



HAL
open science

Un modèle multi-agents de l'apprenant

Stéphane Leman, Pierre Marcenac, Sylvain Giroux

► **To cite this version:**

Stéphane Leman, Pierre Marcenac, Sylvain Giroux. Un modèle multi-agents de l'apprenant. Sciences et Techniques Educatives, 1996, 3 (4), pp.465 - 483. 10.3406/stice.1996.1318 . hal-04016154

HAL Id: hal-04016154

<https://hal.univ-reunion.fr/hal-04016154>

Submitted on 6 Mar 2023

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.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Un modèle multi-agents de l'apprenant

Stéphane Leman, Pierre Marcenac, Sylvain Giroux

Résumé

Notre objectif est de construire un modèle de l'apprenant avec une approche multi-agents. Les connaissances de l'apprenant sont représentées par plusieurs niveaux de granularité. À chaque type de connaissance correspond un type d'agent. La description interne, l'organisation dans des graphes de raisonnement, ainsi que la modification dynamique de ces agents au cours de la résolution des problèmes par l'apprenant sont décrits dans la première partie de cet article. Par ailleurs, lorsque l'apprenant effectue un raisonnement plus complexe, mettant en jeu plusieurs agents, ce raisonnement est reconnu comme sous-ensemble d'un raisonnement expert et est modélisé dans le système par un agent complexe. L'algorithme d'expertisation permettant cette reconnaissance et cette modélisation des raisonnements est détaillé dans la seconde partie de l'article.

Abstract

Our goal in this paper is to propose a generic method to build a student model with a multi-agent approach. Student knowledge are represented in the system with many granularity levels. Agent internal structure, reasoning graph organisation and dynamic agent evolution are describe in the first part. After, when student build a new reasoning with many agents, this reasoning is recognizing and modeling by our system. The 'expertisation' algorithm is detailed in the second part of this paper.

Citer ce document / Cite this document :

Leman Stéphane, Marcenac Pierre, Giroux Sylvain. Un modèle multi-agents de l'apprenant. In: Sciences et techniques éducatives, volume 3 n°4, 1996. pp. 465-483;

doi : <https://doi.org/10.3406/stice.1996.1318>

https://www.persee.fr/doc/stice_1265-1338_1996_num_3_4_1318

Fichier pdf généré le 04/05/2018

Un modèle multi-agents de l'apprenant

Stéphane Leman* — Pierre Marcenac* — Sylvain Giroux**

* IREMIA - Université de La Réunion
BP 7151, 97715 Saint-Denis Messagerie cedex 9, La Réunion

** Centre de recherches LICEF, Télé Université
Montréal, Québec, Canada

RÉSUMÉ. Notre objectif est de construire un modèle de l'apprenant avec une approche multi-agents. Les connaissances de l'apprenant sont représentées par plusieurs niveaux de granularité. À chaque type de connaissance correspond un type d'agent. La description interne, l'organisation dans des graphes de raisonnement, ainsi que la modification dynamique de ces agents au cours de la résolution des problèmes par l'apprenant sont décrits dans la première partie de cet article. Par ailleurs, lorsque l'apprenant effectue un raisonnement plus complexe, mettant en jeu plusieurs agents, ce raisonnement est reconnu comme sous-ensemble d'un raisonnement expert et est modélisé dans le système par un agent complexe. L'algorithme d'expertisation permettant cette reconnaissance et cette modélisation des raisonnements est détaillé dans la seconde partie de l'article.

ABSTRACT. Our goal in this paper is to propose a generic method to build a student model with a multi-agent approach. Student knowledge are represented in the system with many granularity levels. Agent internal structure, reasoning graph organisation and dynamic agent evolution are describe in the first part. After, when student build a new reasoning with many agents, this reasoning is recognizing and modeling by our system. The 'expertisation' algorithm is detailed in the second part of this paper.

MOTS-CLÉS : modélisation du raisonnement, modèle de l'apprenant, systèmes tuteurs intelligents, systèmes multi-agents.

KEY WORDS : reasoning modeling, student model, intelligents tutoring systems, multi-agent systems.

1. Introduction

Cet article s'intéresse à la construction du modèle de l'apprenant d'un système tuteur intelligent. Le principal apport du travail est de proposer une solution à la modélisation d'un apprenant en phase de résolution de problèmes. Cette modélisation de l'apprenant et de ses raisonnements est essentielle dans le cadre d'un système tuteur intelligent car elle permet de personnaliser l'enseignement par le choix de stratégies tutorielles adaptées [GRE 94]. La construction du modèle de l'apprenant reste un problème difficile et complexe [NIC 88], [HOL 94]. Il s'agit de trouver une représentation efficace des connaissances et des raisonnements de l'apprenant. Les raisonnements de l'apprenant n'étant ni figés dans le temps ni nécessairement uniques, le système de représentation doit permettre à la fois une évolutivité et une représentation simultanée de plusieurs raisonnements effectués en parallèle. Par l'autonomie des différents agents qui les composent et par leurs possibilités d'évolutions, les systèmes multi-agents offrent de nombreux avantages pour modéliser les raisonnements d'un apprenant et cet article présente des éléments de solutions originaux au problème de modélisation de l'apprenant grâce à l'approche multi-agents.

Les connaissances de l'apprenant sur le domaine sont vues comme un sous-ensemble des connaissances de l'expert, appelées *connaissances expertes*. Les connaissances expertes sont partagées entre connaissances statiques et connaissances dynamiques représentées dans notre système par des agents de types S et D. Les deux types de connaissances se lient entre eux pour former des raisonnements complexes. Au cours de la résolution des exercices proposés, l'apprenant enrichit ses connaissances. Cette évolution est prise en compte dans notre modèle par la gestion de valeurs de *croiances* attribuées aux agents. Lorsque les actions de l'apprenant ne correspondent plus exactement à une description donnée par l'expert, le système tente alors de reconnaître le raisonnement de l'apprenant comme une portion du raisonnement de l'expert. La modélisation de ce raisonnement de l'apprenant est ensuite introduite dans le système de représentation par un agent particulier qui représente un raisonnement complexe de l'apprenant suffisamment maîtrisé pour être utilisé de façon implicite.

L'article comporte deux parties : la première décrit les composants et l'architecture générale du modèle de l'apprenant, alors que la seconde montre comment le modèle se construit dynamiquement pendant la résolution des problèmes par l'apprenant.

