SIGFAD : un système multi-agents pour soutenir les utilisateurs en formation à distance

Aloys Mbala, Christophe Reffay, Thierry Chanier

▶ **To cite this version:**

Aloys Mbala, Christophe Reffay, Thierry Chanier. SIGFAD : un système multi-agents pour soutenir les utilisateurs en formation à distance. Environnements Informatiques pour l'Apprentissage Humain 2003, Apr 2003, Strasbourg, France. pp.329-330. edutice-00000147

HAL Id: edutice-00000147
https://edutice.hal.science/edutice-00000147v1

Submitted on 4 Nov 2003

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.

SIGFAD : un système multi-agents pour soutenir les utilisateurs en formation à distance

Aloys Mbala*, Christophe Reffay*, Thierry Chanier*

** LIFC – UFR ST*

16 route de Gray

25030 Besançon Cedex

mbala@lifc.univ-fcomte.fr

Christophe.Reffay@univ-fcomte.fr

Thierry.Chanier@univ-fcomte.fr

RESUME. *Cet article décrit la construction de SIGFAD, un système multi-agents destiné à être couplé aux plates-formes informatiques de formation à distance. SIGFAD implémente des fonctionnalités permettant aux tuteurs de maintenir les groupes, les dynamiser et de bien conduire la session de formation à distance (FAD). Nous utilisons une méthodologie d'analyse, de conception et de spécification des systèmes multi-agents (méthodologie MaSE) et nous appuyons sur l'architecture BDI (Beliefs-Desires-Intentions) pour mesurer un certain nombre de critères sur les groupes et les individus engagés dans une session de FAD. Ces critères, bien que construits à posteriori ici, peuvent être calculés au fil d'une session de FAD et mis à la disposition des utilisateurs en temps réel. Nous appliquons SIGFAD à un corpus de données recueillies lors d'une session de FAD grandeur réelle et exhibons quelques résultats.*

MOTS CLES : *formation à distance, apprentissage collaboratif, interactions, EIAH, systèmes multi-agents.*

1 Introduction

Notre recherche concerne la formation à distance (FAD), plus spécifiquement le soutien des interactions dans des groupes de taille restreinte (8 à 15 personnes). Dans ce contexte, nous nous intéressons à la construction d'outils assistant le tuteur du groupe et le coordonnateur de la session de FAD souvent en charge de la cohésion, du dynamisme et de l'existence même des groupes. Partant d'une démarche expérimentale, nous avons mené une expérience en grandeur réelle de formation entièrement à distance et en ligne [CHANIER 2001] [MBALA 2002]. Ceci nous a donné l'occasion d'observer plusieurs phénomènes, d'identifier des problématiques spécifiques à ce mode de formation et de recueillir un corpus important de données qui nous permettent aujourd'hui de définir et de répondre à plusieurs questions de recherche. Dans cet article, nous décrivons un système multi-agents (SMA) destiné à être couplé à des plates-formes de formation à distance afin d'y implémenter des fonctionnalités permettant d'apprécier l'état du groupe : personnes présentes, absentes, dormantes ; l'état du groupe en fonction du pourcentage de personnes actives ; la productivité d'un utilisateur donné ; le niveau de réalisation d'une activité donnée. Ayant choisi de mettre en œuvre le paradigme agent, nous montrons ici à travers des choix particuliers d'architecture d'agents, de méthodologie d'analyse, de conception et de spécification ; qu'il est possible de construire un tel système et nous l'instancions sur le corpus de données recueillies lors de cette expérimentation.

2 Fondements théoriques

Partant de la définition que donne Ferber d'un agent logiciel [FERBER 1995], on peut définir cette entité comme un système informatique situé dans un environnement ; capable de mener de manière autonome des actions sur cet environnement en vue d'accomplir ses objectifs ; possédant en plus les propriétés de réactivité : il perçoit des stimuli provenant de son environnement et réagit en fonction de ceux-ci ; de proactivité : il est mû par un certain nombre d'objectifs qui guident ses actions ; de sociabilité : il communique avec d'autres agents ou des humains et, peut se trouver engagé dans des transactions sociales (négocier ou coopérer pour résoudre un problème) afin de remplir ses objectifs.

