

PROJET ICOGAD

**Interactions et changements COgnitifs dans les Groupes d'Apprentissage à Distance
médiatisé par les systèmes d'information et de communication**

**De l'analyse des données d'interaction de SimuLigne au
développement d'un système multi-agents pour soutenir les
activités des utilisateurs en formation à distance**

Rapport de recherche, Août 2002, Besançon

Aloys Mbala, Thierry Chanier, Christophe Reffay

TABLES DE MATIERES

GLOSSAIRE	4
INTRODUCTION GENERALE	5
CHAPITRE I : FONDEMENTS THEORIQUES.....	9
1 -1 L'IMPORTANCE DU GROUPE ET DES INTERACTIONS DANS L'APPRENTISSAGE COLLABORATIF DISTRIBUE	9
1.2 - LE ROLE DU COORDINATEUR ET DU TUTEUR DANS CE CONTEXTE.....	10
1.3 - L'ECHELLE D'ANALYSE DE LA TACHE DANS NOTRE RECHERCHE	12
CHAPITRE 2 : L'EXPERIMENTATION SIMULIGNE	13
2-1 LES INSUFFISANCES DES PLATES-FORMES DE TELE-FORMATION SUR LE PLAN CONCEPTUEL	13
2-2 L'ANALYSE DES DONNEES D'INTERACTION.....	15
2-3 CONCLUSION	22
CHAPITRE III : LA SPECIFICATION DE SIGFAD	23
3 -1 UN SURVOL DE LA METHODOLOGIE MASE.....	23
3-2 SIGFAD	26
3-4 CONCLUSION	50
CHAPITRE IV : LA PROGRAMMATION DE SIGFAD	51
4-1 L'ARCHITECTURE JAM.....	51
4-2 JATLITE ET KQML	53
4-3 LES AGENTS DE SIGFAD.....	56
4-4 CONCLUSION	57
CONCLUSION GENERALE	58
REFERENCES	61

GLOSSAIRE¹

- **ACAO** : Apprentissage Collaboratif Assisté par Ordinateur
- **AFIT** : Air Force Institute of Technology
- **BDI** : Beliefs–Desires–Intentions. L’architecture BDI est une des architectures d’agent existantes. Elle est basée sur les états mentaux de l’agent qui a ainsi des croyances (beliefs), des désirs (desires) et des intentions (intentions). C’est ce modèle d’architecture que nous avons retenue pour décrire la structure interne de nos agents.
- **FAD** : Formation A Distance
- **FIPA** : Foundation for Intelligent Physical Agents. C’est une fondation qui travaille au développement des applications, des équipements et des services liés au paradigme d’agent.
- **FIPA ACL** : Langage de communication entre agents développé par le FIPA. Il est jugé plus performant que le KQML par le nombre de performatifs qu’il offre, sa structure et sa robustesse.
- **FLE** : Français Langue Etrangère
- **FOAD** : Formation Ouverte et A Distance
- **ICOGAD** : Interactions et Changements cognitifs dans les Groupes d’Apprentissage à Distance. C’est le nom du projet de recherche qui a servi de cadre aux différents travaux de recherche rapportés ici.
- **JAM** : C’est le modèle d’architecture dont nous nous inspirons pour construire la structure interne de nos agents. Il est construit à partir du modèle BDI et des implémentations du système de raisonnement procédural de l’Université de Michigan (UMPRS), et de l’entreprise SRI International (PRS-CL).
- **KQML** : Knowledge Query and Manipulation Language, c’est un langage d’interrogation et de manipulation de connaissances permettant aux agents de communiquer entre eux. Il résulte des efforts de développement du Ministère de la Défense des Etats-Unis à travers le DARPA.
- **KIF** : Knowledge Interchange Format. Langage déclaratif de type logique mis au point au sein du FIPA. Il sert à construire le corps des messages écrits en KQML.
- **MaSE / AgentTool** : Multiagent System Engineering est une méthodologie de développement des systèmes multi-agents constitués d’agents hétérogènes. Cette méthodologie a été mise au point par Scott DeLoach et ses collaborateurs au sein du Laboratoire d’Intelligence Artificielle du Air Force Institute of Technology. MaSE comporte sept étapes réparties en deux étapes et considère un agent comme un ensemble de processus informatiques communiquant entre eux pour atteindre un objectif donné. Les auteurs de MaSE ont développé un environnement d’instanciation de la méthodologie, il s’agit de AgentTool. AgentTool permet de déployer les différentes de MaSE et de procéder à des vérifications de cohérence comme les propriétés de vivacité et de sûreté.
- **PIFAD** : Plate-Forme Informatique pour la Formation A Distance, c’est ainsi que nous désignerons tout au long de ce rapport, la plate-forme utilisée durant Simuligne
- **POO** : Programmation Orientée Objet.
- **SIC** : Systèmes d’Information et de Communication
- **SICAH** : Systèmes d’Information et de Communication d’aide à l’Apprentissage Humain. C’est le nom de notre équipe de recherche au sein du Laboratoire d’Informatique de l’Université de Franche-comté (LIFC). L’équipe est dirigée par le professeur Thierry Chanier.
- **SIGFAD** : Système pour les Interactions dans des Groupes de Formation A Distance
- **SimuLigne** : SIMUlation en LIGNE. C’est le nom de l’expérimentation de formation à distance menée conjointement par l’équipe SICAH du Laboratoire Informatique de l’Université de Franche-Comté et le Département de Langues de l’Open University dans le cadre du projet ICOGAD financé par le Ministère de la Recherche (France).
- **SMA** : Système multi-agents Système informatique constitué d’un ensemble d’agents qui interagissent pour atteindre l’objectif global du système.
- **SQL** : Structured Query Language. C’est un langage de manipulation de données standard dans les bases de données.
- **UML** : Unified Modeling Language. Langage de modélisation utilisé en programmation orientée objet qui développé par l’Object Management Group et qui a la prétention d’unifier toutes les méthodologies objet existantes.
- **TA** : Théorie de l’Activité
- **TIC / TICE** : Technologies de l’information et de la Communication / pour l’Education

¹ Pour un glossaire plus détaillé, se reporter à (Alsic, 2001) et (Torrent, 2001)

INTRODUCTION GENERALE

La formation à distance (FAD) prend une importance de plus en plus grandissante dans la société contemporaine. Ceci correspond certainement à une double évolution sur le plan technologique mais aussi sur celui des besoins pédagogiques et didactiques. Du fait de la complexification de la société, les besoins en formation et en apprentissage obligent d'une part, à imaginer une formation tout au long de la vie et d'autre part, à mettre en place des formations davantage adaptés aux aspirations de leurs bénéficiaires. Les aspirations des apprenants se situent à deux niveaux : il s'agit premièrement de prendre en compte la spécificité des objectifs d'apprentissage et des styles d'apprentissage ; et deuxièmement la formation doit prendre impérativement en compte la variété des espaces et des temps d'apprentissage. Toutes ces évolutions des besoins en enseignement et en formation viennent consacrer mieux encore que par le passé, l'avènement de la FAD. La FAD connaît elle-même des mutations profondes dues à l'avènement des grands réseaux informatiques et des Technologies de l'Information et de la Communication (TIC), on a désormais une FAD en ligne. En même temps qu'il convient de se réjouir de cette évolution, on ne peut manquer de s'interroger sur les mécanismes et les phénomènes qui apparaissent dans les nouveaux types de pédagogie induits par cette forme d'enseignement. La FAD en ligne peut-être précisément parce qu'elle est entièrement à distance vient souligner avec acuité des problèmes d'ores et déjà identifiés à savoir :

- l'isolement sociologique de l'apprenant ;
- la perte de motivation ;
- l'autonomisation de l'apprenant ;
- la prise d'identité dans le groupe ;
- l'appréciation de la progression pédagogique du groupe ;
- etc.

Nos objectifs de recherche concernent la compréhension des facteurs d'échec et de réussite d'une session de FAD. Dans le but de mieux cerner ces questions de recherche, nous avons participé à un projet pluridisciplinaire regroupant des linguistes du Département de Langues de l'Open University (Grande Bretagne), des psychologues de l'interaction du Laboratoire GRIC de Nancy I (France) et des informaticiens du Laboratoire d'Informatique de l'Université de Franche-Comté (France). Ce projet, baptisé ICOGAD pour Interactions et Changements Cognitifs dans des Groupe de Formation à Distance, fait partie du programme Cognitique 2000 lancé et financé par le Ministère de la Recherche (France). Les objectifs du projet ICOGAD étaient :

- grâce à une approche expérimentale, de cerner des phénomènes cognitifs, pédagogiques et sociaux qui apparaissent dans des groupes de travail au cours d'une session de FAD ;
- d'examiner certains de ces phénomènes d'un point de vue psychologique pour essayer de dégager des profils type d'utilisateurs ou de comportement des groupes et des individus ;
- d'analyser les traces des actions des utilisateurs et les objectifs pédagogiques pour proposer des fonctionnalités dont l'ajout dans les plates-formes de FAD permettrait de mieux assister les utilisateurs et mieux atteindre les objectifs d'apprentissage ;
- d'analyser dans quelle mesure les interactions apparaissant dans des groupes de FAD sont-ils porteurs de changements cognitifs dans le groupe et in fine, sur les apprenants et les autres membres du groupe ;

- d'analyser, concevoir et spécifier un système multi-agents (SMA) dont le but est de soutenir les interactions dans des groupes de formation à distance.

Pour notre part, en tant que membres de l'équipe d'informaticiens du LIFC, nous avons dû administrer la plate-forme de formation, mettre en place tous les outils de sauvegarde et plus tard d'analyse des interactions. Nous ne nous attarderons pas sur ces aspects du travail de l'équipe de Besançon dans le présent rapport. Ce rapport s'appesantira sur l'analyse de quelques traces d'interaction, dans le but de justifier les fonctionnalités incluses dans le système multi-agents construit. C'est précisément la construction de notre SMA qui fait l'objet principal du présent rapport.

Dans le cadre du projet ICOGAD, nous avons monté une expérimentation de formation à distance mettant en œuvre des apprenants anglophones engagés dans un exercice de simulation globale pour l'apprentissage du Français. Cette expérimentation dénommée, *Simuligne*, concernait 40 apprenants répartis dans 4 groupes. Chaque groupe disposait en outre d'un tuteur et deux ou trois natifs (étudiants en Maîtrise Français Langue Etrangère) qui servaient de compagnons d'apprentissage. La pédagogie était basée sur une production collaborative dans des groupes de taille réduite (moins de 15 personnes). Le but de l'expérimentation était de recueillir les données concernant les interactions apparues au cours de la formation et de pouvoir par la suite les analyser afin d'identifier quelques facteurs ou phénomènes déterminants pour la réussite ou l'échec d'une telle session de formation.

De façon plus spécifique, notre but est de proposer des outils ou artefacts logiciels capables de soutenir les activités des utilisateurs engagés dans la FAD. Nous axons notre intérêt principalement sur tout ce qui serait de nature à apprécier le volume et la qualité des interactions entre acteurs, à encourager ces interactions au cas où leur volume serait jugé insuffisant, à fournir des indicateurs permettant d'apprécier la progression et la cohésion du groupe. Cet intérêt est accentué par le type de pédagogie choisi, la collaboration ; par l'échelle d'analyse de l'activité, le groupe ; par l'échelle d'analyse de la tâche, nous nous intéressons à des activités se déroulant sur plusieurs semaines et non à des tâches très ponctuelles comme c'est le cas dans plusieurs recherches sur l'apprentissage collaboratif.

La conduite de *Simuligne* nous a permis de constater que les plates-formes informatiques de formation à distance (PIFAD) actuelles ne disposent pas d'outils permettant d'apprécier la cohésion et la durabilité des groupes engagés dans la FAD. Cette insuffisance se pose avec acuité lorsque l'on travaille dans des groupes de taille réduite devant produire collaborativement. La taille du groupe fait en sorte que la défection d'un ou de deux membres peut conduire rapidement à la disparition de celui-ci. Inversement, l'existence au sein du groupe de quelques personnes motivées et dynamiques crée des interactions au sein du groupe très propices à l'effectivité de l'apprentissage collaboratif. Il est alors important de pouvoir former des groupes ou de les redistribuer durant la formation, de façon à obtenir dans chacun d'eux une masse critique d'acteurs actifs propre à générer les effets bénéfiques attendus de ce type de pédagogie. Le rôle d'assistance et d'observation du groupe revient en général dans la FAD à deux types d'acteurs : le tuteur qui intervient dans chaque groupe et le coordonnateur ou superviseur académique qui a une vue sur chacun des groupes et est responsable du bon déroulement de la session.

Nous montrerons dans le présent rapport que durant la session de FAD, l'on peut, à partir des indicateurs assez simples sur le plan conceptuel (volume et fréquence des interactions) prévoir l'évolution et apprécier la progression du groupe. Il s'agit en d'autres termes de fournir au

coordonnateur et au tuteur des outils de gestion des différents groupes. Pour cela, nous analysons des données enregistrées au cours de Simuligne et nous mettons en perspective l'évolution des données d'interaction et le comportement des différents groupes dans un premier temps. Cette mise en perspective nous permettra de voir que l'évolution des différents groupes aurait pu être prévu simplement à partir de ces données. Dans un second temps, nous allons examiner les données d'interactions des apprenants du « mauvais groupe » et nous montrerons que là aussi, l'analyse des interactions liées aux individus permettrait de prédire les cas d'abandon apparus durant l'expérimentation.

Ceci nous conduit à dire que les plates-formes de télé-formation doivent être dotées d'un ensemble de fonctionnalités supplémentaires permettant de réifier en temps réel, la participation et l'activité des utilisateurs engagés dans une session de FAD. Ces fonctionnalités doivent permettre de répondre aux questions qui apparaissent dès que l'on se préoccupe du succès d'une session de FAD à savoir : dans quelle mesure peut-on affirmer que le groupe progresse normalement ? quand est-ce qu'il convient de supprimer une activité pour tout ou partie du groupe ? quels apprenants peut-on dispenser de telle ou telle activité sans nuire à la progression ou à la cohésion du groupe ? quand est-ce qu'il convient de raccourcir ou de prolonger les délais de réalisation d'une activité ? quelle aide pour quels apprenants si l'on ne veut voir la session de FAD s'arrêter ? comment peut-on, à partir des données d'interaction enregistrées, prévoir la durabilité et la cohésion des groupes engagés ?

Nous utilisons le paradigme d'agent pour proposer ces fonctionnalités sous la forme d'un système multi-agents que nous avons baptisé, SIGFAD, acronyme de *Soutien des Interactions dans des Groupes de Formation à Distance*. Le choix du paradigme d'agent nous a été dicté par les caractéristiques des environnements dédiés à la FAD qui sont finalement des systèmes ouverts, complexes, flexibles et très évolutifs.

Il convient néanmoins de dire que malgré le caractère très prometteur et très engageant des systèmes à base d'agents, ce paradigme souffre encore d'une absence cruelle de méthodologies, de langages et d'outils de développement validés. Nous affirmons que l'identification et la validation de ces outils constituent encore une véritable question de recherche. Une réponse satisfaisante ne lui sera apportée que le jour où l'on pourra exhiber des systèmes bâtis, dotés des caractéristiques annoncées et appliqués à plusieurs domaines de recherche.

Nous utilisons une méthodologie de développement des systèmes multi-agents, MaSE (*Multiagent System Engineering*), pour analyser, concevoir et spécifier SIGFAD. Pour cela, nousinstancions toutes les étapes de MaSE. Il convient de signaler ici que MaSE ne fournit pas de méthode pour bâtir l'architecture interne des agents. C'est pourquoi nous avons eu recours à l'architecture JAM, une application du modèle BDI (*belief-desire-intention*) pour spécifier l'architecture interne de nos agents.

Bien que l'analyse des données d'interaction enregistrées au cours de Simuligne nous permette de justifier l'ajout de fonctionnalités aux environnements de FAD existants, il convient cependant de souligner que les systèmes à venir doivent éviter un certain nombre d'inconvénients identifiés dans les systèmes informatiques classiques. Ces systèmes obligent très souvent leurs utilisateurs à prévoir toutes les situations, à coder les programmes, bref à être finalement des experts en informatique. Jusqu'à présent, les logiciels sont des systèmes obéissants, non imaginatifs, effectuant servilement les tâches pour lesquelles ils ont été conçus et développés. Dans un grand nombre d'applications cependant, on a désormais

davantage besoin de systèmes capables de décider par leur propre initiative, des actions à mener afin de remplir leurs objectifs, d'assister véritablement les utilisateurs. En ce qui nous concerne, l'obtention de tels systèmes intelligents, flexibles et rationnels (*goal-directed*) passent par l'élaboration de modèles mathématiques permettant, en analysant de façon autonome les interactions enregistrées, de construire des indicateurs de l'état du groupe, de la progression d'un individu ; de prédire l'évolution de la session de FAD et le cas échéant, de déclencher des mécanismes d'alerte en direction aussi bien des personnes en charge de la cohésion et de la progression des groupes, de l'aboutissement heureux de la session de FAD (tuteurs et coordonnateur) que de chacun des acteurs (particulièrement les apprenants). L'analyse des réseaux sociaux constituent une première réponse à ce besoin, elle peut être utilisée pour modéliser et visualiser la collaboration dans les groupes de FAD (Reffay et Chanier, 2002).

Ce rapport est divisé en quatre chapitres. Le premier chapitre présente les fondements théoriques qui nous ont servi aussi bien à l'analyse des données d'interaction que de la modélisation de la collaboration dans nos groupes. Il insiste particulièrement sur les notions d'apprentissage collaboratif ; le rôle du groupe, des tuteurs et du coordonnateur de la formation, sur la théorie de l'activité qui justifie le modèle conceptuel d'activité de Simuligne. Le deuxième chapitre est plus particulièrement consacré à l'analyse des données d'interaction des quatre groupes et aux interactions des apprenants du groupe dit 'mauvais'. Le troisième chapitre est consacré à la spécification de SIGFAD, notre système multi-agents. On y présente un survol de la méthodologie MaSE et une instanciation des différentes étapes de la méthodologie. Le quatrième chapitre s'intéresse aux aspects concernant la programmation de nos agents, nous avons utilisé le modèle d'architecture JAM avec le langage de développement qui lui associée. En fait, JAM fournit le squelette d'agent, laissant le soin au programmeur d'implémenter toutes les fonctionnalités de ses agents à partir de procédures et fonctions écrites en Java. Le modèle JAM s'interface par ailleurs très bien dans le framework JATLite qui implémente l'échange d'informations et de connaissances entre agents basée sur le langage KQML. Nous présentons aussi dans le chapitre 4 les résultats fournis par notre système multi-agents, simulant ainsi l'utilisation de SIGFAD dans un contexte d'utilisation réelle.

CHAPITRE I : FONDEMENTS THEORIQUES

Dans cette partie, nous allons exposer les principaux fondements théoriques sur lesquels nous nous sommes appuyés pour concevoir le type d'apprentissage mis en œuvre dans SimuLigne et l'analyse des données que nous avons menée. Ces fondements s'articulent autour du rôle du groupe et des personnes en charge de la pérennité des groupes engagés ; le rôle des interactions dans l'apprentissage collaboratif distribué.

1 -1 L'IMPORTANCE DU GROUPE ET DES INTERACTIONS DANS L'APPRENTISSAGE COLLABORATIF DISTRIBUE

La gestion et le suivi des groupes dans la FAD constituent des tâches fondamentales, il existe plusieurs raisons à cela. Le positionnement et l'intégration de l'apprenant dans le groupe permettent de pallier des phénomènes largement identifiés comme les écueils dans la FAD, à savoir : l'isolement sociologique de l'apprenant, la perte de motivation. Le groupe constitue la matérialisation immédiate de l'accompagnement de l'apprenant (ceci est valable pour les autres acteurs, l'apprenant étant pour ainsi dire, le but de tout entreprise d'apprentissage, nous nous restreindrons souvent à son unique évocation).

L'apprentissage collaboratif trouve un contexte naturel en FAD, précisément de par le fait qu'il suppose l'apparition d'interactions sociales entre les membres du groupe, créant ainsi un cadre sociologique de nature à rompre l'isolement de l'apprenant, maintenant un niveau de motivation de l'apprenant suffisant. D'aucuns ont parlé d'apprentissage collaboratif distribué (Fjuk, 1998) pour désigner l'apprentissage collaboratif mis en œuvre en FAD. Certains chercheurs ont noté que dans les environnements informatiques d'apprentissage collaboratif, il est difficile d'évaluer le niveau d'interactions et de communications entre les individus (Haythornthwaite, 1999). Ceci est assez surprenant car on sait bien que dans les PIFAD, toutes les interactions homme-machine sont enregistrées dans les fichiers de traces. Les données enregistrées dans ces fichiers peuvent être analysées et rendre ainsi disponible l'information concernant le niveau d'interaction et de communication entre les acteurs. En fait, cette affirmation découle du fait que les systèmes permettant d'analyser les données d'interaction contenues dans les fichiers de traces et de fournir les interactions dans le groupe sont inexistantes et doivent être construits.

Il a été mis en évidence que les processus cognitifs qui rendent l'apprentissage collaboratif efficace ou effectif sont intimement liés aux interactions qui apparaissent entre les individus engagés dans l'apprentissage (Weiss et Dillenbourg, 1999). L'effectivité de l'apprentissage collaboratif émerge progressivement au cours des interactions à travers les processus comme la résolution de conflits, la régulation mutuelle, l'explication, la justification, l'argumentation (Dillenbourg et al., 1996). Haythornthwaite (1999) souligne que l'apprentissage collaboratif requiert des interactions et des échanges entre les apprenants.

Trois approches théoriques caractérisent les recherches en apprentissage collaboratif. L'approche socio-cognitive s'intéresse à la façon par laquelle les interactions sociales influencent le développement cognitif des individus. L'approche socio-culturelle s'intéresse aux liens de causalité entre les interactions sociales et le développement cognitif des individus. L'approche de la cognition distribuée se focalise davantage sur le groupe, l'environnement et plus particulièrement le contexte social fait partie intégrale de l'activité d'apprentissage. Il faut noter que la dernière approche unifie le social et le cognitif, il est à

présent démontré que les processus cognitifs et sociaux sont intimement liés et que la causalité entre ces deux catégories de processus est circulaire, complexe et qu'il est dès lors superflu d'essayer d'opérer une distinction entre le social et le cognitif.

L'hétérogénéité du groupe est considérée aussi bien du point de vue socio-constructiviste avec la génération du conflit socio-cognitif que du point de vue socio-culturel en suscitant les conditions d'une internalisation des connaissances et savoirs. L'approche de la cognition distribuée (*shared cognition approach*) insiste davantage sur l'aspect social de l'apprentissage collaboratif; alors que les approches socio-constructiviste et socio-culturel voient l'émergence des concepts sur un plan inter-individuel, ici, les concepts sont analysés comme le produit du groupe. Le groupe est vu dès lors comme un système cognitif unique, ceci n'a rien d'anecdotique pour un informaticien (contrairement à un psychologue pour qui l'agent est toujours un être humain), en effet, il est bien connu que la granularité d'un système distribué est un choix de conception purement arbitraire, c'est une variable que le concepteur fixe pour appréhender à une échelle un phénomène qui n'est pas perceptible à une autre échelle. Ceci signifie que l'apprentissage collaboratif ne trouve sa pleine justification que dans le cadre d'un groupe, il n'y a plus d'apprentissage collaboratif à partir du moment où le groupe cesse d'exister.

De manière plus globale, on note une évolution des recherches en apprentissage collaboratif. Alors que les premiers travaux relatifs à ce sujet portaient davantage sur le comportement des individus dans le groupe, on a aujourd'hui un glissement vers la prise en compte du groupe dans sa globalité comme unité d'analyse, cette évolution est étayée par des théories en sciences humaines et sociales comme la théorie de l'activité qui prend en compte l'individu interagissant avec ses pairs, utilisant des outils (ordinateurs, langages symboliques ou naturels) dans le contexte de sa culture (Baker, 2000).

1.2 - LE ROLE DU COORDINATEUR ET DU TUTEUR DANS CE CONTEXTE

Dans le contexte de l'apprentissage collaboratif distribué, la notion de groupe prend une importance singulière. Il est important que le groupe se maintienne et vive au fil de la formation. C'est le rôle du tuteur d'animer et de maintenir le groupe. Le coordinateur de la formation a encore un rôle plus important, il a une vue d'ensemble sur la formation et à ce titre, se préoccupe de la vie de chaque groupe. En fait, il est important que l'on ait dans chaque groupe une quantité suffisante d'interactions, pour cela le coordinateur de la formation peut être amené à redistribuer des groupes ou même lorsque ceci est nécessaire à supprimer des groupes ou à renforcer d'autres. Il semble indispensable d'avoir dans chaque groupe une masse critique suffisante (en termes de personnes capables d'animer le groupe et donc de générer des interactions suffisantes) pour générer des interactions. Plusieurs recherches portent sur les moyens de créer de renforcer les interactions dans des groupes en utilisant des artefacts (Kusunoki et al., 1999).

L'importance du groupe et des interactions apparaissant dans le groupe vient souligner avec davantage de force le rôle du tuteur dans ses tâches d'animation et de maintenance du groupe. Charlier et al. (1999) ont rapporté le rôle que doit jouer le tuteur dans l'apprentissage collaboratif médiatisé dans des campus virtuels à l'issue d'une expérimentation de formation à distance mettant en œuvre des étudiants appartenant à plusieurs universités européennes dans le projet LEARN-NET. Il en ressort que le tuteur doit vis-à-vis du groupe :

- aider à circonscrire le sujet ;
- prendre part à la recherche de la documentation ;

- suggérer des processus de résolution ;
- gérer les communications synchrones ;
- faciliter la communication entre les membres du groupe ;
- être la mémoire du groupe ;

et vis-à-vis de chaque apprenant, il doit :

- lui apporter l'assistance technique ;
- l'aider à formuler son projet ;
- articuler son projet à l'ensemble du groupe ;
- l'aider à mettre en place ses stratégies d'apprentissage.

D'après Donnay et Dreyfus (cités par Charlier et al., 1999), le tuteur peut jouer quatre rôles différents vis à vis de ses apprenants :

- facilitateur : aider les étudiants à choisir leur projet, faciliter leur expression, écouter les autres apprenants, prendre en compte les opinions de leurs pairs ;
- modérateur : synthétiser, critiquer, structurer les contenus, gérer et renforcer les délais de réalisation des activités;
- expert : aider les apprenants à trouver les documents et les ressources, leur apporter son expérience personnelle ;
- soutien affectif : s'engager personnellement, les encourager.