2. Architecture générale

2.1. Un modèle multi-agents

La structure générale du modèle se base sur les idées émises dans [SEL 87] et la construction du modèle de l'apprenant repose sur un algorithme de recouvrement issu de [CAR 77], mais cette fois distribué, car étendu aux systèmes multi-agents. Dans ce type de système, les agents coexistent : ils possèdent chacun des connaissances propres et une autonomie de fonctionnement. Le comportement global du système

multi-agents est la résultante des actions propres des agents et des communications qu'ils établissent entre eux. Le modèle multi-agents proposé ici permet d'apporter certains avantages liés à la représentation distribuée des connaissances dans le domaine des systèmes tuteurs intelligents, où les approches de type multi-agents restent peu nombreuses bien que prometteuses [LEV 94]. Cette représentation permet de réduire la complexité des expertises représentées, puisque celles-ci sont réparties dans des agents autonomes, chacun en possédant une petite partie. Enfin, les avantages obtenus grâce à cette approche concernent également les deux points suivants : évolutivité du modèle au cours des différentes sessions et examen d'hypothèses parallèles lors de la construction du modèle.

2.1.1. *Évolutivité*

Pour conserver sa pertinence, le modèle de l'apprenant doit être évolutif. En effet, les connaissances de l'apprenant évoluent à la suite des différentes sessions d'utilisation du système. C'est d'ailleurs là que réside l'objectif primordial d'un système tuteur. Chaque action de l'apprenant est susceptible de faire évoluer ses connaissances. Ainsi, le même modèle utilisé en début de session ne sera plus aussi fidèle à la fin de celle-ci [IKE 93]. De ce point de vue, un modèle *idéal* doit être modifié en temps réel à chaque action de l'apprenant. L'utilisation d'une approche multi-agents fournit dans ce cas une alternative intéressante. En effet, dans notre modèle, chacune des interactions entre l'apprenant et le système tuteur est prise en compte localement par l'agent concerné et la mise à jour du modèle de l'apprenant se fait ainsi en temps réel sans difficulté.

Un autre aspect de l'évolutivité du système concerne la modification du type des connaissances de l'apprenant. Au fur et à mesure que l'apprenant progresse dans la maîtrise de l'expertise, son comportement se rapproche de celui d'un expert. Il utilise de plus en plus de raisonnements complexes. La reconnaissance de ces raisonnements ainsi que leur modélisation est cruciale : chaque raisonnement complexe maîtrisé par l'apprenant correspond à une élévation de l'expertise globale maîtrisée par celui-ci. Une représentation multi-agents permet de modéliser efficacement cette évolution des connaissances de l'apprenant par agglomération d'un ensemble d'agents simples en un seul agent plus complexe.

2.1.2. *Examen d'hypothèses parallèles*

L'apprenant est une personne humaine dont le comportement n'est pas nécessairement linéaire et cohérent. Aussi, avant d'avoir la certitude de l'intention de l'utilisateur, il est nécessaire d'examiner en parallèle plusieurs hypothèses sur le raisonnement de l'apprenant. Pour être efficace, un modèle de l'apprenant devra prendre en compte ces hypothèses différentes voire contradictoires [LEM 96]. L'utilisation d'un modèle de l'apprenant distribué dans un système multi-agents apporte une solution intéressante pour ce type de problème, puisque par nature les systèmes multi-agents permettent le traitement d'informations en parallèle. Plusieurs possibilités de raisonnement de l'apprenant peuvent ainsi être examinées en parallèle par les agents du système. Le raisonnement effectivement choisi par l'apprenant est reconnu de façon dynamique alors que les autres solutions envisagées sont abandonnées [LEM 95].

2.2. Le modèle d'agent

Le coeur du travail consiste en la définition du modèle d'agent qui spécifie les connaissances, les comportements et les interactions de chaque agent, ainsi que l'évolution dynamique de ses connaissances et de ses comportements au sein de la société. La composition interne des agents est définie par trois caractéristiques essentielles : connaissances, accointances et croyances.

2.2.1. Connaissances

Dans les systèmes tuteurs intelligents, la représentation des connaissances, que ce soient les connaissances à enseigner ou les connaissances maîtrisées par l'apprenant, est l'un des problèmes primordiaux. Dans notre modèle, la représentation des connaissances expertes et des connaissances de l'apprenant est effectuée de manière similaire, en utilisant des agents modélisant chacun une part de l'expertise globale.

Trois types de connaissances sont définies : les *connaissances statiques* qui décrivent les concepts et les *connaissances dynamiques* qui décrivent le savoir-faire dans le domaine. Les connaissances statiques sont représentées par des agents de type **S** et les connaissances dynamiques sont représentées par des agents de type **D**. Ces deux types d'agents sont qualifiés d'agents primitifs car ils modélisent un granule de connaissance indivisible. La connaissance modélisée par un agent correspond à une partie indivisible de l'expertise à enseigner. La granularité de cet agent est fixée conjointement par un expert du domaine et un pédagogue, en fonction du niveau supposé de l'apprenant et du niveau d'expertise à atteindre. Le troisième type de connaissances concerne l'enchaînement du savoir et du savoir-faire dans un *raisonnement* et exprime la capacité à résoudre le problème. Cette modélisation des raisonnements sera abordée dans la section 4.

L'ensemble des agents du système, communiquant entre eux, représente ainsi l'expertise globale. Cette représentation des connaissances sous forme d'un ensemble d'agents n'est pas nouvelle, puisque c'est ainsi que Minsky présentait sa célèbre société de l'esprit [MIN 85]. Toutefois, la modélisation d'une expertise complexe au travers d'un ensemble d'agents communicants est un problème difficile. La recherche d'une démarche de décomposition a été abordée dans [LEM 93] dans le cadre d'une expertise de cryptarithmétique. Il est néanmoins impossible d'affirmer qu'il existe une démarche pouvant s'appliquer à toutes les expertises.

Pour illustrer cette caractéristique des agents, prenons l'exemple de la géométrie euclidienne : la connaissance d'un agent de type statique représente la maîtrise d'un concept comme par exemple un point ou une droite ; la connaissance d'un agent de type dynamique modélise la maîtrise d'un théorème ou d'une construction géométrique.

2.2.2. Accointances

Dans un système multi-agents, chaque agent ne possède que des informations locales. Ainsi, en ce qui concerne les communications, un agent ne peut s'adresser qu'aux agents dont il connaît l'existence : ses accointances. Les communications se font uniquement par envoi de messages asynchrones et il n'y a aucun mécanisme de synchronisation entre les agents.

Dans notre modèle, les agents n'ont qu'une représentation partielle de leur environnement. Ainsi, chacun des agents du modèle possède, au plus, trois accointances. Les accointances d'un agent représentent l'ensemble des pères, l'ensemble des fils et le sommet du raisonnement auxquels l'agent appartient. Hormis ses accointances, un agent ne peut pas communiquer directement avec un autre agent du système.

2.2.3. Croyances

L'introduction de valeurs de croyances dans notre modèle relève de plusieurs observations :

1) La maîtrise d'une connaissance par l'apprenant ne peut être représentée de façon "booléenne". Ainsi chaque connaissance de l'apprenant est modélisée par un agent de type S ou D avec une certaine valeur de croyance représentant le degré de maîtrise de l'apprenant pour cette connaissance.

