



HAL
open science

Analyse, conception, spécification et développement d'un système multi-agents pour le soutien des activités en formation à distance

Aloys Mbala Hikolo

► **To cite this version:**

Aloys Mbala Hikolo. Analyse, conception, spécification et développement d'un système multi-agents pour le soutien des activités en formation à distance. Education. Université de Franche-Comté, 2003. Français. NNT: . edutice-00000429

HAL Id: edutice-00000429

<https://theses.hal.science/edutice-00000429>

Submitted on 20 Apr 2004

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

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

THESE

présentée à

L'Université de Franche-Comté

Par

Aloys MBALA HIKOLO

Pour obtenir le titre de

Docteur de l'Université de Franche-Comté

Spécialité : Automatique et Informatique

Analyse, conception, spécification et développement d'un système multi-agents pour le soutien des activités en formation à distance

soutenue le 16 octobre 2003

Devant la commission d'examen formée de :

Président	:	Jean-Marc Labat	<i>Professeur à l'Université de Paris 5</i>
Rapporteurs	:	Stefano A. Cerri	<i>Professeur à l'Université de Montpellier 2</i>
		Pascal Leroux	<i>Professeur à l'Université du Maine</i>
Examineurs	:	Thierry Chanier	<i>Professeur à l'Université de Franche-Comté</i> <i>Directeur de thèse</i>
		Christian Depover	<i>Professeur à l'Université de Mons-Hainault (Belgique)</i>
		Jacques Julliard	<i>Professeur à l'Université de Franche-Comté</i>
		Christophe Reffay	<i>Maître de Conférences à l'Université de Franche-Comté</i> <i>Responsable scientifique</i>

Remerciements

Cette thèse est le résultat de l'engagement de plusieurs personnes qui ont décidé de m'accompagner résolument dans cet exaltant parcours. Je voudrais profiter de cet espace pour leur exprimer toute ma gratitude et ma reconnaissance.

Mes premières pensées vont à l'endroit du Professeur Awono Onana, Directeur de l'Ecole Nationale Supérieure Polytechnique de Yaoundé, qui le premier a pensé que je pouvais faire une carrière d'enseignant-chercheur. Il n'a pas ménagé sa peine pour me trouver un point d'accueil et les financements nécessaires à l'accomplissement de mes différents séjours à Besançon.

Christophe Reffay est sans aucun doute celui grâce auquel j'ai pu effectuer cette thèse au sein du Laboratoire d'Informatique de l'Université de Franche-Comté. Il a par la suite assumé les délicates tâches de responsable scientifique de mes travaux de thèse avec amitié et indulgence, sans jamais se départir de la rigueur scientifique nécessaire. Je voudrais lui exprimer toute ma reconnaissance.

Je remercie Thierry Chanier, Professeur à l'Université de Franche-Comté, qui a accepté d'être mon directeur de thèse et qui a bien voulu me faire profiter de sa rigueur scientifique et de la pertinence de ses remarques. Il aura su me faire saisir les problématiques liées aux environnements informatiques pour l'apprentissage humain qui constituaient un nouveau champ disciplinaire pour moi.

Je remercie Pascal Leroux, Professeur à l'Université du Maine et Stefano A. Cerri, Professeur à l'Université de Montpellier 2, qui m'ont fait l'honneur d'être rapporteurs de ma thèse et dont les remarques ont contribué à l'améliorer.

Je remercie Jean-Marc Labat, Professeur à l'Université de Paris V, Christian Depover, Professeur à l'Université de Mons-Hainault (Belgique) et Jacques Julliard, Professeur et Directeur du Laboratoire d'Informatique de l'Université de Franche-Comté pour avoir accepté de participer à mon jury.

Je remercie Françoise Simonot, enseignante à l'IUT de Besançon, qui a accepté de relire une première version de mon manuscrit, permettant ainsi de supprimer nombre de coquilles et d'incorrections qui s'y trouvaient. Je tiens néanmoins à souligner que les fautes d'orthographe, de grammaire ou de style qui peuvent encore s'y trouver sont de mon seul fait.

Je tiens à saluer Jérôme Nicolet avec qui j'ai partagé trois années de travail au sein de l'équipe SICAH ainsi que celles et ceux avec qui nous avons travaillé dans le cadre du projet ICOGAD.

Dû à l'éloignement de mon lieu de vie habituel, il est fort compréhensible que cette thèse n'ait pu se dérouler que grâce au soutien affectif et moral de ma famille et de mes amis. Qu'il me soit alors permis de remercier :

- Nathalie, mon épouse qui aura tenu, durant ces dernières années, le rôle d'unique chef de famille en encadrant au mieux nos enfants : Batman et Toutou ;
- Byge, dont l'amitié et le soutien ne m'auront jamais fait défaut tout au long de ces quatre dernières années ;
- Gertrude et Georges, toujours prompts à m'accueillir et à m'héberger lorsque je passais par Paris ;
- Mireille et Michel qui m'ont ouvert leur maison à Besançon, devenant par là-même une véritable famille pour moi ;
- et bien sûr, tous ceux et celles que je ne peux citer ici, faute de place ou par oubli.

Sommaire

REMERCIEMENTS	1
SOMMAIRE.....	3
INTRODUCTION.....	7
CONTEXTE DE NOS TRAVAUX DE RECHERCHE	11
PROBLEMATIQUE	12
OBJECTIFS DE RECHERCHE	13
APPROCHE METHODOLOGIQUE	15
PLAN DU MEMOIRE.....	16
CHAPITRE 1 : FORMATION A DISTANCE, TELEFORMATION ET E-FORMATION.....	19
1.1 - FORMATION A DISTANCE ET E-FORMATION	22
1.1.1 - Définitions.....	22
1.1.2 – FOAD, FAD et e-formation.....	23
1.1.3 - Les enjeux de la FAD.....	25
1.2 - LA TELEFORMATION	29
1.2.1 - Présentation générale.....	30
1.2.2 - Typologie des contextes d'utilisation de la formation en ligne en France.....	32
1.2.3 - Les plates-formes.....	36
1.3 – NORMES ET STANDARDS DE LA E-FORMATION.....	43
1.3.1 - Les travaux de normalisation et de standardisation.....	43
1.3.2 - Le modèle SCORM.....	47
CHAPITRE 2 : FONDEMENTS THEORIQUES	53
2.1 - APPRENTISSAGE COLLABORATIF.....	55
2.1.1 - La variété des échelles.....	56
2.1.3 - La variété des significations données au mot «apprentissage».....	57
2.1.4 - La variété des significations données au mot : collaboration.....	60
2.2 - LE GROUPE ET LES INTERACTIONS.....	62
2.3 - LE ROLE DU COORDONNATEUR ET DU TUTEUR	65
2.4 - LA THEORIE DE L'ACTIVITE.....	68
2.4.1 - Les niveaux d'une activité.....	69
2.4.2 - Les propriétés de l'activité.....	72
2.4.3 - La structure basique d'une activité.....	73
2.5 - L'ECHELLE D'ANALYSE DE LA TACHE DANS NOTRE RECHERCHE.....	76
2.6 - LES RAISONS DU CHOIX DU PARADIGME D'AGENT.....	77
2.7 - L'APPROCHE COMPOSANTS DANS LES EIAH.....	80
2.7.1 Définitions.....	80
2.7.2 Quelques types de composants pour les systèmes d'information	81
2.7.3 Démarches d'ingénierie de systèmes à base de composants.....	84
2.8 - TERMINOLOGIE	86
2.8.1 - PIFAD, SIGFAD, SimuLigne.....	86
2.8.2 - Groupe / Cours:.....	87
2.8.3 - Espaces / e-monde / e-univers.....	87
2.8.4 - e-personnalité.....	88
2.8.5 - Utilisateurs et statuts PIFAD.....	88
2.8.6 - Acteur SimuLigne.....	90
2.8.7 - Implémentation des fonctions et statuts SimuLigne dans PIFAD :	92
2.8.8 - Module de formation / Etape / Activité / Tâche	93
2.8.9 - Outil PIFAD.....	94
2.8.10 - Agent / Système Multi-Agents / rôle MaSE / Tâche informatique concurrente / Objectif MaSE	95
2.8.11 - Interaction / Interactivité / Communication.....	96