Un examen des fonctionnalités des systèmes informatiques dédiés à la FAD montre que finalement, ces systèmes sont ouverts, complexes, évolutifs. Un système ouvert est un système dont la structure peut changer dynamiquement. Ses composants ne sont pas connus à l'avance, changent au cours du temps, et sont essentiellement hétérogènes (en raison du fait qu'ils sont mis au point par différentes personnes, à différentes périodes, utilisant des techniques et outils de développement différents). Le système ouvert le plus connu est Internet. La FAD médiatisée par ordinateur s'appuie essentiellement sur Internet et exige le développement de technologies y

relatives ; il est évident que cette exigence induit de la part des informaticiens la mise en place de systèmes ouverts. La FAD parce qu'elle se trouve au carrefour de plusieurs disciplines de sciences humaines (psychologie, sciences de l'éducation, sociologie, etc.) et des sciences de l'ingénieur (informatique, télécommunications) pose des problèmes complexes, diversifiés, interdépendants et difficilement prévisibles. Les outils adaptés pour faire face à la complexité dans le développement logiciel sont la modularité et l'abstraction. Le paradigme d'agent est un bon outil de modularisation, il permet de résoudre les problèmes posés par la FAD en développant des modules spécialisés (en termes de représentation et de type de problème résolu) dans un aspect particulier. L'interdépendance des problèmes de FAD est surmontée grâce à la coopération entre les agents mis en œuvre.

La FAD en ligne exige de prendre en compte l'hétérogénéité des sources d'informations, des systèmes d'information et des centres de régulation des activités (au niveau de l'apprenant, du tuteur, du formateur, du concepteur, du coordonnateur de la formation) ; l'ouverture liée à l'ajout/retrait de sous-systèmes (logiciels, matériels, acteurs) sans stopper ou réinitialiser le système global ; l'évolutivité pour s'adapter aux changements continuels de l'environnement dans lequel s'inscrit la FAD ; la réactivité pour se configurer en fonction de l'évolution des activités à mettre en œuvre, en fonction des compétences disponibles et des objectifs d'apprentissage ; la coopération entre les différents sous-systèmes pour produire des contenus pédagogiques, administrer, assister, évaluer les apprenants ; la coordination des différentes plates-formes de FAD qui ne manqueront pas de se regrouper en environnements de campus virtuel. Dans ce contexte, l'informatique sur laquelle s'appuie la FAD doit intégrer de nouveaux concepts, en s'appuyant sur des technologies fortement liées à Internet, afin de proposer des architectures logicielles qui permettent la coopération de multiples applications hétérogènes et distribuées.

3 L'expérimentation SimuLigne

Nous avons conduit une expérimentation de FAD qui s'est déroulée du 30 avril au 06 juillet 2001. Le cours était organisé autour d'une simulation globale, comparable à celle que les enseignants de langue utilisent dans leurs classes. Il s'agissait pour les apprenants répartis dans quatre groupes de participer à un concours virtuel dont la ville gagnante accueillerait des étudiants de l'Open University en stage d'été. Chaque groupe était chargé de rédiger une candidature et de faire vivre ses personnages. La plate-forme de télé-formation utilisée était WebCT™ ; dans la suite, nous nous y referons sous l'appellation de PIFAD (plate-forme informatique de formation à distance). SimuLigne s'est déroulée entièrement à distance et en ligne, et la production dans chaque groupe s'est faite de manière collaborative.

Il est démontré dans [MBALA et al. 2002] les insuffisances de PIFAD et des plates-formes existantes de télé-enseignement pour la mise en œuvre de formations telles que SimuLigne. Une analyse des statistiques d'interaction permet de justifier

les fonctionnalités supplémentaires dont il faudrait doter les plates-formes de télé-enseignement afin qu'elles intègrent des outils d'assistance aux tuteurs et au coordonnateur, en charge de la gestion et de la maintenance des groupes. Les statistiques d'interaction peuvent permettre, en s'appuyant sur des théories comme l'analyse des réseaux sociaux, de modéliser et de visualiser la collaboration dans des groupes de FAD [REFFAY & CHANIER 2002]. Notre expérience en FAD nous a donné l'occasion de constater que cette simple visualisation peut être déterminante pour le tuteur et le coordonnateur dans leurs prises de décision.