2) Lors de la résolution d'un problème, plusieurs solutions envisageables n'ont pas nécessairement le même poids en terme de valeur pédagogique de l'exercice. Chacune des solutions se voit donc attribuer une valeur de croyance relative par rapport aux autres solutions. Cette valeur propre à chaque solution est répercutée au niveau de tous les agents qui la composent.

3) Dans un même raisonnement, toutes les étapes n'ont pas la même importance. Ainsi certaines étapes représentées par des agents ont plus d'importance que d'autres. Par conséquent, chaque agent se voit donc attribuer une valeur de croyance pour représenter son importance relative dans le raisonnement.

Pour plus de précisions dans la représentation des connaissances, trois valeurs de croyances sont ainsi introduites dans le système au niveau de chaque agent :

1) Croyance **propre** (C_p) : C'est la valeur de confiance attribuée à un agent et qui indique le degré de maîtrise par l'apprenant de la connaissance que représente l'agent.

2) Croyance **globale** (C_g) : Elle représente l'importance relative de la solution à laquelle appartient l'agent par rapport aux autres solutions possibles.

3) Croyance **relative** (C_r) : A l'intérieur de la solution, elle mesure l'importance relative de l'agent dans la solution.

Les valeurs initiales de C_g et C_r sont attribuées par des experts du domaine qui définissent les différentes solutions aux problèmes. Au départ, la valeur attribuée à C_p est indéterminée. Ensuite cette croyance évolue en fonction des actions effectuées par l'apprenant.

Les agents S et D permettent de représenter la maîtrise des connaissances par l'apprenant sur le domaine enseigné. Cette représentation des connaissances sur le domaine constitue l'ébauche de la modélisation d'un apprenant, mais les connaissances statiques et dynamiques prises individuellement ne suffisent pas pour résoudre un problème. Aussi, pour une utilisation plus efficace de cette représentation, il est fondamental de pouvoir modéliser l'enchaînement des savoir-faire de l'apprenant, ce qui constitue le raisonnement. Le raisonnement est représenté dans notre modèle par une société.

2.3. *Le modèle de société*

Si les agents modélisent les connaissances primitives, une société d'agents représentent un raisonnement utilisé pour résoudre un problème. Les raisonnements sont initialement donnés par des experts qui décomposent ainsi le problème général en une succession de sous-problèmes qui peuvent être résolus par l'apprenant. Pour représenter ces raisonnements, les agents S et D sont organisés suivant un *graphe de raisonnement* qui décrit l'ensemble des connaissances à enchaîner pour résoudre un problème posé. Un graphe de raisonnement est un graphe hiérarchique dans lequel chaque étape de la progression dans la résolution du problème est signifiée par l'un des noeuds du graphe. A chacun de ces noeuds correspond un agent S ou D du système.

En général, pour la résolution d'un problème proposé à l'apprenant, il existe plusieurs solutions. Pour chacune de ces solutions, l'expert propose un sous-graphe de raisonnement correspondant. Le système est alors composé de plusieurs sous-systèmes multi-agents correspondant aux différentes possibilités de raisonnement. Lors de la résolution du problème, l'apprenant ne suit pas forcément un seul raisonnement, il peut ainsi commencer plusieurs solutions possibles. Chacune des actions de l'apprenant est traitée en parallèle dans les sous-systèmes multi-agents. De cette façon, la solution choisie par l'apprenant n'est pas prédéfinie dans le système. Les solutions sont examinées en parallèle et celle qui est effectivement choisie par l'apprenant soit reconnue, alors que les autres solutions entamées n'aboutissent pas nécessairement.

Par ailleurs, la société d'agents n'est pas figée. Les accointances entre les agents qui définissent la structure du graphe de raisonnement peuvent évoluer de façon à intégrer de nouveaux agents correspondant à des nouveaux raisonnements de l'apprenant. Cette évolution se fait au travers d'un algorithme implanté au niveau de la société : l'*expertisation* (Cet algorithme sera détaillé en section 4).

3. **Évolution dynamique des agents**

Pour la construction dynamique du modèle, nous nous plaçons dans l'hypothèse de travail suivante : l'apprenant travaille à la résolution d'un problème au travers d'une interface informatique, appelée l'application hôte. Pendant son travail, il est observé grâce un système qualifié d'épiphyte, c'est-à-dire un système qui se greffe sur une application existante, l'hôte, de façon à l'espionner sans en modifier le fonctionnement.

L'évolution dynamique des agents a deux aspects : d'une part, un agent peut être modifié par une sollicitation extérieure correspondant à une action de l'apprenant ; d'autre part, un agent peut subir des modifications de son état par l'intermédiaire d'autres agents du système, selon un mécanisme qualifié de *propagation*.

3.1. *Évolution par sollicitation extérieure*

Initialement, les connaissances de l'apprenant sur le domaine enseigné sont complètement inconnues dans le modèle. On ne dispose pas d'informations a priori sur le niveau de l'apprenant. De ce fait, le système est alors constitué d'un ensemble d'agents correspondant aux différentes solutions expertes, mais dans lequel tous les coefficients de croyances propres ont des valeurs inconnues.

Lorsque l'apprenant effectue une opération, celle-ci modifie le modèle de l'apprenant. Le modèle de l'apprenant est alors construit selon un algorithme de recouvrement dont le motif est un agent S ou D. En effet, une action de l'apprenant correspond à l'activation d'une connaissance qui est reconnue par l'agent (ou les agents) la modélisant. La valeur de croyance propre de ces agents est alors modifiée selon une loi de mise à jour. Ainsi, toutes les actions de l'apprenant sont traitées dynamiquement par le système pour construire, au cours de la résolution par l'apprenant de l'exercice proposé, le modèle de l'apprenant.

La définition d'une loi de mise à jour de la valeur de croyance propre d'un agent est un problème complexe. Cette loi modélise le renforcement des connaissances par répétition et le réajustement des valeurs de croyances à chaque nouvelle utilisation de l'agent, c'est-à-dire à chaque sollicitation par l'apprenant de la connaissance modélisée par cet agent. Pour un agent de type S, chaque fois que l'agent est sollicité dans des contextes différents, sa valeur de croyance propre (C_p) est renforcée. On suppose en effet que l'apprenant maîtrise de mieux en mieux cette connaissance. Pour un agent de type D, la nouvelle valeur de croyance propre est la résultante de l'ancienne valeur et du résultat de l'évaluation par le système de la maîtrise de la connaissance. Par exemple, si l'apprenant doit fournir le résultat d'un calcul, l'évaluation de la valeur de croyance pourra se faire en calculant la différence entre le résultat fourni par l'apprenant et le résultat réel calculé par le système.