CHAPITRE 3 - L'EXPERIMENTATION SIMULIGNE : LE MODELE CONCEPTUEL D'ACTIVITE ET L'ANALYSE DES DONNEES D'INTERACTIONS DANS LES GROUPES	99
3.1 - SIMULIGNE	101
3.1.1 - Contexte de départ	101
3.1.2 - Principes et objectifs pédagogiques de SimuLigne	103
3.2 - LES INSUFFISANCES CONCEPTUELLES DES PLATES-FORMES DE TELE-FORMATION	116
3.2.1 - Le modèle conceptuel d'activité dans SimuLigne	117
3.3 - L'ANALYSE DES DONNEES D'INTERACTION OBTENUES	124
3.3.1 - Les données d'interaction relatives aux groupes	125
3.3.2 - Les données d'interaction relatives aux utilisateurs du groupe Lugdunensis	130
3.4 - LIMITES DE PIFAD ET PROPOSITION D'UN SMA : SIGFAD	135
3.4.1 - Les limites des plates-formes existantes	135
3.4.2 - Les fonctionnalités de SIGFAD	138
CHAPITRE 4 : AGENTS, SYSTEMES MULTI-AGENTS ET METHODOLOGIES	141
4.1 - AGENTS	144
4.1.1 - Définitions	144
4.1.2 - Typologie des agents	148
4.2 - LES SYSTEMES MULTI-AGENTS	154
4.2.1 - Définitions	156
4.2.2 - Raisonnement et communication dans les SMA	156
4.2.3 - Méthodologies et plates-formes de développement des systèmes à base d'agents	161
4.3 - APPRENTISSAGE MACHINE ET SYSTEMES MULTI-AGENTS	167
4.3.1 Généralités	167
4.3.2 - Définition et principes	168
4.3.3 - L'apprentissage par les agents	170
4.3.4 - Les techniques d'apprentissage	171
4.4 - LES OUTILS DE DEVELOPPEMENT DE SIGFAD	172
4.4.1 - La méthodologie Multiagent System Engineering (MaSE)	172
4.4.2 - L'architecture JAM	179
4.4.3 - La communication entre les agents	183
CHAPITRE 5 : ANALYSE, CONCEPTION, SPECIFICATION ET IMPLEMENTATION DE SIGFAD	189
5.1 - SIGFAD	192
5.1.1 - Analyse, conception et spécification de SIGFAD	192
5.1.2 - Implémentation de SIGFAD	213
5.2 - L'APPLICATION DE SIGFAD AUX DONNEES DE SIMULIGNE	224
5.2.1 - Les informations fournies sur les apprenants	228
5.2.2 - Les informations portant sur le groupe tout entier	229
5.2.3 - Les informations portant sur les activités	231
5.3 - VERS UNE PRISE EN COMPTE DES DONNEES D'INTERACTION DANS LA NORMALISATION ET LA STANDARDISATION DE LA E-FORMATION	232
BILAN, PERSPECTIVES ET CONCLUSION	235
BILAN DES TRAVAUX	238
RESULTATS OBTENUS	240
APPORTS DE NOS TRAVAUX DE THESE	241
PERSPECTIVES	243
CONCLUSION	244
ANNEXES	247
A - LES PRODUCTIONS DE SIMULIGNE	249
B - APPLICATION DE LA METHODOLOGIE MASE A L'ANALYSE, LA CONCEPTION ET LA SPECIFICATION DE SIGFAD	251
B-1 Les diagrammes de séquence décrivant les cas d'utilisation	251
B.2 Les automates à états finis décrivant les tâches concurrentes	253
C - APPLICATION DE SIGFAD AUX DONNEES DE SIMULIGNE	259
D - CODE SOURCE DE SIGFAD	260

<i>D – 1 Code source de l'agent superviseur des interactions</i>	261
<i>D.2 - Exemple de code source d'un agent tuteur : L'agent du tuteur du groupe Aquitania</i>	266
<i>D.3 - Code de ReceiveQueryTutor.java</i>	274
<i>D.4 – Un exemple de la primitive : AbsentsInGroup</i>	275
<i>D.5 - Code de la primitive SendResultToTutor.java</i>	277
LISTE DES ACRONYMES	279
LISTE DES FIGURES	283
LISTE DES TABLEAUX	284
INDEX DES MOTS CLES ET D'AUTEURS CITES	285
REFERENCES BIBLIOGRAPHIQUES	287
BIBLIOGRAPHIE.....	289
SITES INTERNET	294
TABLE DES MATIERES	295

Introduction

On assiste à l'apparition de nouveaux besoins dans la formation et l'enseignement : d'une part ces besoins se manifestent dans trois directions : les personnes, les lieux et les moments ; c'est ce que Denis et Leclercq (1995) simplifient en disant : *tous, tout le temps, partout* ; d'autre part, les apprenants ont de plus en plus besoin d'un apprentissage centré sur eux-mêmes, ils demandent que leur vision soit prise en compte. L'une des réponses à l'émergence des nouveaux besoins de formation est la formation à distance (FAD) médiatisée par ordinateur. Dès lors, il n'est pas surprenant de constater que la FAD prend une importance de plus en plus grandissante dans la société contemporaine. En gardant à l'esprit ce qui vient d'être dit, on peut affirmer que l'intérêt pour la FAD correspond 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 adaptées 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. Un collectif de professionnels et de chercheurs rassemblés au sein d'une conférence de consensus sur les formations ouvertes et à distance (FOAD) explique l'essor de ce type d'enseignement et de formation par la montée de plusieurs phénomènes : *diffusion des technologies de l'information et de la communication dans l'ensemble du tissu social et professionnel, accroissement des besoins de compétence face à des effets de seuil dans les financements publics et privés de la formation, démultiplication des réseaux d'accès à l'information, critique des formes canoniques de la formation, poussée des thématiques de l'organisation apprenante, de l'autoformation et de la société dite "cognitive", etc.* (Collectif de Chasseneuil, 2000).

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), l'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.

Beaucoup d'auteurs (Winn, 1993), (Montgomery, 1998), (Furness et al., 1998) soulignent que l'enseignement à l'école se faisait jusqu'à présent en utilisant un symbolisme abstrait qui reflète davantage la perception et la vision du formateur (enseignant). Aujourd'hui, face à la complexification de notre société et des savoirs (Lebrun, 1999), l'apprenant doit avoir la possibilité de mener des expériences directes avec son environnement pour mettre en avant sa propre perception des choses, c'est ce que Clancey (1993) appelle les « *first-hand experiences* ». Cette exigence constitue l'un des fondements du constructivisme. Un autre principe sur lequel s'appuie le constructivisme est que l'apprentissage se fait de façon collaborative, c'est une activité (sociale) de groupe (Winn, 1997). Il est d'ailleurs important de noter avec Depover et al. (1999) que l'enseignement à distance a vu son modèle pédagogique évoluer du transfert de connaissances à l'interaction dynamique. Cette nouvelle manière de conceptualiser la relation pédagogique vient aussi bien du constructivisme que de la prise en compte du rôle essentiel joué par le contexte d'apprentissage. Les mêmes considérations voient la banalisation du rôle de l'enseignant qui évolue du statut de pourvoyeur de connaissances vers celui d'acteur parmi d'autres au sein d'un système global. Ce système global intègre des partenaires humains (entre autres, apprenants au niveau de connaissances variable) et des artefacts ou outils cognitifs *qui, en modifiant la représentation de la tâche, supportent l'apprenant dans son traitement cognitif de la situation.*

L'apparition des TICE (Technologies de l'Information et de la Communication appliquées à l'Enseignement) avec l'explosion des ressources informatiques et télématiques constitue, là aussi, une chance pour obtenir un enseignement davantage centré sur l'apprenant. Dans le même temps, on note un intérêt de plus en plus important pour l'apprentissage collaboratif (René et Chanier, 1996) ; en effet, non seulement plusieurs problèmes se posent de manière distribuée, mais encore, dans les administrations, les entreprises et la société en général, la résolution des problèmes est faite de manière distribuée (Ferber, 1995). L'intérêt pour l'apprentissage collaboratif est étayé par des travaux très anciens, notamment la théorie de

l'apprentissage de Vygotsky selon laquelle la construction des connaissances se réalise d'abord durant l'interaction sociale avant d'être internalisée (Charlier et al., 1998). La théorie de la cognition distribuée va dans le même sens puisqu'elle situe l'apprentissage non pas au niveau individuel mais au niveau d'une communauté ou d'une organisation.

Contexte de nos travaux de recherche

Nos travaux se situent dans le champ de l'apprentissage collaboratif à distance. Nous nous intéressons à la construction d'outils et d'artefacts informatiques capables de fournir un soutien et une assistance aux personnes engagées dans cette forme d'enseignement. C'est dire que nous nous situons à l'intersection des sciences humaines, en ce sens qu'il faut identifier, comprendre et modéliser les phénomènes observés, et des sciences de l'ingénieur, car nous devons rechercher dans l'informatique et la télématique des paradigmes susceptibles d'aider à la résolution des problèmes identifiés.

De manière plus générale, nos thématiques de recherche touchent à des communautés d'ores et déjà organisées comme les communautés CSCL (*Computer-Supported Collaborative Learning*) et ITS (*Intelligent Tutoring Systems*) regroupées en France au sein de la communauté EIAH (Environnements Informatiques pour l'Apprentissage Humain). L'utilisation de paradigmes informatiques comme les agents et les systèmes multi-agents (SMA) nous plongent au cœur de la communauté AIED (*Artificial Intelligence in Education*). Notre démarche consiste à mener des expérimentations grandeur réelle, à identifier des problèmes à un niveau macro et à proposer des modèles et problématiques informatiques capables d'accompagner les différents utilisateurs dans la réalisation de leurs activités d'apprentissage. L'originalité de notre démarche nous amène à intégrer :

- des théories issues des sciences humaines et sociales comme la théorie de l'activité (TA), et l'analyse par les réseaux sociaux (SNA pour *Social Network Analysis*), aussi appelée théorie de l'interactionnisme structural ;
- des paradigmes informatiques comme l'analyse, la conception, la spécification et le développement des systèmes à base d'agents ;
- des problématiques inhérentes à l'intelligence artificielle distribuée (IAD) comme la répartition, la communication entre les systèmes, la modélisation de l'activité humaine et des entités intelligentes.

Problématique

La formation à distance pose des problèmes spécifiques. Au nombre de ces problèmes, on note le sentiment d'isolement éprouvé par les utilisateurs et le déficit d'accompagnement des apprenants ; tout ceci conduit à un grand taux de déperdition et accroît sensiblement les risques d'éclatement des sessions de formation à distance. Devant le fort taux d'abandon observé, il serait particulièrement intéressant de dégager les facteurs de réussite et d'échec d'une session de formation à distance : quels éléments dans les dispositifs informatiques, dans les approches pédagogiques ou dans le vécu de la formation, font qu'une formation atteint ses objectifs et va jusqu'à son terme ? Quelles stratégies convient-il de mettre en place pour faire diminuer les taux d'abandon inhérents à ce type de formation ? Quels sont les indicateurs à mettre en place et à surveiller lorsque l'on est chargé du bon déroulement d'une session de FAD ?