En définitive, les statistiques d'interaction permettent d'apprécier la vie du groupe et son évolution. Nous sommes plus spécialement intéressés par les indicateurs de l'état de progression et de durabilité du groupe ; l'identification d'une surcharge de travail pour un apprenant donné afin éventuellement, de le dispenser d'effectuer certaines activités ; la possibilité d'allonger ou de raccourcir les délais de réalisation d'une activité (modifier le calendrier de la formation) ; l'appréciation de l'état de réalisation d'une activité ou d'une tâche ; l'évaluation de la sociabilité et de la productivité d'un apprenant, etc. En fin de compte, il s'agit d'identifier des outils d'aide qui pourraient être couplés à la plate-forme afin de juguler nombre de problèmes spécifiques à la FAD. Nous avons choisi de proposer ces outils d'aide sous la forme d'un système multi-agents implémentant un ensemble de fonctionnalités supplémentaires dont nous doterions les plates-formes de FAD existantes.

4 Les outils de développement

4.1 La méthodologie MaSE et l'environnement agentTool

Le développement de SIGFAD, notre système multi-agents (SMA) exige plusieurs outils de développement permettant de passer en revue toutes les étapes du cycle de développement logiciel. En premier lieu, il convient de disposer d'une méthodologie d'analyse, de conception et de spécification du SMA. Pour cela, nous avons eu recours à la méthodologie MaSE (Multiagent System Engineering) mise au point par Scott DeLoach et ses collègues au sein du Laboratoire d'Intelligence Artificielle de l'AFIT (Air Force Institute of Technology) [DELOACH et al. 2001]. MaSE est couplée à agentTool, un environnement de développement permettant de l'implémenter [DELOACH 2001]. Cet environnement offre l'avantage de vérifier les dialogues entre les agents afin d'éviter des situations d'interblocage (*deadlock* ou *livelock*) et peut générer du code.

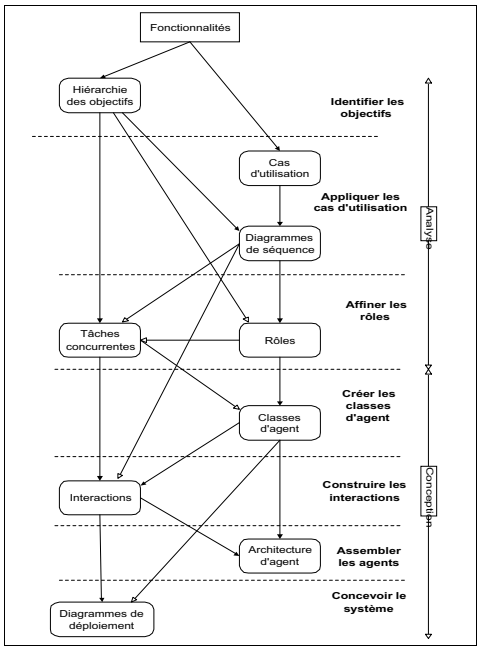

Figure 1. *La méthodologie MaSE*

MaSE comporte sept étapes regroupées en deux phases : une phase d'analyse et une phase de conception. Le but de la phase d'analyse est de produire un ensemble de rôles que le système doit remplir pour atteindre ses objectifs. Un rôle est une entité accomplissant une tâche dans le système ; chaque rôle est chargé d'accomplir un objectif ou un sous-ensemble d'objectifs du système. La phase de conception a pour but de passer à la construction concrète du SMA en assignant des objectifs à des classes d'agents spécifiques ; en construisant les communications entre agents ; en proposant une architecture interne des agents où sont implémentées les processus de raisonnement ; en définissant le nombre et la localisation des agents dans le système.