Dans les deux cas, après la mise à jour de sa valeur de croyance propre, l'agent va diffuser cette valeur aux agents qui modélisent la même connaissance dans les différents graphes de raisonnement. Cette diffusion se fait de proche en proche et va aboutir à une harmonisation des différentes valeurs de croyances. La valeur de croyance propre des agents évolue ainsi en permanence au fur et à mesure de la progression de l'apprenant dans la résolution du problème.

3.2. *Évolution par propagation*

Afin de conserver la cohérence du système, il est impératif qu'à chaque fois qu'un agent modifie sa valeur de croyance propre, il informe les autres agents du système qui modélisent la même connaissance, afin que ceux-ci mettent à jour leurs propres croyances. Comme l'agent ne les connaît pas directement, il le fait par l'intermédiaire de ses accointances. Le message de mise à jour circule ainsi dans le système en suivant le graphe de raisonnement. Lorsque ce message parvient à un agent modélisant la même connaissance que l'agent émetteur, celui-ci prend en compte l'information pour modifier sa propre valeur de croyance et diffuse alors lui aussi cette nouvelle valeur.

La nouvelle croyance propre de l'agent dépend alors de son ancienne croyance ($Cp1$), de l'importance relative de sa croyance globale ($Gg1$) et de la croyance globale ($Cg2$) et propre ($Cp2$) de l'autre agent. La nouvelle valeur de croyance propre peut être formulée par :

$$Cp = \frac{Cg1Cp1 + Cg2Cp2}{Cg1 + Cg2}$$

Néanmoins, cette propagation des valeurs de croyances pose le problème de l'équilibre et de la cohérence des informations:

Étudions par exemple le cas simplifié suivant :

Deux agents 1 et 2 représentent la même connaissance utilisée dans des raisonnements différents. Pour l'agent 1, la croyance propre est de 80% et la croyance globale de 30%. A la suite d'une action de l'apprenant, la croyance propre de l'agent 2 devient 60% avec une croyance globale de 70%.

Cet agent 2 communique, par l'intermédiaire des accointances successives, sa nouvelle valeur de croyance propre au premier agent qui recalcule alors sa propre valeur : $Cp1=66\%$.

L'agent 1 communique cette nouvelle information à l'agent 2 qui se doit alors de calculer à nouveau sa nouvelle valeur de croyance propre : $Cp2=61.6\%$.

Cette nouvelle valeur de $Cp2$ est alors elle aussi diffusée par l'agent 2 et l'agent 1 modifie à nouveau sa valeur, ...

Il est essentiel que la propagation des nouvelles valeurs de croyances entre deux agents différents modélisant la même connaissance ne soit pas infinie. Pour cela, il convient de démontrer la convergence des deux valeurs de croyances vers une même valeur λ .

La démonstration de cette convergence peut être établie facilement en étudiant la suite définie ci-dessous qui représente les valeurs successives de croyances propres des agents 1 (terme pair) et 2 (terme impair) [LEM 96]. Ainsi U_2 correspondra à la valeur de croyance propre de l'agent 1 après avoir reçu l'information de l'agent 2.

$$\begin{cases} U_0 = Cp1 \\ U_1 = Cp2 \\ U_{2n} = \frac{Cg1}{Cg1 + Cg2} U_{2n-2} + \frac{Cg2}{Cg1 + Cg2} U_{2n-1} \\ U_{2n+1} = \frac{Cg2}{Cg1 + Cg2} U_{2n-1} + \frac{Cg1}{Cg1 + Cg2} U_{2n} \end{cases}$$

Les valeurs de croyances propres de deux agents convergeant vers une même valeur, la cohérence des informations de notre système est donc assurée. Par ailleurs, pour limiter les échanges de messages, nous proposons les hypothèses suivantes : un agent dont la nouvelle valeur de croyance propre différera très faiblement de son

ancienne valeur (moins de 3%) ne propagera pas cette nouvelle valeur aux autres agents du système et les valeurs de croyances propres des agents sont arrondies aux pourcentages entiers les plus proches. Avec ces hypothèses et en reprenant l'exemple décrit ci-dessus, les valeurs de croyances finales des deux agents, convergeant *théoriquement* vers 62,3%, sont $Cp1=66\%$ et $Cp2=62\%$ et seulement deux messages ont été échangés :

A2 -> A1 $Cp=60$ $Cg=70$

Pour A1, $Cp1=80 \times 30 + 60 \times 70 = 66$

A1 -> A2 $Cp=66$ $Cg=30$

Pour A2, $Cp2=60 \times 70 + 66 \times 30 = 62$

$Cp2$ passe alors de 60 à 62. La différence est inférieure à 3%, on ne diffuse plus la nouvelle valeur.

Il est important de remarquer qu'il y a également convergence des valeurs de croyances dans le cas où plus de deux agents sont impliqués. Toutefois, dans le cas où plusieurs agents sont concernés, il est cette fois impossible de déterminer l'ordre dans lequel les messages sont échangés, et les valeurs de croyances des différents agents concernés ne sont donc pas déterministes à chaque instant.

Soit l'exemple suivant où deux agents A1 et A2 possèdent une croyance $Cp1=Cp2=80$. Un troisième agent A3 reçoit comme nouvelle valeur de croyance $Cp3=60$, les messages échangés peuvent être :

Soit au départ¹ :

A1,80,60 - A2,80,60 - A3,60

On ne peut pas déterminer qui de A1 ou A2 va traiter le message en premier. Supposons que ce soit A1 (les deux agents jouent dans cet exemple un rôle symétrique).

A1 passe de 80 à 70 et l'état des agents est alors le suivant

A1,70 - A2,80,60,70 - A3,60,70

La encore, deux choix sont possibles pour l'ordre d'exécution : A2 ou A3. Les résultats correspondants sont :

Cas A2 : A1,70,70 - A2,70,70 - A3,60,70,70

puis A1,70,65 - A2,70,65 - A3,65,70

Cas A1 : A1,68 - A2,70,65,68 - A3,65,70,68

Cas A3 : A1,70,65,68 - A2,70,65,68 - A3,68

puis finalement A1,68 - A2,68 - A3,68

Cas A3 : A1,70,65 - A2,80,60,70,65 - A3,65

Cas A1 : A1,68 - A2,80,60,70,65,68 - A3,65,68

Cas A2 : A1,70,65,70 - A2,70,70,65 - A3,65,70

etc...

¹ La notation suivante $Ax, Cp, x1, x2, \dots, xi$ signifie que l'agent X possède actuellement une valeur de croyance propre de Cp et qu'il doit intégrer successivement les valeurs de croyances $x1, x2, \dots, xi$. (Hypothèse : les croyances générales sont égales).