Tous ces aspects identifiés comme faisant partis du rôle du tuteur montrent la complexité et la difficulté de sa tâche. En tant qu'informaticiens, nous sommes interpellés par le besoin de mettre à disposition de ces derniers, des systèmes capables de les assister efficacement dans leurs activités.

Naturellement, ce qui vient d'être dit pour le tuteur prend une résonance particulière pour le coordonnateur de la formation en ce sens, qu'il est chargé de la pérennité de tous les groupes de FAD. Le rôle du coordonnateur en est donc plus crucial. Torrent (2001) présente les différents aspects de ce rôle tels qu'elle les a perçus personnellement :

- les aspects organisationnels et administratifs : distribution des ressources humaines dans les groupes et gestion des inscriptions, récupération des informations personnelles des personnes concernées ;
- intermédiaire entre les différents acteurs : apprenants et tuteurs, techniciens et tuteurs/apprenants ;
- supervision et contrôle de l'environnement.

Il apparaît évident que le rôle du coordonnateur dans l'apprentissage collaboratif distribué et très complexe et comporte plusieurs facettes. Aussi, il serait très utile pour ce dernier de disposer d'outils automatiques capables de l'aider à apprécier l'état des différents groupes et dans une certaine mesure, de prévoir leur évolution.

Il est intéressant de noter que dans la perspective d'unification des trois approches théoriques courantes en apprentissage collaboratif (approche socio-constructiviste, approche socio-culturelle, approche de la cognition distribuée), Koshmann (1996) insiste sur l'évolution du rôle de l'enseignant de pourvoyeur d'informations vers celui de facilitateur et de guide dans l'utilisation des ressources.

1.3 - L'ECHELLE D'ANALYSE DE LA TACHE DANS NOTRE RECHERCHE

Plusieurs recherches en apprentissage collaboratif concernent la gestion des interactions dans des groupes. Une classification y relative distingue les systèmes réflexifs qui réifient les actions des utilisateurs engagés dans le groupe ; les outils méta-cognitifs qui représentent l'état des interactions à travers des indicateurs clés et les systèmes tutoriels qui guident les utilisateurs à partir de ces indicateurs. Jermann et al. (2001) ont proposé une classification de ces systèmes.

Notre démarche se démarque de ces recherches dans la mesure où l'échelle d'analyse dans celles-ci est différente de la nôtre. Ces recherches situent l'analyse des interactions au niveau de tâches ponctuelles et très limitées dans le temps. Ici, les tâches ne sont pas intégrées dans un ensemble d'activités d'apprentissage, elles sont étudiées comme si elles étaient isolées. Pour notre part, nous nous intéressons à des tâches et des activités se déroulant sur une période significative et faisant partie d'un processus d'apprentissage. Cette approche nous semble plus adaptée à la FAD où l'on a un éclatement des espaces et des temps qui rend caduque des conclusions faisant suite à des microanalyses de tâches brèves, ponctuelles et prises isolement. L'approche décrite plus haut suppose implicitement que les utilisateurs travaillent en même temps, hypothèse non envisageable en FAD. Nous souhaitons comprendre les raisons pour lesquelles une session de FAD n'arriverait pas à son terme, c'est-à-dire, les obstacles à l'effectivité de l'apprentissage collaboratif distribué.

CHAPITRE 2 : L'EXPERIMENTATION SIMULIGNE

L'expérimentation de formation à distance en groupe a duré dix semaines et s'est déroulée du 30 avril au 07 juillet 2001. La formation, qui a pour nom SimuLigne (SIMUlation en LIGNE), a impliqué quarante apprenants britanniques adultes de français langue étrangères (FLE) de l'Open University (OU), quatre tuteurs-formateurs FLE du même établissement et une dizaine de natifs (francophones résidents en France). Le but de cette expérimentation était de recueillir des données dont l'analyse permettrait de mieux cerner des phénomènes qui apparaissent dans la mise en œuvre de l'apprentissage collaboratif distribué et notamment de comprendre les facteurs de réussite ou d'échec d'une session de FAD. Dans le type de pédagogie qui nous intéresse, nous sommes plus spécialement intéressés par :

- les indicateurs de l'état de progression et de cohésion 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 ;
- la possibilité de doter les plates-formes de télé-formation d'outils automatiques permettant d'apprécier pratiquement en temps réel les paramètres ci-dessus énumérés.

En fin de compte, il s'agit pour nous d'identifier dans les activités des apprenants, des tuteurs et du coordonnateur, le type d'aide que nous pourrions, en tant qu'informaticiens, leur apporter. Cette aide, nous avons choisi de la proposer au travers d'un ensemble de fonctionnalités supplémentaires dont nous doterions les plates-formes existantes.

Dans la suite de cette section, nous affirmons que les plates-formes existantes ne sont pas adaptés au modèle conceptuel d'activité que nous avons mise en œuvre au cours de Simuligne. Ensuite, à travers l'analyse de quelques données d'interaction enregistrées, nous montrerons qu'une analyse se basant sur le volume et le type d'interactions dans les groupes permet d'identifier les apprenants ou les groupes « peu dynamiques » et dans une certaine mesure de prévoir leur comportement et leur évolution dans la session de formation. L'idée étant de prévoir les cas d'abandon, finalement si courants en FAD, et de réorganiser les effectifs de groupes afin d'obtenir le maximum d'interactions entre leurs membres. Nous montrerons aussi comment l'analyse par les réseaux sociaux des interactions permet de modéliser et de visualiser la collaboration dans les groupes de Simuligne.

2-1 LES INSUFFISANCES DES PLATES-FORMES DE TELE-FORMATION SUR LE PLAN CONCEPTUEL

La théorie de l'activité (TA) proposée par Leontiev (cité par Fjök et al., 1997) nous semble la plus appropriée pour expliquer la structure d'une activité dans Simuligne. Ceci dû au fait qu'elle prend en compte le groupe comme unité de base de l'analyse et donne une place centrale au concept d'activité. Fjök et al. (1997) affirment que la TA permet une prise en compte conceptuelle du travail et du développement ainsi que du rôle des artefacts (comme l'ordinateur) dans les contextes sociaux.

Durant Simuligne, nous avons utilisé la plate-forme WebCT™, nous nous y référons dans la suite de ce rapport par PIFAD (plate-forme informatique de formation à distance).

Nous présentons à la figure 1 le modèle conceptuel d'activité de Simuligne adapté de la structure de base d'une activité tel que proposé par Engelström (1987). La figure montre le modèle conceptuel d'activité de Simuligne. Le module collaboratif (qui forme en fait la session de FAD dans notre cas) est centré autour du concept d'activité. L'activité est réalisée par un sujet, membre d'un groupe. Elle a un objet et fait partie d'une étape ; ceci signifie que le module collaboratif est composé d'étapes, chacune composée d'un ensemble d'activités. Une activité est constituée d'un ensemble de tâches, chaque sujet utilisant un outil pour accomplir une tâche. Un sujet peut être membre de plusieurs groupes, mais dispose dans chacun d'eux d'un statut unique. L'activité spécifie l'espace (dont le groupe pour nous) dans lequel le sujet doit la réaliser, on spécifie ainsi le statut que doit avoir le sujet pour réaliser ladite activité. Les concepts habituels de « division de travail » et de « règles » dans la TA sont implémentés ici à travers le statut et la tâche tels qu'indiqués dans la figure. Notre modèle est plus adapté à la FAD car il prend en compte la notion de temps : une activité possède une date de début et une date de fin. La notion de temps n'est pas prise en compte dans le modèle originel d'Engelström, il est inapplicable tel quel en FAD. En résumé, le modèle montre que chaque sujet possède un statut dans un groupe, utilisent des outils (chaque tâche liée à un outil) pour réaliser l'objet d'une activité dans un délai.

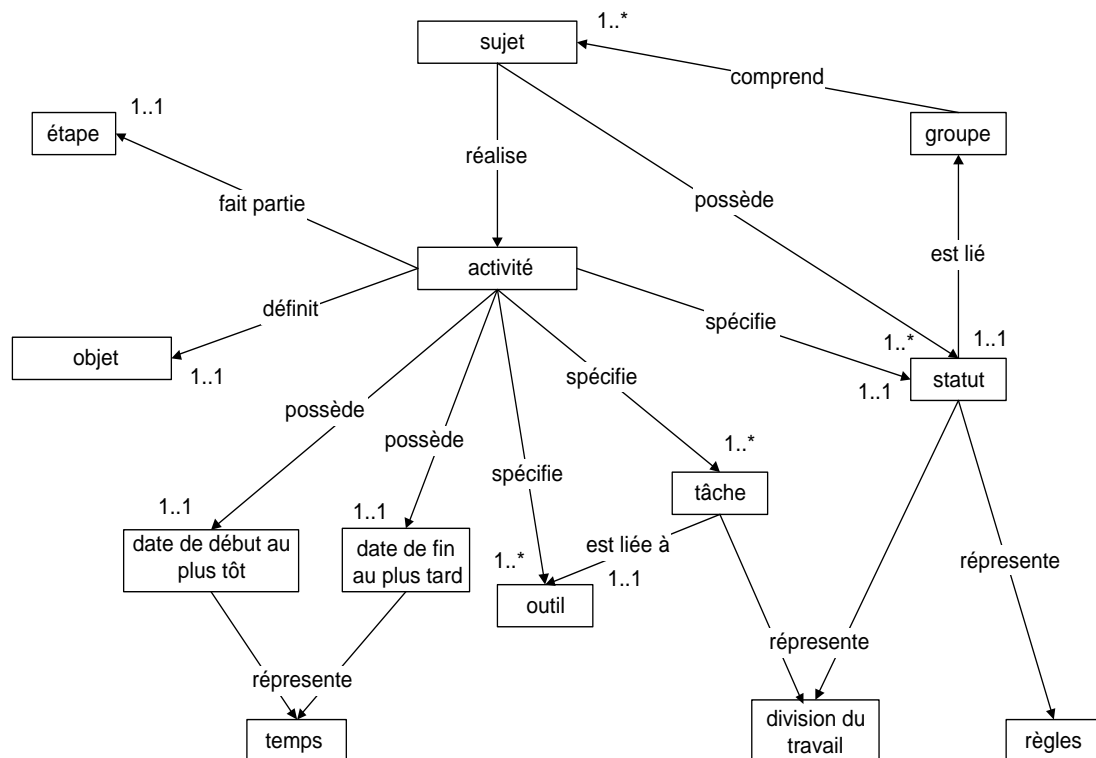


Fig. 1. Le modèle conceptuel d'activité de Simuligne

La mise en relief de ce modèle et du schéma général de PIFAD permet de voir que sur un plan purement formel, PIFAD se prête mal à la mise en œuvre de Simuligne. Le schéma général de PIFAD est montré à la figure 2. PIFAD est construit sur une architecture de statuts. Chaque utilisateur (ensemble constitué d'un nom de connexion et d'un mot de passe) est membre d'un cours - *cours* (que nous avons utilisés comme espace caractérisant le groupe, il est impossible d'implémenter autrement le concept de groupe) et y a un statut permanent. PIFAD permet de définir quatre statuts : l'apprenant, l'administrateur, le concepteur et l'assistant pédagogique. Pendant Simuligne, nous avons utilisé trois statuts dans chacun des 4 groupes de base qui nous concernent ici : les apprenants et les natifs ayant le statut d'apprenant

(*learner*) ; le tuteur du groupe ayant le statut de concepteur (*designer*) et le coordonnateur ayant le statut d'administrateur. Le coordonnateur avait aussi la possibilité de se connecter dans chaque groupe comme un utilisateur anonyme afin d'observer ce qui s'y passe. Le même compte était utilisé par les chercheurs qui souhaitaient observer la vie du groupe.

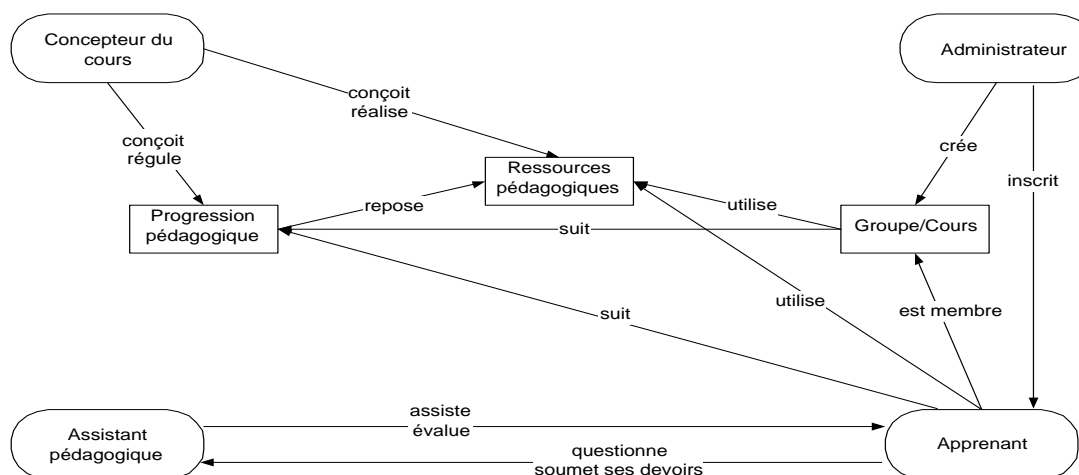


Fig. 2. Le schéma général de PIFAD

Un examen rapide de la figure 2, représentant le schéma général de PIFAD révèle une absence totale de référence au temps. Ceci empêche de rendre compte des différents phases que l'on observe dans la FAD : les ressources et la progression pédagogique sont conçus et mis en ligne par les enseignants-concepteurs à l'avance, les enseignants-tuteurs régulent l'activité pédagogique au cours de la session en assistant, évaluant, modifiant les parcours pédagogiques et les ressources en même temps que les apprenants qui réalisent leurs activités. Nous constatons que les statuts définis dans PIFAD ne sont pas tout à fait compatibles avec ceux nécessaires au cours de Simuligne. Nous n'avons par exemple pas utilisés le statut d'assistant pédagogique présent dans PIFAD. A l'opposé, plusieurs personnes ayant joué un rôle dans la mise en œuvre de l'expérimentation ne sont pas représentables dans PIFAD. Nous pensons aux personnes en charge de la médiatisation des ressources, de l'assistance technique aux utilisateurs, à la conception pédagogique de la formation, au compagnonage (natifs de Besançon), à la gestion de la formation sur les plans administratif et didactique. L'on peut aussi noter qu'alors même que le concept d'activité occupe une place centrale dans Simuligne, il occupe une place beaucoup moins prépondérante dans PIFAD. On peut en outre noter l'absence de référence à la notion de groupe dans PIFAD, nous avons dû utiliser les cours pour implémenter nos différents groupes. L'inconvénient de ceci étant qu'un même utilisateur appartenant à plusieurs groupes différents apparaît comme autant d'utilisateurs différents. Tout ce qui vient d'être dit montre que PIFAD n'a pas été conçu pour répondre particulièrement aux besoins de la FAD et encore moins à la mise en œuvre de l'apprentissage collaboratif distribué.

2-2 L'ANALYSE DES DONNEES D'INTERACTION

Durant la formation Simuligne qui s'est déroulée sur dix semaines entre le 30 avril et le 07 juillet 2001, nous avons observé une baisse régulière du niveau d'activités dans le groupe Lugdunensis à tel point que nous avons dû le fermer le 30 mai ; deux apprenants identifiés comme actifs ont été affectés au groupe Aquitania pour poursuivre la formation. Dans la suite

de ce paragraphe, nous allons montrer que les données d'interaction enregistrées constituaient des éléments statistiques suffisants pour éclairer le tuteur et le coordonnateur en temps réel, sur le comportement des apprenants et l'évolution du groupe.

2-2-1 LES DONNEES D'INTERACTION RELATIVES AUX GROUPES

Nous présentons les tableaux résumant les statistiques d'interaction obtenues au cours des quatre premières semaines. Cette période nous intéresse car elle est celle qui a vu coexister les quatre groupes de base. Nous rappelons que notre objectif est de montrer au lecteur que la simple mise à disposition des données concernant la fréquence et le type d'interactions pourrait permettre au tuteur et au coordonnateur de la formation d'apprécier l'état de son groupe et d'anticiper dans une certaine mesure les cas d'abandon.

Les tableaux 1 et 2 présentent les pourcentages cumulés du nombre de fois où les membres du groupe se sont connectés sur PIFAD et le temps passé en ligne. Cette information est importante car la plupart des activités se passent en ligne même s'il est vrai qu'une partie non négligeable peut être réalisée hors ligne. On peut aussi dire que la quête d'une identité dans le groupe devrait pousser, tout au moins au début, les apprenants et les natifs à se connecter régulièrement. De même peut-on affirmer que le caractère occasionnel des connexions traduit probablement un positionnement insuffisant dans le groupe et dénote d'une faible motivation. Le dynamisme des membres du groupe se traduit aussi par le temps par lequel il reste en ligne.

Groupe de base	Semaine 1	Semaine 1+2	Semaine 1+2+3	Semaine 1+2+3+4
Aquitania	35,39%	31,21%	30,47%	29,73%
Gallia	22,17%	27,34%	29,01%	30,15%
Lugdunensis	21,75%	20,95%	20,16%	19,44%
Narbonensis	20,68%	20,50%	20,36%	20,68%
Total	100,00%	100,00%	100,00%	100,00%

Tableau 1. Pourcentage cumulé du nombre de connexions à PIFAD

Groupe de base	Semaine 1	Semaine 1+2	Semaine 1+2+3	Semaine 1+2+3+4
Aquitania	42,03%	34,99%	33,47%	32,31%
Gallia	18,09%	23,16%	23,93%	25,55%
Lugdunensis	17,94%	17,54%	17,72%	17,27%
Narbonensis	21,93%	24,31%	24,88%	24,87%
Total	100,00%	100,00%	100,00%	100,00%

Tableau 2. Pourcentage cumulé des temps de connexion à PIFAD

Le tableau 2 nous montre qu'alors que tous les autres groupes ont un temps de connexion à l'issue de la troisième semaine qui est proche ou supérieur à 25 %, le groupe Lugdunensis présente une proportion, moins de 18 %, beaucoup plus faible.

Les tableaux 3 et 4 montrent les pourcentages cumulés du nombre de courriels émis et lus dans le groupe. Dans PIFAD, les groupes sont étanches du point de vue des communications, en ce sens que les courriels ne sont émis ou recevables qu'au sein du groupe. Un utilisateur ne peut émettre ou recevoir de courriels qu'à l'intérieur de son groupe, c'est aussi ce qui justifie l'idée de considérer le groupe comme un espace, un lieu où se déroulent des activités prescrites à l'avance. Tout ceci rend pertinent l'information contenue dans les tableaux sus-cités, elle traduit en quelque sorte le dynamisme et la vitalité du groupe.

Groupe de base	Semaine 1	Semaine 1+2	Semaine 1+2+3	Semaine 1+2+3+4
Aquitania	29,15%	23,61%	21,81%	20,92%
Gallia	13,70%	20,69%	25,65%	31,15%
Lugdunensis	19,24%	19,88%	18,84%	17,40%
Narbonensis	37,90%	35,82%	33,70%	30,53%
Total	100,00%	100,00%	100,00%	100,00%

Tableau 3. Pourcentage cumulé du nombre de courriels lus à l'intérieur du groupe

Groupe de base	Semaine 1	Semaine 1+2	Semaine 1+2+3	Semaine 1+2+3+4
Aquitania	27,62%	24,09%	22,88%	22,24%
Gallia	20,95%	25,41%	29,01%	35,12%
Lugdunensis	17,14%	15,51%	15,09%	14,05%
Narbonensis	34,29%	34,98%	33,02%	28,60%
Total	100,00%	100,00%	100,00%	100,00%

Tableau 4. Pourcentage cumulé du nombre de courriels postés à l'intérieur du groupe

On peut se rendre compte que là aussi, le nombre de courriels postés par les membres du groupe Lugdunensis à l'intérieur de leur groupe est très bas par rapport à ce qui se passe dans les autres groupes ; il en va de même pour le nombre de courriels lus. Il faut d'ailleurs remarquer que le pourcentage cumulé diminue au fil des semaines, ce qui devrait alerter sur la baisse d'activités dans le groupe.

Les tableaux 5 et 6 présentent respectivement, les pourcentages cumulés du nombre de messages lus et du nombre de messages postés dans les forums par les membres des groupes de base. En raison de la conception même de Simuligne dont la majorité des activités étaient basés sur l'utilisation des forums, tout au moins, pendant la période couverte par la présente étude, ces tableaux présentent probablement les indicateurs les plus significatifs du niveau d'activité dans un groupe. Rappelons que toutes les activités de l'étape E1 demandaient l'utilisation du forum. La toute première activité, E0A1, consistait en une discussion dans un forum afin de permettre une parfaite maîtrise de cet outil par l'ensemble des utilisateurs. Durant la période couverte par notre étude, huit forums ont été ouverts dans chaque groupe de base.

Groupe de base	Semaine 1	Semaine 1+2	Semaine 1+2+3	Semaine 1+2+3+4
Aquitania	50,65%	42,39%	40,27%	38,16%
Gallia	25,26%	28,27%	26,62%	26,26%
Lugdunensis	12,66%	13,55%	13,69%	13,94%
Narbonensis	11,43%	15,79%	19,43%	21,64%
Total	100,00%	100,00%	100,00%	100,00%

Tableau 5. Pourcentage cumulé du nombre de messages lus dans les forums

Groupe de base	Semaine 1	Semaine 1+2	Semaine 1+2+3	Semaine 1+2+3+4
Aquitania	57,75%	50,26%	49,46%	47,71%
Gallia	19,01%	23,28%	20,43%	19,82%
Lugdunensis	8,45%	9,26%	9,32%	9,45%
Narbonensis	14,79%	17,20%	20,79%	23,02%
Total	100,00%	100,00%	100,00%	100,00%

Tableau 6. *Pourcentage cumulé du nombre de messages postés dans les forums*

L'observation des données contenues dans les deux tableaux ci-dessus montrent que les pourcentages d'interactions liées aux forums sont très faibles dans le groupe Lugdunensis. Dès lors, il n'est pas surprenant que ce groupe ait été amené à disparaître à la fin de la quatrième semaine. On peut d'ailleurs observer que ces pourcentages étaient déjà très faibles à l'issue de la première semaine, les semaines suivantes ne venant que le confirmer. La légère évolution des pourcentages indiquent probablement des tentatives de motivation de la part du tuteur du groupe et du coordonnateur de la formation, restées manifestement infructueuses. Ceci nous emmène à penser qu'une action plus volontariste aurait probablement dû être entreprise dès la première semaine si ces données d'interaction avaient été disponibles et fournies au tuteur et au coordonnateur. Le coordonnateur aurait pu entreprendre soit de réorganiser les groupes, de manière à les re-équilibrer, soit pourquoi pas, fermer ce groupe dès cet instant, identifier les apprenants motivés et dynamiques, les réaffecter à d'autres groupes. Ces actions ont été entreprises à la fin de la quatrième semaine de formation car les données présentées ci-dessus n'étaient pas disponibles, le coordonnateur ne disposait que d'indicateurs subjectifs ou intuitifs. Ces données auraient constitué des indicateurs plus objectifs capables de guider et de justifier de telles actions.

Les données contenues dans les tableaux précédents ont été extraites et construites du fichier des traces de Apache (Apache étant le serveur HTML utilisé). Ce fichier s'appelle */icogad/.../server/logs/access_log*. Ce fichier enregistre chaque requête sur le serveur HTTP sur une ligne différente contenant : le nom de connexion à PIFAD de l'utilisateur, son adresse IP, la date et le type d'action, la taille du fichier utilisé, le code de retour de la requête effectuée.

Nous présentons dans le tableau 7, les durées de connexion aux pages de consignes du début du 30 avril au 30 mai 2001. Ces durées ne sont pas présentées de manière hebdomadaire comme pour les données précédentes mais indiquent une fois encore les durées les plus faibles pour le groupe Lugdunensis. Les consignes des différentes étapes de Simuligne ont été implémentées dans PIFAD comme les contenus des cours. Ces durées ont été calculées à partir des fichiers spécifiques de PIFAD. Par exemple, les données concernant la cinquième activité de la seconde étape sont enregistrées dans le fichier *EtapeE2_e2a5.htm.log* du répertoire *.../<group>/database/page_tracking*. *<group>* est le nom du groupe de base concerné.

Groupe de base	Nombre d'accès	Durée de connexion	Durée de connexion (en secondes)	Pourcentage
Aquitania	626	103h06m03s	371 163	32,26%
Gallia	364	99h18m26s	357 506	31,08%
Lugdunensis	379	56h22m43s	202 963	17,64%
Narbonensis	390	60h46m40s	218 800	19,02%
Total	1759		1150432	100,00%

Tableau 7. *Durées de connexion aux pages de consignes*

Les différents tableaux montrent des pourcentages hebdomadaires d'interaction relativement faibles dans le groupe Lugdunensis. Ceci permet de comprendre pourquoi ce groupe a été fermé le 30 mai 2001.

Rendu à ce stade, nous voyons que les informations enregistrées dans les fichiers de traces permettent de fournir aux utilisateurs (principalement aux tuteurs et au coordonnateur qui

nous intéressent ici) des données d'interaction de telle manière qu'on puisse savoir quotidiennement et même pratiquement en temps réel, la situation des groupes et prédire leur évolution. Pour des raisons de concision, nous avons présenté ces données de façon hebdomadaire, on pourrait les présenter quotidiennement et même dans une unité de temps plus petite comme l'heure. Nous voulons souligner ici, que si des chercheurs comme Haythornthwaite (1999) affirment :

“in computer-supported distance learning classes it is often difficult to know to what extent individuals are interacting and how much they communicate with other class members”

ceci indique simplement une absence de fonctionnalités y afférentes dans les plates-formes de formation à distance. Les plates-formes existantes doivent être couplées à systèmes permettant de collecter les données d'interaction dans les endroits appropriés, les calculer automatiquement et fournir des indicateurs de l'état et de la durabilité des groupes concernés.