Dans cette catégorie de problèmes qui sont finalement très généraux à la FAD, nous nous intéressons en particulier à une pédagogie en situation de distance, basée sur une production collaborative dans des groupes de taille restreinte. Nous empruntons à Fjök et Ludvigsen (2001) l'expression «apprentissage collaboratif distribué» pour désigner ce type de pédagogie. On peut observer que ceci induit, en plus des questionnements dégagés plus haut, d'autres préoccupations spécifiques à l'approche pédagogique choisie. Finalement, la question centrale de cette thèse concerne la mise au point d'outils informatiques pouvant faciliter le suivi de l'apprentissage collaboratif dans le contexte de la formation à distance. Il est, de ce point de vue, important de comprendre le rôle du groupe d'une part, en tant qu'entité et d'en faire l'unité d'analyse, d'autre part, le rôle de personnes en charge de l'existence et surtout de la vitalité de ces groupes.

Il est aussi important de comprendre le rôle des interactions qui apparaissent entre les membres du groupe. Quels sont les effets de ces interactions dans l'existence du groupe, quels changements cognitifs induisent-elles sur les apprenants ? En d'autres termes, quel est le rôle des interactions dans la progression et la réussite d'une session de FAD ? On est ainsi amené à s'intéresser très rapidement au comportement de l'apprenant, du tuteur, ... et ainsi à modéliser le comportement des acteurs engagés dans la FAD.

La variété des espaces et des temps dans la FAD complique singulièrement les activités des utilisateurs, en même temps que les supports et les systèmes informatiques utilisés peuvent être porteurs de complexité et de difficultés supplémentaires. Il est toutefois important de bien cerner l'activité des différents utilisateurs et de soutenir celle-ci en proposant des systèmes informatiques disposant d'outils et d'artefacts appropriés. Cette préoccupation interpelle les informaticiens et les engage à identifier des outils, concepts et paradigmes capables de les aider à analyser, concevoir, spécifier et développer ces systèmes informatiques. Les plateformes informatiques adaptées à la mise en œuvre et au suivi de l'apprentissage collaboratif distribué sont des systèmes complexes, ouverts, évolutifs et flexibles. Les paradigmes et outils de développement habituellement utilisés en informatique ne sont pas toujours appropriés pour répondre à la mise en place de tels systèmes.

La FAD vient nourrir de façon significative le débat et l'ambition d'avoir une nouvelle informatique : celle où l'ordinateur ne serait plus un exécuter servile et obéissant mais deviendrait un partenaire capable d'assister l'utilisateur dans la résolution des tâches (Negroponte, 1997), (Leroux, 1995). On comprend bien que l'approche classique de l'ordinateur servile oblige le programmeur à mettre en œuvre des capacités d'analyse, de conception, de codage des logiciels avérées, alors que la démocratisation de l'informatique emmène de plus en plus des non informaticiens à manipuler les ordinateurs. Ceci serait certainement un moindre mal s'il n'était désormais établi que nombre de problèmes sont tellement complexes et évolutifs qu'il est hors de propos d'envisager une analyse et une modélisation complètes débouchant à la mise en place d'applications appropriées (Ferber, 1995).

Objectifs de recherche

Les objectifs de nos travaux de thèse sont de répondre aux différents éléments de problématique dégagés plus haut. De façon plus précise, il s'agit à travers la conduite d'une expérimentation de FAD, de cerner le rôle du tuteur et des autres utilisateurs dans la réussite d'une session de formation à distance basée en particulier sur l'apprentissage collaboratif ; de construire des outils informatiques (pour le tuteur en particulier) capables de faciliter le suivi de l'apprentissage collaboratif distribué. Nos travaux concernent aussi la compréhension et la mise en place d'interfaces entre les plateformes informatiques d'e-formation (les anglo-saxons parlent de *Learning Management Systems* ou LMS, nous utiliserons souvent cette

dernière appellation dans la suite) et les utilisateurs. Ceci s'inscrit dans les efforts de normalisation et de définition de standards pour la e-formation entrepris aux niveaux mondial (ISO, IMS, LTSC, IEEE), européen (ARIADNE, CEN) et français (AFNOR).

Nous avons indiqué plus haut que notre démarche nous conduit à dégager des questions inhérentes à la conduite des sessions de formation grandeur réelle. En d'autres termes, notre souci est de comprendre ce qui se passe dans une session de formation étalée sur plusieurs semaines, engageant de vrais apprenants avec des objectifs pédagogiques réels. L'un de nos premiers objectifs était donc de concevoir, spécifier, et mettre en œuvre une expérimentation avec, comme stratégie pédagogique, l'apprentissage basé sur une production collaborative dans des groupes de taille réduite (8 à 15 personnes).

Le second objectif consiste à recueillir, analyser le corpus de données issues de l'expérimentation en vue d'identifier des éléments de suivi de l'activité des groupes en FAD, des indices ou indicateurs à mettre en place et à surveiller lorsque l'on gère une session de FAD. Il s'agissait en fin de compte, à partir du vécu de la formation et des données recueillies, d'identifier les facteurs ou phénomènes déterminants pour la réussite ou l'échec d'une session de FAD.

Le troisième objectif de nos travaux de thèse est de proposer des outils informatiques dont on doterait les LMS afin de contribuer à la réussite des sessions de FAD. 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 de la session de formation (réalisation des activités pédagogiques) et l'état du groupe.

Notre quatrième objectif de recherche est, en tant qu'informaticiens, d'identifier et de mettre en œuvre des concepts informatiques adaptés à la construction des systèmes informatiques supports de la FAD. Nous nous sommes orientés vers le paradigme d'agent, qui souffre encore d'une absence cruelle de méthodologies, de langages et d'outils de développement validés. Nous affirmons que l'identification et la validation d'outils de développement des systèmes à base d'agents constituent une véritable question de recherche. Une réponse

satisfaisante ne lui sera apportée que le jour où l'on disposera de systèmes bâtis, dotés des caractéristiques annoncées et appliqués à plusieurs domaines, dont les EIAH.

Approche méthodologique

L'approche méthodologique choisie visait à apporter une réponse aux différents éléments de problématique dégagés plus haut et à remplir nos objectifs de recherche. Pour cela, nous avons participé à la mise en place d'une expérimentation réelle de formation à distance. Ceci a été notamment possible grâce à un programme de recherche financé par le Ministère français de la Recherche (Programme Cognitique 2000) dont le but était de former des équipes pluridisciplinaires qui travailleraient dans le domaine des sciences cognitives (Icogad, [http](http://www.icogad.fr)). Nous avons ainsi mené, avec notre équipe de recherche, une session réelle de formation à distance baptisée SimuLigne. Nous avons recueilli un corpus de données d'interaction et avons aussi vécu au quotidien la progression de la session de formation durant les dix semaines au cours desquelles elle s'est déroulée. A l'image de la composition des équipes participant à ce projet de recherche, nous avons eu une approche méthodologique pluridisciplinaire, entre les sciences de l'ingénieur et les sciences humaines et sociales. En effet, notre souci préalable aura été de nous assurer du bon déroulement de SimuLigne, cette expérimentation constituant la base à partir de laquelle nous pourrions plus tard dégager des questions de recherche intéressantes et valider les solutions afférentes. Nous nous sommes ainsi intéressés à la mise en œuvre d'une pédagogie basée sur la production collaborative dans des groupes de taille restreinte. De ce point de vue, nous avons fourni l'assistance technique à l'ensemble des acteurs de la formation et nous avons ainsi cerné différentes préoccupations liées à l'utilisation de la technologie et des difficultés supplémentaires que celle-ci peut introduire.

L'autre souci crucial aura été de construire d'abord, au cours de la formation des outils capables de nous aider à apprécier le déroulement de la formation, le niveau de réalisation des activités prévues ou l'opportunité de modifier le calendrier pédagogique.

Une fois la formation achevée, il a fallu procéder au recueil des données d'interaction enregistrées dans la LMS durant la formation. Pour cela, il a fallu comprendre la façon dont les données sont stockées dans la plate-forme, en comprendre la signification et organiser ces données afin d'en faciliter l'exploitation (Reffay, 2001).

Au niveau informatique, nous nous sommes orientés vers l'utilisation du paradigme d'agent et des systèmes multi-agents pour la construction des outils supplémentaires dont il faudrait doter les plates-formes informatiques. Nous avons procédé à une étude bibliographique afin de mieux nous familiariser avec les concepts d'agent et les problématiques de construction de systèmes à base d'agents. Très rapidement, il nous est apparu qu'il existe peu d'implémentations réelles et on note une abondante littérature soulignant les promesses et les avantages de ce que d'aucuns appellent « Programmation Orientée-Agent » (Shoham, 1993). Notre souci aura donc été d'identifier des outils de développement relatifs à la programmation orientée-agent (POA) et surtout de les appliquer concrètement à la construction d'un système multi-agents.