Le comportement exact des agents reste inconnu, et donc il en est de même de l'évolution de la croyance propre de tous les agents. Ce problème est inhérent au parallélisme et au traitement asynchrone des messages, mais il offre l'avantage d'une plus grande souplesse dans le traitement des informations : il est assuré que les messages émis par les agents sont traités dans un temps fini.

4. Modélisation des raisonnements complexes de l'apprenant

4.1. Nécessité d'un nouveau type d'agent : le R-agent

Les exemples précédents ont montré comment s'effectuait la construction du modèle de l'apprenant dans les cas simples, c'est-à-dire lorsque l'action effectuée par l'apprenant correspondait bien à une action attendue par l'un des agents non encore activé. L'apprenant suivait, étape par étape, l'une des solutions décrite par un expert. Toutefois, il est possible que l'apprenant effectue un saut dans le raisonnement lorsqu'il maîtrise bien une partie de celui-ci. Dans ce cas, une ou plusieurs étapes de raisonnement données par l'expert peuvent être omises. Ce saut dans le raisonnement correspond à la maîtrise par l'apprenant d'un nouveau type de connaissance. Pour modéliser cet état, des agents de type **R** sont introduits.

Un agent de type R possède les mêmes caractéristiques que les agents primitifs S ou D en terme de connaissances, d'accointances et de croyances. Toutefois, la granularité de la connaissance qu'il représente est supérieure. Cette possibilité de représenter les connaissances de l'apprenant avec plusieurs niveaux de granularité apporte un avantage indéniable pour l'exploitation par des stratégies tutorielles du modèle de l'apprenant [CAL 94].

Toutefois, il n'est pas approprié de créer un agent de type R à chaque fois que l'apprenant effectue un saut dans le graphe de raisonnement. Il est au préalable important de vérifier que les connaissances sautées par l'apprenant dans ce raisonnement sont bien maîtrisées. Si tel est le cas, l'apprenant a acquis une nouvelle connaissance plus complexe que celles qu'il maîtrisait jusqu'alors, et un agent de type R doit être créé dans le modèle pour représenter cette nouvelle connaissance. Pour résoudre le problème de la validation de l'expertisation de l'apprenant, l'algorithme comprend deux parties. Dans un premier temps, il convient de reconnaître le raisonnement de l'apprenant dans l'un des graphes de raisonnement. Ensuite, et seulement dans le cas où le raisonnement est reconnu, il convient de le modéliser par la création d'un agent R.

4.2. Reconnaissance du Raisonnement

Reconnaître le raisonnement de l'apprenant revient donc à rechercher si l'apprenant a ou n'a pas effectué un saut dans le graphe de raisonnement expert. Toutefois, nous ne disposons pas au niveau des agents d'une vue globale sur les graphes de raisonnement. La détection d'un saut de raisonnement se fait donc de façon locale. Lorsqu'un agent reçoit une information correspondant à une action de l'apprenant, si cette information correspond à la connaissance modélisée par l'agent

récepteur, alors celui-ci met à jour sa nouvelle valeur de croyance propre et met à jour sa vision propre de l'apprenant. Il transmet ensuite cette nouvelle information aux autres agents qui modélisent la même connaissance de façon à conserver la cohérence du modèle.

Par contre, si l'action effectuée par l'apprenant et transmise à l'agent ne correspond pas à la connaissance qu'il modélise, celui-ci consulte les autres agents pour une expertisation. Dans ce cas, la première étape consiste à reconnaître le raisonnement de l'apprenant dans l'une des solutions de l'expert. Pour cela, il faut s'assurer que l'action effectuée par l'apprenant correspond à l'un des agents situés au-dessus de l'agent demandeur dans le même graphe de raisonnement. Si tel est le cas, il faut alors déterminer l'ensemble des agents qui ont été sautés par l'apprenant et qui constituent le raisonnement.

La reconnaissance du raisonnement de l'apprenant se fait donc de façon locale, de proche en proche. L'agent demandeur communique sa demande à ses pères, qui la retransmettent à leur tour à leurs pères, et ceci tant que le sommet n'est pas atteint ou le raisonnement reconnu.

La liste des agents constituant le raisonnement de l'apprenant est construite au fur et à mesure du parcours vers le sommet du graphe de raisonnement. Si l'agent recherché est trouvé, le raisonnement est reconnu comme une portion de la solution de l'expert et la liste des agents la constituant est retournée au demandeur. Un tel cas est illustré par la figure 1 :

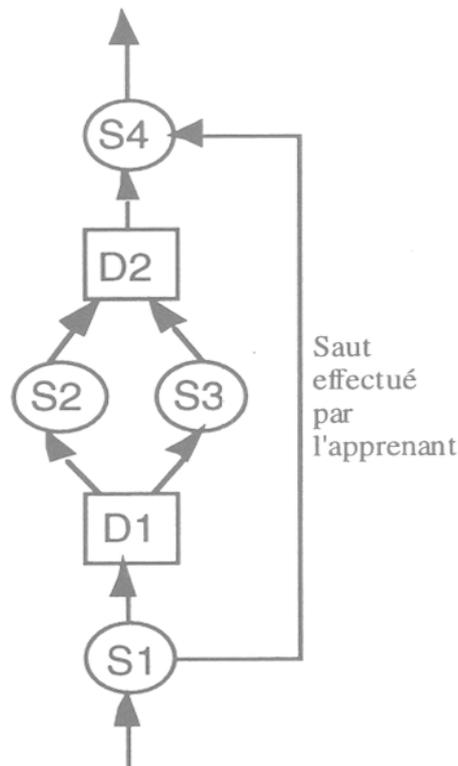


Figure 1. Un exemple de saut dans le raisonnement

Par contre, si le sommet du graphe est atteint sans trouver d'agents correspondant à la recherche, la reconnaissance du raisonnement échoue. L'apprenant a effectué une action non reconnue dans la solution experte concernée. Cet échec de la reconnaissance du raisonnement doit alors être exploité par une stratégie tutorielle.

A l'opposé, si le raisonnement de l'apprenant est reconnu, il faut dans le cas où l'apprenant maîtrise effectivement les connaissances qu'il contient, le modéliser par un agent R. C'est l'instanciation de cet agent R, c'est-à-dire la définition de la connaissance qu'il représente et des accointances et croyances qu'il contient que nous allons aborder maintenant.