2-2-2 LES DONNEES D'INTERACTION RELATIVES AUX UTILISATEURS DU GROUPE LUGDUNENSIS

Dans cette section, nous allons nous focaliser sur les données d'interaction concernant les apprenants du groupe Lugdunensis. En fait, deux apprenants de ce groupe ont été affectés à un autre groupe pour continuer la session de formation. Comme nous l'avons vu précédemment, l'analyse quotidienne des données d'interaction aurait pu permettre de prédire cette évolution. Nous allons tenter de le montrer dans cette section.

Le tableau 8 montre les pourcentages cumulés des messages lus dans les forums par chaque apprenant du groupe. Nous présentons le pourcentage par apprenant de messages lus durant la semaine et le même pourcentage cumulé depuis le début de la formation. Afin de préserver l'anonymat des personnes, nous désignons les apprenants par LA_i ; LA_i représente l'apprenant (A) numéro *i* du groupe Lugdunensis (L).

	Semaine 1		Semaine 2		Semaine 3		Semaine 4	
	Sem. 1	Sem. 2	Sem. 1+2	Sem. 3	Sem. 1+2+3	Sem. 4	Sem. 1+2+3+4	
LA1	19,42%	15,06%	16,58%	0,62%	11,96%	2,38%	10,71%	
LA2	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	
LA3	0,00%	23,17%	15,08%	8,02%	13,04%	8,33%	12,42%	
LA4	2,16%	10,42%	7,54%	0,00%	5,36%	7,14%	5,59%	
LA5	19,42%	4,25%	9,55%	27,16%	14,64%	32,14%	16,93%	
LA6	0,00%	0,00%	0,00%	19,75%	5,71%	14,29%	6,83%	
LA7	17,27%	23,17%	21,11%	16,05%	19,64%	13,10%	18,79%	
LA8	27,34%	19,31%	22,11%	18,52%	21,07%	22,62%	21,27%	
LA9	14,39%	4,63%	8,04%	9,88%	8,57%	0,00%	7,45%	
Total	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%	

Tableau 8. Pourcentages cumulés du nombre de messages lus par les apprenants du groupe Lugdunensis

L'analyse des informations contenues dans le tableau 8 donne une idée précise du comportement des apprenants du groupe. Il n'est pas surprenant que les deux apprenants ayant survécu à la disparition du groupe soient LA5 et LA8 qui présentent les meilleurs pourcentages à la fin de la quatrième semaine. LA7 avait des raisons personnelles très contraignantes qui l'ont obligé à arrêter la formation, il aura été néanmoins présent jusqu'au bout. On peut observer que LA1 qui était très actif au début de la formation est progressivement devenu moins dynamique et a fini donc par abandonner.

Nous montrons dans la suite que les données d'interaction peuvent aussi permettre de savoir si un apprenant est en train oui ou non de réaliser une activité. Nous allons le montrer sur un exemple. Prenons l'activité E1A2 (deuxième activité de la première étape), le but de cette activité est pour chaque apprenant de choisir un thème (parmi les thèmes pour l'élaboration de la candidature du groupe au concours *Open City Competition*) dans un forum consacré à cet effet : '*e1a2_choisir_theme*' ouvert du 11 au 15 mai 2001. Après avoir effectué son choix, l'apprenant doit développer son thème (en interagissant éventuellement avec les natifs pour de possibles corrections) et soumettre son travail au tuteur avant le 25 mai. Cette activité illustre par ailleurs, très bien les questions qui interpellent le coordonnateur et les tuteurs dans ce type de pédagogie : comment savoir si les apprenants réalisent correctement ou pas leurs activités ? La réponse à cette question est cruciale mais peut trouver des éléments de réponse dans les données d'interaction. Un apprenant réalisant correctement les tâches liées à cette activité, doit se connecter un certain nombre de fois à PIFAD ; il doit lire et poster des messages dans le forum '*e1a2_choisir_theme*' entre le 11 et le 15 mai. La consultation des pages de consignes fournit aussi une bonne indication sur les efforts de l'apprenant dans la réalisation de l'activité. Toutes les données relatives à ces thèmes peuvent être obtenues quotidiennement. Le tableau 9 montre le nombre d'entrées dans le groupe entre le 11 mai et le 25 mai. De l'observation des informations contenues dans cette table, on peut dire qu'un maximum de sept apprenants sont susceptibles de réaliser cette activité. LA2 et LA5 ne se sont pas connectés au groupe entre le 11 et le 15 mai. On peut aussi remarquer que les apprenants LA1 et LA4 n'entrent pas une seule fois dans le groupe entre le 15 mai et le 25 mai, ce qui indique qu'ils ne sont pas probablement en train de réaliser correctement l'activité. Ceci aurait permis au tuteur de prédire qu'un maximum de cinq personnes réaliseraient l'activité concernée et finalement iraient jusqu'au bout de la simulation en suivant toute la progression pédagogique envisagée au départ.

	LA1	LA2	LA3	LA4	LA5	LA6	LA7	LA8	LA9
11/05/2001	3		4	2			7	3	
12/05/2001	3		1				2	3	
13/05/2001	9		3					3	
14/05/2001	1		1			2	2	5	3
15/05/2001						1	1	5	
16/05/2001			1			2	2	1	1
17/05/2001			1				4	5	1
18/05/2001			1			1	4	7	11
19/05/2001							1	1	1
20/05/2001					2	2	1	1	
21/05/2001						1	1	1	
22/05/2001					4		1	7	
23/05/2001						2		4	
24/05/2001			1		2		1	3	2
25/05/2001					5	2		2	

Table 9. Nombre d'entrées dans le groupe Lugdunensis entre le 11 mai et le 25 mai

La mise à la disposition du tuteur de ces informations aurait pu lui permettre de constater que deux apprenants du groupe ne sont manifestement pas en train de réaliser les tâches liées à l'activité E1A2 et d'envisager des actions de remédiation dès le 13 ou le 14 mai par exemple. Il est aussi évident qu'une action de la part du tuteur du groupe vis à vis de ces apprenants leur montrerait qu'ils sont suivis et accompagnés et leur ôterait tout éventuel sentiment

d'isolation, maintiendrait à un niveau satisfaisant leur motivation. Ces apprenants auraient bien le sentiment qu'ils font partie d'un groupe et que des défaillances et des manquements de leur part affaibliraient le groupe et en tous cas, ne manqueraient pas d'être remarqués.

Les tableaux 10 et 11 montrent respectivement le nombre de messages lus et postés dans le groupe entre le 11 et le 15 mai. Ils permettent de constater que finalement, le nombre d'apprenants susceptibles d'être en train de réaliser l'activité E1A2 est de quatre. Seuls les apprenants LA1, LA3, LA6 et LA8 ont posté un message dans les forums entre le 11 et le 15 mai. En fait, seuls LA3, LA6 et LA9 ont posté un message lié à l'activité concernée.

	LA1	LA2	LA3	LA4	LA5	LA6	LA7	LA8	LA9
11/05/2001	2		3	6			5	0	
12/05/2001	21						3	17	
13/05/2001	0		6					3	
14/05/2001	1		2			8		2	11
15/05/2001						23		7	

Tableau 10. Nombre de messages lus par les apprenants dans les forums du Groupe entre le 11 et le 15 mai 2001

	LA1	LA2	LA3	LA4	LA5	LA6	LA7	LA8	LA9
11/05/2001									
12/05/2001	2								
13/05/2001			1						
14/05/2001			1			1			
15/05/2001								1	

Tableau 11. Nombre de messages postés par les apprenants dans les forums du Groupe entre le 11 et le 15 mai 2001

Le tableau 12 montre le nombre de fois où un apprenant a accédé aux pages de consignes entre le 11 et le 23 mai. Parmi les quatre apprenants ayant posté un message dans les forums, seuls trois ont accédé aux pages de consigne entre le 11 et le 23 mai, ceci nous emmène à dire qu'un maximum de trois personnes ont pu réaliser correctement cette activité. On arrive à la même conclusion si l'on rapproche le nombre de personnes ayant posté un message dans les forums du 11 au 15 mai du nombre de personnes étant entrées dans le groupe du 15 au 25 mai.

	LA1	LA2	LA3	LA4	LA5	LA6	LA7	LA8	LA9
11/05/2001	4			5				1	
12/05/2001									
13/05/2001									
14/05/2001						1		1	
15/05/2001									
16/05/2001									
17/05/2001							2		
18/05/2001						1			
19/05/2001									
20/05/2001						1			
21/05/2001									
22/05/2001					3			4	
23/05/2001									

Tableau 12. Nombre de fois où les apprenants du groupe ont consulté les pages de consignes entre le 11 mai et le 23 mai

Nous voyons qu'à partir des données d'interaction dans les groupes, il est possible d'identifier les apprenants 'faibles' et déclencher des actions de rémediation appropriées. Nous avons eu l'occasion de nous rendre compte de l'importance de cette question durant cette expérimentation. On peut aussi constater que l'apprenant LA5 qui n'a pas effectué correctement l'activité E1A2 s'est néanmoins révélé être parmi les apprenants les plus dynamiques et est allé jusqu'au bout de la session de formation. Ceci loin d'être contradictoire, montre qu'en FAD, un réel besoin d'accompagnement existe. Un apprenant peut être incapable de réaliser une activité ou d'accomplir certaines tâches sans que sa motivation ou sa volonté soient moindres. Ceci souligne encore une fois la perspective qui doit être laissée au tuteur ou au coordonnateur de dispenser certains apprenants de certaines activités, de modifier les échéances de certaines activités, quand ceci ne nuit pas à la progression du groupe.

Dans tous les cas, il est indispensable de pouvoir identifier le plus tôt possible des apprenants en difficulté et de leur proposer des mécanismes d'assistance afin qu'ils restent dans la formation. Nous soulignons encore une fois que cette identification presque en temps réel éloignerait des apprenants le sentiment d'isolation sociologique qui est un des principaux écueils en FAD et combattrait la perte de motivation qu'on observe souvent dans les acteurs de la FAD. Ceci renforcerait aussi l'idée d'appartenance au groupe, idée centrale dans le type de pédagogie mise en œuvre ici.

2-3 CONCLUSION

Dans l'analyse des données d'interaction présentée ci dessus, nous nous sommes focalisés d'une part sur le groupe, ensuite sur les apprenants. Nous n'avons pas pris en compte le comportement des tuteurs, il semble toutefois que la même approche puisse être utilisée pour mesurer quotidiennement l'activité des tuteurs. Nous sommes persuadés qu'un examen des données d'interaction relatives aux tuteurs donnerait là aussi, des éléments révélateurs du comportement et de la progression de nos groupes.

Nous avons montré dans ce chapitre qu'en analysant les données d'interaction enregistrées dans les fichiers de traces, on peut fournir aux personnes en charge la gestion des groupes en FAD des indicateurs permettant d'apprécier leur état, de prédire leur évolution, de les réorganiser de façon à obtenir un volume minimal d'interactions dans chaque groupe. Ces outils permettraient aussi d'apprécier la quantité et le type d'interactions dans les groupes et de prendre des mesures nécessaires pour les soutenir. Dans toutes les questions de recherches en apprentissage collaboratif, les interactions jouent un rôle majeur et sont porteurs de mécanismes d'apprentissage spécifiques (Dillenbourg et al., 1996).

En montrant que les interactions permettent d'apprécier la vie du groupe et son évolution, nous avons justifié la nécessité d'enrichir les plates-formes de formation à distance d'un certain nombre de fonctionnalités relatives à l'analyse des interactions et à la réification de la participation des acteurs aux activités d'apprentissage.

CHAPITRE III : LA SPECIFICATION DE SIGFAD

Dans le chapitre précédent, nous sommes arrivés à la conclusion qu'il fallait doter les plateformes de formation à distance d'un certain nombre de fonctionnalités supplémentaires en vue d'offrir aux personnes en charge de la gestion des groupes des outils leur permettant, quotidiennement à défaut de le faire en temps réel, d'apprécier l'état des groupes et leur évolution. Nous avons montré à travers un certain nombre de tableaux que les données d'interaction enregistrées dans PIFAD auraient permis de prédire à l'avance le comportement des utilisateurs et l'évolution des groupes engagés dans Simuligne. Naturellement, les tableaux présentés ont été construits, pour ainsi dire, à posteriori.

Dues aux caractéristiques que doivent posséder les systèmes informatiques dédiés à la FAD, nous avons eu recours au paradigme d'agent et nous proposons d'offrir les fonctionnalités supplémentaires sous forme de système multi-agents (SMA). Toutefois et malgré les promesses fort attrayantes rapportées dans la littérature et attribuées aux systèmes à base d'agents, force est de constater que ce paradigme souffre encore d'un manque de méthodologies, de langages et d'outils de développement. Dans ce contexte, l'identification et la validation de méthodologies permettant de développer les systèmes à base d'agents restent de véritables questions de recherche. En tant qu'informaticiens, nous devons grâce à des réalisations concrètes montrer que l'implémentation des caractéristiques liées au paradigme d'agent peuvent être obtenues réellement dans des systèmes informatiques.

Nous avons utilisé une méthodologie de développement des systèmes multi-agents mise au point à l'Air Force Institute of Technology par Scott DeLoach et ses collègues pour spécifier SIGFAD, notre système multi-agents. La méthodologie utilisée se nomme, MaSE pour Multiagent System Engineering. C'est une méthodologie qui part d'une définition de l'agent très générale et surtout, qui est couplée avec AgentTool, un environnement de développement qui lui est dédié. MaSE bénéficie aussi d'une abondante littérature mise à disposition par ses auteurs et a déjà été utilisée semble-t-il avec un certain succès, pour développer nombre de systèmes multi-agents.

Toutefois, MaSE n'impose aucune méthodologie particulière pour la construction de l'architecture interne des agents. Ceci, bien que pouvant être vu comme un avantage en ce sens la méthodologie reste ouverte à tout modèle de construction des agents, pousse le concepteur à rechercher une architecture particulière pour remplir cette fonction. Pour notre part, nous avons eu recours à l'architecture BDI (*belief-desire-intention*) pour construire l'intérieur de nos agents. En fait, nos agents seront construits suivant l'architecture JAM mise au point par Marcus Huber. JAM est une architecture qui est construite à partir d'un noyau BDI et qui intègre des idées et théories venant du système de raisonnement procédural (PRS en anglais pour Procedural Reasoning System) et des sémantiques des circuits structurés (SCS en anglais pour Structured Circuit Semantics).

Pour un survol des définitions des concepts d'agent, de système multi-agents et d'autres considérations générales, voir (Mbala, 2001).

3-1 UN SURVOL DE LA METHODOLOGIE MASE

MaSE est une méthodologie pour analyser, concevoir et développer des systèmes multi-agents hétérogènes ; elle offre un cycle de vie logiciel complet. La méthodologie se focalise sur les

communications entre les agents afin d'obtenir un comportement cohérent du système global. Pour atteindre cet objectif, MaSE utilise un certain nombre de modèles graphiques pour décrire les objectifs du SMA, le comportement des agents, la typologie des agents, les interfaces de communication entre agents ; elle offre par ailleurs la possibilité de définir une architecture interne des agents. MaSE permet au concepteur, à partir d'un ensemble de fonctionnalités très générales, d'analyser, concevoir et implémenter un système multi-agents fonctionnel. Ceci est fait en suivant sept étapes réparties dans deux phases telles que représentées sur la partie droite de la figure ci-dessous. Les rectangles aux bords arrondis indiquent les modèles MaSE utilisés pour matérialiser le résultat de chaque étape, les flèches indiquent comment les modèles sont reliés entre eux.

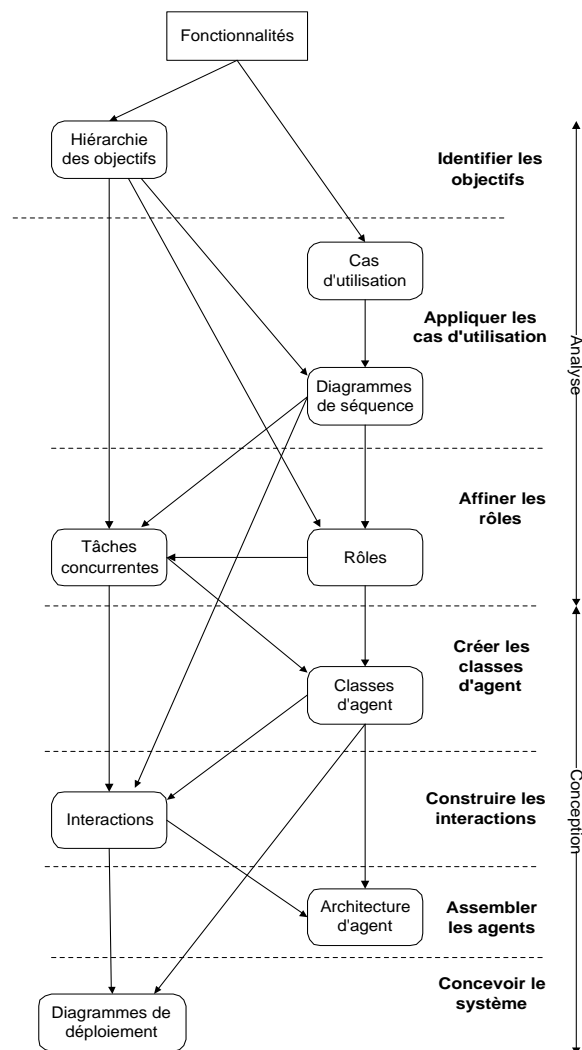


Fig. 3. Les étapes et les phases de la méthodologie MaSE

Cette méthodologie 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 système multi-agents 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.

La phase d'analyse est composée de trois étapes :

- **identifier les objectifs du SMA** : cette étape consiste à produire un ensemble structuré d'objectifs à partir du cahier de charges du SMA ; les objectifs sont identifiés du point de vue du système et non de l'utilisateur. Le résultat attendu de cette étape est la production d'un diagramme hiérarchisé d'objectifs.
- **appliquer les cas d'utilisation** : il s'agit de dégager à partir du cahier de charges, un ensemble de cas d'utilisation (à la UML – *Unified Modeling Language*) du système ; ces cas d'utilisation sont matérialisés par des diagrammes de séquence qui constituent le résultat attendu de cette étape. Les diagrammes de séquence permettent d'identifier déjà les interactions présentes dans le système en termes de communications entre les différents rôles ;
- **affiner les fonctionnalités** : ici, le concepteur part du diagramme hiérarchisé des objectifs et des différents diagrammes de séquence pour produire un diagramme présentant d'une part les rôles avec les tâches qui leur sont associés en vue de remplir leurs objectifs et d'autre part, les communications entre les différents rôles sous la forme d'interactions entre les tâches. Ceci permet d'identifier aussi bien les communications entre rôles que celles à l'intérieur d'un même rôle. Le concepteur peut utiliser des automates à états finis pour spécifier les interactions entre les tâches, ceci décrit les communications devant apparaître dans le système en vue de lui permettre d'accomplir ses objectifs. Ici chaque tâche est exécutée dans le cadre d'un processus léger indépendant (*thread*), c'est pourquoi on parle dans MaSE de tâche concurrente. Les résultats attendus de cette étape sont le diagramme de rôles de MaSE et les différents diagrammes décrivant chaque tâche concurrente. Les résultats de cette étape servent de point de départ pour les étapes de la phase suivante.

La phase de conception comporte quatre étapes :

- **créer les classes d'agents** : ici, le concepteur assigne chaque rôle précédemment identifié à au moins une classe d'agent, on est ainsi sûr que le système remplira tous ses objectifs. On doit aussi exhiber l'ensemble des interactions dans lesquelles une classe d'agent sera impliquée, ceci se fait à partir des communications externes des rôles que remplit la classe d'agent. Une classe d'agent est à cette étape, définie par l'ensemble des objectifs qu'elle remplit et l'ensemble des interactions qui la concernent. Tous les détails concernant aussi bien l'architecture interne des agents que la description des interactions sont décrites dans les étapes suivantes. Le résultat de cette étape est un diagramme de classes d'agent reprenant pour chaque classe d'agent, les rôles qu'elle remplit et les interactions avec les autres classes d'agent ;
- **construire les communications entre agents** : le concepteur définit le détail des communications entre les agents en décrivant à l'aide d'un diagramme d'automate à états finis le comportement de chaque agent engagé dans chaque communication avec un autre agent. Ceci se fait à partir des diagrammes de tâches concurrentes. Le résultat de cette étape est un ensemble de diagrammes à états finis représentant chaque classe d'agent engagé dans une communication ;
- **assembler les agents** : c'est ici que le concepteur construit l'architecture interne des agents. Il est laissé au concepteur la possibilité soit de partir d'une architecture

existante comme l'architecture BDI, soit de construire lui-même sa propre architecture. Les concepteurs de MaSE se proposent d'offrir prochainement la possibilité de construire une architecture interne à partir de l'ensemble des objectifs et des tâches dégagés lors de la phase d'analyse. L'idée explorée serait de définir chaque tâche comme un élément interne de l'architecture de la classe d'agent. Chaque tâche concurrente serait transformée en une combinaison d'éléments du diagramme d'états internes et en un ensemble de conversations. Les activités identifiées dans une tâche concurrente devenant les méthodes associées à l'élément (représentant la tâche dans l'architecture interne). Dans le cadre de SIGFAD, nous allons utiliser l'architecture JAM pour définir l'intérieur de nos agents ;

- **concevoir le système** : ici, le concepteur instancie les classes d'agents définis précédemment. Ceci est comparable à l'instanciation des objets à partir des classes d'objets en POO. On utilise un diagramme de déploiement pour montrer le nombre, le type et la localisation des agents dans le système.

La méthodologie MaSE utilise une abstraction du paradigme d'objet considérant les agents comme des objets spécialisés. En effet, plutôt que d'être de simples objets avec des méthodes pouvant être invoquées par d'autres objets, les agents coordonnent leur comportement en communiquant entre eux. Les agents ont un comportement rationnel dans la mesure où ils essaient de remplir leurs objectifs propres et les objectifs du système tout entier. Les agents sont vus comme une abstraction convenable des objets et peuvent posséder ou pas des caractéristiques d'intelligence. Le concepteur quand il applique MaSE, n'est pas obsédé par l'implémentation des caractéristiques d'intelligence toujours difficiles à mettre en œuvre. Une des principales forces de MaSE est qu'elle offre un cycle de développement itératif et incrémental, tout élément identifié dans l'une des étapes de la phase d'analyse ou de conception peut être suivi en traçant les éléments qui lui sont associés dans les autres étapes soit en amont, soit en aval. MaSE associée à AgentTool ont été utilisés avec succès pour développer une douzaine de systèmes multi-agents comprenant parfois des centaines d'agents.

Le lecteur intéressé par une présentation plus détaillée de MaSE, d'AgentTool et de leurs applications peut consulter (DeLaach, 2001) ; (DeLoach et al., 2001) ; (Wood et DeLoach, 2000).

3-2 SIGFAD

Nous présentons dans cette section la spécification de SIGFAD en déroulant les différentes étapes de la méthodologie MaSE. Nous utilisons l'architecture JAM pour spécifier l'intérieur de nos agents.

Afin de permettre au lecteur de bien comprendre notre système, nous présentons dans la figure 5 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 pour accéder à la plate-forme et réaliser ses activités d'apprentissage. SIGFAD s'appuie sur la même structure, nous y distinguons trois types d'utilisateurs : les apprenants, les tuteurs, le coordonnateur et les natifs qui servent de compagnons d'apprentissage aux apprenants et sont classés comme apprenants. Chaque utilisateur a un agent, localisé sur le serveur, cet agent migre sur son poste dès qu'il se connecte. L'agent sera 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. Un agent superviseur des interactions est défini, il réside sur le serveur et est chargé des transactions avec la base de données. 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 construction de la base de données des interactions ;
- un agent superviseur des interactions.

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. C'est un langage basé sur les actes du langage naturel qui est utilisé surtout pour la communication entre agents. 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, c'est un langage déclaratif de type logique, utilisé pour l'échange de connaissances entre différents programmes. Les utilisateurs communiquent avec leurs agents à travers des interfaces graphiques (GUI pour *Graphic User Interface*). a_SI communique avec la base de données en utilisant des requêtes SQL (Structured Query Language).

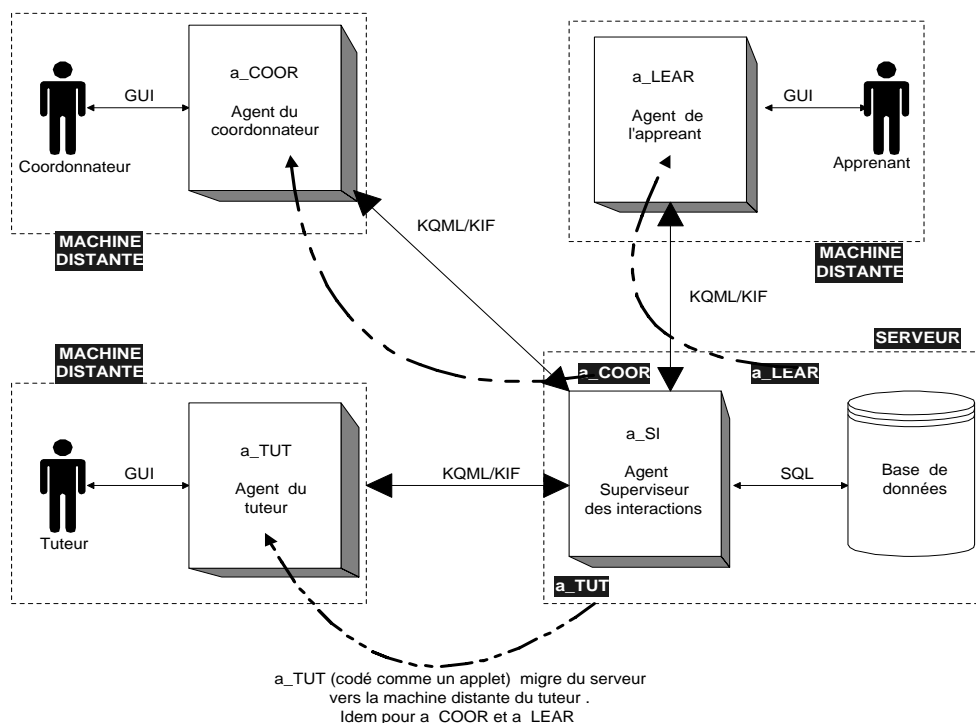


Fig. 4. Une vue d'ensemble de SIGFAD

3-2-1 LE DIAGRAMME HIERARCHISE DES OBJECTIFS DE SIGFAD

Le diagramme hiérarchisé des objectifs a pour principal objectif de résumer et d'organiser les fonctionnalités du système sous la forme d'une arborescence des objectifs et sous-objectifs. Les objectifs principaux de SIGFAD sont l'analyse des données enregistrées dans PIFAD et l'assistance aux utilisateurs engagés dans la session de FAD.