4.2 L'architecture BDI et le modèle JAM

MaSE n'offre pas une architecture interne des agents. Elle permet de bien définir et de spécifier les interactions dans lesquelles les agents sont engagés. Pour la construction de l'intérieur des agents (« Assembler les agents » de la phase de conception de MaSE), il existe plusieurs architectures d'agent qui se distinguent généralement par le mode de prise de décision [WOOLDRIDGE 1999] :

- les agents logiques où le processus de décision est basé sur la logique déductive ;
- les agents réactifs où le processus de décision est basé sur les événements rencontrés ;
- les agents BDI où le processus de décision est basé sur la manipulation des structures de données représentant les états mentaux (croyances, désirs, intentions) de l'agent ;
- les architectures à couches où le processus de décision est basé sur plusieurs couches logicielles qui mènent chacune un raisonnement à propos de leur environnement avec différents niveaux d'abstraction.

Nous avons utilisé le modèle JAM comme architecture des agents de SIGFAD. L'architecture JAM est construite autour d'un noyau BDI (le modèle BDI est basé

sur les états mentaux de l'agent, qui peut ainsi avoir des croyances, des désirs et des intentions. C'est un modèle assez bien accepté car sa compréhension est presque intuitive ; il a d'autre part, fait l'objet de plusieurs travaux de recherche l'architecture interne des agents.) et se base sur les implémentations PRS (*Procedural Reasoning System*) de l'Université de Michigan et de SRI International connues respectivement sous les sigles UMPRS et PRS-CL. Une présentation plus détaillée de l'architecture JAM est disponible dans [HUBER 1999].

4.3 La communication entre les agents et le langage de programmation : les sockets et Java

Afin de permettre à nos agents de s'échanger des messages et des informations, nous avons implémenté la communication entre les agents sous la forme de sockets bien qu'il existe des infrastructures de communication inter-agents comme JATLite [MBALA 2002]. Ceci nous a, du reste, permis de mieux cerner plusieurs aspects de la communication inter-agents et de résoudre différents autres préoccupations liées à l'implémentation réelle de SIGFAD comme la définition des interfaces utilisateur-SMA. Nous avons développé l'ensemble de SIGFAD en Java. Ce choix n'est pas fortuit, le modèle JAM offre en effet, pour chaque agent un squelette initial qu'il appartient au programmeur de compléter avec un ensemble de procédures et de fonctions écrites en Java pour construire ses agents. Java permet d'implémenter les communications inter-processus sous forme de sockets.

5 SIGFAD

Nous présentons dans la figure 2 une vue d'ensemble de SIGFAD, avec les types d'agents présents dans le système. Dans les plates-formes classiques, chaque utilisateur est connecté à son ordinateur et accède aux programmes et à toutes les ressources pédagogiques à travers le protocole HTML et les autres protocoles Internet. L'utilisateur a besoin pour cela d'un navigateur Internet afin d'accéder à la plate-forme et de réaliser ses activités pédagogiques. SIGFAD s'appuie sur la même structure, nous y distinguons trois types d'utilisateurs : les apprenants, les tuteurs, le coordonnateur. Chaque utilisateur a un agent, localisé sur le serveur, cet agent migre sur son poste dès qu'il se connecte. L'agent est codé comme une applet Java ; les applets sont des programmes résidant sur le serveur et qui s'exécutent sur le poste client. Cette technologie permet à un utilisateur d'utiliser son agent quelque soit le poste à partir duquel il accède à la plate-forme. L'agent Superviseur des interactions réside sur le serveur et est chargé des transactions avec la base de données. Bien que nous ayons construit la base de données actuelle de façon manuelle, nous envisageons l'existence d'un agent Constructeur de la base de données. L'agent Coordonnateur est chargé de fournir au SMA des meta-informations sur les activités (date de début, date de fin, personnes concernées, outils, etc.) et sur les forums (date de début, date de clôture, etc.). Les agents communiquent entre eux en utilisant des requêtes KQML (Knowledge Query and Manipulation Language). KQML est un langage d'interrogation et de manipulation de connaissances. Bien que ce langage ne soit pas normalisé, il est devenu une norme officieuse à cause de sa forte utilisation.