4.3. Instanciation de l'agent R

Nous sommes dans le cas où le raisonnement de l'apprenant est reconnu. Il convient alors d'examiner tous les agents pour déterminer si l'apprenant maîtrise effectivement les connaissances représentés par les agents sautés. L'agent demandeur se met en attente de réponse après avoir lancé une demande d'information sur la liste des agents sautés. Cette information est demandée directement au sommet qui la dirige alors vers l'ensemble de ses fils à l'exception de celui qui est l'ancêtre de l'agent demandeur. Un agent qui reçoit un message de demande d'informations répond s'il est concerné, c'est-à-dire si la connaissance statique ou dynamique qu'il modélise correspond à celle du demandeur. Dans ce cas, il communique ses valeurs de croyance à l'agent demandeur par l'intermédiaire du sommet. Nous sommes maintenant dans la phase descendante de l'algorithme, si l'agent n'est pas concerné, il se contente de transmettre la demande à ses fils. Ainsi, de proche en proche, tous les agents susceptibles de fournir une information sont interrogés. L'agent demandeur reçoit ainsi des informations concernant tous les agents qui modélisent les mêmes connaissances que celles des agents qui composent le raisonnement de l'apprenant. La création de l'agent R ne se fait que lorsque des informations ont été reçues sur tous les agents sautés. Il s'agit alors de traiter les informations reçues, et d'attribuer connaissances, accointances et croyances à l'agent R :

4.3.1. Calcul de la connaissance

La connaissance associée au nouvel agent R créé correspond à l'agrégat des connaissances modélisées par les différents agents que R va généraliser. La granularité de cette connaissance est donc supérieure à celle des autres agents ce qui correspond bien à l'idée que l'on se fait des connaissances d'un expert par rapport aux connaissances d'un novice.

4.3.2. Calcul des accointances

L'agent R nouvellement créé possède lui aussi trois accointances. L'agent qui a déclenché l'algorithme sera le fils, l'agent qui a reconnu le raisonnement sera le père et le sommet de l'agent R est le même que celui de l'agent fils et de l'agent père, puisque ces trois agents appartiennent à la même solution.

4.3.3. Calcul des croyances

Comme un agent R intervient à l'intérieur d'un seul raisonnement, la croyance globale de tous les agents que généralise l'agent R est la même. La croyance globale attribuée à l'agent R est donc définie comme étant égale à cette croyance.

Par ailleurs, un agent R remplaçant une partie de raisonnement, on lui attribue une valeur de croyance relative égale à la somme des croyances relatives des agents qu'il remplace.

Enfin, pour déterminer la valeur de croyance propre associée au nouvel agent R, il convient tout d'abord de calculer les croyances propres des agents X_1, X_2, \dots, X_n que R remplace. Il est alors aisé de calculer la valeur de C_p pour l'agent R en pondérant les différents $C_p(X_i)$ par l'importance relative des agents X_i dans la solution :

$$C_p(R) = \frac{\sum_i Cr(X_i)C_p(X_i)}{\sum_i Cr(X_i)}$$

Notre modèle de l'apprenant permet ainsi de représenter à la fois les connaissances primitives de l'apprenant par l'intermédiaire des agents S et D, mais aussi et surtout, il offre deux possibilités de représentation des raisonnements de l'apprenant avec deux niveaux de granularité différents, modélisant la maîtrise et le degré d'expertisation de l'apprenant sur une partie du domaine enseigné.

5. Éléments d'implémentation

Le système TREMMA, acronyme pour TRansfert d'Expertise avec un Modèle Multi-Agents, est écrit en Smalltalk et se base sur la plate-forme de programmation Epitalk permettant de décrire et de générer des systèmes épiphytes. Nous décrivons tout d'abord brièvement l'environnement de programmation, puis le système en lui-même et enfin nous donnons les deux exemples d'expertises avec lesquelles nous avons testé TREMMA.

5.1. Environnement de programmation

Notre environnement de programmation est la plate-forme Epitalk. Epitalk est le terme d'une succession de couches au dessus de Smalltalk-80 et notre système TREMMA hérite automatiquement d'un ensemble de propriétés issues de ces plates-formes successives.

5.1.1. Smalltalk-80

C'est un des précurseurs des langages à objets [GOL 83], dont il reste l'une des références. C'est à la fois un langage de programmation, un environnement graphique et un système d'exploitation. Il est principalement utilisé dans les applications industrielles comme un outil de prototypage d'applications. La raison essentielle du

succès de Smalltalk vient de la conception du langage qui permet la réutilisation effective de classes et de méthodes existantes.

5.1.2. *Actalk*

Le langage Actalk écrit par Jean Pierre Briot [BRI 88] est conçu comme une couche acteur au dessus de Smalltalk-80. L'avantage d'un tel environnement de programmation est de proposer de programmer avec des acteurs tout en disposant de toutes les facilités offertes par l'environnement Smalltalk. Pour transformer les objets de Smalltalk-80 en acteurs, l'auteur utilise deux techniques de programmation, d'une part pour rendre les objets actifs et d'autre part pour transformer les communications entre ces différents objets actifs et les rendre asynchrones. Un acteur est ainsi composé de deux objets, l'un décrivant son comportement (*behavior*) et l'autre sa boîte aux lettres (*mailbox*). Par ailleurs, une pseudo-variable *aself* permet de référencer l'acteur au niveau de son comportement.

5.1.3. *ReActalk*

Smalltalk-80 sert de niveau de base. Au niveau de la couche Actalk, les capacités des objets passifs de Smalltalk-80 sont redéfinies pour en augmenter l'autonomie et pour obtenir une sémantique asynchrone de la transmission de messages. Les objets passifs accèdent au statut d'acteurs. Cette approche ne permet toutefois pas les modifications dynamiques du comportement d'un acteur pris isolément. Sa vie durant, un acteur reste lié à sa classe, et par conséquent se comporte en stricte conformité avec les connaissances et comportements hérités. La couche ReActalk [GIR 93] introduit la réflexivité et complète l'autonomie en libérant les acteurs de la hiérarchie d'héritage. La réflexivité autorise les changements, même temporaires, du comportement par l'intermédiaire d'un méta-acteur.

5.1.4. *Epitalk*

C'est le dernier maillon de la chaîne d'évolution successive de Smalltalk-80 que nous utilisons. Epitalk [PAQ 94] permet de générer des systèmes épiphytes multi-agents, c'est-à-dire des systèmes qui se greffent sur une application existante de façon à l'espionner sans en modifier le comportement. Ainsi, tandis que l'apprenant travaille dans l'application hôte, le système épiphyte garde dans le même temps une trace de l'ensemble des interactions.

Pour implémenter notre système, les différents agents S et D sont définis comme des agents de Epitalk, c'est-à-dire qu'ils sont actifs et peuvent communiquer avec des agents avec lesquels ils sont en relation. Ces relations entre les différents agents forment un graphe isomorphe au graphe de raisonnement. Pour réaliser l'espionnage de l'apprenant, Epitalk utilise un autre type d'agent : les agents *espions*. Ces derniers recueillent les informations concernant les actions effectuées par l'apprenant dans le système hôte et les transmettent aux agents S ou D correspondant.