L'expérimentation Simuligne nous a donné l'occasion de mettre en relief les limites des plates-formes de formation à distance existantes. Nous avons noté qu'elles ne disposent pas d'outils automatiques capables d'afficher des informations sur la participation des acteurs. Nous avons particulièrement insisté sur le besoin de fournir aux personnes en charge de la cohésion, la maintenance et la durabilité des groupes, des outils capables de leur permettre d'apprécier le niveau d'activité dans les groupes et de déclencher des mécanismes de remédiation et d'assistance nécessaires.

La figure 6 présente le diagramme hiérarchisé des objectifs de SIGFAD. SIGFAD a des objectifs liés à l'assistance aux utilisateurs (objectif 1.1 et les sous-objectifs associés) et des objectifs dédiés à la sauvegarde et à l'extraction des données d'interaction (objectif 1.2 et les sous-objectifs associés). L'objectif *Analyse des interactions* consiste à examiner automatiquement les données d'interaction dans le but d'alerter des apprenants présentant un risque d'échec. Les objectifs d'assistance aux utilisateurs concernent : la fourniture des informations sur leurs interactions, l'évaluation de la charge de travail sur une période donnée, la réorganisation des groupes, le recadrage du calendrier des activités dans les groupes.

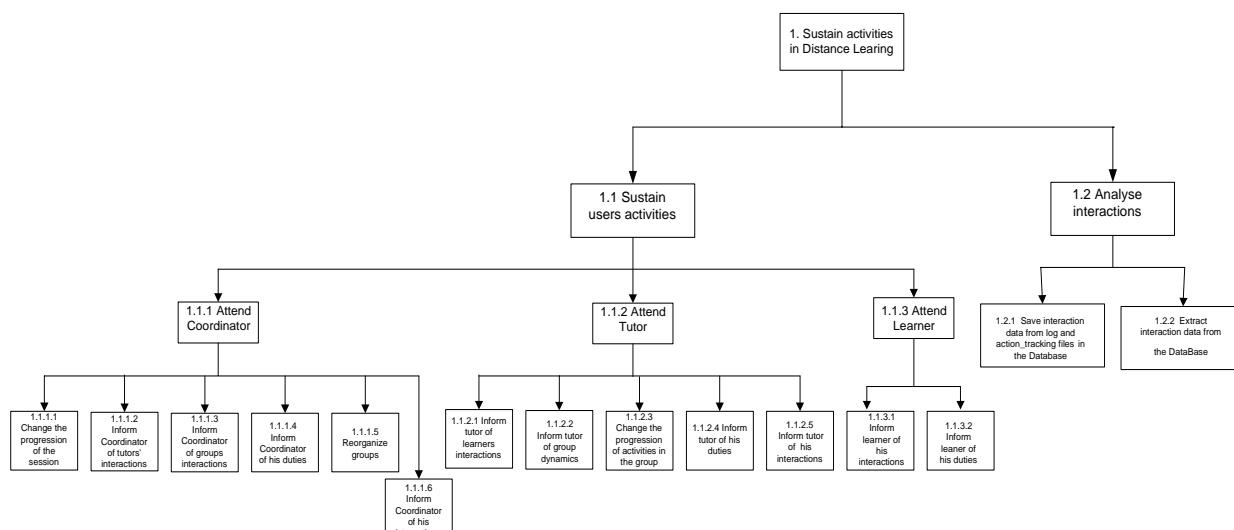


Fig. 5. Le diagramme hiérarchisé des objectifs de SIGFAD

En vue de mieux cerner la faisabilité de SIGFAD, nous avons dégagé un sous-ensemble de fonctionnalités pour lesquels nous sommes allés jusqu'à l'implémentation. La base de données des interactions ayant été construites au long du déroulement de SimuLigne, nous ne nous sommes attardés sur la construction de cette base de données par l'agent qui en est chargé dans SIGFAD. Donc dans ce qui suit, il ne sera pas fait mention de la construction de la base de données par notre SMA. Rappelons que cette base de données a été construite par des applications écrites en PHP intégrant le langage SQL et qu'il s'agit d'une base de données MySQL.

Les fonctionnalités implémentées sont les suivantes :

- a) **Surveiller l'activité du groupe** : cet objectif consiste à partir des données d'interaction construites quotidiennement (cumulées ou non sur une période) telles que le nombre de membres du groupe connectés, le pourcentage de temps passé en ligne dans le groupe, le nombre de courriels lus/émis, le nombre de messages lus/postés dans les forums à

apprécier l'état d'activité et/ou de cohésion du groupe, à savoir si une activité est correctement réalisée par le groupe ou pas, à déterminer le nombre de personnes actives dans le groupe.

- b) **Repérer les utilisateurs dormants** : catégoriser sur une période donnée, les utilisateurs actifs, les dormants – qui se sont connectés mais qui ne réalisent aucune activité, absents – non connectés.
- c) **Indiquer la proximité des interventions des tuteurs dans les forums** : l'idée étant de voir si les interventions sont judicieusement placées dans le temps pour permettre le bon déroulement du forum.
- d) **Fournir les ratios temps de connexion effectif/prévu** : indiquer les temps de connexion des apprenants/utilisateurs et comparer ces temps avec ce que prévoit le contrat pédagogique

Nous présentons donc le diagramme hiérarchisé des objectifs issus des fonctionnalités ci-dessus énumérées. Les autres étapes de la méthodologie MaSE s'appliquent elles aussi à ce sous-ensemble de fonctionnalités de SIGFAD.

Les fonctionnalités ci-dessus peuvent être organisées en objectifs :

1. *Déterminer les apprenants présents/absents sur une période, dans un groupe donnée.* Cette fonctionnalité permettrait par exemple au tuteur de savoir à tout moment et une période spécifiée les apprenants de son groupe qui se sont connectés. Il convient de préciser que notre PIFAD permet à un utilisateur d'entrer dans la plate-forme sans nécessairement entrer dans son groupe de base. Cette fonctionnalité permet en fait de déterminer les apprenants qui ne sont pas entrés dans leur groupe pendant une période donnée. Le tuteur ayant en charge le déroulement des activités qui se déroulent dans son groupe pourra donc à partir des utilisateurs présents dans le groupe, déterminer les apprenants encore engagés dans la formation.
2. *Déterminer les apprenants actifs (communiquent et réalisent les activités) pour un groupe donné.* Nous entendons par apprenant actif, un apprenant qui est présent dans son groupe et réalisent les activités prévues. N'accédant pas au contenu des productions et n'en faisant dans tous les cas une analyse sémantique, cette fonctionnalité a simplement pour but de déterminer les traces des activités des utilisateurs repérables à travers les outils utilisés. Nous considérons en général qu'un apprenant actif est un apprenant qui ne se contente pas simplement de lire les productions des autres et des messages dans les courriels et les forums. Un apprenant actif est une personne dont l'activité est perceptible au sein du groupe, par les autres membres du groupe et dont la production contribue à la réalisation de l'activité et des objectifs d'apprentissage. Cet objectif est central dans le rôle du tuteur, c'est grâce à lui qu'il pourra le dynamisme des apprenants et la vitalité de son groupe. C'est aussi l'objectif qui s'adresse le mieux à l'une des difficultés majeures que l'on rencontre en FAD : comment savoir ce que fait un apprenant ? la charge de travail est-elle supérieure aux capacités de l'apprenant et nécessite-t-elle de lui fournir de l'aide ou de modifier le calendrier de la formation, la scénarisation de la formation ou même les objectifs d'apprentissage ?
3. *Déterminer les apprenants dormants (se connectent mais ne communiquent pas et ne réalisent pas les activités attendues) dans un groupe donné.* Cet objectif est intimement lié au précédent en ce sens qu'on pourrait dire en première approximation qu'un apprenant dormant est un apprenant non actif. Toutefois, il convient certainement de nuancer un peu cette première approximation. Un apprenant sera dit actif au sens de la justesse de ses actions par rapport aux objectifs pédagogiques. On

pourrait dire qu'un apprenant actif est un apprenant qui agit et réalise correctement les activités programmées. Un apprenant dormant est un apprenant qui ne produit rien, indépendamment de la justesse de son activité. Un apprenant qui communique par exemple énormément avec les autres membres du groupe en utilisant les outils de communication disponibles n'est pas actif si ces interactions ne rentrent pas dans la satisfaction des objectifs pédagogiques. Par contre cet apprenant n'est pas un apprenant dormant en ce sens qu'il a bien une production perceptible par le reste du groupe.

4. *Déterminer les groupes dynamiques (par rapport au nombre de personnes actives qu'ils contiennent).* Cet objectif situe l'échelle d'analyse au niveau du groupe tout entier. Il indiquera le caractère "dynamique" d'un groupe en fonction du pourcentage de personnes actives sur une période donnée.
5. *Déterminer si un groupe réalise ou non une activité donnée.* Cet objectif situe l'échelle d'analyse au niveau de l'activité à réaliser et/ou des tâches qui la composent. L'idée est de permettre au tuteur ou au coordonnateur de savoir si une activité est réalisée dans les délais prévus par le groupe ou le sous-groupe concerné. Ceci permettrait aussi de modifier la progression pédagogique pour tenir compte du niveau de réalisation d'une activité. Là aussi, ne procédant à aucune analyse sémantique des contenus, il s'agira simplement à partir des traces de l'utilisation des outils de la PIFAD et des dates de réalisation de faire une prédiction sur la réalisation d'une activité donnée.
6. *Déterminer si un tuteur a bien démarré/clôturé un forum : temps entre la date d'ouverture d'un forum et la date d'envoi du premier message ; temps entre la date du dernier message et la date de clôture du forum.* Le forum est l'un des outils les plus adaptés à l'animation et à la gestion des groupes d'apprentissage en FAD. Ceci prend une importance singulière lorsqu'il s'agit d'un apprentissage collaboratif. Il est donc très important que le tuteur puisse gérer très efficacement les forums ouverts durant la session de formation. Il en est très souvent le modérateur. Il doit donc pouvoir intervenir à des moments bien précis en vue d'assurer la réussite de tous les forums. Le forum est aussi probablement l'un des outils majeurs de création et de maintien du sentiment d'appartenance au groupe, à la communauté. Son caractère asynchrone et collectif lui assure une grande efficacité comparé à d'autres outils de communication comme le bavardage, le tableau blanc, le courriel, l'envoi de fichiers.
7. *Fournir pour chaque utilisateur le ratio temps de connexion effectif/prévu sur une semaine.* Cet objectif permet de réaliser les écarts entre le contrat pédagogique et la réalité de la formation. Il s'agit de savoir si l'apprenant passe un temps en ligne en rapport avec ce qui est prévu dans les consignes pédagogiques. Ceci permet aussi au tuteur d'apprécier le niveau d'engagement de chaque apprenant.

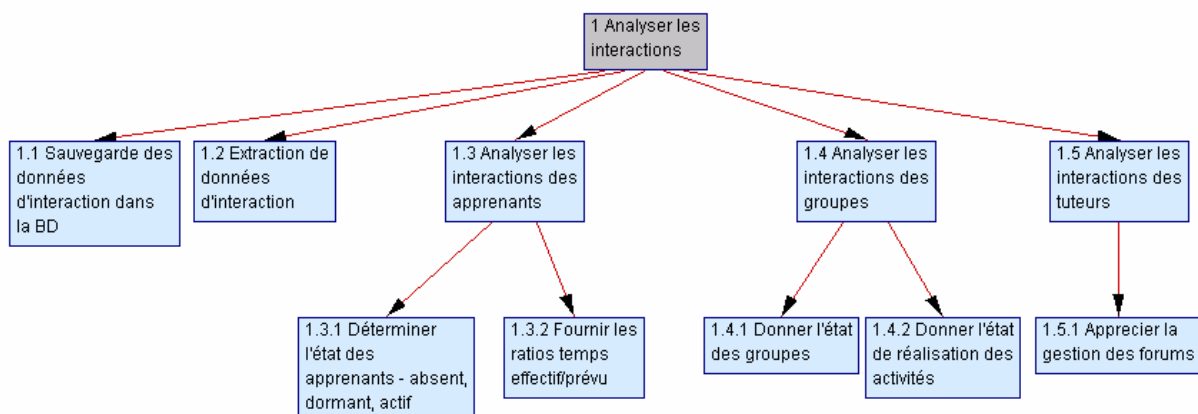


Fig. 6. Le diagramme hiérarchisé des objectifs de SIGFAD

3-2-2 L'APPLICATION DES CAS D'UTILISATION

L'objectif de cette étape de MaSE est d'extraire les différents cas d'utilisation du cahier de charges initial du système. Les cas d'utilisation sont explicitées dans des diagrammes de séquence. Cette étape a aussi pour but de permettre au concepteur d'identifier un ensemble initial de rôles et d'interactions (entre ces rôles) qui apparaîtront dans le système multi-agents. Les différents diagrammes matérialisent l'ensemble des rôles et des interactions. Dans la spécification de SIGFAD, nous avons identifié neuf cas d'utilisation représentés neuf cas diagrammes de séquence.

Nous présentons dans cette section la description de tous les cas d'utilisation de SIGFAD en précisant les données à partir desquelles le SMA réalisera les objectifs associés.

Les rôles définis dans SIGFAD sont les suivants :

- H_COOR encapsule l'interface entre le coordonnateur et son agent.
- H_TUT encapsule l'interface entre le tuteur et son agent.
- DataBase encapsule l'interface SIGFAD- BDD (la base de données contenant les données d'interaction)
- r_COOR est le rôle que jouera l'agent du coordonnateur
- r_SI est le rôle que jouera l'agent superviseur des interactions.
- r_TUT est le rôle qui sera joué par l'agent du tuteur.

3-2-2-1 Fournir la liste des apprenants connectés durant une période

Cas d'utilisation n° 1 : Lister les apprenants absents du groupe

Ce cas d'utilisation décrit la façon dont le système fournit au tuteur la liste des apprenants absents de son groupe. La demande est initiée par l'agent du tuteur en fonction de la sa représentation du monde ; elle est adressée à l'agent superviseur des interactions. Ce dernier scrute les données d'interaction et fournit l'information demandée à l'agent du tuteur qui peut la transmettre au tuteur. Les rôles concernés sont : r_SI, r_TUT, H_TUT, DataBase

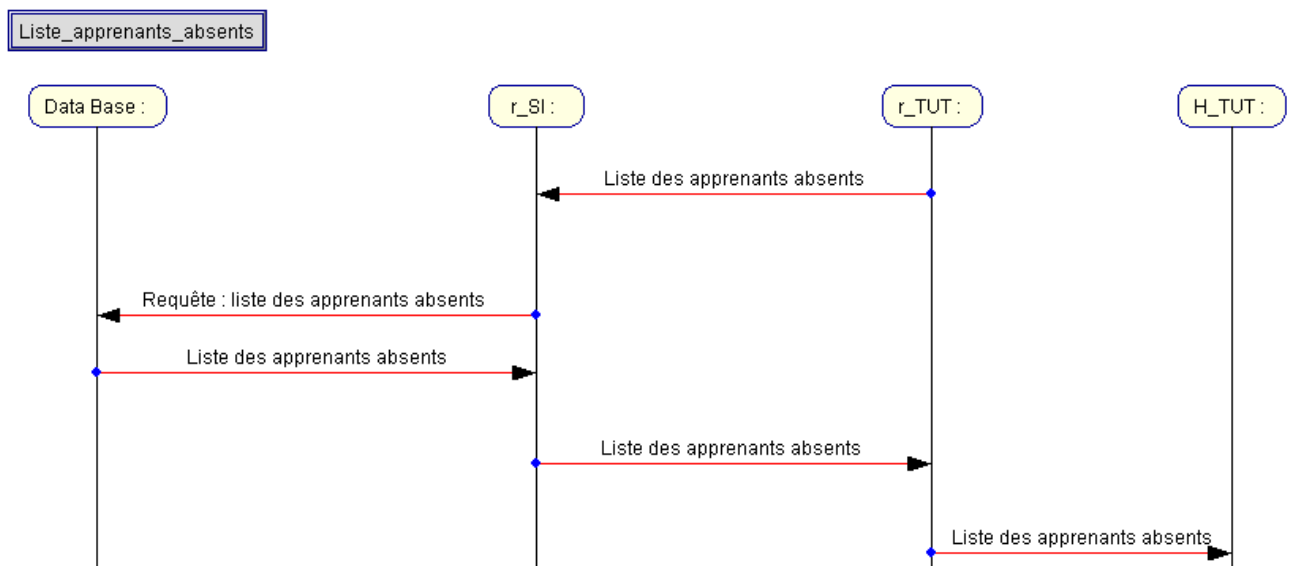


Fig. 7. Le diagramme de séquence fournissant la liste des apprenants absents

SIGFAD utilisera les traces fournies par le serveur HTTP pour déterminer les utilisateurs qui sont entrés dans leur groupe. Ces informations sont déjà sauvegardées dans la base de données et accessibles au moyen de requêtes SQL contenues dans des formulaires de requêtes HTML/PHP.

3-2-2-2 Fournir la liste des apprenants dormants durant une période

Cas d'utilisation n° 2 : Lister les apprenants dormants du groupe

Ce cas d'utilisation décrit la façon dont le système fournit au tuteur la liste des apprenants dormants de son groupe. La demande est initiée par l'agent du tuteur en fonction de la sa représentation du monde ; elle est adressée à l'agent superviseur des interactions. Ce dernier scrute les données d'interaction et fournit l'information demandée à l'agent du tuteur qui peut la transmettre au tuteur. Les rôles concernés sont : r_SI, r_TUT, H_TUT, DataBase

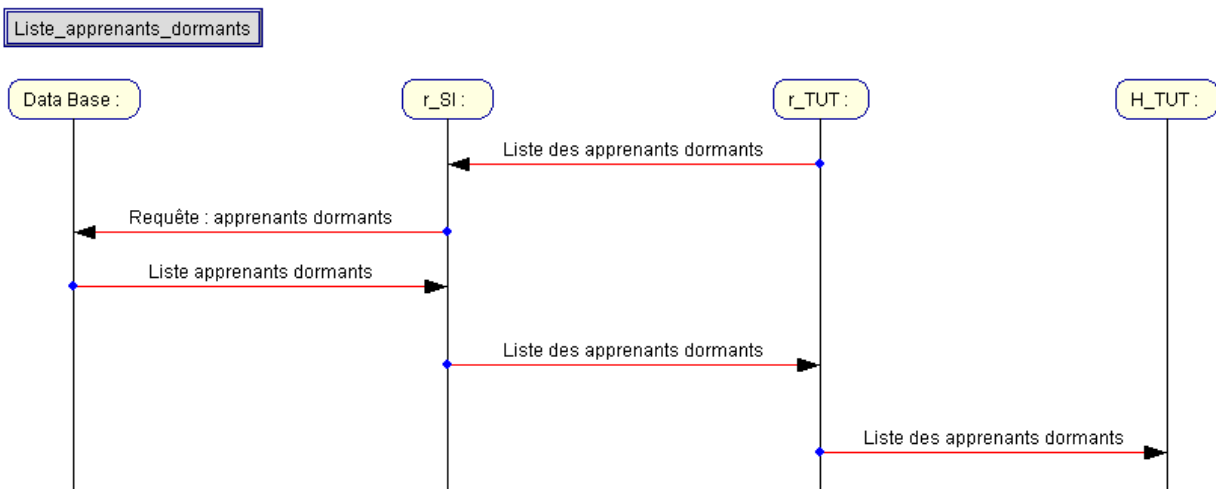


Fig. 8. Le diagramme de séquence fournissant la liste des apprenants dormants

SIGFAD utilisera les traces fournies par le serveur HTTP pour déterminer les utilisateurs qui sont entrés dans leur groupe mais ne font rien d'autre que lire des courriels et des forums ; nous considérons un apprenant qui ne fait rien d'autre que lire des messages comme un apprenant dormant. Ces informations sont déjà sauvegardées dans la base de données et accessibles au moyen de requêtes SQL contenues dans des formulaires de requêtes HTML/PHP. Ces requêtes permettent d'obtenir sur une période la liste des personnes ayant posté un message dans les forums, écrit un courriel, ayant participé à un bavardage, ayant publié un article, ayant téléchargé un fichier, ayant accédé à des pages de contenus (consignes), ayant consulté l'agenda, ayant lu des messages dans les forums ou lu des courriels.

3-2-2-3 Fournir la liste des apprenants actifs durant une période

Cas d'utilisation n° 3 : Lister les apprenants actifs du groupe

Ce cas d'utilisation décrit la façon dont le système fournit au tuteur la liste des apprenants actifs de son groupe. La demande est initiée par l'agent du tuteur en fonction de la sa représentation du monde ; elle est adressée à l'agent superviseur des interactions. Ce dernier scrute les données d'interaction et fournit l'information demandée à l'agent du tuteur qui peut la transmettre au tuteur. Les rôles concernés sont : r_SI, r_TUT, H_TUT, DataBase

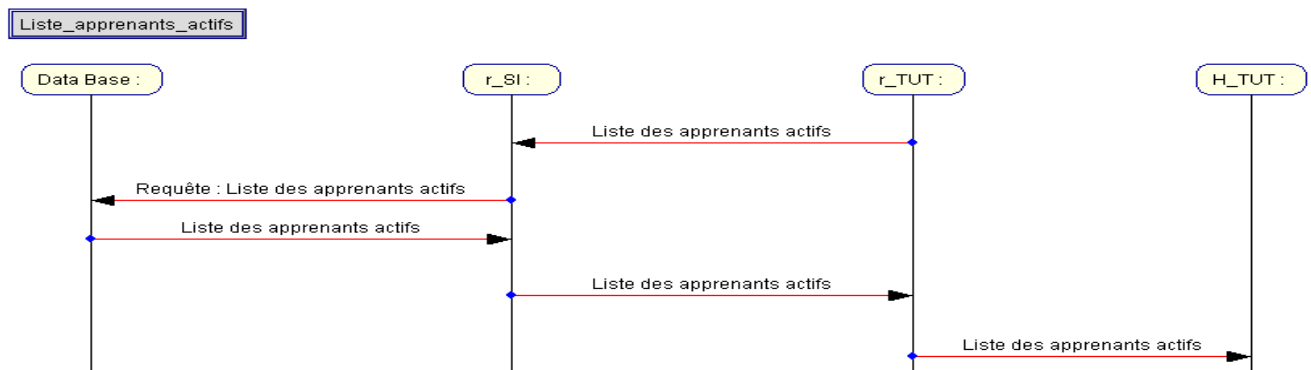


Fig. 9. Le diagramme de séquence fournissant la liste des apprenants actifs

SIGFAD utilisera les traces fournies par le serveur HTTP pour déterminer les utilisateurs qui sont entrés dans leur groupe et les actions qu'ils effectuent. Ces informations sont déjà sauvegardées dans la base de données et accessibles au moyen de requêtes SQL contenues dans des formulaires de requêtes HTML/PHP. Ces requêtes permettent d'obtenir sur une période la liste des personnes ayant posté un message dans les forums, écrit un courriel, ayant participé à un bavardage, ayant publié un article, ayant téléchargé un fichier, ayant accédé à des pages de contenus (consignes), ayant consulté l'agenda, ayant lu des messages dans les forums ou lu des courriels.

3-2-2-4 Fournir les ratios temps effectif/prévu par semaine

Cas d'utilisation n° 4 : Fournir les ratio temps effectif/prévu

Ce cas d'utilisation décrit la façon dont le système calcule pour chacun des apprenants du groupe, le ratio entre le temps passé en ligne pendant une semaine et le temps prévu dans le contrat pédagogique (consignes). Ce calcul est fait en fin de semaine (le dimanche à 24 H GMT) par l'agent superviseur des interactions. Les ratios des apprenants du groupe sont envoyés à l'agent du tuteur qui peut les communiquer au tuteur. Les rôles concernés sont : r_SI, r_TUT, H_TUT, DataBase

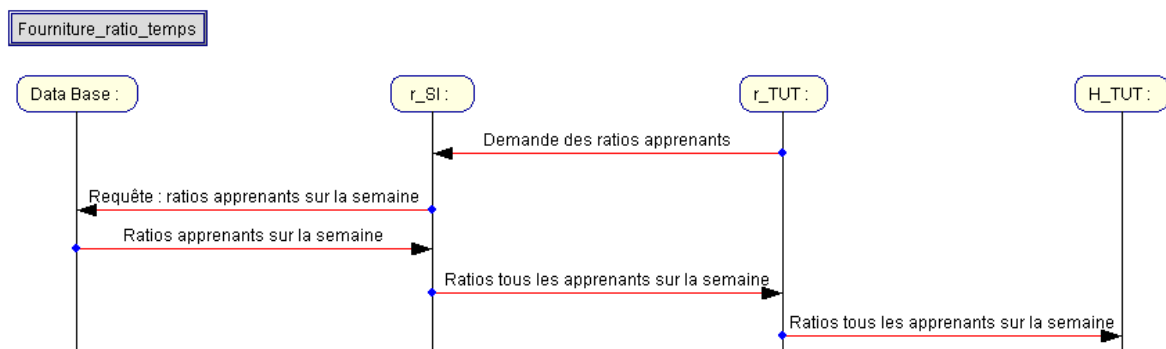


Fig.10. Le diagramme de séquence fournissant les ratio de temps effectif/prévu

SIGFAD utilisera les traces fournies par le serveur HTTP pour déterminer les temps de connexion des apprenants dans leur groupe. Ces informations sont déjà sauvegardées dans la base de données et accessibles au moyen de requêtes SQL contenues dans des formulaires de requêtes HTML/PHP.