KIF (Knowledge Interchange Format) est utilisé pour construire le corps du message KQML. Les utilisateurs communiquent avec leurs agents à travers des interfaces graphiques (GUI pour *Graphic User Interface*). a_SI et a_BDD communiquent avec la base de données en utilisant des requêtes SQL (Structured Query Language).

Nous présentons à la figure 3 le diagramme hiérarchique des objectifs de SIGFAD. C'est la première étape de MaSE dont l'application [MBALA 2002] nous a permis de définir le diagramme des classes d'agent de notre SMA ; ce diagramme est présenté à la figure 4. Les classes d'agent encapsulent les rôles qui seront joués par chaque agent. Le diagramme de classes d'agent est comparable au diagramme de classes d'objet que l'on obtient en programmation orientée-objet. Il donne une idée très précise de l'architecture du SMA. Les classes d'agent de SIGFAD sont les suivantes :

- un agent Apprenant (un pour chaque apprenant) ;
- un agent Tuteur (un pour chaque tuteur) ;
- un agent Coordonnateur ;
- un agent Constructeur de la base de données des interactions ;
- un agent Superviseur des interactions.

SIGFAD fournit trois grandes catégories d'appréciations :

- au niveau du groupe entier : en indiquant les personnes présentes, absentes, dormantes, l'état du groupe par rapport à la réalisation des activités pédagogiques ;
- au niveau d'un individu : en appréciant sa productivité en termes de réalisation des activités pédagogiques, sa sociabilité qui indique son niveau de communication avec les autres membres du groupe ;
- au niveau de l'activité : en indiquant le niveau de réalisation d'une activité par l'ensemble des participants, ceci permet au Coordonnateur de prendre des décisions objectives de modification du calendrier des activités.

Un utilisateur dormant est un utilisateur qui se connecte à la plate-forme sans rien faire d'autre que lire des messages, son passage ne laisse pas de trace pour les autres membres du groupe. Le passage d'un apprenant à l'état dormant doit alerter le tuteur car très souvent, un tel utilisateur se trouve en situation d'échec et abandonne la formation dans les jours suivants. Nous avons défini une personne active comme une personne qui en plus des activités de communication (lire et écrire des messages dans des courriels, forums ou bavardages) réalisent les activités pédagogiques prévues ; ce critère permet d'apprécier la productivité réelle de l'utilisateur qui en général, est à dissocier de la sociabilité. Un utilisateur présent est celui qui s'est connecté à PIFAD au moins une fois pendant la période spécifiée ; un absent est celui qui ne s'est pas connecté.

Les figures 5 et 6 représentent des captures d'écran illustrant les possibilités de SIGFAD, la première indique la liste des utilisateurs présents du groupe 'Aquitania' entre le 23 mai 2001 et le 31 mai 2001 ; la deuxième indique l'état du groupe 'Aquitania' au cours de la même période, ce calcul se fait à partir du nombre de personnes actives dans le groupe, le pourcentage de personnes actives est indiqué. Avec notre SMA, l'on peut aussi apprécier la participation d'un individu ; celle-ci se fait à partir de la réalisation ou non des tâches et activités pédagogiques.

Un des points essentiels de SIGFAD est la construction de la base de données des interactions. Ceci se comprend aisément lorsqu'on se souvient qu'une base de données est construite pour répondre à des besoins spécifiques. La base de données est constituée d'informations issues du fichier des traces du serveur HTTP (Apache dans notre cas) mais aussi de la base de données interne à PIFAD. Du fait de l'absence de normalisation des plates-formes de formation, cette base de données interne ne permettait pas de répondre immédiatement à nos préoccupations. Il a fallu recourir à un travail préalable de compréhension et de recueil des différentes données éparpillées dans PIFAD. Des groupes de travail au sein de l'AFNOR et des organismes de normalisation européens et mondiaux se préoccupent des questions d'interopérabilité (SCORM) entre les plates-formes de formation et d'interfaçage avec les utilisateurs.