5.2. *Le système TREMMA*

La programmation de notre modèle consiste en la définition de la classe générique des agents du système décrivant les connaissances des agents comme des variables

d'instances de cette classe, et en la définition des méthodes correspondant aux comportements. De même, il convient de définir l'organisation des agents au sein de la société. Les comportements et les connaissances de cette société sont décrits au travers des variables d'instances et des méthodes appartenant à la classe définissant le modèle générique de société.

Un agent TREMMA est principalement caractérisé par les connaissances propres qu'il contient. En particulier, quelle que soit l'application pour lequel il va être utilisé, un agent de notre modèle est une instance de la classe *TREMMA* et il doit posséder un nom (*name*), un identificateur unique (*id*) et un ensemble d'acointances (*acointances*). Ces différentes informations sont stockées dans le comportement de l'agent. Chaque agent possédant également une boîte aux lettres. Comme les agents TREMMA bénéficiaient de l'héritage des acteurs Actalk et agents ReActalk, une société TREMMA est une descendante des sociétés d'agents définies dans ReActalk et dont la caractéristique principale est d'être un écosystème qui peut se définir par un ensemble d'agents structuré selon un graphe organisationnel représentant le réseau des relations entre les agents. Cette notion d'écosystème, telle qu'elle existe déjà en ReActalk convient parfaitement pour représenter une société TREMMA.

L'application au transfert d'expertise utilise le modèle multi-agents générique. La programmation consiste donc à enrichir les classes génériques de TREMMA. Cet enrichissement se fait en créant les classes d'agents et en introduisant les connaissances et les comportements spécifiques au transfert d'expertise. Dans le cadre d'une application de transfert d'expertise, il est nécessaire de créer les classes correspondant aux agents S, D et R qui sont des sous-classes de *Tremma*. Toutefois, dans un souci de simplification de la programmation, il est intéressant de factoriser des comportements communs dans une classe intermédiaire *TremmaTE*.

Il convient aussi de spécifier les connaissances d'un agent : ainsi, par exemple, la variable générique *acointances* de l'agent Tremma peut être décrite comme une liste de trois variables. On a donc *acointances* := (*pères*, *fil*s, *soc*). D'autres connaissances spécifiques d'un agent sont introduites ici et vont dépendre du type d'agent. Par exemple, un agent D reçoit des variables *entree* et *sortie* qui représenteront les agents S qu'il va accepter en entrée et en sortie. C'est également à ce niveau que sont implémentés les comportements génériques des agents et de la société liés au transfert d'expertise, mais pas à l'expertise elle-même. On retrouve en particulier ici tous les comportements implémentant les algorithmes de recouvrement distribué, de modification des croyances par mise à jour et par propagation et d'expertisation.

5.3. Exemples d'expertises utilisées

Deux expertises ont pour le moment été utilisées pour tester le modèle. La première concerne la géométrie et nous a permis de construire un modèle de l'apprenant avec des agents S et D, et la seconde le diagnostic d'un jeu de tarot permettant d'illustrer la construction d'un R-agent. Les applications développées autour de ces expertises sont très restreintes et ne constituent pas de véritables systèmes tuteurs mais elles permettent néanmoins de valider les hypothèses prises quant à la représentation de l'apprenant par un système multi-agents. Une validation

sur une expertise plus complexe est actuellement à l'étude et devra ensuite être expérimentée sur de vrais apprenants.

5.3.1. *Géométrie*

L'objectif est de faire construire des figures à l'apprenant uniquement à l'aide de la règle et du compas. L'apprenant évolue dans une interface, issue de [AUB 90], lui permettant de construire point, droite et cercle et le système lui propose une construction à réaliser.

Nous nous plaçons dans le cas où l'apprenant en est à sa première séance de travail. Tous les agents ont alors un coefficient de croyance inconnu. L'exercice proposé est celui de la construction d'un carré à partir de deux de ses points.

Voici trois solutions données par des experts en géométrie :

- Construire 1 cercle et 3 droites perpendiculaires.
- Construire 2 cercles et 2 droites perpendiculaires.
- Construire 3 cercles et 1 droite perpendiculaire.

Dans ce cas, le système multi-agents représente trois graphes de raisonnement. Chacun de ces graphes de raisonnement étant lui-même composé de plusieurs agents. Si la première action de l'apprenant est la construction d'un cercle de centre A passant par B, trois agents seront activés, puisque dans chacune des solutions, il faut construire un cercle de ce type. Par la suite, les actions effectuées par l'apprenant dans l'application de géométrie sont reconnues par le système comme concourant à l'une des trois solutions. Si après avoir construit la perpendiculaire à (AB) en A, le point D, puis le cercle de centre D passant par A, l'apprenant se trouve alors dans une situation lui permettant de terminer son carré par la construction d'un cercle de centre B passant par A ou d'une perpendiculaire. Cette double possibilité est modélisée, sans choix à priori, dans notre modèle par le fait qu'un agent D activé par l'apprenant permettra d'obtenir le quatrième sommet du carré.

5.3.2. *Tarot*

Il s'agit d'une expertise simplifiée sur l'évaluation d'un jeu de tarot : la méthode enseignée demande à l'apprenant d'évaluer son jeu en fonction de plusieurs critères afin de choisir son enchère. L'apprenant doit évaluer successivement plusieurs paramètres avant de proposer une annonce. Cette méthode est très didactique. Chacune des évaluations à effectuer par l'apprenant est décomposée en évaluations plus simples : ainsi l'évaluation d'un jeu consiste à évaluer les atouts et les couleurs, et l'évaluation d'une des quatre couleurs consiste en l'évaluation de la force et de la longueur de cette couleur.

Lorsque l'apprenant progresse dans la maîtrise de l'expertise, il va naturellement sauter des étapes de raisonnement. Par exemple, il pourra arriver à une évaluation globale de la couleur en sautant la décomposition force-longueur. Ainsi après une évaluation décomposée et réussie des trèfles et des carreaux, l'apprenant peut évaluer globalement les coeurs. Dans ce cas, une expertisation est nécessaire et se traduit dans notre système par la création d'un nouvel agent R correspondant à l'évaluation globale d'une couleur. Avant de créer cet agent R, il convient tout d'abord d'identifier la partie de raisonnement sauté par l'apprenant, et de vérifier que les agents

modélisant cette partie de raisonnement représentent des connaissances maîtrisées par l'apprenant. Ensuite le nouvel agent R est créé et pourra être directement utilisé par la suite pour modéliser un raisonnement de l'apprenant.

6. Conclusion

Cet article propose une architecture originale de type multi-agents pour modéliser les connaissances et les raisonnements d'un apprenant.

Un modèle d'agent définit les connaissances, les comportements et les interactions de chaque agent, ainsi que l'évolution dynamique de ses connaissances et de ses comportements au sein de la société. Ce modèle d'agent spécifie les connaissances d'un agent grâce à des types d'agents (S, D ou R) ainsi que ses comportements. Un modèle de société d'agents est également défini. Il décrit la structure des agents dans un graphe de raisonnement et l'évolution de la structure par un algorithme d'expertisation.