3-2-2-5 Donner l'état du groupe

Cas d'utilisation n° 5 : Donner l'état du groupe

Ce cas d'utilisation décrit la façon dont le système détermine et indique l'état du groupe. La demande est initiée par l'agent du tuteur en fonction de la sa représentation du monde ; elle est adressée à l'agent superviseur des interactions. Ce dernier scrute les données d'interaction et détermine le pourcentage d'apprenants actifs sur les 3 derniers jours. En fonction de ce pourcentage, il envoie une des mentions suivantes :

- pas dynamique – moins de 20% des apprenants sont actifs,
- peu dynamique – entre 20 et 40% des apprenants sont actifs,
- assez dynamique – entre 40 et 60% des apprenants sont actifs,
- dynamique – entre 60 et 80 % des apprenants sont actifs,
- très dynamique – plus de 80% des apprenants sont actifs.

Les rôles concernés sont : r_SI, r_TUT, H_TUT, DataBase

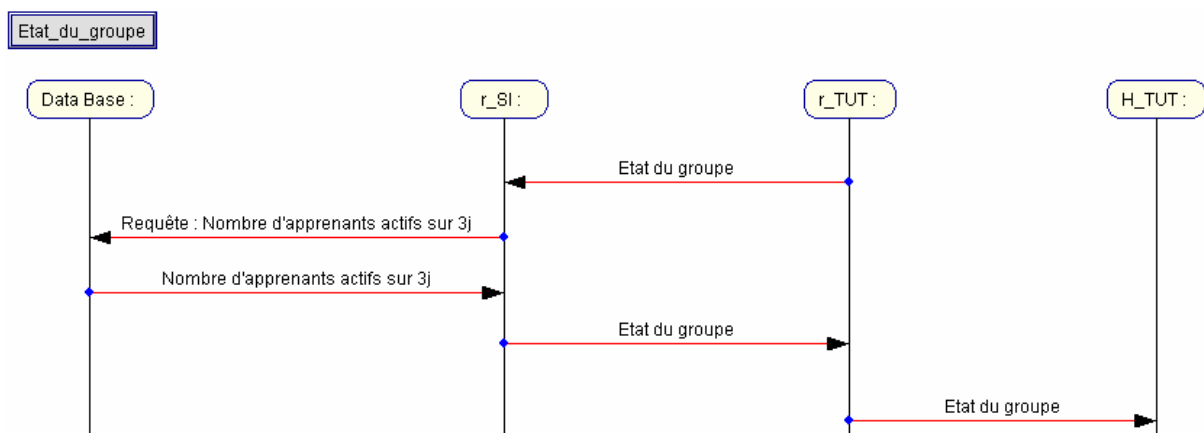


Fig.11. Le diagramme de séquence fournissant l'état du groupe

L'appréciation de l'état du groupe se fera à partir du nombre d'apprenants actifs, dormants ou absents du groupe. Les données utilisées sont donc celles issues des traces fournies par le serveur HTTP, sauvegardées dans la base de données et accessibles au moyen de requêtes SQL contenues dans les formulaires HTML/PHP.

3-2-2-6 Donner l'état de réalisation d'une activité dans un groupe

Ce cas d'utilisation décrit la façon dont SIGFAD indiquera si une activité donnée est potentiellement en cours de réalisation ou non dans un groupe.

A la date de début de l'activité et pour chacune des activités, le coordinateur (via son agent qui communiquera les informations à l'agent superviseur des interactions) indiquera la date de début de l'activité, la date de fin de l'activité, les actions repérables dans la plate-forme que l'apprenant est censé effectuer durant la réalisation de l'activité :

- poster un message dans un forum ;
- lire des messages dans un forum
- poster un courriel ;
- lire un courriel ;
- publication d'un document ;
- remise d'un document par l'outil à rendre ;
- utilisation du tableau blanc ;

- téléchargement d'un fichier ;
- télécharger les consignes ;
- accéder à la page des consignes en ligne ;
- etc.

à mi-parcours de la réalisation de l'activité, l'agent superviseur des interactions établira la liste des apprenants ayant effectué des actions prévues et repérables afin de dire si oui ou non, l'activité ou la tâche est en train d'être réalisée par l'apprenant. Il fournira à l'agent du tuteur la liste des membres, qui d'après son raisonnement, effectuent l'activité ou non.

Cette vérification sera de nouveau effectuée à la date de clôture de l'activité, elle permettra à l'agent superviseur des interactions de fournir une liste des apprenants ayant oui ou non effectué l'activité.

Cas d'utilisation n° 6 : Enregistrer les méta-informations sur les activités

Ce cas d'utilisation décrit la façon dont le système enregistre les informations nécessaires au suivi des activités – date de début, date de fin, outils à utiliser, tâches à effectuer, temps nécessaires estimés en ligne et hors ligne, dates limite intermédiaires. Les informations sont enregistrées par l'agent du coordonnateur et communiquées à l'agent superviseur des interactions. Les rôles concernés sont : H_COOR, r_COOR, r_SI.

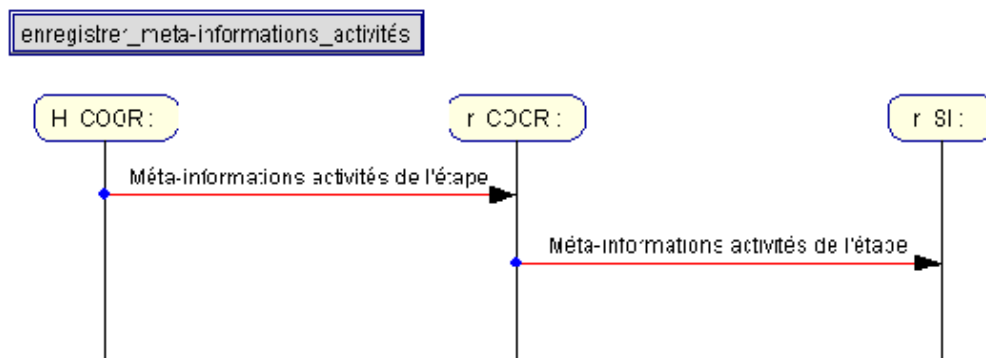


Fig.12. Le diagramme de séquence enregistrant les méta-informations sur les activités

Cas d'utilisation n° 7 : Donner l'état de réalisation d'une activité

Ce cas d'utilisation décrit la façon dont le système suit et indique l'état de réalisation d'une activité. A partir des informations méta sur l'activité, enregistrées au début de l'étape – date de début, date de fin, tâches, etc. – l'agent superviseur des interactions scrute les données d'interaction et indique si une activité est bien réalisée au sein du groupe ou pas. Le système détermine la durée T de l'activité. L'évaluation de l'état de réalisation de l'activité est effectuée au début de l'activité (date de début + T/10), à la mi-parcours et à la fin de l'activité (date de fin prévue). Les rôles concernés : r_SI, r_TUT, H_TUT, Database.

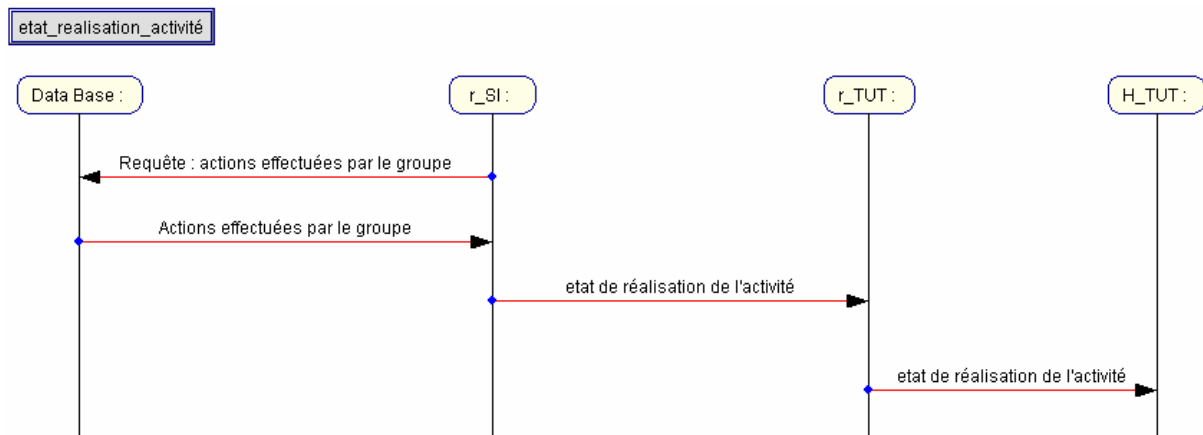


Fig.13. Le diagramme de séquence fournissant l'état de réalisation d'une activité dans un groupe

SIGFAD utilisera les traces fournies par le serveur HTTP pour déterminer si une activité est potentiellement en train d'être réalisée. Ces informations sont déjà sauvegardées dans la base de données et accessibles au moyen de requêtes SQL contenues dans des formulaires de requêtes HTML/PHP. Ces requêtes permettent de savoir si un apprenant a posté un message dans un forum, a écrit un courriel, a participé à un bavardage, a publié un article, a téléchargé un fichier, a accédé à des pages de contenus (consignes), a consulté l'agenda, a remis un document par l'intermédiaire de l'outil à rendre, a lu des messages dans les forums ou lu des courriels.

3-2-2-7 Indiquer si un forum est bien suivi par le tuteur

Ce cas d'utilisation décrit comment SIGFAD déterminera si un forum est bien suivi ou non par le tuteur. Ceci se fait en déterminant l'ordre de postage du premier message du tuteur et en comparant la date de postage de ce premier message et d'ouverture prévu du système. Une autre fonctionnalité permettra de déterminer si le tuteur lit régulièrement les messages du forum. A partir de ces informations déterminées au début du forum, à la mi-parcours du forum et à la date de clôture du forum, SIGFAD pourra dire si un tuteur anime bien un forum donné. Les informations concernant le forum – date d'ouverture et de clôture du forum – devront au préalable avoir été communiquées à l'agent superviseur des interactions, elles seront fournies par le coordonnateur de la formation à son agent qui les communiquera à l'agent superviseur des interactions.

A partir de ces informations, l'agent superviseur des interactions utilisera les données d'interaction pour déterminer si un forum est bien administré ou pas, avec des mentions :

- mal ouvert : si le tuteur n'a posté aucun message dans le forum à l'ouverture ;
- bien ouvert : si le tuteur a posté son premier message avant tous les autres et si celui-ci est proche du jour d'ouverture prévu ;
- mal suivi : si le tuteur ne lit pas les messages dans le forum ;
- bien suivi : si le tuteur lit régulièrement les messages postés dans le forum ;

A la date de clôture prévue d'un forum, l'agent superviseur des interactions vérifiera si le forum a été bien clôturé par le tuteur. Il déterminera l'ordre du dernier message posté par le tuteur et en fonction de la réponse obtenue, fournira la mention :

- bien clôturé : si le dernier message du tuteur est bien le dernier message dans le forum et si la date d'émission de ce message est proche de la date de clôture prévue du forum ;

- mal clôturé : si le message du tuteur n'est pas le dernier de la liste ou si la date d'émission de ce dernier est éloignée de la date de clôture prévue.

Cas d'utilisation n° 8 : Enregistrer les méta-informations sur les forums

Ce cas d'utilisation décrit la façon dont le système enregistre les informations qui lui permettront d'indiquer si un forum est bien géré par un tuteur. Les informations sont recueillies par l'agent du coordonnateur au début de chaque étape. Ces informations sont ensuite communiquées à l'agent superviseur des interactions. Les rôles concernés sont : H_COOR, r_COOR, r_SI.

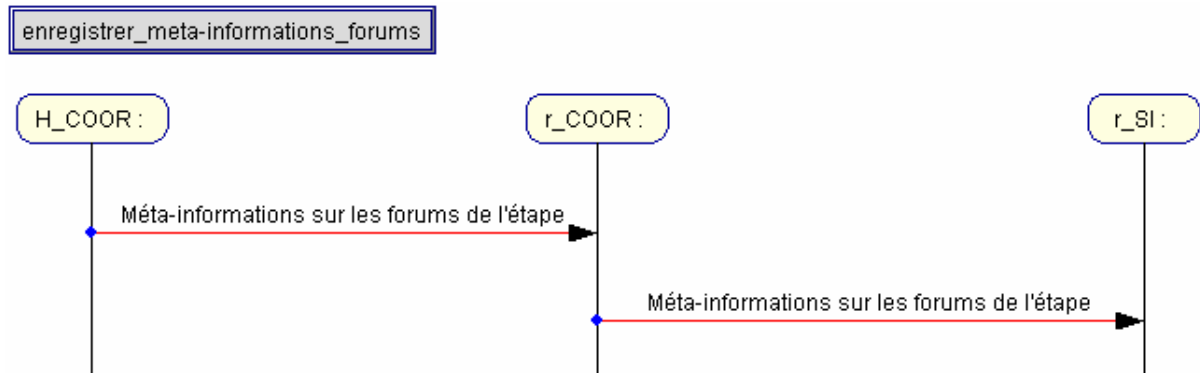


Fig.14. Le diagramme de séquence enregistrant les méta-informations sur les forums

Cas d'utilisation n° 9 : Apprécier la gestion des forums

Ce cas d'utilisation décrit la façon dont le système apprécie et indique si un forum est bien géré par un tuteur. Pour chaque forum, l'agent superviseur des interactions ira scruter les données pour voir si le tuteur a bien posté un message à l'ouverture du forum, s'il lit régulièrement les messages dans le forum. Le système détermine la durée de vie du forum T. L'évaluation de la gestion du forum est faite à l'ouverture du forum (date de début + T/10), à la mi-parcours (date de début + T/2). A la clôture du forum, l'agent superviseur des interactions regardera la proximité et la position du dernier message du tuteur (fil de discussion) pour indiquer si le forum a été bien clôturé ou pas par le tuteur. L'appréciation de la gestion d'un forum est communiquée à l'agent du tuteur. Les rôles concernés sont : r_SI, r_TUT, H_TUT, Database.

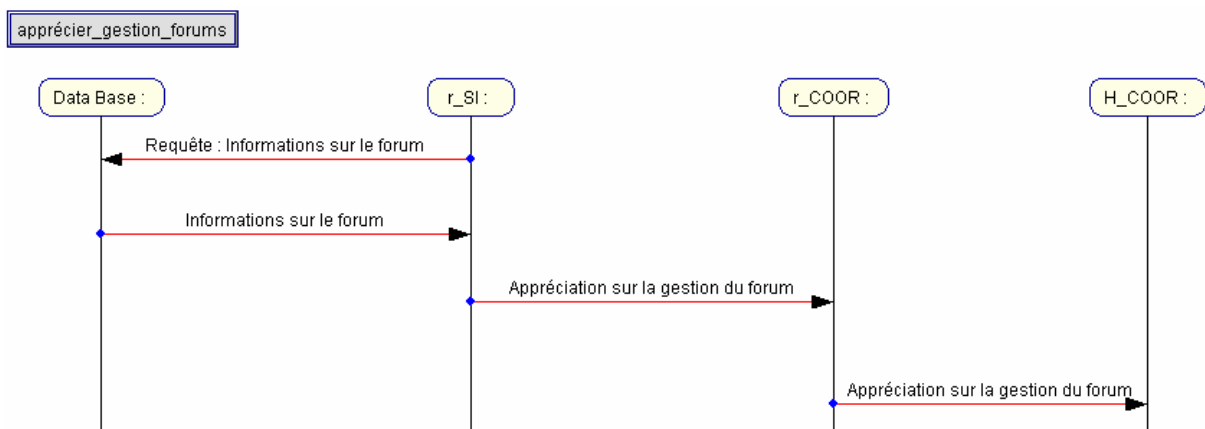


Fig.15 Le diagramme de séquence permettant d'apprécier la gestion des forums

SIGFAD utilisera les traces contenues dans les fichiers spécifiques de WebCT pour savoir l'heure de postage d'un message dans un forum par le tuteur, la date et l'heure d'émission du

message, la date d'ouverture effective d'un forum, la date de fermeture d'un forum. Ces fichiers permettent de savoir dans quel fil de discussion le tuteur poste un fichier. Les données utilisées sont donc essentiellement contenues dans les fichiers spécifiques de WebCT.

3-3-3 LE DIAGRAMME DE ROLES DE MASE ET LES TACHES CONCURRENTES

L'objectif de cette étape est de transformer le diagramme hiérarchisé des objectifs (figure 6) et les diagrammes de séquence (figures 7 à 15) issus des différents cas d'utilisation, en un diagramme de rôles associés aux tâches concurrentes. Les tâches sont des processus indépendants que le rôle doit exécuter pour remplir ses objectifs ; étant indépendants, ils s'exécutent de façon parallèle, c'est pourquoi l'on parle de tâches concurrentes. Les interactions inter et intra-rôles sont matérialisées par des chemins de communication entre les tâches, ces communications étant elles aussi supposées concourir à la réalisation des objectifs assignés à chaque rôle. Afin de s'assurer que tous les objectifs du système seront remplis, chacun des objectifs ou sous objectifs du diagramme hiérarchisé des objectifs est assigné à un rôle particulier. La méthodologie MaSE indique que chaque objectif doit être assigné à au moins un rôle, laissant ainsi supposer qu'un objectif peut être réalisé par plusieurs rôles mais AgentTool, l'environnement mis au point pour instancier MaSE, oblige à assigner un objectif à un rôle et un seul. Chaque tâche concurrente doit être spécifiée par un diagramme d'automate à états finis qui en fait décrit le comportement du rôle (auquel la tâche appartient) pendant la communication.

3-3-3-1 Le diagramme des rôles de SIGFAD

Dans la figure 16, nous présentons le diagramme des rôles de SIGFAD. Le diagramme des rôles de SIGFAD comprend six rôles auxquels sont associés dix-huit tâches concurrentes, neuf protocoles de communication. Par souci de simplicité, nous n'avons pas représenté les protocoles de communication intra-rôles. Nous parlons de protocole de communication pour utiliser le vocabulaire anglo-saxon des auteurs de MaSE, il faudrait comprendre cette expression comme un chemin de communication. Plus simplement, on pourrait dire que le diagramme nous a permis d'exhiber neuf communications entre rôles.

Comme nous l'avons dit plus haut, les rôles exhibées dans cette étape proviennent des différents diagrammes de séquence obtenus lors de l'étape précédente. Il en de même pour les communications entre tâches. L'assignation des objectifs aux rôles permet de définir les tâches associées aux différents rôles.

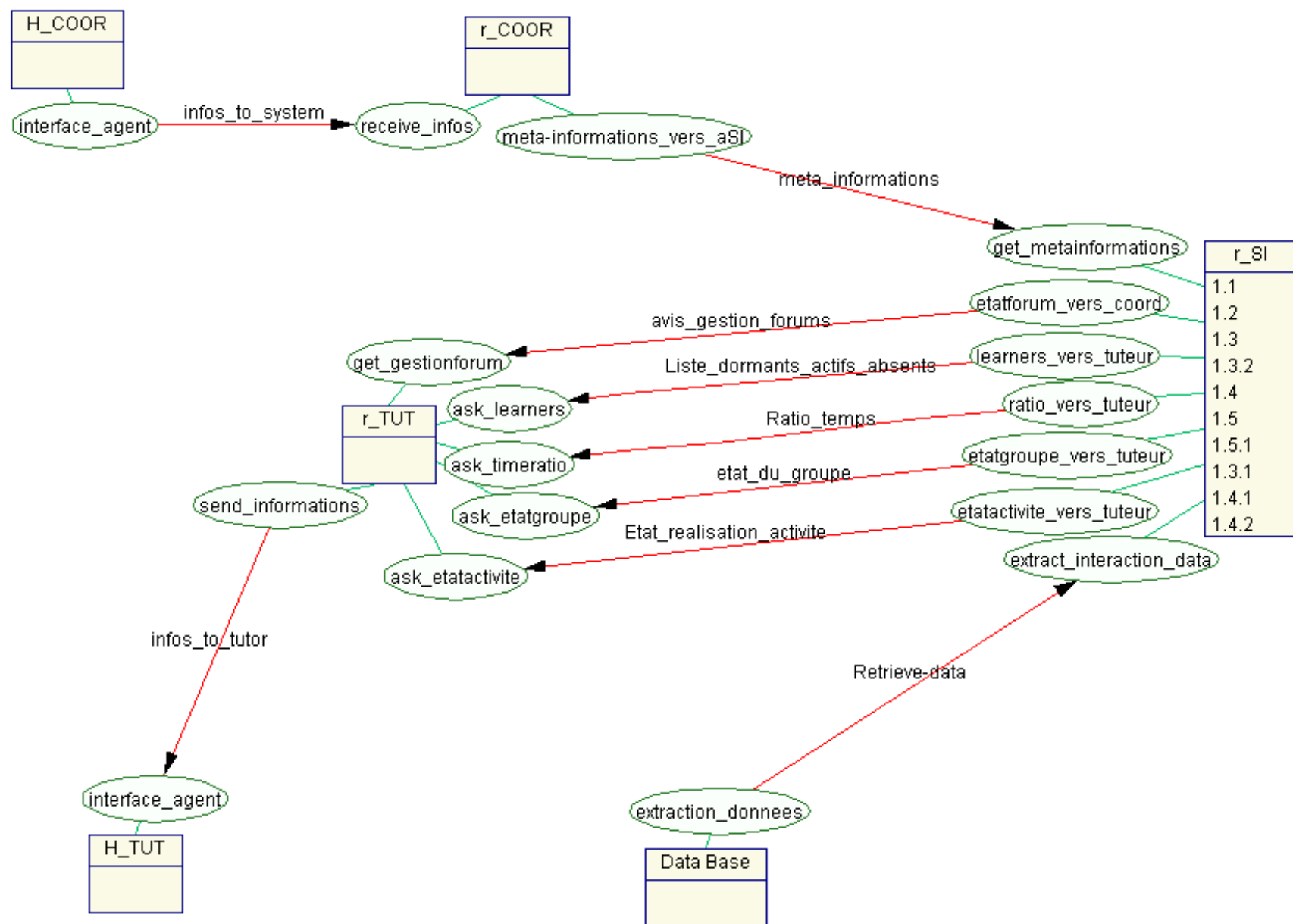


Fig. 16. Le diagramme de rôles (à la MaSE) de SIGFAD

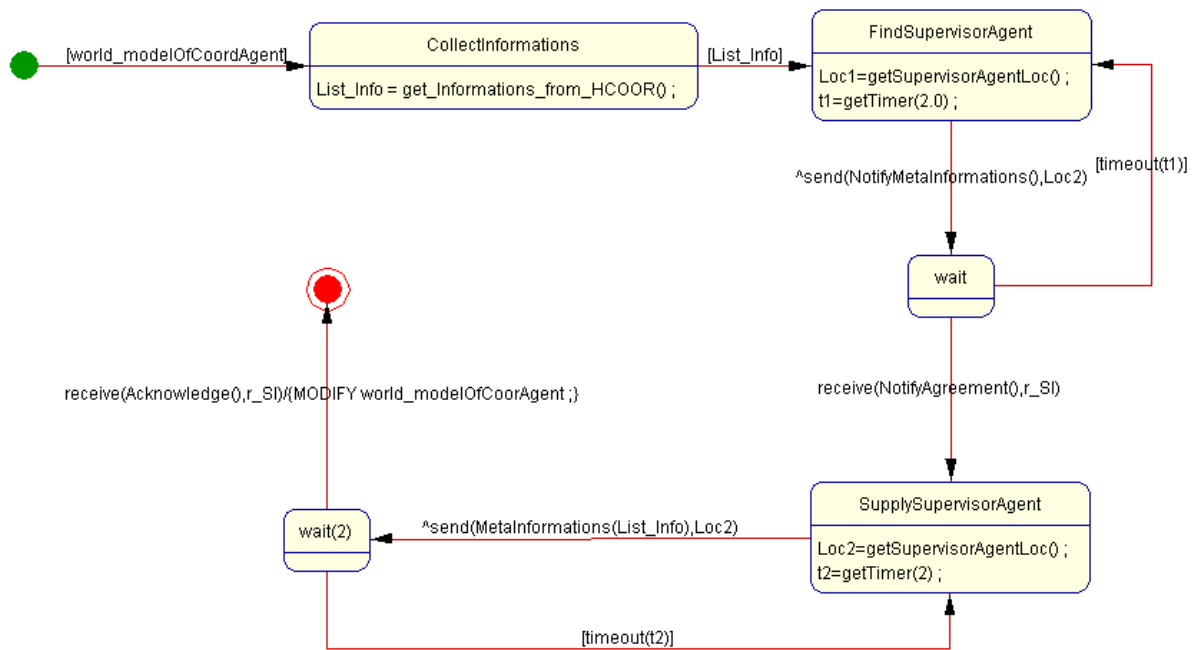
3-3-3-2 La description des tâches concurrentes

a - La fourniture des meta-informations concernant les activités et les forums

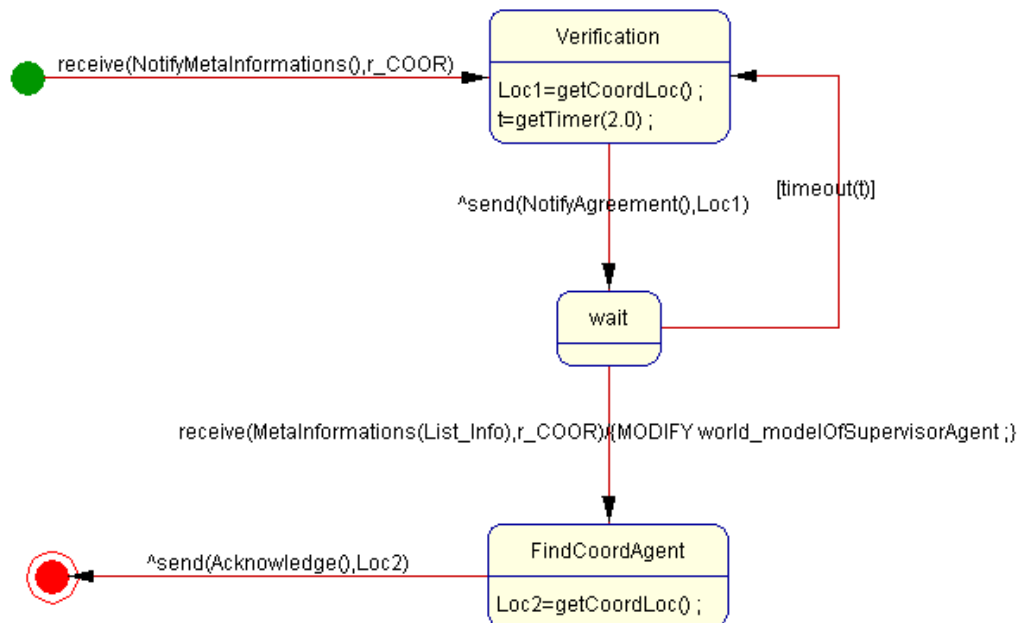
Ce protocole de communication décrit la façon dont l'agent du coordonnateur (r_COOR) fournira les méta-informations à l'agent superviseur des interactions (r_SI). Il met en œuvre les rôles r_COOR et r_SI avec leurs tâches concurrentes respectivement, *meta-informations_vers_aSI* et *get_metainformations*.