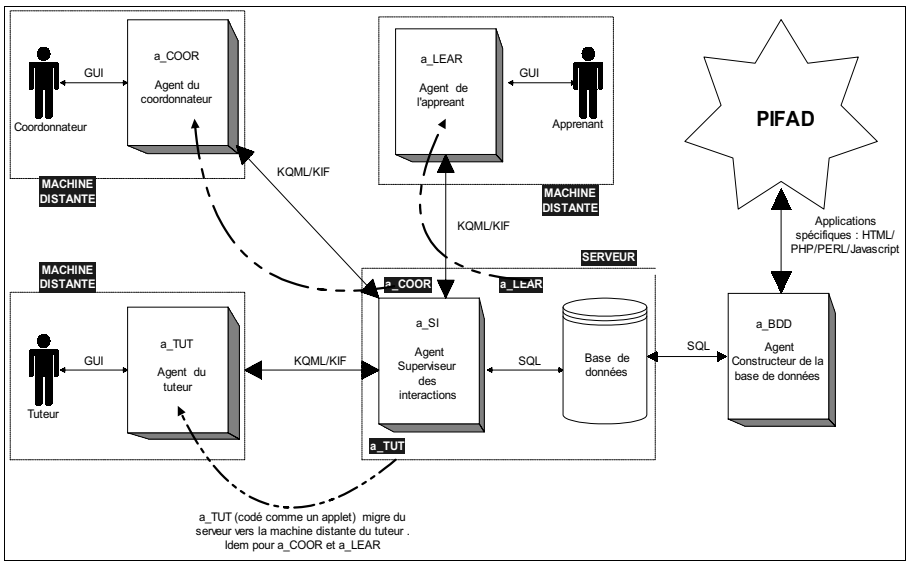

Figure 2. *Une vue d'ensemble de SIGFAD. PIFAD regroupe le fichier des traces du serveur HTTP et la base de données interne.*