Plan du mémoire

Le premier chapitre de cette thèse présente le contexte de notre recherche. L'évolution des besoins d'apprentissage et l'éclatement des espaces et des lieux ont conduit à l'émergence d'une FAD en ligne. Nous présentons cette nouvelle forme d'enseignement, appelée téléformation. Nous mettons en perspective la téléformation par rapport à la FAD, la FOAD, les TICE, la e-formation (*e-learning* en anglais). Quelques chiffres permettant d'apprécier la montée en puissance de la e-formation en général sont présentés. Nous portons notre intérêt sur les outils à greffer aux plates-formes existantes pour les voir devenir de véritables environnements informatiques supportant la FAD, c'est-à-dire des environnements intégrant des outils permettant d'apprécier l'activité des acteurs, d'évaluer la progression pédagogique de la session de FAD. La recherche des informations nécessaires à l'appréciation de l'activité des utilisateurs et la progression de la session de FAD nous ont amenés à nous intéresser à l'organisation des données d'interaction dans les LMS. Ceci nous a donné l'occasion de nous préoccuper des problèmes de normalisation des LMS et du *e-learning* (dans la suite, nous utiliserons l'équivalent français de ce terme, la e-formation) en général. Nous présentons l'évolution des EIAH vers une normalisation.

Le deuxième chapitre présente les fondements théoriques sur lesquelles nous nous appuyons pour bâtir toute notre recherche. Nous présentons les questions de recherche inhérentes à l'apprentissage collaboratif distribué, le rôle des interactions, du groupe, du tuteur et du coordonnateur de la formation. Nous présentons diverses théories issues des sciences

humaines et sociales sur lesquelles nous nous sommes basés pour modéliser l'activité pédagogique. Nous justifions aussi le recours au paradigme d'agent et faisons une mise en perspective par rapport à l'approche dite par «composants» adoptée dans plusieurs recherches menées dans les EIAH.

Nous décrivons l'expérimentation de formation à distance à laquelle nous avons participé dans le troisième chapitre. Nous présentons le modèle conceptuel d'activité adapté de la théorie de l'activité de Engelström. Nous commentons les résultats issus de l'expérimentation, à savoir, les productions issues des groupes de travail et les données d'interaction intra et inter-groupes. Nous analysons les données d'interaction issues de l'expérimentation. Nous mettons ces données en perspective pour montrer que le vécu d'une session de FAD peut être prédit en les analysant en temps réel. Nous mettons l'accent sur la fermeture d'un groupe survenue au cours de l'expérimentation et montrons que ceci aurait pu être prédit à partir de la qualité et la quantité des interactions apparues lors des précédents jours. Ce chapitre nous donne l'occasion de souligner les limites des LMS existantes dans la mise en œuvre de l'apprentissage collaboratif distribué et la nécessité de leur ajouter des fonctionnalités pour les transformer en véritables plates-formes de téléformation. Ce chapitre s'achève par la description de fonctionnalités que nous souhaitons implémenter dans notre système multi-agents (SMA).

Dans le quatrième chapitre, nous présentons des considérations générales sur les notions d'agent et de systèmes multi-agents. Nous parcourons les méthodologies existantes pour l'analyse, la conception, la modélisation et la spécification des systèmes multi-agents. Nous évoquons les possibilités d'implémenter les algorithmes d'apprentissage machine au sein des systèmes multi-agents. Nous pensons en effet que ce n'est que dotés des mécanismes d'apprentissage machine que les agents logiciels peuvent réellement être des entités intelligentes, dotées de capacités d'autonomie, de prise d'initiative et qu'ils pourraient apprendre à partir du comportement de leurs utilisateurs. Nous présentons à la fin de ce chapitre les outils de développement utilisés pour l'analyse, la conception, la spécification et le développement de SIGFAD, notre SMA.

Le cinquième chapitre concerne la mise en œuvre concrète de SIGFAD. Nous décrivons toutes les étapes du cycle de développement logiciel, en allant de l'analyse à la programmation de SIGFAD. Ceci nous donne l'occasion de souligner le déficit qui caractérise

la programmation orientée-agent en termes d'outils de développement. Ensuite nous montrons les résultats que peut fournir notre système multi-agents en l'instanciant par rapport au corpus de données de SimuLigne. C'est une illustration de l'utilité de SIGFAD au cours d'une session de FAD. Le chapitre s'achève par une présentation de données qui, de notre point de vue, doivent être disponibles au sein des plates-formes afin de permettre une mise en œuvre de dispositifs d'apprentissage basés sur la collaboration en réseau plus efficace. C'est une contribution aux réflexions en cours sur la normalisation et la standardisation de la e-formation.

En guise de conclusion, nous présentons un bilan de nos travaux de thèse, les résultats obtenus et traçons des perspectives futures à nos travaux. Nous évoquons notre contribution dans l'avancement de la compréhension de notre champ de recherche.

Chapitre 1 : Formation à distance, téléformation et e-formation

Nos travaux de recherche se situent dans le domaine des EIAH (Environnements Informatiques pour l'Apprentissage Humain). Les EIAH ont conduit, avec l'émergence des Technologies de l'Information et de la Communication appliquées à l'Education (TICE) et l'apparition des grands réseaux d'information, à la naissance de la e-formation. La formation à distance, qui est notre point d'intérêt focal, trouve un renouveau avec la e-formation. On assiste à une FAD entièrement médiatisée par les ordinateurs et les réseaux informatiques. Notre souci est donc de nourrir la réflexion sur la construction de systèmes informatiques capables de soutenir cette nouvelle forme de FAD. Les LMS existantes présentent des fonctionnalités très intéressantes. Il ne nous semble pas judicieux d'essayer d'en développer de nouvelles, mais au contraire, de greffer aux plates-formes existantes des outils et artefacts implémentant des fonctionnalités prenant en compte les besoins spécifiques de la FAD. Ceci permettra une mise en œuvre plus efficace de la FAD. Notre démarche méthodologique nous amène naturellement aux questions de normalisation des LMS car, il est clair que les outils à greffer doivent être suffisamment génériques pour être intégrés à différentes LMS.

Dans ce chapitre, une première section présente des généralités sur la FAD à savoir les définitions, les enjeux. Ensuite nous élargissons la réflexion à la e-formation en général. En effet, la FAD médiatisée par les ordinateurs et les réseaux informatiques constitue un volet important de la e-formation. Devant la diversité des plates-formes de téléformation, il est apparu indispensable de s'acheminer vers une normalisation de la e-formation. La dernière partie de ce chapitre présente en particulier le modèle de référence SCORM et résume les travaux de normalisation entrepris au niveau français.

Pour une meilleure compréhension de la façon dont nous percevons les différents modes de formation que nous examinons dans la suite, on peut résumer en disant que pour nous :

e-formation	=	FOAD + TICE
téléformation	=	FAD + TICE.

1.1 - Formation à distance et e-formation

1.1.1 - Définitions

Le Collectif de Chasseneuil (2000) sur les formations ouvertes et à distance, a proposé une définition de la formation à distance, en l'élargissant à la formation ouverte et à distance (FOAD).

La formation ouverte et à distance...

- *est un dispositif organisé, finalisé, reconnu comme tel par les acteurs ;*
- *qui prend en compte la singularité des personnes dans leurs dimensions individuelle et collective ;*
- *et repose sur des situations d'apprentissage complémentaires et plurielles en termes de temps, de lieux, de médiations pédagogiques humaines et technologiques, et de ressources.*

Cette définition nous semble parfaitement convenable pour la FOAD, mais eu égard à la spécificité de la FAD, il nous semble important d'essayer d'en donner une pour cette forme d'enseignement. Keegan (1996) justifie la nécessité d'arriver à une définition consensuelle de la FAD en ces termes :

« The need to clarify terminology in this field is urgent. No progress can be made in formulating the theoretical underpinnings of an area of educational endeavour or in developing guides to good practice if there is no agreement on the area of education under discussion. » (PP 33 et 34).

On peut remarquer que peu d'auteurs se lancent dans l'exercice difficile qui consiste à proposer une définition de la FAD. Nous pensons que ceci est révélateur de l'amalgame qui est fait entre toutes les pratiques et les usages en e-formation et ce qui relèverait réellement de la FAD. Ceci doit d'ailleurs nous rendre vigilants face à des études qui prétendent relever de la FAD mais sont réellement axées sur la FOAD et sur toutes les pratiques de e-formation en général (centre de ressources, formation mixte présentiel et distance, enseignement assisté par ordinateur, formation en présentiel s'appuyant sur des ressources multimédia et des LMS etc.).

Keegan (1996) après avoir examiné des définitions proposées par plusieurs spécialistes de la FAD (précisons que Keegan s'intéresse ici à la FAD indépendamment de l'avènement des

TICE et d'Internet), propose une définition que nous retiendrons ici. La formation à distance est une forme d'enseignement caractérisée par :

- la séparation quasi-permanente entre le formateur et l'apprenant tout au long du processus d'apprentissage (ceci différencie la formation à distance de la formation présentielle) ;
- l'influence d'une organisation administrative aussi bien en ce qui concerne la planification et la préparation des matériaux pédagogiques que la mise à la disposition des apprenants des services d'accompagnement et de support (ceci différencie la FAD de l'autoformation) ;
- l'utilisation de médias techniques (imprimerie, audio, vidéo, ordinateurs) pour assurer le lien entre le formateur et l'apprenant et médiatiser le contenu de la formation ;
- l'existence de mécanismes de communication bidirectionnelles afin que l'apprenant bénéficie mais prenne aussi l'initiative de dialogues avec le formateur (ceci distingue la FAD des autres usages de la technologies dans l'éducation) ;
- la quasi-absence de la notion de groupe tout au long du processus d'apprentissage, de sorte que les apprenants sont toujours vus comme des individus isolés et non comme faisant partie d'un groupe, avec la possibilité d'organiser occasionnellement des rencontres, soit en présentiel, soit via des moyens électroniques à des fins didactiques ou de socialisation.