Nous présentons ci-après les deux diagrammes à automates finis décrivant ces deux tâches.

Task : meta-informations_vers_aSI

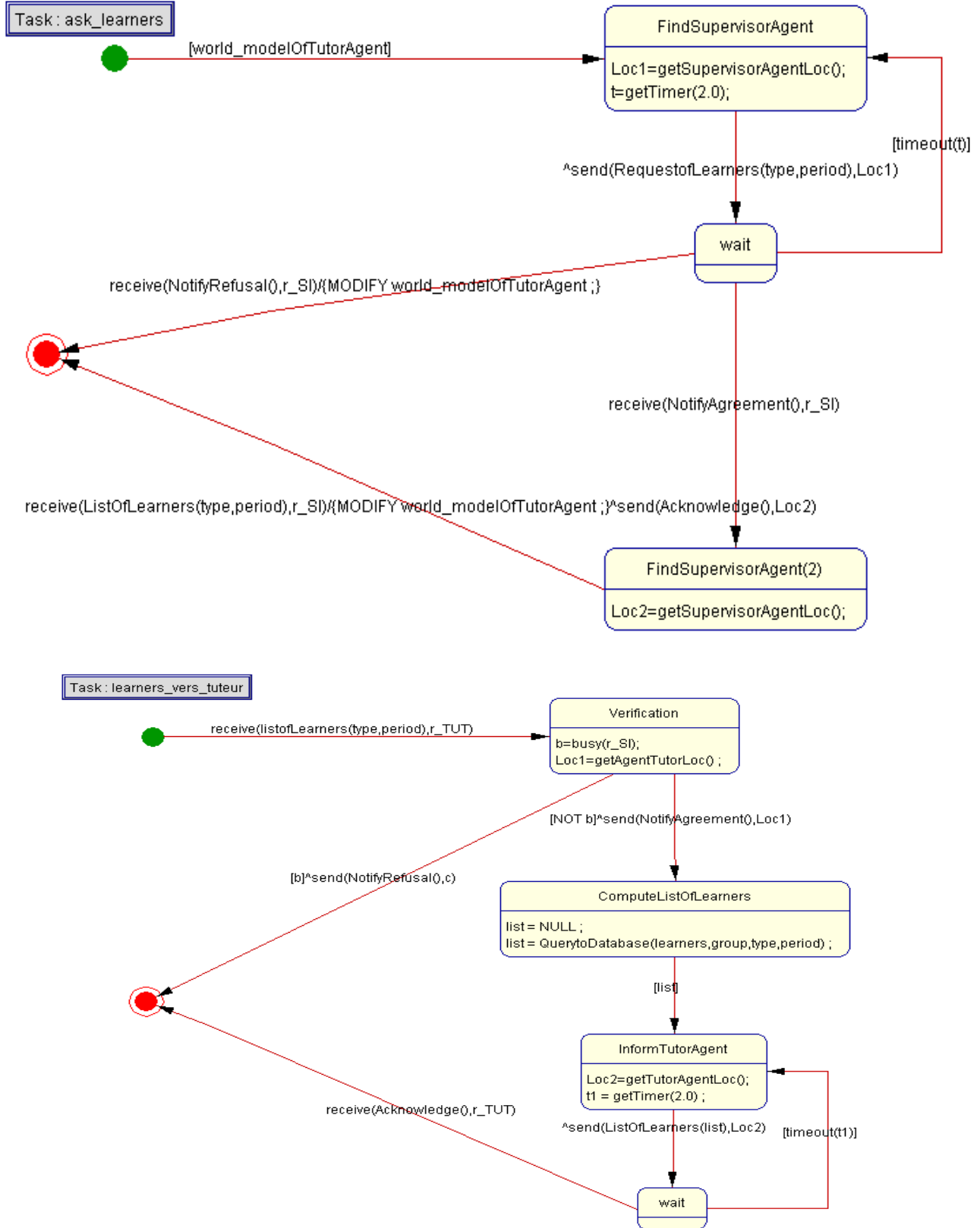


Task : get_metainformations



b – La fourniture de la liste des apprenants actifs, dormants et absents

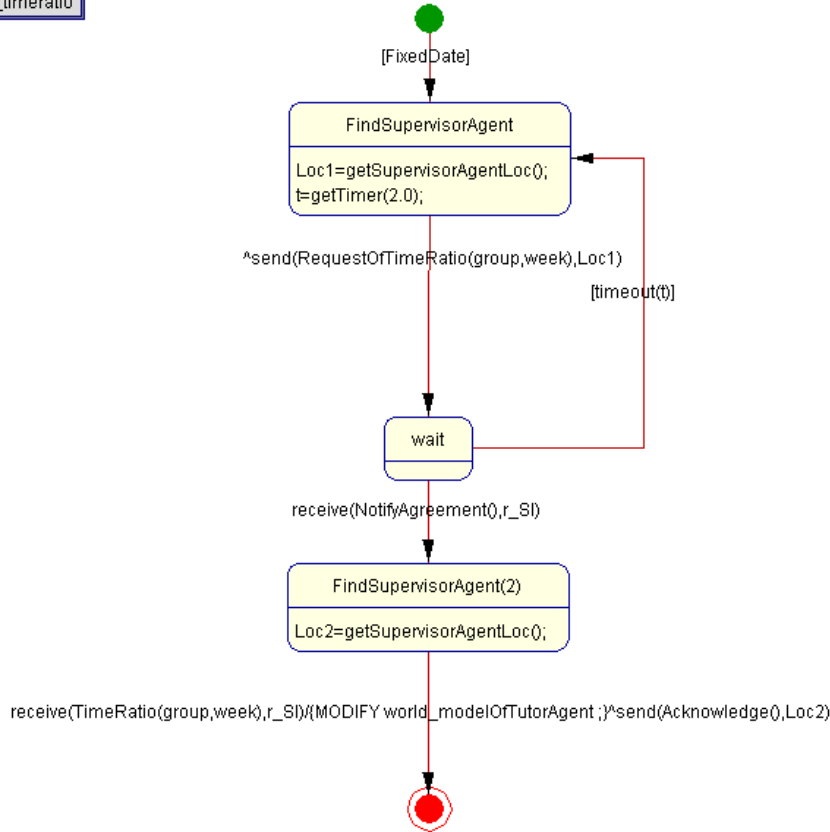
Ce protocole de communication décrit la façon dont l'agent du tuteur (r_TUT) demande et obtient liste des utilisateurs dormants, absents et actifs de la part de l'agent superviseur des interactions (r_SI). Il met en œuvre les rôles r_TUT et r_SI avec leurs tâches concurrentes respectivement, *ask_learners* et *learners_vers_tuteur*. Nous présentons ci-après les deux diagrammes à automates finis décrivant ces deux tâches.



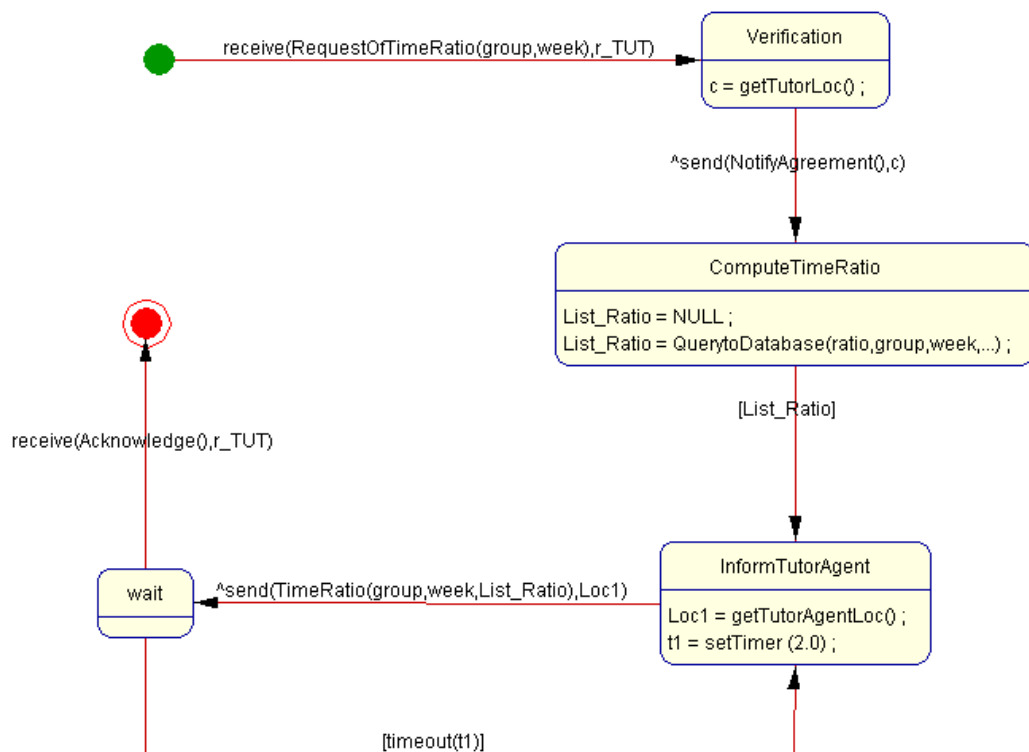
c – La fourniture des ratios de temps du groupe sur la semaine

Ce protocole de communication décrit la façon dont l'agent du tuteur (r_TUT) demande et obtient les ratios de temps passé en ligne par les apprenants de son groupe sur la semaine de la part de l'agent superviseur des interactions (r_SI). Il met en œuvre les rôles r_TUT et r_SI avec leurs tâches concurrentes respectivement, *ask_timeratio* et *ratio_vers_tuteur*. Nous présentons ci-après les deux diagrammes à automates finis décrivant ces deux tâches.

Task : ask_timeratio



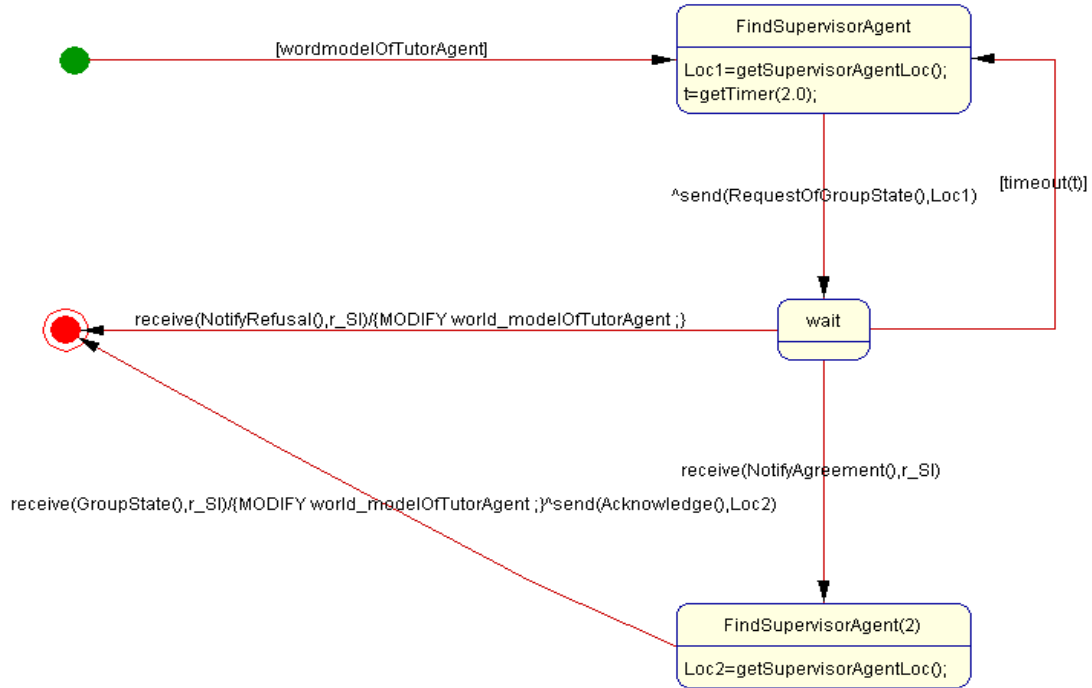
Task : ratio_vers_tuteur



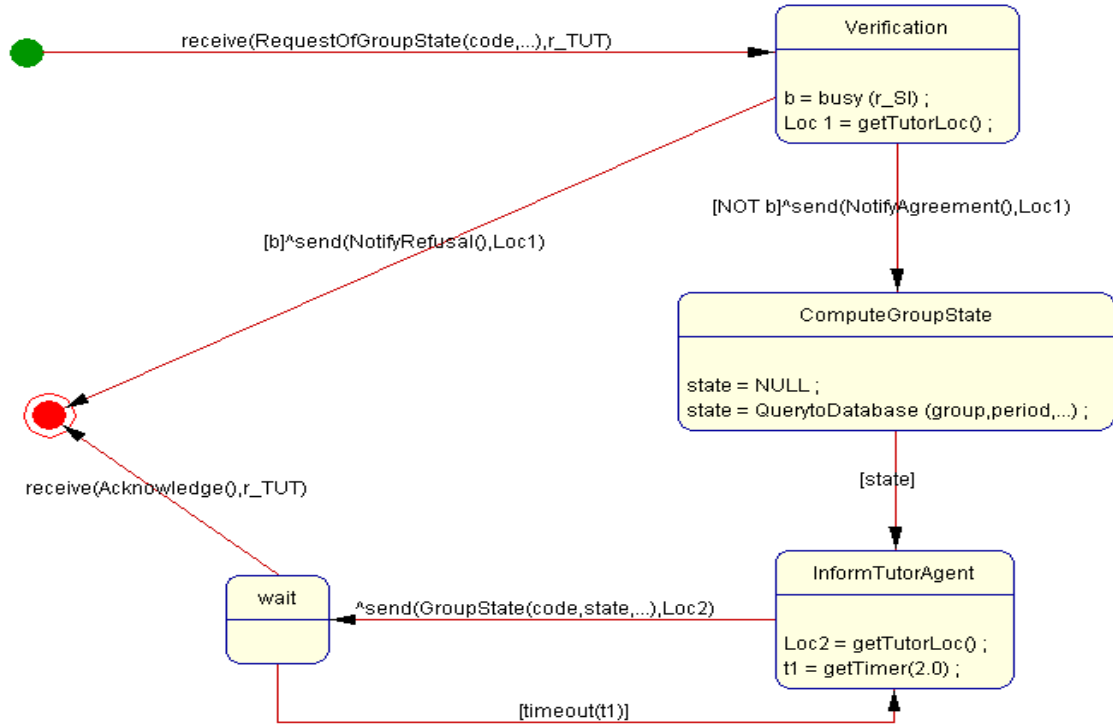
d – La fourniture de l'état de son groupe

Ce protocole de communication décrit la façon dont l'agent du tuteur (r_TUT) demande et obtient l'état de son groupe de la part de l'agent superviseur des interactions (r_SI). Il met en œuvre les rôles r_TUT et r_SI avec leurs tâches concurrentes respectivement, *ask_etatgroupe* et *etatgroupe_vers_tuteur*. Nous présentons ci-après les deux diagrammes à automates finis décrivant ces deux tâches.

Task : ask_etatgroupe

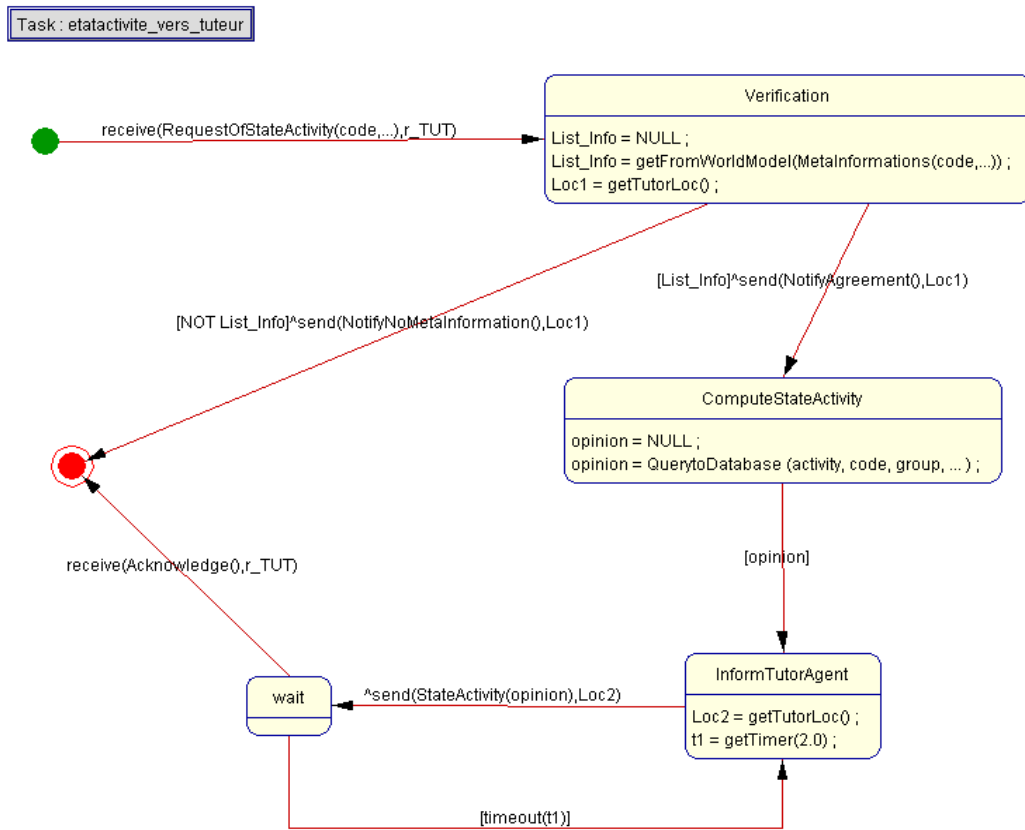
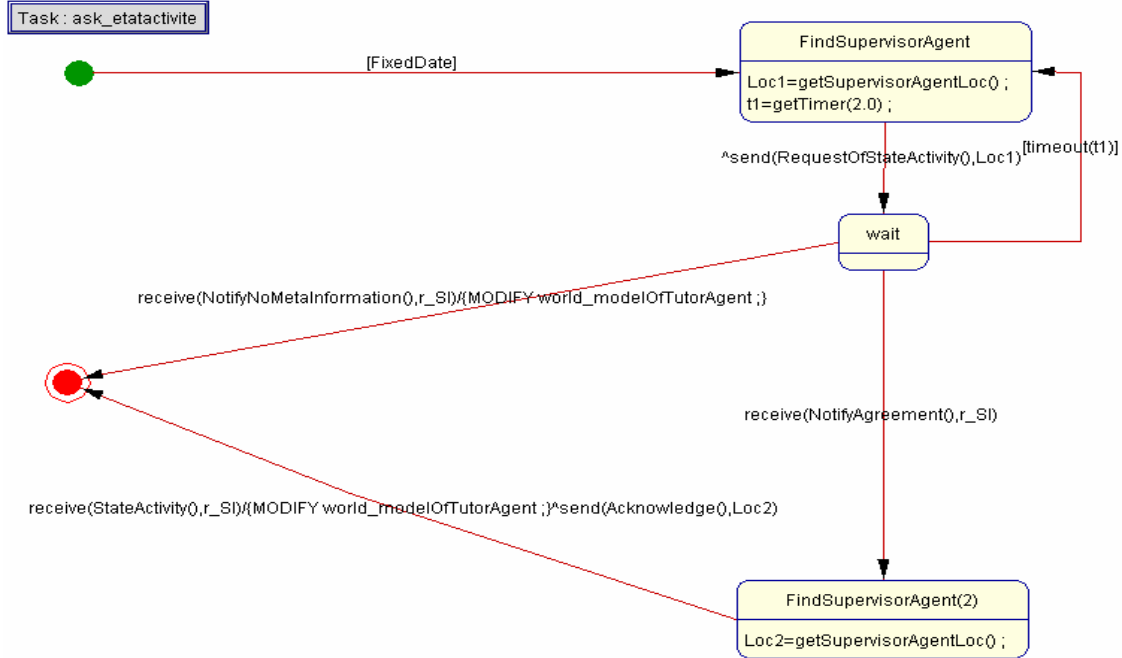


Task : etatgroupe_vers_tuteur



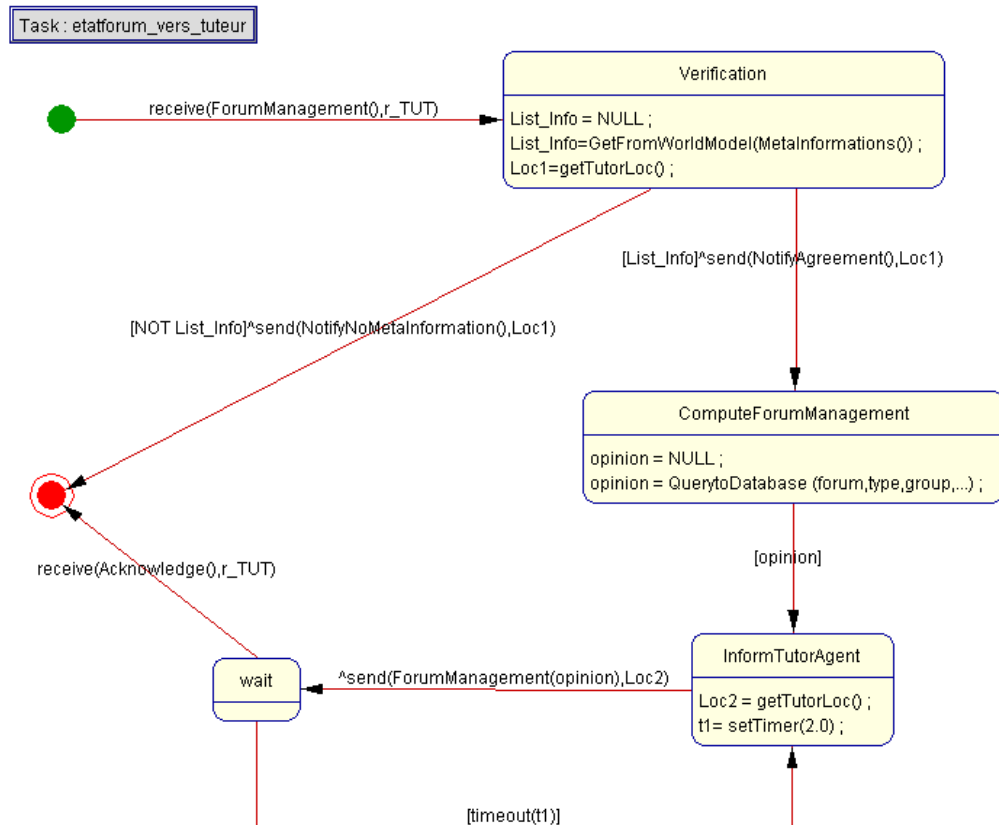
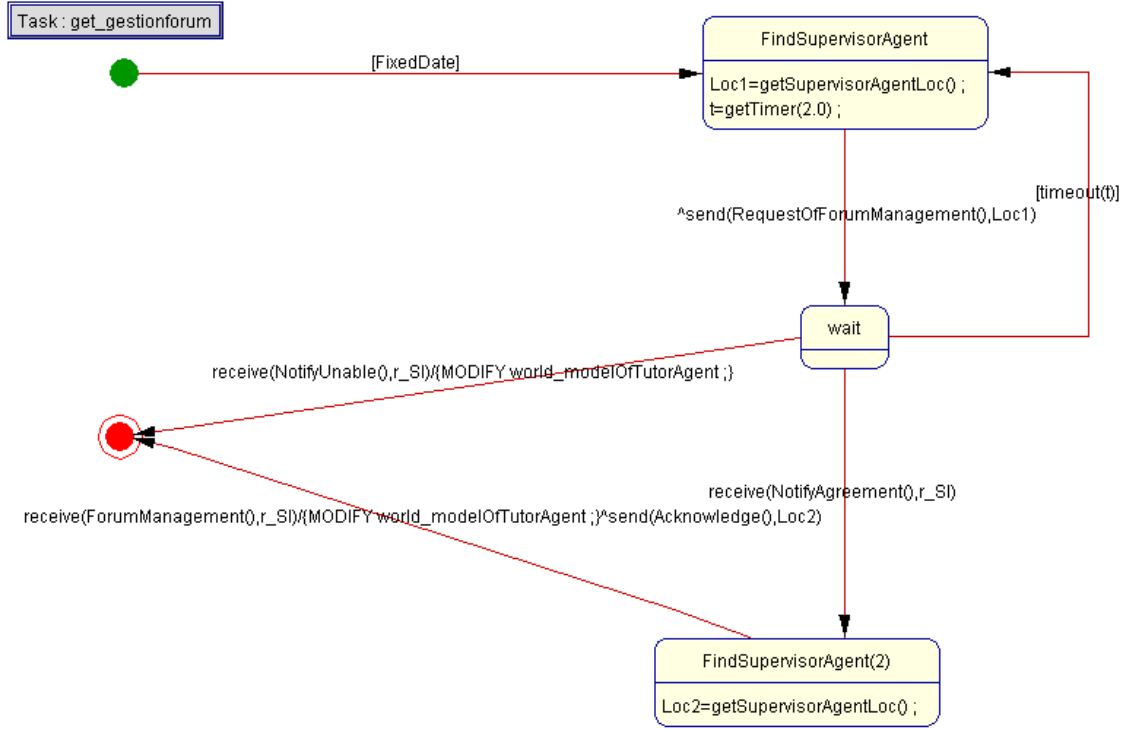
e – La fourniture de l'état de réalisation d'une activité

Ce protocole de communication décrit la façon dont l'agent du tuteur (r_TUT) demande et obtient l'état de réalisation d'une activité par son groupe de la part de l'agent superviseur des interactions (r_SI). Il met en œuvre les rôles r_TUT et r_SI avec leurs tâches concurrentes respectivement, *ask_etatactivite* et *etatactivite_vers_tuteur*. Nous présentons ci-après les deux diagrammes à automates finis décrivant ces deux tâches.



f – La fourniture de l'avis de gestion d'un forum

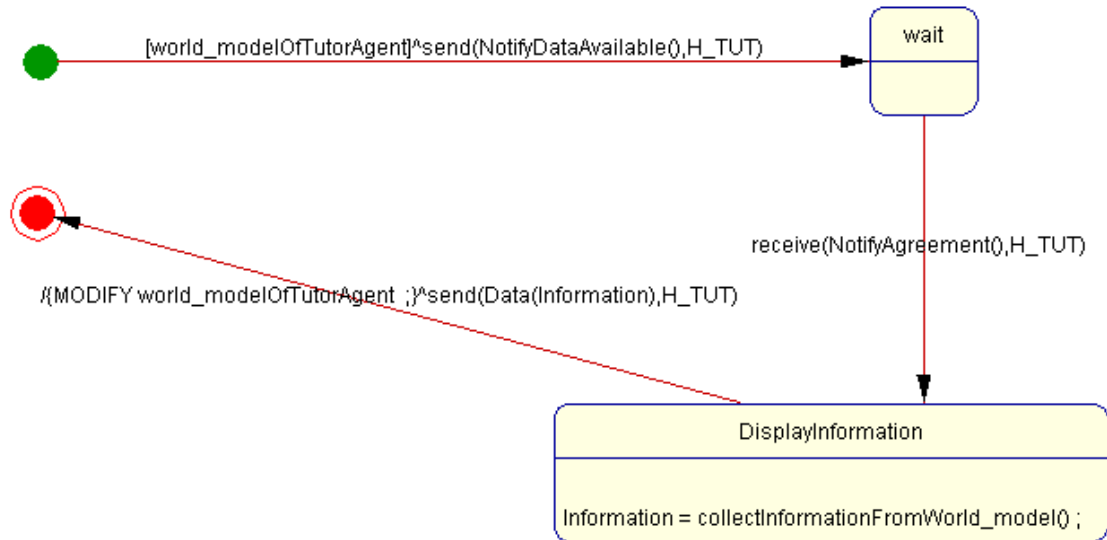
Ce protocole de communication décrit la façon dont l'agent du tuteur (r_TUT) demande et obtient l'avis sur la gestion d'un forum par le tuteur de la part de l'agent superviseur des interactions (r_SI). Il met en œuvre les rôles r_TUT et r_SI avec leurs tâches concurrentes respectivement, *get_gestionforum* et *etatforum_vers_tuteur*. Nous présentons ci-après les deux diagrammes à automates finis décrivant ces deux tâches.



g – L’interface agent - tuteur

Ce protocole de communication décrit la façon dont l’agent du tuteur (r_TUT) communique les informations à son propriétaire, le tuteur (H_TUT). Nous ne décrivons que la tâche concurrente associée côté agent à savoir : *send_informations*.