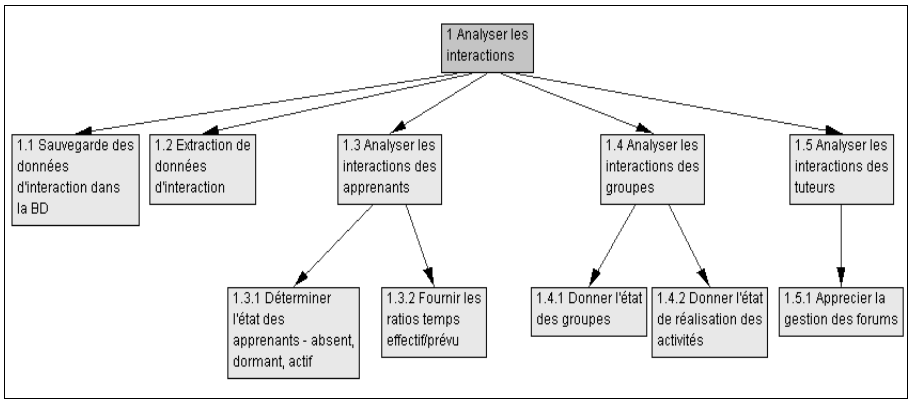

Figure 3. *Le diagramme hiérarchisé des objectifs de SIGFAD*

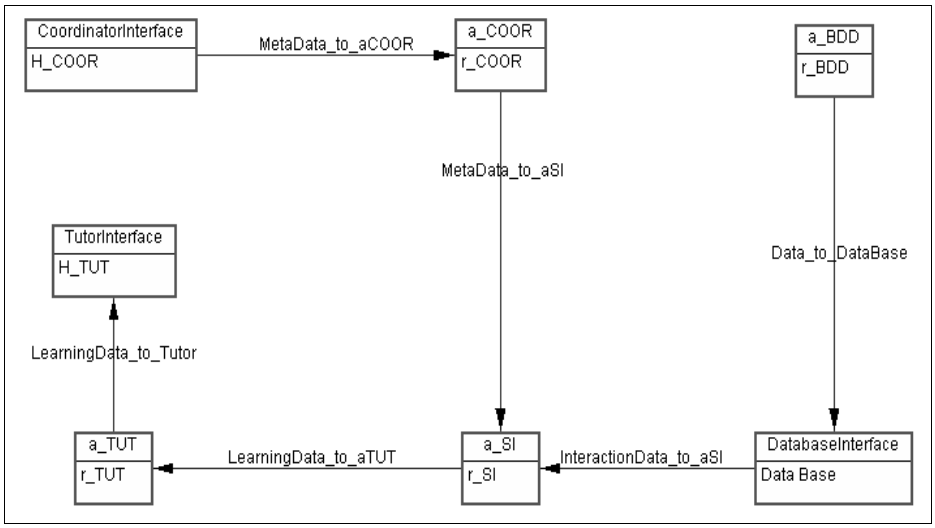

Figure 4. *Le diagramme des classes d'agent de SIGFAD*

6 Travaux futurs et perspectives

L'application de la méthodologie MaSE, du modèle BDI et l'implémentation des mécanismes de communication à travers les sockets nous ont permis de vérifier la faisabilité de la mise en œuvre d'un SMA tel que SIGFAD ; et partant, de valider notre démarche conceptuelle. Notre démarche conceptuelle aura consisté à montrer que le soutien des interactions dans des groupes de FAD exige des systèmes informatiques flexibles, dynamiques et évolutifs. Nous avons donc ensuite identifié

le paradigme agent comment pouvant répondre à ces exigences. Il restait à identifier et à appliquer des outils de développement pour construire un tel système.

Au stade actuel, plusieurs points restent à explorer et à implémenter. Sans être exhaustif, on peut citer l'ajout des caractéristiques d'apprentissage machine qui permettra de rendre SIGFAD plus autonome et proactif ; il serait ainsi capable tout seul de scruter périodiquement les données d'interaction et de déclencher des alertes pour prévenir le tuteur, le coordonnateur et l'apprenant lorsque le groupe court un risque d'éclatement ou est en situation d'échec pédagogique. Un point imminent de notre agenda de recherche concerne la construction en temps réel de la base de données des interactions et l'intégration d'une infrastructure de communication inter-agents comme JATLite et d'un langage de communication entre agents comme KQML ou FIPA-ACL.

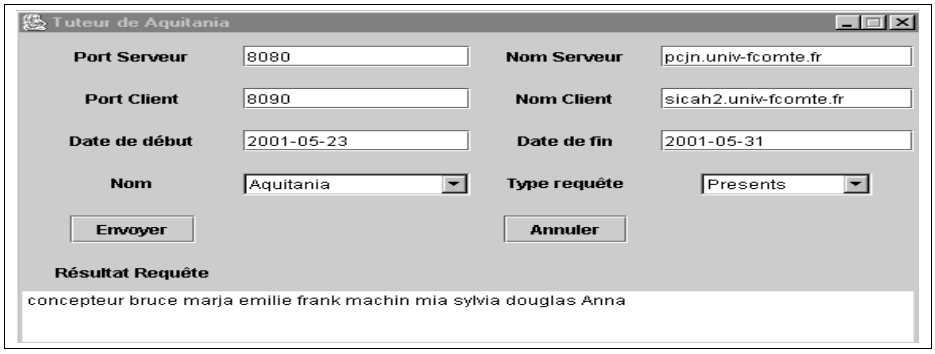

Figure 5. *SIGFAD fournit au tuteur la liste de membres (prénoms) du groupe 'Aquitania' présents entre le 23 mai et le 31 mai 2001*

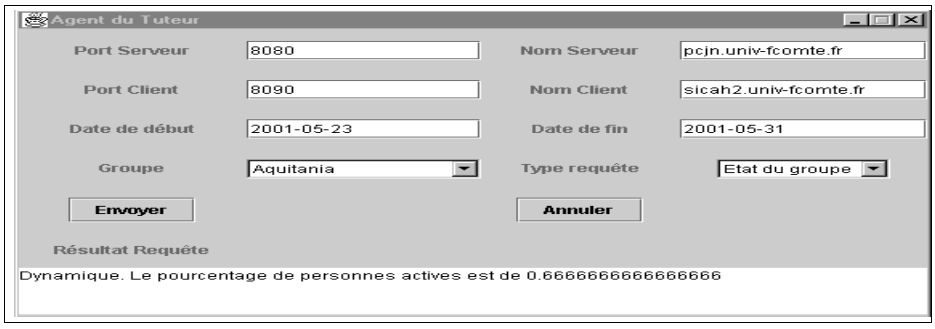

Figure 6. *SIGFAD fournit l'état du groupe 'Aquitania' entre le 23 et le 31 mai 2001*