Il convient néanmoins de souligner que, pour notre part, nous nous intéressons à la formation à distance à travers les TICE, donc médiatisée par les réseaux informatiques et entièrement à distance.

1.1.2 – FOAD, FAD et e-formation

Nous avons rapporté en ouverture du paragraphe précédent, une définition de la FOAD dans laquelle on note bien l'étendue des situations de formation qui en relèvent. La FOAD offre un cadre adéquat pour prendre en compte l'introduction des technologies (cédéroms, Internet, Intranet, etc.) dans la formation, la mixité des situations d'apprentissage (présentielle, distance, autoformation). La FOAD, lorsqu'elle s'appuie sur Internet et les TICE constitue ce que l'on désigne couramment par *e-learning* ou e-formation.

La e-formation comprend à la fois des situations où la technologie complète un apprentissage en classe, présentiel, en centre de ressources et relevant ou non de l'autoformation par des composants basés sur le réseau Internet et en particulier la Toile, et par des situations où l'ensemble du processus d'enseignement est vécu en ligne. Plus généralement, on peut définir la e-formation comme toute utilisation de l'Internet dans la gestion et la diffusion de la formation. Grâce aux TIC, la FOAD devrait pouvoir aujourd'hui concilier économie (réduction des frais indirects), réponse aux contraintes (individualisation de la formation) et efficacité (rapidité du déploiement à grande échelle). Ne se contentant plus d'être une réponse à des problèmes strictement géographiques par une diffusion extensive de la formation, la FOAD devenue e-formation est un ensemble de méthodes qui visent à répondre de manière adaptée à plusieurs types de besoins. Ainsi les qualités essentielles recherchées par les concepteurs de formation en ligne sont par exemple une distribution plus large et plus efficace de la formation, l'individualisation et l'adaptation de parcours de formation aux besoins individuels des apprenants, des possibilités de suivi et de tutorat, la rapidité, enfin la démultiplication de l'offre grâce à la mise en commun de ressources, de modules de formation, de tutorat.

La e-formation permet différentes formules de cours :

- L'autoformation individuelle dont le contenu des cours est disponible en ligne, en libre service, à tout moment.
- La formation individuelle en ligne avec tutorat asynchrone, pendant laquelle le travail de l'apprenant est suivi par un tuteur, qui répond à ses questions et contrôle sa progression en différé mais dans un délai très court.
- Les classes virtuelles en ligne avec tutorat synchrone, où les apprenants se retrouvent en ligne pendant une plage horaire déterminée pour un séminaire ou un groupe de travaux dirigés, avec ou sans vision directe de l'intervenant.
- L'accompagnement en ligne qui personnalise entièrement la formation à l'occasion du tutorat synchrone où le tuteur et l'apprenant échangent en temps réel sur le contenu présenté.

On voit bien que la e-formation permet de répondre à un nombre important de besoins de formation ; toutefois, elle pose des problèmes très généraux qui apparaissent aussi bien dans la formation en présentiel que pour certains dans la FAD. Il nous apparaît donc important de nous restreindre aux situations spécifiques à la FAD et de nous intéresser à des

problématiques qui lui sont propres. Dans toute la suite, nous qualifions de « FAD » les seules situations où la distance intervient, en excluant tout autre mode de formation qui inclurait simplement les technologies dans l'enseignement ou l'apprentissage.

1.1.3 - Les enjeux de la FAD

Le développement de la FAD présente plusieurs enjeux (nous ne prenons pas en compte ici les enjeux financiers et les enjeux de positionnement stratégique dans ce secteur économique ou de rééquilibrage par rapport à l'Amérique du Nord ou à d'autres régions du monde). Nous pouvons en retenir :

- l'autonomie de l'apprenant et des autres acteurs des dispositifs de la FAD ;
- l'accompagnement de l'apprenant ;
- le développement des dispositifs de support pour la FAD.

1.1.3.1 - L'autonomie de l'apprenant

L'autonomie de l'apprenant (comme celle des autres acteurs du dispositif) peut être examinée sous deux angles :

- la motivation : sens, projet, finalités, enjeux personnels ;
- les capacités métacognitives : apprendre à apprendre, identifier et gérer des ressources, travailler avec des pairs, maîtriser les techniques et outils d'apprentissage, etc.

L'autonomie ne peut être considérée comme un préalable, un prérequis, une caractéristique fixe, éventuellement discriminante. Elle est conjoncturelle, relative aux finalités de formation perçues par le sujet ; elle est contingente au dispositif pédagogique et évolutive dans le déroulement de l'apprentissage. Il n'y a, de ce point de vue, pas de bon ou de mauvais dispositif, il y a des dispositifs plus ou moins adaptés au degré d'autonomie de l'apprenant.

La prise en compte des caractéristiques de l'autonomie que nous avons indiquées plus haut nous permet de caractériser le rapport individuel à une formation donnée, dans un contexte donné, à un moment donné, par l'un des quatre cas de figure suivants (Collectif de Chasseneuil, 2000) :

- Type 1 : motivation faible / métacompétence faible ;
- Type 2 : motivation forte / métacompétence faible ;
- Type 3 : motivation faible / métacompétence forte ;

- Type 4 : motivation forte / métacompétence forte.

L'autonomie dans un dispositif de FAD s'évalue aussi en termes de prise de contrôle de l'apprenant sur les ressources offertes par le dispositif. Le besoin d'autonomie est causé par l'isolement et l'absence de relation (face-à-face) avec les autres acteurs. L'autonomie s'exprime aussi par la possibilité de choisir son temps de formation et entraîne une interaction différée avec les autres acteurs (synchronie/asynchronie), elle se traduit par une diversification des interactions pédagogiques accessibles (apprentissage par l'action, apprentissage par la découverte, apprentissage à base de cas, etc.) et des modes d'apprentissage (formation tuteurée, formation coopérative, autoformation accompagnée, formation auto-dirigée). L'autonomie dans la FAD se traduit aussi par la possibilité de négocier les objectifs de la formation, les modalités de conduite des apprentissages et les types de régulation entre les différents acteurs de la FAD. La réponse face à cette exigence d'autonomie se trouve dans une ingénierie de dispositifs qui visent à solliciter l'apprenant pour la gestion des temps, des espaces, et des activités pédagogiques, pour tirer le meilleur parti de la diversité de ses capacités d'apprentissage ; et qui proposent des modalités de médiation / régulation / négociation / évaluation qui opèrent sur la dimension cognitive (par ex. : capacité d'induction, analogie), la dimension technique (par ex. : maîtrise d'outils documentaires) et sur la dimension sociale (par ex. : travail avec ses pairs). D'autre part, les modes d'accompagnement pédagogique (prise en main du dispositif, degré de médiation, de guidance, d'intervention du tuteur, de contrainte du parcours, fréquence du suivi, rythme et nécessité de la relance, etc.) varieront en fonction du degré d'autonomie de l'apprenant, ouvrant sur une pédagogie "contingente".

1.1.3.2 - L'accompagnement de l'apprenant

L'accompagnement humain et technique de l'apprenant mobilise ses relations interpersonnelles. L'accompagnement s'inscrit à la fois dans cette partie de l'histoire de l'apprenant qu'est le processus de formation, et dans le tissu des situations sociales qu'il vit.

On assiste à un éclatement et à une tension des différents espaces-temps sociaux :

- ceux liés aux situations individuelles et collectives de formation proprement dites ;
- ceux liés aux situations de relations avec l'institution de formation ;
- ceux liés aux situations professionnelles ;
- ceux liés aux situations personnelles.

L'accompagnement dans les dispositifs de FAD doit à la fois prendre en compte ces tensions et aider l'apprenant à les gérer en cohérence avec son projet. En conséquence, l'accompagnement interpersonnel doit intégrer des dimensions non seulement pédagogiques mais aussi organisationnelles, professionnelles et personnelles.

La multiplicité des intervenants (formateurs, techniciens, “ dépanneurs ” techniques, experts contenus, tuteurs, gestionnaires de la formation, autres apprenants, autres personnes de son environnement, ...) et des fonctions de l'accompagnement nécessitent :

- une évolution des compétences des formateurs vers davantage de polyvalence ;
- une reconnaissance et une valorisation de la médiation assurée par les différents collectifs ;
- une coordination, un pilotage pour intégrer et réguler les accompagnements.

La mise en œuvre de l'accompagnement nécessite aussi l'utilisation d'artefacts pour assurer les fonctions suivantes :

- information sur le processus, les ressources, les modalités d'apprentissage (bases de données, catalogues, serveur de Toile, ...) ;
- communication et coordination entre les acteurs (forum, bavardage, courriel, ...) ;
- ajustement du planning des activités au cours de la formation ;
- suivi, guidage et régulation (carnets de bord, aide à l'articulation entre les apprentissages, aide à l'allocation des ressources, ...) ;
- construction et gestion de parcours (outils d'autodiagnostic, logiciels de gestion de parcours, ...) ;
- bilan et validation (aide aux bilans individuels et collectifs, ...).