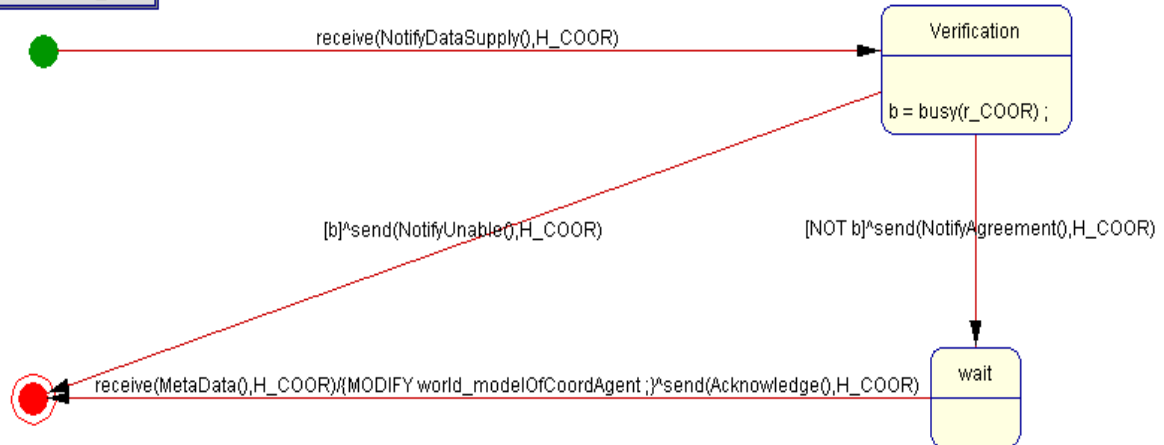
Task : send_informations



h – L’interface agent - coordonnateur

Ce protocole de communication décrit la façon dont l’agent du coordonnateur (r_COOR) reçoit les méta-informations concernant les activités et les forums de la part de son propriétaire, le coordonnateur (H_COOR). Nous ne décrivons que la tâche concurrente associée côté agent à savoir : *receive_infos*.

Task : receive_infos



En plus des concepts de *move activity* (pour indiquer qu’un agent a changé de location ou d’adresse) et de *timeout activity* (pour indiquer que le temps d’attente est épuisé) définis par les auteurs de MaSE (DeLoach et al., 2001), nous définissons un *busy state* pour indiquer qu’un agent (un rôle à ce stade de l’analyse) est très occupé et qu’il ne peut exécuter à une requête reçue. La syntaxe de cet état est :

$Boolean = busy(agent) ;$

Cette expression renvoie la valeur TRUE lorsque l'agent est occupé et ne peut satisfaire une requête et FALSE sinon. Nous illustrons cet état dans la tâche *learners_vers_tuteur*. Des situations où r_{SI} est très sollicité peuvent arriver, dans l'état *Verification* (figure 9), r_{SI} évalue l'expression :

$b = busy(r_{SI}) ;$

Si b contient la valeur TRUE, alors r_{SI} notifie à r_{TUT} son refus d'exécuter la requête par lui émise.

Le concept de *busy state* permet d'implémenter concrètement une des caractéristiques reconnues aux agents, la capacité de refuser d'exécuter une requête, capacité qui l'en distingue d'un objet au sens classique de la programmation orientée-objet où ce dernier ne peut refuser d'exécuter une méthode une fois invoquée.

3-3-4 LE DIAGRAMME DES CLASSES D'AGENTS DE SIGFAD

Durant cette étape, première de la phase de conception, les classes d'agent sont créées à partir des rôles définis à l'issue de la phase d'analyse. Ceci se fait de manière relativement simple, le concepteur crée les classes d'agents de manière à ce que chaque rôle soit rempli par au moins une classe d'agent. Là aussi, alors que la méthodologie laisse la possibilité de faire jouer un rôle par plusieurs agents, AgentTool réduit cette possibilité en obligeant à assigner chaque rôle à une et une seule classe d'agent.

Toutefois, dans la spécification de SIGFAD, nous nous sommes légèrement écartés de l'esprit de MaSE. En fait nous avons transformé directement nos rôles en agents, c'est à dire que nous avons affecté chacun des rôles de l'étape précédente à un agent différent. Ceci signifie finalement que nous avons déjà nos agents en tête au moment de la définition des rôles dans la phase d'analyse. Ceci n'est pas une application exacte de MaSE. La méthodologie recommande en outre de noter dans chaque rectangle représentant une classe d'agent, les numéros des rôles remplis par cette classe. Pour notre part, en raison de l'assignation de chaque rôle à un agent unique, nous avons noté les tâches remplis par la classe d'agent. Cette façon de procéder nous semble plus expressive.

Dans SIGFAD, nous avons huit classes d'agent. a_{COOR} (qui remplit le rôle r_{COOR}) est l'agent du coordonnateur, il est chargé de fournir au coordonnateur les données d'interaction, la liste des tâches à effectuer sur une période donnée. a_{TUT} (qui remplit le rôle r_{TUT}) est l'agent du tuteur (un agent pour chaque tuteur) qui remplit des tâches similaires à celles de a_{COOR} vis-à-vis de son propriétaire. a_{LEAR} (qui remplit le rôle r_{LEAR}) est l'agent de l'apprenant (un agent par apprenant) assiste son propriétaire en lui fournissant les données concernant ses interactions et la liste des tâches à effectuer sur une période donnée. a_{SI} (qui remplit le rôle r_{SI}) est chargé de fournir les données d'interaction à tous les autres agents, il assure l'interfaçage de la base de données avec tout le reste de SIGFAD. *CoordInterface*, *TutInterface*, *LearnerInterface* et *DBInterface* encapsulent les interfaces du système avec respectivement : le coordonnateur, les tuteurs, les apprenants et la base de données. Le diagramme de classes d'agent de SIGFAD est représenté sur la figure 10.

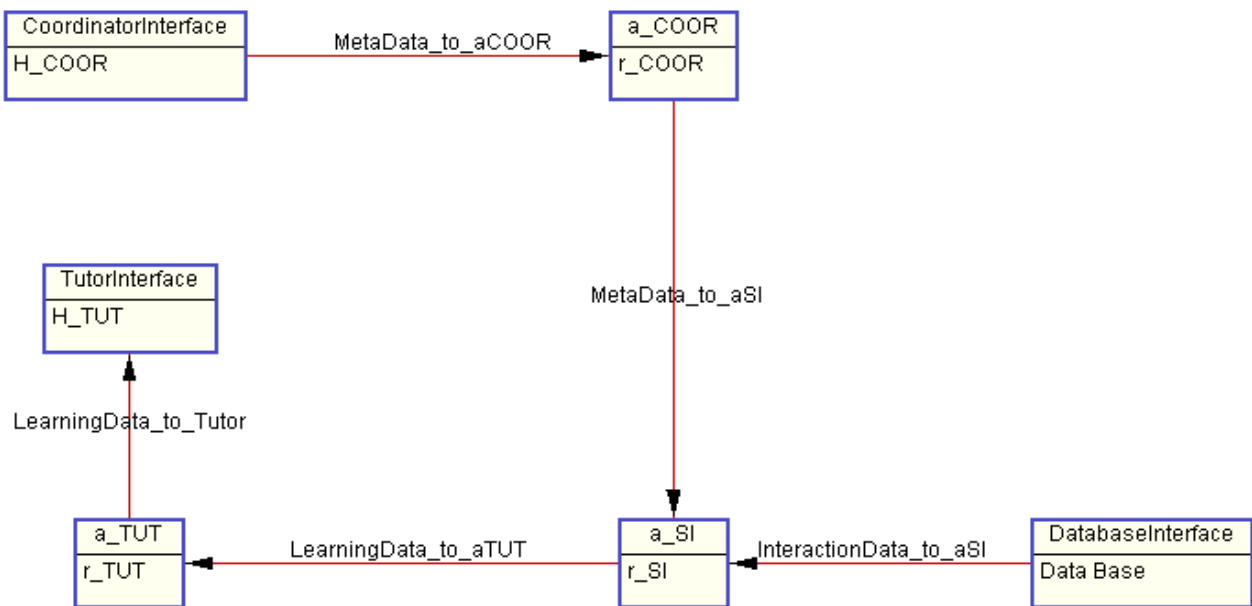


Fig.17. Le diagramme de Classes d'agent de SIGFAD

3-3-5 LA CONSTRUCTION DES COMMUNICATIONS INTER-AGENTS DE SIGFAD

Cette étape a pour but de construire le détail des interactions entre agents. Ces interactions sont spécifiées grâce à des diagrammes d'automates à états finis selon le même principe que la spécification des tâches concurrentes de la troisième étape. Pour chaque protocole de communication entre agents, le concepteur doit définir deux diagrammes de communication de classes : l'un pour l'initiateur et l'autre pour le destinataire.

Le fait que nous ayons assignés directement chaque rôle à un agent différent nous dispense de spécifier explicitement les communications inter-agents. En fait, les communications inter-agents correspondent exactement aux communications entre rôles définies plus haut. Nous rappellerons que ceci est dû à la façon particulière dont nous avons appliqué la méthodologie MaSE. Dans un cas plus général, le concepteur aurait certainement à construire les diagrammes de communication des classes d'agent tel que demandé par la méthodologie.

3-3-6 LA CONSTRUCTION DES COMMUNICATIONS INTER-AGENTS DE SIGFAD

Cette étape dont le but est de construire l'architecture interne des agents nous semble cruciale. Jusqu'à présent, nous nous sommes focalisés sur les interactions et les communications entre les agents de notre système multi-agents. Rien n'a été dit sur les comportement intelligent de nos agents : comment les agents représentent-ils leurs connaissances ? Comment sont implémentées les caractéristiques de flexibilité, d'autonomie, de rationalité, de dynamisme ?

Notre système multi-agents pour bien atteindre les fonctionnalités attendues des plates-formes dédiées à la FAD doit contenir des agents capables d'assister les utilisateurs dans leurs activités. Pour cela, ces agents doivent être capables de représenter leur connaissance de l'environnement, les données, leurs tâches, le comportement des autres agents et leur propre comportement, autant de choses qui caractérisent réellement les systèmes d'Intelligence Artificielle Distribuée (IAD). Plusieurs architectures dédiées à la réalisation de tels systèmes existent. Nous pouvons citer ici entre autres (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 comme les croyances, les désirs et les 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.

Pour notre part, nous avons choisi de construire l'intérieur de nos agents en utilisant une architecture BDI. Plusieurs raisons ont présidé à ce choix. Le modèle BDI se comprend assez facilement si l'on se réfère au fait que chacun a une compréhension même intuitive des notions de croyance, désir et intention. Ce modèle offre en outre une décomposition fonctionnelle qui indique quels sont les briques nécessaires pour la construction d'un agent. Le modèle BDI bénéficie d'un large effort de normalisation de la part de la communauté en intelligence artificielle. Nos agents seront construits suivant l'architecture JAM.

3-3-7 LA CONCEPTION DU SYSTEME

La dernière étape de MaSE consiste à instancier les classes d'agent construites en définissant les agents du système. Le résultat en est un diagramme de déploiement qui montre le nombre, le type et la localisation des agents du système. Le lecteur voudra bien se référer à la figure 4 qui fournit toutes ces informations.

3-4 CONCLUSION

Au stade actuel de la construction de SIGFAD, il existe encore des fonctionnalités à ajouter à SIGFAD afin qu'il puisse remplir toutes les exigences de la FAD telles que nous les avons identifiées durant Simuligne. En particulier, la nécessité pour le système d'analyser de façon automatique les données d'interaction, d'informer les personnes appropriées lorsque des risques d'échec de la formation apparaissent et de déclencher tous les mécanismes de remédiation n'a pas encore été réellement implémentée. SIGFAD doit aussi pouvoir en se basant par sur l'analyse par les réseaux sociaux calculer des paramètres d'indication de la cohésion du groupe ou de la centralité d'un membre du groupe, nous pensons au tuteur en particulier.

Ces fonctionnalités n'ont pas été implémentées pour plusieurs raisons. La première en est que nous devons poursuivre l'analyse des données d'interaction pour établir de façon plus formelle que la cohésion du groupe influence la progression de la session ou la réalisation des activités. De même devons-nous établir l'impact de la centralité du tuteur sur la réussite ou non du groupe. De manière plus générale, nous pensons que l'implémentation des fonctionnalités évoquées plus haut passe par l'élaboration de modèles mathématiques permettant d'apprécier le niveau d'engagement et de participation des utilisateurs dans leurs activités d'apprentissage en FAD. L'élaboration de ces modèles mathématiques est un de nos objectifs de recherche. Comme nous l'avons dit, l'analyse par les réseaux sociaux nous semble constituer une bonne piste en ce qu'elle permet de construire des variables telles que la cohésion d'un groupe ou la centralité du tuteur.

Les travaux de l'équipe de Besançon sont déjà assez avancés sur ce point avec notamment la construction d'outils d'analyse et de représentation des groupes d'apprentissage se basant sur l'analyse des réseaux sociaux (Jouchet, 2002) et (Reffay et Chanier, 2002).

CHAPITRE IV : LA PROGRAMMATION DE SIGFAD

4-1 L'ARCHITECTURE JAM

JAM est une architecture hybride qui se fonde sur les idées et les théories du système de raisonnement procédural, de la sémantique des circuits structurés et du Act plan interlingua. JAM offre des représentations procédurales riches et étendues, un raisonnement au niveau méta et rationnel prenant en compte plusieurs objectifs en même temps, un comportement basé sur les objectifs poursuivis et la réaction aux événements. L'architecture JAM permet aussi d'implémenter des capacités de mobilité.

L'architecture JAM est construite autour d'un noyau BDI et se base sur les implémentations PRS de l'Université de Michigan et de l'entreprise SRI International connues respectivement sous les sigles UMPRS et PRS-CL.

Chaque agent JAM est composé de cinq composants primaires représentés à la figure ci-après : un modèle du monde, une bibliothèque de plans, un interpréteur, une structure intentionnelle, un observateur. Le modèle du monde est une base de données représentant les croyances de l'agent. La bibliothèque de plans est un ensemble de plans que l'agent utilise pour atteindre ses objectifs. La structure intentionnelle est un modèle interne des objectifs actuels de l'agent ; elle garde les traces de l'engagement et de la progression dans la réalisation des objectifs. L'observateur est une procédure déclarative spécifiée par l'utilisateur qui permet à l'agent de relier les étapes d'un plan afin de remplir des fonctionnalités simples et périodiques (ex. transférer les messages entrants au modèle du monde). L'interpréteur est le cerveau de l'agent, il permet à l'agent de raisonner sur ce qu'il doit faire, quand et comment.

L'agent JAM fonctionne en utilisant les sémantiques d'exécution et de comportement, c'est une combinaison de UMPRS et de SCS : les changements dans le modèle du monde et l'apparition de nouveaux objectifs déclenchent un processus de raisonnement afin de rechercher les plans à appliquer à la nouvelle situation ; l'interpréteur sélectionne un plan de la liste des plans applicables en faisant un raisonnement de niveau méta ou d'utilité maximale.

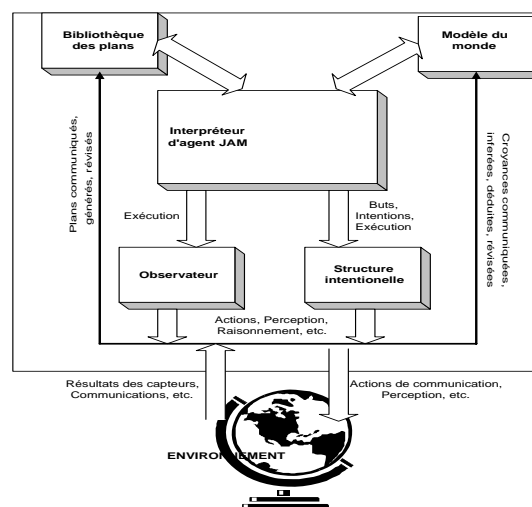


Fig. 18. L'architecture JAM

4-1-1 L'INTERPRÉTEUR

L'interpréteur est responsable de la sélection et de l'exécution des plans sur la base des intentions, plans, objectifs et croyances de l'agent. L'interpréteur est associé à la structure intentionnelle, c'est une pile d'exécution des objectifs associés ou non à des plans instanciés. L'agent contrôle les plans applicables à un objectif. Une variable d'utilité est associée à chaque plan. S'il n'est pas possible de sélectionner un plan unique à partir du critère d'utilité, l'interpréteur met en œuvre un raisonnement de niveau méta pour sélectionner le plan à exécuter. L'interpréteur est une procédure écrite en Java, comprenant une boucle qui s'exécute jusqu'à ce que l'agent réalise tous ses objectifs.

4-1-2 LES BUTS

Le comportement d'un agent JAM est motivé par les buts. On a trois types de buts, chacun ayant une sémantique différente : ACHIEVE (*réaliser*), PERFORM (*se comporter*), MAINTAIN (*maintenir*). Le but ACHIEVE spécifie que l'agent désire atteindre un état, il s'agit typiquement du but associé aux architectures BDI et aux systèmes générant des plans. Le but PERFORM spécifie que le comportement que l'agent désire avoir ; il ne s'agit pas de réaliser un but mais d'exhiber un comportement particulier. Le but MAINTAIN indique qu'un but particulier doit être réalisé de nouveau à chaque fois qu'il viendrait à ne plus l'être, c'est un but homéostatique qui ne doit pas être retiré de la liste de buts à moins que ce soit du fait exprès de l'agent.

Les buts sont assignés aux agents en précisant, le type du but, son nom, des paramètres, un critère d'utilité et une expression (optionnels). L'utilité du but est associée à l'utilité du plan pour déterminer l'utilité totale de l'intention. L'agent essaye toujours de réaliser le but pour lequel le critère d'utilité est le plus élevé, ceci permet d'obtenir des agents rationnels. Un ou plusieurs buts initiaux sont assignés à l'agent au moment de son invocation. La liste des objectifs peut être augmentée durant son exécution à travers les interactions avec d'autres agents, ou des raisonnements internes de l'agent ou encore par d'autres moyens. Les buts initiaux de l'agent sont dits persistants en ce sens qu'ils sont poursuivis jusqu'à ce qu'ils soient réalisés grâce à des plans bien appliqués ou par d'autres agents ou jusqu'à ce qu'ils soient retirés du plan par exemple quand l'agent estime qu'ils ne peuvent plus être atteints. Si le plan associé à la réalisation des buts initiaux échoue, l'agent renonce à son engagement à réaliser ce but en retirant son intention ; le but reste toutefois dans la structure intentionnelle pour des tentatives ultérieures de réalisation de l'objectif. Par contre si un plan associé à un sous-objectif échoue, l'interpréteur considère que l'action associée à ce sous-but a échoué (en supposant qu'il existe un autre plan pour réaliser ce sous-but).

4-1-3 LES PLANS

Un plan JAM spécifie une procédure (écrite en Java) pour réaliser un but, réagir à un événement ou avoir un comportement donné. Les agents JAM peuvent dès lors être guidés par des buts (rationnels) ou simplement réagir à des événements (réactifs). Un ou plusieurs plans sont affectés à l'agent au moment de son invocation ; des plans peuvent être ajoutés durant l'exécution soit à travers les interactions avec les autres agents, soit générés à partir d'un processus de raisonnement interne de l'agent, soit par d'autres moyens. L'applicabilité d'un plan est limitée soit à un objectif, soit à la réaction à une combinaison donnée d'événements. Chaque plan est contraint par des pré-conditions, conditions devant apparaître avant le début de son exécution ; et un contexte, conditions que l'on doit avoir avant et pendant l'exécution du plan. Chaque plan doit comporter une procédure de calcul des critères d'utilité implicite et explicite afin de permettre la sélection d'un plan à partir d'un ensemble donné de plans. JAM

offre plusieurs structures itératives comme DO...WHILE, WHILE, et des structures de branchement conditionnel comme OR, AND, DO_ALL, WHEN, des structures d'initialisation comme ASSIGN. Les croyances d'un agent JAM peuvent être modifiées ou contrôlées en utilisant les primitives ASSERT, FACT, RETRACT, RETRIEVE, UPDATE. JAM permet des exécutions parallèles avec PARALLEL et une primitive de synchronisation : WAIT.

Le concepteur peut augmenter les fonctionnalités offertes par JAM en écrivant des fonctions en Java, plusieurs méthodes d'accès au code Java sont offertes par JAM. C'est à travers ces mécanismes que JAM permet de construire des applications spécifiques (par exemple l'interfaçage avec des bases de données) et d'implémenter des caractéristiques de sociabilité (comme la communication ou la collaboration entre agents).

4-1-4 LE MODELE DU MONDE

Le modèle du monde de JAM contient les données représentant l'état courant du monde tel que perçu par l'agent. Les informations contenues dans cette base de données concernent : les variables d'état, les informations sensorielles, les conclusions tirées des déductions et des inférences, la modélisation des autres agents, etc. Une grande flexibilité est laissée au programmeur pour définir l'ordre, le type et la sémantique des arguments utilisés.

4-1-5 L'OBSERVATEUR

L'observateur est une procédure déclarative optionnelle que l'interpréteur JAM exécute après chaque action contenue dans le plan. L'observateur est une architecture-crochet que le programmeur peut utiliser pour implémenter plus facilement des caractéristiques qui ne sont pas prises en compte dans les plans et les buts normaux de JAM. Le programmeur peut utiliser l'observateur pour vérifier par exemple le tampon des messages entrants. Cette procédure est utilisée pour observer les événements asynchrones.

4-2 JATLITE ET KQML

Après la construction de l'architecture interne des agents, l'une des préoccupations les plus cruciales dans la construction d'un SMA est la communication entre les agents. Cette communication est nécessaire aussi bien pour échanger les données entre les agents que les connaissances. Pour cela, plusieurs moyens existent : au niveau le plus bas, il existe des sockets qui permettent aux différents agents codés en Java de communiquer entre eux ; il existe aussi d'autres technologies (en Java notamment). On peut citer l'invocation de méthodes distantes, RMI (Remote Method Invocation) et la technologie CORBA (Common Request Broker Architecture).

4-2-1 SOCKETS, RMI ET CORBA

Une socket est une abstraction de programmation représentant les extrémités d'une connexion entre deux ordinateurs. Pour chaque connexion donnée, il existe une socket sur chaque machine, on peut imaginer un câble virtuel reliant les deux ordinateurs, chaque extrémité étant enfichée dans une socket. En Java, on crée un objet Socket pour établir une connexion vers une autre machine, puis on crée un InputStream et un OutputStream à partir de ce Socket, afin de traiter la connexion en tant qu'objet flux d'entrée/sortie.

L'invocation de méthodes distantes se base sur le raisonnement selon lequel la meilleure manière d'exécuter des instructions à travers un réseau sur d'autres ordinateurs est de

considérer qu'un objet est présent sur une autre machine et qu'on lui envoie un message et l'on obtient le résultat comme si cet objet était instancié sur la machine locale. L'invocation de méthodes distantes exige de créer une interface distante (remote interface) sur la machine serveur ; ainsi, l'implémentation sous-jacente est masquée aux différents clients. Les étapes suivantes de la création des objets RMI sont l'implémentation de l'interface distante, la mise en place du registre, la création des stubs et des skeletons qui assurent les opérations de connexion réseaux et permettent de donner l'illusion que l'objet distant est situé sur la machine locale. L'utilisation de l'objet distant consiste simplement pour le programme client à rechercher et à rapatrier depuis le serveur, l'interface distante.

