour exécuter une tâche complexe, mais ne sait pas comment effectuer ces actions. Pour cela, on ajoute aux *peut-effectuer* du chef les *peut-effectuer* de chacun des membres de son équipe, en masquant les parties *conditions* et *garde* de ces interactions. De ce fait, il peut utiliser ces interactions dans le chaînage arrière, mais comme elles ne comportent aucune condition, le chaînage ne va pas plus loin : le plan est abstrait. Dans une seconde phase, le chef distribue aux agents les actions abstraites du plan. Les agents se réfèrent alors à leur version de l'interaction (qui contient les conditions et la garde) et établissent un plan concret.

### 4. Affectation et réaffectation des actions

Le second rôle du chef est d'affecter les actions à effectuer aux membres de l'équipe, en tentant de maximiser le nombre de tâches effectuées simultanément. Nous utilisons pour cela l'algorithme de Ford-Fulkerson [FOR 56] après avoir transformé notre problème d'affectation en graphe de transport. Il est ainsi possible, en recherchant le flot maximal dans le réseau, de trouver une affectation optimale des agents.

Durant la simulation, la liste des agents et des tâches est susceptible d'évoluer et il est nécessaire de procéder à une réaffectation des agents. Or il est coûteux en temps de changer l'affectation d'un agent car il doit se déplacer jusqu'à sa nouvelle cible. Aussi nous devons prendre en compte les modifications dans la liste des agents et la liste des tâches en faisant un minimum de changements d'affectation. Pour cela, nous utilisons l'algorithme de Edmonds-Karp qui est une optimisation de l'algorithme de Ford Fulkerson. Cet algorithme nous permet d'ajouter ou de retirer une tâche ou un agent et de recalculer l'affectation de manière incrémentale, tout en garantissant une solution qui minimise le nombre de réaffectations.



**Figure 1.** notre environnement de simulation permet de créer indépendamment les agents, les interactions et les environnements afin de les intégrer dans différentes simulations.

## 5. Conclusion et perspectives

Ce travail s'appuie sur notre modèle d'agents basé sur les interactions. Sur ce modèle, nous proposons une méthode permettant de regrouper des agents en équipe afin de tirer profit de toutes leurs capacités. Le chef d'équipe joue un rôle particulier en établissant un plan abstrait pour l'équipe, ce qui permet de conserver l'autonomie de décision des agents en répartissant l'effort de planification. La méthode d'affectation des tâches que nous utilisons nous permet de tirer partie des capacités de l'équipe de manière optimale tout en prenant en compte efficacement la dynamique de l'équipe. Il reste toutefois un problème dur qui est la remontée d'information des agents vers le chef, pour éviter que le blocage local d'un agent n'aboutisse au blocage de toute l'équipe. Une de nos autres perspectives est la gestion dynamique des capacités de l'équipe par le biais du recrutement.

## 6. Bibliographie

- [DEV 04] DEVIGNE D., MATHIEU P., ROUTIER J.-C., « Planning for Spatially Situated Agents in simulations », *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)*, IEEE Press., 2004, p. 385-388.
- [DEV 05] DEVIGNE D., MATHIEU P., ROUTIER J.-C., « Interaction-based Approach for Game Agents », *proceedings of ECMS'05*, 2005.
- [FOR 56] FORD L., FULKERSON D., « Maximal flow through a network », *Canadian Journal of Mathematics*, vol. 8, 1956, p. 399-404.