1.1.3.3 - Le développement des dispositifs de support pour la FAD

Le développement des dispositifs de support pour la FAD est un enjeu majeur sur le plan technologique. C'est ainsi que l'on voit se développer des plates-formes de téléformation. A ce jour, on dénombre plusieurs plates-formes dédiées d'après leurs auteurs à la e-formation et singulièrement parfois, à la formation à distance. On peut se poser la question d'un tel foisonnement dans ce domaine. La multiplicité de plates-formes existantes semble témoigner d'une tendance pernicieuse des informaticiens : chacun essaye de développer une plate-forme répondant à ses besoins spécifiques sans chercher à adapter l'une ou l'autre plate-forme existante. Il n'est pas utile de revenir sur les inconvénients de l'attitude qui consiste à

continuellement «réinventer la roue». Il est néanmoins clair que l'on épuise ainsi une quantité très importante de ressources pour arriver à un résultat qui est très souvent en dessous de ce qui existe déjà.

On peut ainsi noter, dans deux des dernières thèses dans notre domaine de recherche, que George (2001) développe un environnement pour soutenir une pédagogie à base de projet et Després (2001) développe un environnement pour le suivi d'activités en mode synchrone à l'usage des tuteurs. Sébastien George (George, 2001) motive ses travaux par le souci de mettre en place une forme de pédagogie favorisant les interactions entre les apprenants dans le contexte d'un apprentissage à distance. Pour sa part, Christophe Després (Després, 2001) développe un environnement de suivi pédagogique synchrone dans un EIAO, Roboteach. Cet EIAO a été développé pour supporter des activités de projets et des travaux pratiques mettant en jeu des micro-robots. Tous ces travaux ne conduisent pas d'emblée à des systèmes suffisamment génériques pour être couplés à d'autres plates-formes ou à des types d'apprentissage différents. Il convient néanmoins de nuancer quelque peu nos propos. Le souci de ces deux chercheurs était de simplifier la tâche du formateur et d'enrichir les conversations, on peut s'en rendre compte par l'existence d'actes de langage introductifs dans les interfaces de dialogue mises en place. Ceci signifie que la problématique de ces travaux de thèse était beaucoup axée sur le contenu de la formation et des interactions. Ce type de problématique conduit nécessairement à une plus grande contextualisation des environnements. Soulignons que Després (2001) propose un modèle de suivi d'activités d'apprentissage à distance susceptible d'être utilisé dans d'autres contextes de formation à distance. De même convient-il de préciser que dans les travaux précités, il y a certes la distance, mais cette distance existe principalement entre le formateur et la communauté d'apprenants. L'existence d'apprenants dans le groupe n'est pas primordiale, d'autant que la pédagogie de projet utilisée oblige de toutes les façons les apprenants à communiquer entre eux pour accomplir leur tâche.

Le problème du développement des dispositifs de support à la FAD peut être abordé dans une autre perspective : développer des outils suffisamment génériques pour être couplés à des plates-formes existantes et favoriser l'apprentissage collectif à distance. C'est dans cette deuxième perspective que nous nous situons. Ceci nous conduit naturellement à nous intéresser aux problèmes de normalisation des LMS et d'interfaçage des outils que nous nous proposons de mettre en place avec les LMS existantes. C'est le même type de problèmes que

le modèle SCORM (ADLnet [http](http://adlnet.org)), mis en place grâce par l'initiative ADL tente de résoudre mais concernant la caractérisation des ressources pédagogiques.

1.2 - La téléformation

Le concept de téléformation est aujourd'hui largement décliné, tout en recouvrant des pratiques et en mobilisant des acteurs dont les objectifs, bien qu'identiques en apparence (former à distance), obéissent cependant à des logiques différentes, selon le contexte dans lequel ils sont poursuivis : s'il y a évidemment dans tous les cas une démarche pédagogique, il est clair qu'elle pourra avoir, selon le cas, une finalité commerciale, scientifique ou liée à des impératifs de gestion de ressources humaines. Cette section permettra de définir synthétiquement le concept de téléformation, avec notamment quelques éléments de comparaison entre les Etats-Unis (pays ayant le plus tôt largement développé ce mode d'apprentissage) et la France. Sera ensuite esquissée une brève typologie des opérateurs de téléformation, afin de souligner que les acteurs, malgré des appellations communes, n'exercent pas exactement le même rôle selon le contexte. La dernière partie sera consacrée à la présentation de l'outil privilégié de formation à distance utilisant les TICE, à savoir les plates-formes de téléformation.

Les chiffres rapportés dans les paragraphes qui suivent concernent non pas des situations réelles de téléformation mais davantage des situations de e-formation car dans la majorité des cas, des situations de formation mixte (présentiel, présentiel/distance, centre de ressources, usage des technologies dans la formation, etc.) comme on en rencontre dans la e-formation sont indûment qualifiées de situation de formation à distance. Nous aurions voulu donner des chiffres relevant exclusivement de la téléformation, mais ceci est impossible, car les enquêtes dédiées exclusivement à l'usage de la FAD en ligne sont pratiquement inexistantes. Nous rapportons néanmoins ces chiffres pour que le lecteur se fasse une idée de l'importance et de la montée en puissance de ce secteur. Cette absence d'enquêtes sur la téléformation indique aussi à notre avis, le faible niveau de pratiques de la FAD en ligne aussi bien dans les entreprises que dans les universités. Très souvent, la téléformation est assimilée à l'introduction des technologies informatiques dans la formation, ce qui est bien entendu une idée fausse.

1.2.1 - Présentation générale

La téléformation, formation à distance « en ligne », telle qu'elle se développe actuellement est le résultat d'une adaptation aux Technologies de l'Information et de la Communication (TIC) qui ont été progressivement adoptées par la formation. Ces évolutions se sont accélérées depuis l'arrivée d'Internet. La téléformation correspond donc à tout dispositif de formation qui s'exerce par Internet, Intranet ou Extranet et a pour but de fournir un enseignement ou un apprentissage à tout individu ou groupe d'individus distants de l'organisme prestataire, du service de formation, selon des modalités souples adaptées au mieux aux possibilités de chacun des individus en formation. Ce mode d'enseignement-apprentissage qui s'appuie sur le réseau pour communiquer est de plus en plus utilisé pour favoriser l'apprentissage actif et collaboratif, mais variable selon les spécificités des contextes éducatifs. Qu'il s'agisse d'entreprises, d'organismes de formation ou d'universités, les solutions de téléformation sont guidées par un triple objectif : répondre aux besoins, présenter des avantages concrets immédiats et être ainsi acceptées par les partenaires, produire des effets permettant de justifier des investissements importants.

En France, la formation par Internet en est encore à un stade émergent. Les pratiques de téléformation sur Intranet et Internet ont commencé à se déployer en 1998. L'arrivée de l'offre de formation et de solutions, l'augmentation du nombre des utilisateurs d'Internet, l'évolution de pratiques liées à l'usage d'Internet dans de multiples domaines ; les besoins de formations plus souples et diverses, moins contraignantes, plus rapides à déployer, conciliant temps professionnel et temps personnel (notamment l'application des accords sur la réduction du temps de travail) contribueront largement encore à son développement. On peut d'ores et déjà noter l'utilisation de plus en plus répandue de vocables anglo-saxons tels que *e-learning*, l'apparition de néologismes mêlant langue française et langue anglaise ou la prononciation à l'Anglaise du préfixe e - comme dans «e-formation». Cette observation linguistique permet de mettre l'accent sur le fait que le monde anglo-saxon, et plus particulièrement les Etats-Unis d'Amérique, semble avoir une avance en matière de formation à distance fondée sur les TIC. Ils ont déjà atteint, dans le domaine, un stade de maturité et de banalisation qui permet de disposer d'une forme de recul nécessaire à l'analyse. Elle nous oblige par ailleurs à préciser que la France se situe encore, notamment dans le secteur universitaire, dans une phase que l'on pourrait qualifier d'expérimentale.

Selon une étude réalisée en avril 2000 par le cabinet Arthur Andersen (Andersen, 2000), la formation dans sa forme traditionnelle, encore très largement dominante, semble avoir encore de beaux jours devant elle en France. La formation en salle reste «la solution» pour 92% des entreprises. 88% des entreprises n'utilisent pas Internet dans le cadre de leurs formations et 78% n'utilisent pas non plus leur Intranet à cet effet. Les lieux de pratiques innovantes sont majoritairement les grandes entreprises. Mais les entreprises françaises qui utilisent la formation en ligne restent encore assez discrètes sur leurs initiatives, aussi est-il encore assez difficile de bénéficier de la capitalisation de leurs expériences.

Une enquête réalisée par l'Observatoire de la Formation, de l'Emploi et des Métiers en juillet 2002 auprès de 100 établissements (OFEM, 2002) qui pratiquent la e-formation en France, montre une progression de la pratique de la e-formation dans les entreprises par rapport à une précédente enquête menée en 2001 (OFEM, 2001) :

- 66% d'entreprises de plus de 1000 salariés utilisent la e-formation ;
- 13% d'entreprises de 500 à 999 salariés utilisent désormais la e-formation et ce mode de formation est adopté par deux fois plus d'entreprises de moins de 1000 salariés qu'en 2001 ;
- 46% des entreprises ont formé plus de 200 salariés en 2002, contre 29% en 2001.
- 6,8% des entreprises utilisaient la e-formation avant 1998, elles sont 54,4% à l'avoir introduites depuis l'an 2000 ;
- 82% des entreprises ont utilisé le tutorat en 2002, contre 70% en 2001.

S'agissant des lieux de dispense de la formation, on note qu'elle se passe pour

- 56% dans un lieu dédié de l'entreprise ;
- 43% sur le poste de travail ;
- 1% en dehors de l'entreprise.