Lorsque les applications distribuées deviennent plus importantes et exigent de fonctionnalités telles que l'intégration de bases de données existantes, l'accès aux services d'un objet serveur sans se soucier de sa location physique, l'on a besoin d'effectuer des appels à des procédures distantes ou RPC (*Remote Procedure Call*) et il peut être nécessaire d'avoir une indépendance par rapport au langage. Dans ces cas, il est nécessaire de recourir à une technologie d'intégration comme CORBA. Il s'agit d'une spécification que les fabricants peuvent suivre afin d'implémenter des produits supportant une intégration CORBA. La norme CORBA fait partie du travail réalisé par l'OMG (Object Management Group) pour définir des standards d'interopérabilité des objets distribués et indépendants du langage. Elle fournit la possibilité de faire des appels à des procédures distantes dans des objets Java et non-Java, d'interfacer des systèmes existants sans se soucier de leur emplacement.

La spécification de l'interopérabilité objet est généralement désignée comme l'Object Manager Architecture (OMA). L'OMA définit deux composants : le Core Object Model et l'OMA Reference Architecture. CORBA est un raffinement du Core Object Model dans ce sens qu'il met en place les concepts de base d'objet, d'interface, d'opération, etc. L'OMA Reference Architecture définit les services et les mécanismes qui permettent aux objets d'inter-opérer, il contient l'Object Request Broker (ORB), les services CORBA et les outils communs. L'ORB est le bus de communication par lequel les objets peuvent réclamer des services auprès des autres objets, sans rien connaître de leur localisation physique. La norme CORBA n'a pas pour but d'expliquer comment implémenter les ORBs, mais permet une compatibilité entre les différents ORBs des fournisseurs. Ainsi, l'OMG définit un ensemble de services qui sont accessibles par l'intermédiaire d'interfaces standards. Afin d'assurer l'indépendance vis-à-vis du langage, CORBA dispose d'un langage de définition d'interface, IDL (Interface Definition Language), qui précise les types de données, les attributs, les opérations, les interfaces, etc. Le compilateur IDL génère le code stub et le code skeleton utilisé pour réunir les arguments des méthodes et réaliser les appels distants.

En résumé, on pourrait dire que CORBA est le support de RPC, qui permet à des objets locaux d'appeler des méthodes d'objets distants. Cette fonctionnalité est native à Java à travers les RMI ; toutefois, là où RMI rend possible les RPC entre des objets Java, CORBA le fait entre des objets implémentés dans n'importe quel langage.

4-2-2 JATLITE

Bien que les objets Java RMI et la technologie CORBA puissent être utilisés pour permettre à nos agents de communiquer entre eux, il nous a semblé préférable de recourir à JATLite (Java Agent Template, Lite). Il s'agit d'un paquetage de programmes écrits en Java qui permettent de créer rapidement des agents logiciels qui communiquent à travers Internet. JATLite fournit une infrastructure de base qui permet aux agents de s'inscrire auprès d'un routeur de messages, AMR (Agent Message Router) en utilisant un nom et un mot de passe, de se

connecter et se déconnecter, d'envoyer et de recevoir des messages, de transférer des fichiers par FTP et en général, d'échanger des informations avec d'autres agents localisés sur différents ordinateurs. Avant de présenter plus en détail JATLite, soulignons que l'utilisation de CORBA ou de RMI en lieu et place de JATLite présenterait deux inconvénients majeurs : le premier est que tous les modules devraient envoyer des objets CORBA et le second serait la nécessité de devoir écrire son propre routeur de messages (AMR). L'autre grand avantage de JATLite est qu'il supporte le langage KQML qui permet aux agents de s'échanger des connaissances et qu'il supporte par ailleurs d'autres langages de requêtes comme SQL et FIPA ACL.

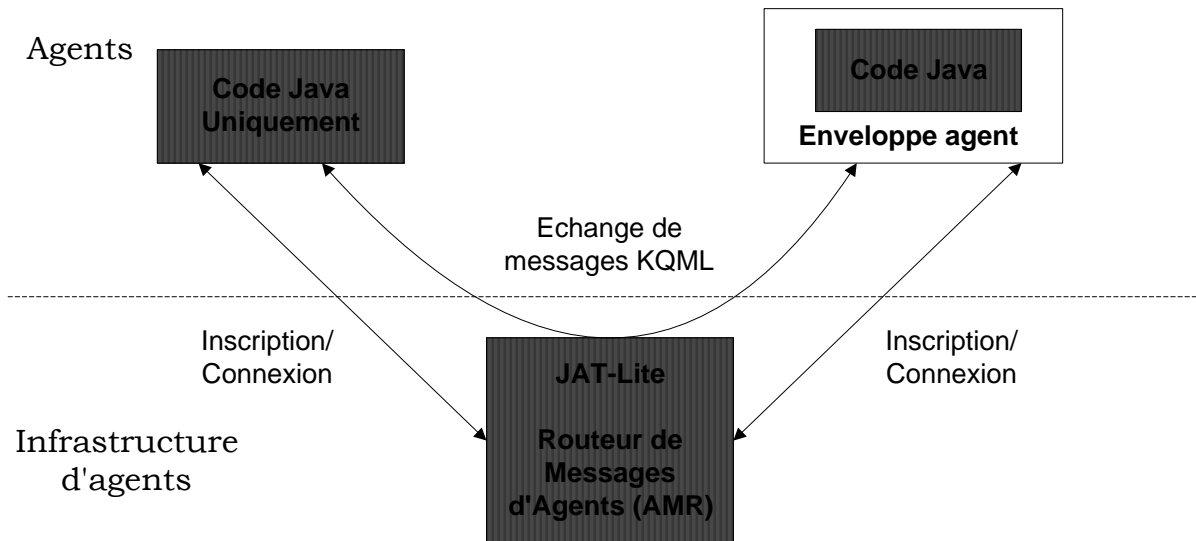


Fig. 19. Le Routeur de messages de JATLite

JATLite a été développé à partir de 1996 à l'Université de Stanford aux Etats-Unis par Jeon Heecheol et quelques autres étudiants sous la supervision du Professeur Mark Cutkosky et du Docteur Charles Petrie. JATLite est basé sur les considérations générales suivantes :

- les communications sont basées sur le protocole TCP/IP ;
- les agents communiquent à travers l'envoi de messages ;
- une connexion unique pour chaque agent connecté au routeur de messages ;
- une adresse unique (nom, hôte, port, méthode d'envoi de message, description) est assignée à chaque agent.

L'architecture JATLite est organisée en un certain nombre de couches hiérarchisées et spécialisées qui constituent des interfaces de programmes d'application. Ceci permet au programmeur, en fonction des protocoles de communication qu'il souhaite utiliser, de se positionner sur la couche adéquate.

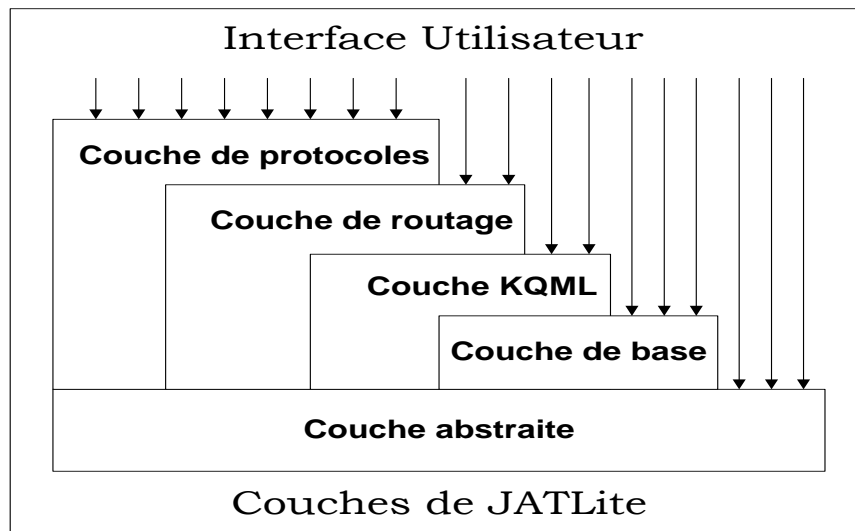


Fig. 20. L'architecture de JATLite est basée sur un ensemble de couches spécialisées et hiérarchiques

- La *couche abstraite* fournit un ensemble de classes abstraites nécessaires à l'implémentation de JATLite. Bien que toutes les communications se basent sur le protocole TCP/IP, il est laissé la possibilité au programmeur d'implémenter d'autres protocoles comme UDP.
- La *couche de base* fournit des communications TCP/IP et donne accès à la couche abstraite. Cette couche peut être étendue pour permettre de recevoir des informations à partir des sockets et rediriger des informations de sortie vers des fichiers. Elle peut aussi permettre de fournir plusieurs ports de communication aux agents.
- La *couche KQML* fournit le stockage et l'analyse des messages KQML. Des extensions au KQML standard proposées par l'Université de Stanford (<http://cdr.stanford.edu/ProcessLink/kqml-proposed.html>) ont été implémentées afin de permettre un protocole standard pour l'inscription, la connexion, la déconnexion, etc.
- La *couche de routage* offre aux agents l'inscription, le routage et la mise en file des messages. Lorsqu'un agent est déconnecté, le routeur stocke tous ses messages jusqu'à ce qu'il se connecte de nouveau.
- La *couche de protocoles* supporte différents protocoles et services Internet comme SMTP, FTP, POP3, HTTP. La version actuelle de JATLite supporte les protocoles SMTP et FTP et peut être facilement étendue pour supporter les autres protocoles.

4-3 LES AGENTS DE SIGFAD

De toutes les parties de l'architecture d'un agent JAM, le programmeur ne se préoccupe de l'écriture des fichiers textes contenant les buts, les faits représentant l'état du monde, les plans et ses différentes parties. Le programmeur n'a pas à se soucier de l'interpréteur et de la structure intentionnelle, ces parties ont déjà été implémentées dans JAM. L'ensemble des plans définis par le programmeur constitue la bibliothèque de plans. L'état du monde, les buts et les critères d'utilité associés peuvent être définies dans un ou plusieurs fichiers texte. A l'invocation du ou des fichiers définissant un agent, l'interpréteur JAM se charge d'analyser les différents fichiers et construit l'agent spécifié. Il existe plusieurs modes d'exécution des agents JAM. Celui que nous avons utilisé jusqu'à présent dans la simulation des agents de

SIGFAD est le mode commande en ligne. Ce mode nous permet de simuler les différentes fonctionnalités de SIGFAD mais demande à évoluer très rapidement vers une interface plus proche de celle qui pourrait être utilisée en situation réelle par des utilisateurs engagés dans une session de FAD.

Les résultats obtenus par les différentes simulations sont très concluants. Nous avons ainsi installé la base de données des interactions sur un micro-ordinateur qui simule un serveur. Des agents tuteurs adresser des requêtes à l'agent Superviseur des Interactions qui, en accédant à la base de données, fournit des réponses demandées. L'on peut ainsi :

- obtenir la liste des utilisateurs présents/absents/dormants pendant une période donnée dans un groupe donnée ;
- obtenir l'état du groupe sur une période donnée.

Nous présentons les codes source de nos différents agents en annexe. Les agents JAM utilisent des fonctions écrites en Java par le programmeur en plus de celles fournies par la distribution originelle de JAM, nous en avons écrit un certain nombre dont quelques-unes ont leur code source présenté en annexe.

4-4 CONCLUSION

La programmation de SIGFAD est en cours. Notre objectif à ce stade était de simuler les fonctionnalités de notre système multi-agents. Cette simulation donne des résultats très prometteurs et nous a déjà permis de réfléchir aux différents critères de l'évaluation de l'activité des utilisateurs dans une session de FAD, notamment en ce qui concerne le qualificatif de 'dormant', l'état du groupe avec la notion d'apprenant ou d'utilisateur 'actif'.

Nous devons à présent intégrer dans notre programmation les éléments intrinsèques de la programmation orientée-agent en usant de différents plans alternatifs pour atteindre des objectifs en fonction du contexte de départ. Il est aussi urgent que nous puissions mettre au point une interface qui permette d'exécuter les agents comme des applets Java, et qui permette le passage des arguments à nos agents. Cette interface servira d'interface pour les chercheurs désireux de tester les fonctionnalités de SIGFAD.

Au stade actuel de notre programmation, nous avons utilisé les sockets pour assurer la communication entre les agents, avec l'outil JATLite, nous devrions nous situer à un niveau beaucoup plus élevé et permettre à nos agents d'échanger véritablement des connaissances en utilisant pleinement le langage KQML ou FIPA ACL.

CONCLUSION GENERALE

Le projet ICOGAD nous a permis de mener une expérimentation de FAD en grandeur réelle, expérimentation comprenant de vrais apprenants, travaillant pour atteindre des objectifs pédagogiques sur une période significative de dix semaines. Simuligne nous a permis de recueillir un corpus très important de données d'interaction et de production. L'analyse des données ainsi recueillies et le vécu de la formation Simuligne nous auront permis d'appréhender un certain nombre de phénomènes en formation à distance et en apprentissage collaboratif, on peut citer ici :

- le rôle crucial du tuteur dans la gestion des groupes et des activités ;
- le rôle du groupe comme cadre d'expression et d'accompagnement de l'apprenant ;
- la dimension cognitive de la résolution de la tâche en apprentissage collaboratif ;
- le modèle conceptuel d'une activité en apprentissage collaboratif sur réseau ;
- les limites des environnements de campus virtuel dans la médiatisation de la FAD et les caractéristiques dont il faudrait doter ce type de plates-formes pour leur permettre de devenir de véritables plates-formes informatiques de formation à distance ;
- les paradigmes informatiques et les outils de développement à mettre en œuvre pour répondre aux exigences de la FAD.

D'autre part, le projet ICOGAD en permettant de travailler dans une approche pluridisciplinaire nous a permis d'appréhender les apports des psychologues et de comprendre la nécessité d'exhiber le profil cognitif du tuteur et de l'apprenant dans la résolution de ses tâches en FAD.

L'examen des consignes et recommandations fournies aux tuteurs, telle qu'effectuée par Brixhe (2002) indique que les concepteurs de Simuligne ne se sont pas attardés sur les mécanismes qui auraient permis au tuteur d'évaluer l'état du groupe, d'apprécier le niveau de réalisation d'une activité donnée, d'indiquer et de savoir si le groupe ou une partie se retrouvent débordés et en marge de la formation. On note que des indications et recommandations claires concernent l'animation du groupe en supposant que ce dernier existe mais rien n'est dit pour savoir si le groupe existe toujours ou si des individus, membres du groupe sont en train de décrocher. C'est ainsi que la question suivante est abordée dans les consignes fournies aux tuteurs (voir consignes, p.) : *Que faire si les individus n'interagissent pas comme un groupe ?* alors que de notre point de vue, il existe une question plus lancinante encore : *Que faire si les individus n'interagissent pas ?* et comporte une question sous-jacente, qui prend une très grande importance en FAD : *Comment savoir que les individus n'interagissent pas ?*

Cette situation traduit simplement l'état des connaissances des concepteurs de Simuligne lors de la conception de l'expérimentation. Cette phase qui a été menée par l'équipe de Besançon et celle de l'Open University qui en cette date, ont intégrés des consignes tenant compte de leur expérience et connaissance de l'animation des groupes en formation ouverte et à distance (il faut noter que l'Open University qui délivre des formations ouvertes à ses étudiants allient souvent une phase à distance couplée avec du présentiel). La conduite de Simuligne nous a montré que la FAD 100 % à distance introduit des questions spécifiques déplaçant par exemple les préoccupations de la qualité des interactions dans des groupes à la quantité des interactions dans le groupe. La question du volume des interactions devient absolument cruciale d'autant qu'il caractérise l'existence même du groupe ou non.

L'appréciation des facteurs que nous venons de décrire n'aurait pu être possible sans la conduite de SimuLigne, de ce point de vue, le projet ICOGAD nous aura permis de nous mettre au cœur de certains phénomènes spécifiques à la FAD ou en tous cas, qui prennent une résonance particulière dans ce contexte.

Nous avons présenté ici l'analyse de certaines données d'interaction enregistrées dans la plate-forme au cours de l'expérimentation SimuLigne. Notre but dans cette analyse aura été de montrer que les informations enregistrées dans les fichiers de traces du serveur HTTP utilisé (il s'agissait d'un serveur Apache dans notre cas) et dans des fichiers spécifiques de notre plate-forme (nous avons utilisé WebCT au cours de l'expérimentation) peuvent permettre au prix de certaines opérations informatiques de construire des données d'interaction reflétant le comportement des utilisateurs. Nous avons ainsi montré que finalement, la baisse d'activité dans le groupe Lugdunensis qui a conduit à la fermeture de ce groupe le 30 mai 2001 (soit un mois après le début de la formation Simuligne) aurait pu être prévu d'une certaine manière au fil des jours et des semaines. Nous avons aussi montré que le comportement des apprenants du groupe Lugdunensis peut aussi très bien être observé à partir des données concernant leurs interactions, de même avons-nous mis en perspective des données d'interaction pour montrer que le tuteur peut grâce à celles-ci conjecturer sur la bonne réalisation des activités par les apprenants du groupe. Toutes les données fournies dans les tableaux n'existent pas à l'heure actuelle dans PIFAD, elles demandent à être construites et pour cela, nous avons dû récupérer des informations dans les fichiers de traces, les mettre dans des bases de données et obtenir ces données en exécutant des requêtes SQL appropriées. Ces manipulations sont indispensables pour l'obtention de ces données et ne peuvent en aucun cas être fournies par les logiciels commerciaux d'analyse de fichiers de traces des serveurs HTTP. C'est pourquoi nous pensons que ces fonctionnalités sont absentes à l'heure actuelle et doivent être ajoutées aux plates-formes de télé-formation. Pour notre part, nous avons recours au paradigme d'agent pour les proposer et les construire car pensons-nous par ailleurs, les systèmes informatiques dédiés à la FAD sont finalement des systèmes ouverts, complexes, évolutifs et doivent être dotés des caractéristiques de flexibilité, d'autonomie, d'intelligence. Cette exigence nous conduit naturellement vers ce nouveau paradigme qui proclame l'avènement de nouveaux systèmes où l'utilisateur ne sera plus emmené à procéder à une analyse fine de son système avant de l'implémenter suivant toutes les règles de l'art de la programmation informatique !

Toutefois, il convient de noter que pour autant que le paradigme d'agent est prometteur, il souffre à l'heure actuelle d'une insuffisance notoire de méthodes, outils et langages de développement qui permettraient de construire des systèmes à base d'agents présentant effectivement les caractéristiques annoncées dans la littérature. Dans cette situation, nous pensons que la recherche et la validation des méthodologies et outils pour développer des systèmes à base d'agents demeurent de véritables questions de recherche.

Nous avons identifié pour notre part, une méthodologie de développement des systèmes multi-agents. Nous en avons instancié les différentes étapes de celle-ci pour spécifier notre système multi-agents, SIGFAD. Ceci nous a amené à nous rendre qu'un aspect fondamental, la construction de l'architecture interne de nos agents, n'est pas pris en compte par cette méthodologie. Nous avons eu recours à une architecture dédiée à cet effet, son application est en cours et elle nous permettra notamment de montrer comment les données issues de PIFAD peuvent être utilisées concrètement pour atteindre les objectifs de SIGFAD.

En marge de ceci, il convient toutefois de poursuivre l'analyse des données au niveau par exemple des interactions des tuteurs pour voir si là aussi, leur comportement dans le déroulement de la formation n'aura pas influencé son cours. De même, sera-t-il nécessaire de prendre tout l'ensemble du groupe dans sa globalité et de définir des observables susceptibles d'expliquer la vie du groupe au cours de la formation. Nous plaçons de grands espoirs dans l'analyse par réseaux sociaux qui permet de définir des concepts comme la cohésion du groupe. Cette théorie nous semble d'autant plus prometteuse qu'elle a déjà été utilisée par plusieurs chercheurs pour modéliser et visualiser la collaboration dans les groupes de FAD. Voir pour cela (Reffay et Chanier, 2002). Rappelons qu'un simple outil de représentation de la collaboration dans les groupes serait d'un grand apport pour le tuteur et le coordonnateur dans le type de pédagogie mise en œuvre au cours de Simuligne.

REFERENCES

- Alsic (2001) "Glossaire français-anglais sur l'apprentissage des langues et les SIC". *Revue ALSIC (Apprentissage des Langues et Système d'Information et de Communication)*. Consulté en septembre 2001 à <http://alsic.org>
- Baker, M., (2000). The roles of models in Artificial Intelligence and Education research : a prospective view. *International Journal of Artificial Intelligence in Education*, 11. PP 122-143.
- Bradshaw, J. M., (1997). An Introduction to Software Agents. In *Software Agents*. Ed. J.M. Bradshaw. Menlo Park, Calif. : AAAI Press.
- Chanier, T. (2001). *Les dossiers de l'Ingénierie Educative, no 36 sur "Les communautés en ligne", octobre*. Centre National de Documentation Pédagogique(CNDP): Montrouge. pp 56-59.
- Charlier, B., Daele, A., Cheffert, J-L., Peeters, R., Lusalusa, S. (1999). Learning collaboratively in a virtual campus : teachers' experiences. ISATT 99. Dublin
- Deloach, S., A. (2001). Analysis and Design using MaSE and agentTool. The 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 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.
- Denis, B., Leclercq, D. (1995). Apprentissage et multimédia, Actes de la Journée d'information sur le multimédia, Presses Universitaires de Namur, PP 119-141
- Dillenbourg, P. (1999) What Do You Mean By "Collaborative Learning" in P. Dillenbourg (Ed.) *Collaborative Learning : Cognitive and Computational Approaches*, pp 1-19. Amsterdam : Pergamon/Elsevier Science.
- Dillenbourg, P., Baker, M., Blaye, A., O'Malley, C., (1996). The evolution of research on collaborative learning. In E. Spada & P. Reiman (Eds) *Learning in Humans and Machine : Towards an interdisciplinary learning science*. pp 189-211. Oxford. Elsevier.
- Engelström, Y. (1987). Learning by expanding. An activity-theoretical approach to developmental research. Orienta-Konsultit Oy, Helsinki.
- Ferber, J., (1997) Les systèmes multi-agents : vers une intelligence collective. Intereditions.
- Fjåk, A. (1998). Computer Support for Distributed Collaborative Learning. Exploring a Complex Problem Area. Dr. Scient. Thesis5. Department of Informatics. University of Oslo.
- Fjåk, A., Nurminen, M. I., Smordal, O. (1997). Taking Articulation Work Seriously – an Activity Theoretical Approach. TUCS Technical Report N° 120. 16 p.
- Haythornthwaite, C. (1999). Networks of Information Sharing among Computer-Supported Distance Learners. *Proceedings of the Third International Conference on Computer-Supported Collaborative Learning (CSCL'99)*. pp 218-222.
- Jermann, P., Soller, A., Muehlenbrock, M. (2001). From Mirroring to Guiding : A Review of State of the Art Technology for Supporting Collaborative Learning. *Proceedings of the First European Conference on Computer-Supported Collaborative Learning (CSCL'01)*.
- Kusunoki, F., Sugimoto, M., Hashizume, H. (1999) A System for Supporting Group Learning that Enhances Interactions. *Proceedings of the Third International Conference on Computer-Supported Collaborative Learning (CSCL'99)*. pp 323-327.
- Maes, P. (1997). Agents that Reduce Work and Information Overload. In *Software Agents*, ed. J.M. Bradshaw. Menlo Park, Californie, AAAI Press.

- Mbala, A. (2001). SIGFAD : un système multi-agents pour le soutien des interactions en formation à distance. Rapport 5/0110 du Projet ICOGAD. Octobre 2001. 95 p
- Mbala, A. (2002). De l'analyse des données d'interaction de Simuligne à la spécification d'un système multi-agents pour soutenir les activités des utilisateurs en formation à distance. Rapport 6/0110 du Projet ICOGAD. Avril 2002. 40 p
- Mbala, A., Reffay, C., Chanier, T. (2002). Integration of Automatic Tools for Displaying Interaction Data in Computer Environments for Distance Learning. Accepted for Intelligent Tutoring Systems Conference 2002 (ITS'2002) to be held in June 2002 in Lyon (France).
- Nurmela, K., Lehtinen, E., Palonen, T. (1999). Evaluating CSCL Log Files by Social Network Analysis. Proceedings of the Third International Conference on Computer-Supported Collaborative Learning (CSCL'99). pp 434-442.
- Reffay, C., Chanier, T. (2002). Social Network Analysis used for modelling collaboration in distance learning groups. Accepted for Intelligent Tutoring Systems Conference 2002 (ITS'2002) to be held in June 2002 in Lyon (France).
- Torrent, S., (2001). "Le questionnaire d'une plate-forme de téléformation. Présentation et analyse de la fonction dans un contexte de formation en ligne : Simuligne". Rapport de stage DESS, juin 2001. Besançon : Laboratoire LIFC, Université de Franche-Comté. 97 p..
- Weiss, G., Dillenbourg (1999). What is 'multi' in multi-agent learning ? In P. Dillenbourg (Ed) Collaborative – Learning : Cognitive.
- Wertsch, J.V. (1991). A socio-cultural approach to socially shared cognition. In Resnick, L., Levine, J., Teasley, S. (Eds). Perspectives on Socially Shared Cognition. PP 85-100. Hyattsville, MD: American Psychological Association.
- Whatley, J., Staniford, G., Beer, M., Scown, P. (1999). Intelligent Agents to Support Students Working in Group Online. Journal of Interactive Learning Research. Vol. 10, N° 3 /4. AACE, Charlottesville.
- Wood, M. F., Deloach, S. A. (2000). An overview of the Multiagents Systems Engineering Methodology. Proceedings of the First International Workshop on Agent-Oriented Software Engineering. Ciancarini P., Wooldridge, M. (Eds). Lectures Notes in Computer Science, Vol. 1957. Springer Verlag. Berlin.
- Wooldridge, M. (1999) Intelligent Agents In G. Weiss (Ed). Multiagent Systems, The MIT Press, April 1999.