La modélisation de l'apprenant présentée dans cet article concerne la résolution de problèmes par l'apprenant. Un transfert d'expertise ne se limite pas à la résolution de problèmes. Il faut prévoir d'enchaîner différents problèmes au sein de sessions, puis ces sessions au sein d'un cursus. Cette organisation du transfert d'expertise en trois niveaux hiérarchiques permet de montrer un autre avantage de notre modèle : son auto-similarité. En effet, les systèmes multi-agents décrits dans cet article et représentant la résolution d'un problème sont alors vus comme des agents primitifs au niveau session. Ces agents possèdent alors les mêmes caractéristiques que les agents S et D et il est donc aisé de construire de manière similaire un modèle de l'apprenant au niveau d'une session puis d'un cursus.

L'utilisation du modèle par des stratégies tutorielles est une autre prochaine étape de la recherche. Deux types d'interventions sont envisagées dans un premier temps. Une première stratégie tutorielle concerne le cas où l'apprenant ne progresse plus dans la résolution du problème (c'est-à-dire qu'il n'a ni progressé dans aucune des solutions, ni exploré aucune autre solution référencée). Dans ce cas, la stratégie consiste à proposer à l'apprenant l'étape ou les étapes suivantes possibles. Une seconde stratégie consiste à proposer à l'apprenant une aide pas à pas, en lui offrant les prochaines étapes de résolution de chacune des solutions possibles.

7. Bibliographie

- [AUB 90] M. AUBÉ, Utilisation d'un langage orienté objets pour la compréhension de concepts de géométrie, Congrès de l'AMQ, Collège de Sherbrooke, 20 octobre 1990.
- [BRI 88] JP. BRIOT, Actalk : une plateforme de modélisation de langages d'acteurs en Smalltalk-80, Rapport de Recherche, Université de Paris VI, Paris, 1988.
- [CAR 77] B. CARR, I. GOLDSTEIN, *Overlays, a theory of modelling for CAI*, AI memo 406, MIT Press, 1977.
- [CAL 94] G. MC CALLA, J. GREER, Granularity-Based Reasoning and Belief Revision in Student Models, *Student Models : The key to Individualized Educational Systems*, pages 39-62, Springer-Verlag, New-York, USA, 1994.

- [GIR 93] S. GIROUX, Agents et Acteurs : une nécessaire unité, Thèse de l'Université de Montréal, Mars 1993.
- [GOL 83] A. GOLDBERG, D. ROBSON, Smalltalk-80. *The Language and its Implementation*, Addison Wesley, 1983.
- [GRE 94] J. GREER, G. MC CALLA, *Student Models : The key to Individualized Educational Systems*, Springer Verlag, 1994.
- [HOL 94] P. HOLT, S. DUBS, M. JONES, J. GREER, The state of Student Modelling, *Student Models : The key to Individualized Educational Systems*, pages 39-62, Springer-Verlag, New-York, USA, 1994.
- [IKE 93] M. IKEDA, Y. KONO, R. MIZOGUCHI, Nonmonotonic Model Inference. A Formalization of Student Modeling, *IJCAI'93, 13th International Joint Conference on Artificial Intelligence*, Pages 467-473, Chambéry, France, 1993.
- [LEM 93] S. LEMAN, P. MARCENAC, Modélisation sous forme d'agents et implémentation en acteurs d'une expertise arithmétique complexe, Actes de la Journée Systèmes Multi-Agents du PRC-IA, Montpellier, France, 17 Décembre 1993.
- [LEM 95] S. LEMAN, S. GIROUX, P. MARCENAC, A Multi-Agent Approach to Model Student Reasoning Process, *7th World Conference on Artificial Intelligence in Education*, pages 258-265, Washington D.C. , USA, 16-19 Août 1995.
- [LEM 96] S. LEMAN, P. MARCENAC, S. GIROUX, Maintien dynamique de la cohérence d'un modèle de l'apprenant par coopération d'agents cognitifs, Actes de la Journée Systèmes Multi-Agents du PRC-IA, Pages 93-103, Toulouse, 2 février 1996.
- [LEM 96] S. LEMAN, TREMMA : Transfert d'Expertise avec un Modèle Multi-Agents, Thèse de l'Université de La Réunion, 24 Juillet 1996.
- [LEV 94] S. LEVESQUE, C. FRASSON, J. GESCEI, Perspectives multi-agents des systèmes tuteurs intelligents , Actes des 2° Journées Francophones IAD & SMA, Pages 55-66, Voiron, France, 9-11 Mai 1994.
- [MIN 85] M. MINSKY, *The Society of Mind*, Simon and Schuster, 1985.
- [NIC 88] JF. NICAUD, M. VIVET, Les tuteurs intelligents : réalisations et tendances de recherches, *Technique et Science Informatiques*, Volume 7, numéro 1, Pages 21-45, 1988.
- [PAQ 94] G. PAQUETTE, F. PACHET, S. GIROUX, EpiTalk, un outil générique pour la construction de systèmes conseillers, *Sciences et Techniques Educatives*, Vol 1, n° 3, Pages 305-336, Novembre 1994.
- [SEL 87] J. SELF, Student models : what use are they ? Artificial Intelligence tools in Education, *proceedings of the IFIP TC3 working conference on Artificial Intelligence tools in Education*, Frascati, Italie, mai 1987.

Stéphane Leman est docteur en informatique de l'université de la Réunion depuis juillet 1996. Ses travaux de recherches, menés à l'Institut de REcherche en Mathématiques et Informatique Appliquées (IREMIA), s'intéressent aux apports d'une approche multi-agents pour les systèmes tuteurs intelligents et plus particulièrement pour la modélisation de l'apprenant. Sa thèse a principalement débouché sur la définition d'un modèle multi-agents pour la représentation d'une expertise en vue de son transfert.

Pierre Marcenac est actuellement maître de conférences à l'université de la Réunion, où il travaille depuis 1992 à l'élaboration de plates-formes à base d'agents. Ses centres d'intérêt sont la modélisation des processus de raisonnements des apprenants en phase de résolution de problèmes et la modélisation de systèmes complexes.

Sylvain Giroux est actuellement professeur substitut à la Télé-université, université du Québec. Après avoir obtenu un Ph. D. en informatique à l'université de Montréal en 1993, il a fait ses études postdoctorales à l'université de la Réunion où il a travaillé sur l'adaptation dans les systèmes ouverts et sur la simulation d'édifices volcaniques à l'aide de systèmes multi-agents. Ses travaux de recherches actuels portent sur les systèmes conseillers épiphytes multi-agents et sur l'utilisation des hypermédias dans la réalisation de cours pour la formation à distance.