Le cabinet d'étude IDC (International Data Corporation), prévoit une croissance du marché mondial de la e-formation atteignant 250 millions d'euros en 2004. Si l'offre est à présent bien réelle, il lui reste encore à se structurer sous de multiples aspects pour répondre aux problématiques des entreprises.

Aux Etats-Unis, le marché de la téléformation par Intranet et Internet semble avoir atteint un stade de massification aussi bien dans le secteur des universités que de l'entreprise. De 1995 à 1998, les universités américaines ont dispensé 52 270 cours à distance à 1,6 million d'étudiants (Paquette, 2002). Les universités américaines sont pour 72% d'entre elles engagées dans des offres de formation en ligne (RH Info, 2001). Pour les entreprises, le volume d'affaires réalisé par la formation sur Intranet / Internet s'élevait en 1999 à un tiers du marché total des formations utilisant les technologies. De grands organismes de prévision américains, en particulier le cabinet d'étude IDC, prévoient que le marché de l'apprentissage par les technologies dans les entreprises devrait passer devant le marché traditionnel de la formation aux alentours de 2003, et le marché de la formation en ligne devrait approcher 50% du marché en 2005. On envisage en fait une décroissance spectaculaire de la formation traditionnelle à partir de 2001 : dans le monde de l'entreprise, la formation traditionnelle en salle passerait de 80% en 2000 à 60% en 2003.

1.2.2 - Typologie des contextes d'utilisation de la formation en ligne en France

1.2.2.1 - Les sociétés spécialisées

Elles ont su, pour certaines, occuper rationnellement, méthodiquement le terrain, et acquérir parfois une notoriété ainsi qu'une reconnaissance implicite, en misant sur les concepts ayant émergé voici quelques années, tels que la formation tout au long de la vie, la formation « juste à temps », où et quand l'apprenant le veut, la formation individualisée, le soutien pédagogique, l'accompagnement, le tutorat. Il est frappant de constater la singulière disproportion entre l'audience, réelle ou annoncée, de ce type d'organisme et leur importance sur le plan logistique. Ceci semble tenir au fait que, dans un marché émergent, ces opérateurs ont su habilement se rendre en quelque sorte incontournables et apparaître comme la référence forte, y compris parfois aux yeux des institutions publiques de formation, faute probablement, pour ces dernières, d'avoir eu la même réactivité par rapport aux potentialités de la formation en ligne.

Pédagogiquement le schéma de fonctionnement est assez simple : la société dispose d'une plate-forme à partir de laquelle elle injecte des contenus, standardisés ou conçus à la demande d'un client institutionnel, et met en avant les fonctionnalités de son outil, notamment la possibilité pour l'apprenant de se construire un parcours individuel de formation et de

bénéficier d'un «tutorat» inclus dans le montant acquitté lors de la commande. Celui-ci s'exerçant, au choix, par téléphone, messagerie ou visioconférence sur ordinateur en cas de besoin, sur demande de l'apprenant.

1.2.2.2 - La téléformation en entreprise

Deux études de RH Info de décembre 2000 et Juillet 2001 sur l'état du *e-learning* (RH Info, 2001) en France font émerger quelques points essentiels concernant l'introduction de la formation en ligne dans le monde de l'entreprise. Le marché de la téléformation peut encore être considéré comme relativement inconsistant : d'après le sondage effectué en septembre 2000 par ce cabinet, 80,5% des directeurs des ressources humaines pensent que le « *e-learning* » est avant tout à envisager en complément d'approches traditionnelles. Cependant, RH info ne doute pas que le « *e-learning* » soit promis à un bel avenir : « Les gains économiques et opérationnels sont trop importants ». Sur ce secteur d'activité, l'enquête mentionne l'introduction de solutions « modernes » d'apprentissage à distance, en particulier l'autoformation individuelle, la formation en ligne avec tutorat asynchrone, et les classes virtuelles en ligne avec tutorat synchrone. Il est intéressant de noter que 73,6% des décideurs en ressources humaines affirment ne pas avoir une vision claire de l'offre actuelle en matière de « *e-learning* » (toujours d'après le sondage RH Info). Cependant, ils semblent avoir perçu que les TICE constituent une solution efficace dans l'accès au savoir, facteur de développement de l'entreprise, et à ce titre, transforment la conception traditionnelle de la formation. La gestion de la charge de formation s'en trouve elle aussi redistribuée. Les formations à distance permettent la mise en place de parcours individualisés, déterminés à l'aide d'une évaluation initiale. L'évaluation peut avoir lieu à chaque étape du parcours et non plus seulement en fin de stage. La redistribution est aussi temporelle. La formation n'impose plus la mobilisation de groupes lors de plusieurs journées, l'apprenant avance à son rythme et en fonction de ses disponibilités. Ceci a pour effet d'adapter la formation aux exigences concrètes de l'activité de l'entreprise.

Il s'agit là d'un environnement en plein essor qui, nécessairement, concerne les groupes industriels ou structures professionnelles dont l'envergure est suffisante pour justifier l'investissement de moyens humains, financiers et techniques conséquents pour la formation, en ligne, de leurs salariés. On citera le secteur bancaire, les grands groupes industriels, des sociétés issues du secteur public mais désormais inscrites dans une logique concurrentielle,

comme la Poste, EDF ou France-Télécom. Citons notamment l'expérimentation d'envergure de quatre entreprises françaises (EDF, GDF, France-Télécom et Renault) ; expérimentation réalisée en 1997-1998 et qui a consisté en la mise en œuvre de dispositifs éducatifs faisant appel aux TICE. Ces organismes tendent à développer leur propre dispositif de téléformation, autrement dit des intranets de formation. Aux Etats-Unis le phénomène se répand : de grosses entreprises mettent en place ce qu'on appelle des universités d'entreprises (*corporate university*). Deux paramètres peuvent être perçus comme conduisant au choix d'un outil propriétaire :

- le souci de maîtriser le processus de formation, le développement des compétences internes étant liées aux stratégies de développement de l'organisme ;
- mais aussi, voire surtout, la possibilité de gérer la formation de la façon la plus rationnelle possible, en adaptant son temps et son mode de mise en œuvre aux contingences propres de l'entreprise.

Il y a là une double logique, dont les deux volets se complètent : logique de valorisation des compétences des agents, objectif de fond, garant du développement de l'entreprise ; et logique économique, en adaptant le déroulement de la formation aux impératifs de gestion et de fonctionnement, avec le pari que le retour sur investissement initial se traduira, à terme, par une croissance maîtrisée des coûts de formation et des gains de productivité induits.

Il est à prévoir enfin que les entreprises travailleront en collaboration avec des organismes de formation spécialisés dans la formation en ligne, s'associeront avec des fournisseurs de téléformation ou feront appel à des éditeurs de contenus pédagogiques en ligne.

1.2.2.3 - L'université et la téléformation

Nous avons indiqué qu'aux Etats-Unis, ce mode d'enseignement est prévu comme devant dépasser à brève échéance, en volume, le secteur de la formation universitaire classique, en présentiel. Nous pourrions faire également référence aux universités canadiennes qui se sont toutes dotées d'une politique et d'une infrastructure de développement en matière de TICE. Cette révolution programmée n'est pas encore envisagée à l'identique en France, au moins à moyen terme. Mais on peut imaginer que le recours aux TICE va certainement prendre de l'ampleur et modifier le rôle et le mode de fonctionnement des institutions d'enseignement.

Des groupements d'offres universitaires se font jour (36 projets¹ ont été retenus dans le cadre du troisième appel à projets «Campus numériques français»), des alliances inter-universitaires sont en cours au plan régional. Mais pour de multiples raisons, dont la première semble de nature culturelle et sociologique (réticences diverses à la remise en cause des schémas traditionnels de formation, tant chez nombre de formateurs qu'au sein même du public étudiant, proportion encore limitée de la population ayant une utilisation régulière d'Internet), la formation en ligne n'a pas encore été pleinement adoptée.

Il n'empêche que l'université recèle d'innombrables structures de recherche et formation au sein desquelles se mène une réflexion de fond sur les potentialités aujourd'hui offertes par les environnements technologiques d'apprentissage. L'atout majeur réside en particulier dans la capacité interne à mener une double démarche dans la mise en œuvre de ce type d'environnements : démarche pédagogique, certes, mais parallèlement analyse critique par la dimension recherche qui doit accompagner l'action. Ce recul et cette observation scientifique sont ce qui fait précisément la spécificité de l'enseignement supérieur, en le différenciant des autres opérateurs nécessairement engagés dans une logique plus fonctionnelle et dictée par des impératifs d'abord économiques. S'agissant des modalités concrètes selon lesquelles peut se décliner la formation universitaire en ligne, un exemple parmi d'autres permet d'en appréhender la spécificité : il s'agit de la notion de tutorat. Il semble bien que ce terme recouvre plusieurs acceptions. Dans une logique «commerciale», on peut définir le tuteur comme la stricte personne ressource au moment où l'apprenant se trouvera en difficulté ; mais on imagine vite certaines limites : en aucun cas le tuteur ne sera en situation d'anticiper les besoins de l'apprenant, par une présence volontariste, par une démarche de veille du processus d'apprentissage. Dans une expérimentation de formation en ligne dispensée par des équipes universitaires, au contraire, on s'attachera à observer comment ce rôle de facilitateur, accompagnateur, animateur, fréquemment mis en avant dans de tels dispositifs, pourra concrètement être tenu, afin justement d'en mesurer l'apport concret en termes de valorisation du processus d'apprentissage. Certes, à grande échelle et pour des flux importants, les ambitions peuvent avoir à être tempérées, en fonction du potentiel humain déployé, mais il n'empêche que des expérimentations de cette nature apportent un éclairage irremplaçable pour évaluer les conditions requises pour le développement de dispositifs de formation en ligne.