7 Conclusion

La conduite de SimuLigne a servi à identifier un certain nombre de critères cruciaux pour le bon déroulement d'une session de formation à distance et en ligne basée sur l'apprentissage collaboratif. Nous avons construit un système multi-agents calculant à posteriori ces critères. Ce calcul peut néanmoins être fait en temps réel ; ceci exige de construire la base de données des interactions au fil de la session de formation. La conduite d'une prochaine expérimentation de FAD nous permettra de le faire. Nous avons insisté sur l'assistance à fournir aux tuteurs car l'expérimentation SimuLigne nous a permis d'apprécier la complexité et l'importance de leur tâche. La construction de SIGFAD nous a permis d'identifier et de mettre en œuvre un certain nombre d'outils de développement. Ceci est important car en programmation orientée agent, on rencontre plus souvent des *déclarations d'intention* que des implémentations réelles de systèmes à base d'agents.

Remerciements *: Ces travaux ont été menés dans le cadre du Programme Cognitique 2000 (projet ICOGAD) financé par le Ministère de la Recherche français.*

Bibliographie

[CHANIER 2001] CHANIER, T. "Créer des communautés d'apprentissage à distance". *Les dossiers de l´Ingénierie Educative, N° 36 sur "Les communautés en ligne"*, octobre. Centre National de Documentation Pédagogique (CNDP) : Montrouge (2001) PP56-59.

[DELOACH 2001] DELOACH, S., A.. Analysis and Design using MaSE and agentTool. *The 12th Midwest Artificial Intelligence and Cognitive Science Conference*. 7p

[DELOACH et al. 2001] DELOACH, S. A., WOOD, M. F., SPARKMAN, C. H. (2001). Multiagent Systems Engineering. *International Journal on Software Engineering and Knowledge Engineering*. World Scientific Publishers. Vol. 11 n° 3 (2001). PP 231-258.

[FERBER 1995] FERBER, J., Les systèmes multi-agents : vers une intelligence collective. Intereditions (1995)

[HUBER 1999] HUBER M. J., "JAM: A BDI Theoretic Mobile Agent Architecture". *Proceedings of the Third International Conference on Autonomous Agents (Agents' 99)*, Seattle, WA, May, (1999). PP 236-243.

[MBALA 2002] MBALA, A. "Specifying a Multiagent System to Support Users' Activities in Computer-Supported Distance Learning". *Proceedings of the Starting Artificial Intelligence Researchers Symposium (STAIRS 2002)*. Thierry Vidal, Paolo Liberatore (Eds). IOS Press. (2002). PP 33-34.

[MBALA et al. 2002] MBALA, A., REFFAY, C., CHANIER, T. "Integration of Automatic Tools for Displaying Interaction Data in Computer Environments for Distance Learning ". *6th International Conference of Intelligent Tutoring Systems (ITS'2002)*. S. A. Cerri, G. Gouardères, F. Paraguaçu (Eds). LNCS 2363. Springer. (2002). PP 841-850.

[REFFAY & CHANIER 2002] REFFAY, C., CHANIER, T. "Social Network Analysis used for modelling collaboration in distance learning groups". *Proceedings of the 6th International Conference of Intelligent Tutoring Systems (ITS'2002)*. S. A. Cerri, G. Gouardères, F. Paraguaçu (Eds). LNCS 2363. Springer. (2002). PP 31-40.

[WOOLDRIDGE 1999] WOOLDRIDGE, M. "Intelligent Agents". In G. Weiss (Ed). *Multiagent Systems*, The MIT Press. (1999)