¹ Ces trente-six projets ont été retenus parmi 82 présentés par plus de 200 établissements d'enseignement supérieur en partenariat avec 64 établissements étrangers, entreprises et associations des secteurs concernés.

1.2.3 - Les plates-formes

La FAD est, à l'évidence, désormais un domaine privilégié de mise en œuvre des TICE et elle semble s'engager dans un processus d'industrialisation à tel point d'ailleurs que l'on commence à parler d'un marché de la téléformation. Pour concilier ce déploiement industriel et la flexibilité de l'offre, des moyens techniques, des solutions logicielles adaptées, ont été imaginés et mis en œuvre ; ils concernent à la fois les ressources pédagogiques (supports multimédias), les moyens de gestion des publics et de leur progression et les moyens de communiquer avec eux. Plusieurs solutions de téléformation ont donc vu le jour, mais la solution technique qui retiendra notre attention ici est celle des plates-formes de téléformation.

1.2.3.1 - Présentation générale

Produire puis intégrer, gérer et administrer des contenus de formation et des ressources pédagogiques, diffuser et permettre l'accès à ces contenus et ressources, présenter des programmes de formation, permettre un positionnement puis un suivi d'apprenants, construire des parcours de formation individualisés, gérer ces parcours individualisés ou des parcours communs, encadrer des individus et animer des groupes (ceci a trait au tutorat), administrer financièrement ou du point de vue technique ; voilà quelques fonctions parmi celles impliquées par la formation à distance qui supposent d'avoir recours à une catégorie de logiciels spécialisés appelés plates-formes de téléformation ou encore, en anglais, *Learning Management Systems* (LMS).

Une plate-forme est un progiciel, autrement dit, un ensemble complexe de logiciels rassemblés dans un environnement cohérent à point d'entrée unique. Les logiciels internes à la plate-forme permettent en particulier de :

- mettre en ligne les ressources pédagogiques : documents ou programmes informatiques ;
- communiquer de façon asynchrone (courriels, forum de discussion) ou synchrone (bavardage, tableau blanc, etc.) ;
- gérer un agenda collectif ;
- créer des espaces de travail individuel ou des espaces collectifs ;

- utiliser des procédures spécifiques pour soumettre les travaux, les corriger, les partager avec d'autres.

Sont exclus de cette définition les aspects liés à la gestion de la formation (inscription, paiement, etc.) (Chanier, 2001b).

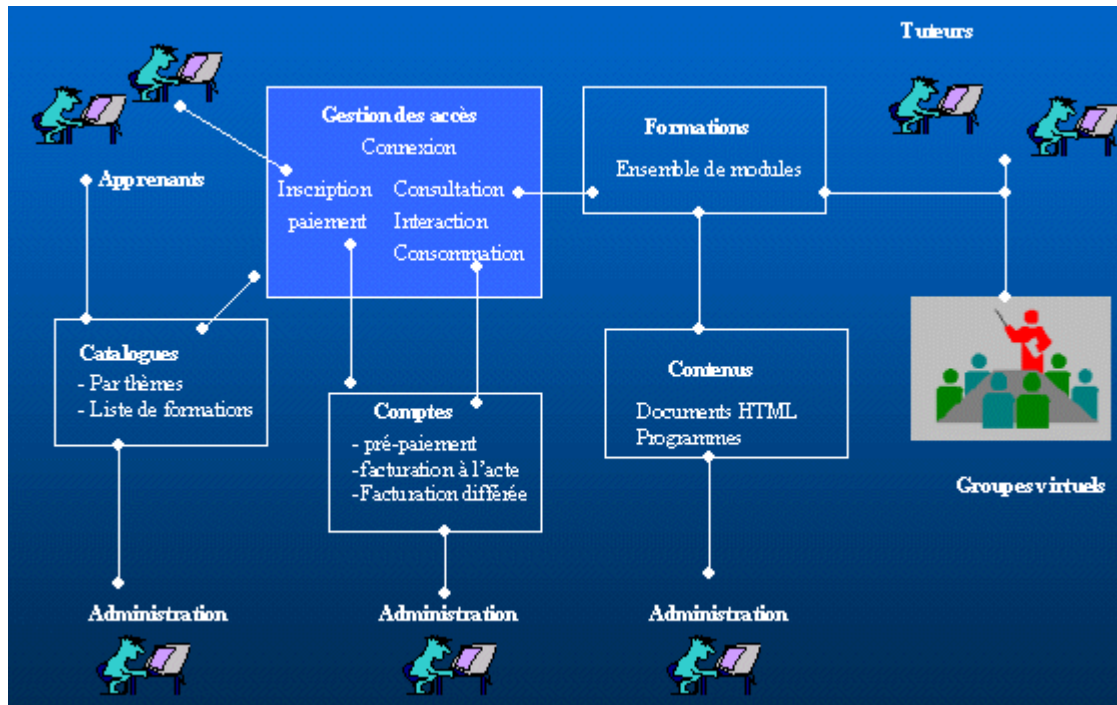


Figure 1.1 : Vue d'ensemble d'un système de téléformation

Les plates-formes sont un outil relativement récent pour la formation à distance puisque, si l'on se réfère à l'étude des propriétés des onze plates-formes les plus performantes déjà utilisées dans l'enseignement supérieur et dans le secteur de la formation professionnelle (Le Preau, 2000), la plupart des produits présentés sont apparus sous leur version commerciale dans les années 1990 pour aider les concepteurs et formateurs à mener à bien l'essentiel de leurs fonctions pédagogiques. On situe trois étapes dans le développement des plates-formes. Dans un premier temps, il s'agissait d'assurer un meilleur accompagnement de la formation à distance et de la rendre moins anonyme. Plusieurs outils différents, les systèmes de tutorat asynchrone et/ou synchrone par exemple, furent ainsi utilisés. Dans un deuxième temps, sont apparus des outils qui permettaient de gérer de plus grands nombres de stagiaires avec la même qualité d'accompagnement, et qui liaient les composantes du système (par exemple, la base de données pédagogique, les parcours pédagogiques, le tutorat) et permettaient de rendre plus efficace l'intervention des acteurs de la formation. La gestion des groupes de stagiaires suivant des formations individualisées a notamment retenu l'attention des formateurs, dirigeants, pédagogues, d'autant que cette notion est à l'ordre du jour depuis quelques années.

Dans un troisième temps, l'intégration de ces fonctions dans un système d'information et de formation unique a permis de rationaliser les différents processus : inscription, positionnement, évaluation, qui régissent les activités de formation, et de les automatiser tout en conservant une grande souplesse. Cette dernière évolution observée depuis 1998 a permis une meilleure intégration de la formation dans son environnement (en entreprise en particulier), un meilleur suivi de l'information et une économie considérable des tâches de gestion.

En résumé, nous sommes passés en peu de temps de la formation à distance à la formation en ligne, puis des plates-formes de publication et de diffusion de la formation, qui prévoyaient un nombre restreint d'intervenants (apprenant, formateur et administrateur technique), à des plates-formes de gestion des apprentissages de plus en plus sophistiquées. Nous arrivons à celles qui intègrent la gestion des connaissances et des compétences dans les organisations et qui se fondent sur des processus opérationnels de gestion des ressources humaines. Les grandes organisations ont incité d'importants éditeurs à s'intéresser à ce marché. L'offre d'outils de téléformation est aujourd'hui en pleine expansion, avec une multiplication des produits en 1998 et 1999.

Les plates-formes de téléformation ont été mises au point aussi bien en Amérique du Nord qu'en Europe. En France, certains organismes de formation se sont intéressés à ces outils. Ils ont créé des produits ad hoc ou ont acquis ceux développés par d'autres² ou par des sociétés spécialisées. Leurs expériences diverses, parfois anciennes (le CNAM, par exemple, pratique la téléformation depuis 1987) ont permis de mettre au point des méthodes et des dispositifs originaux, mais constituent un acquis encore peu exploité. Les grandes entreprises ont, elles, découvert la téléformation récemment et ont tendance à développer leur propre plate-forme Intranet. Le choix d'une plate-forme se fera en fonction de l'objectif de formation poursuivi, en fonction du type de formation ouverte et à distance que l'on souhaite mettre en place en distinguant notamment le «degré» de la distance (formation individualisée à domicile, en centre de ressources, avec séances de regroupement ou non, utilisant d'autres matériels - visioconférence..., la gestion du synchrone et de l'asynchrone dans la relation apprenant / formateur...). Il s'agira en outre de repérer l'ensemble des personnes qui seront impliquées

²Citons trois fameux exemples : Plei@d (développée par le CNAM des Pays de la Loire), Sylfide développée et commercialisée par le groupe Initiatives Développement Formation adoptée par le Réseau Miriad, et Campus Virtuel développé par le laboratoire Trigone de l'Université de Lille 1.

dans l'offre de formation (personnel administratif, techniciens de maintenance, formateurs, tuteurs, concepteurs de cours, responsables de formation, personnes des centres ressources ou d'accueil qui donnent l'accès à la formation, et apprenants), leurs compétences vis à vis de la formation individualisée à distance, comme en matière de nouvelles technologies. Devra enfin être pris en considération le fait que l'arrivée de ce nouvel outil de mise à distance de la relation triangulaire apprenant–formateur-savoir change considérablement les données de la formation. Formateurs et apprenants ont à se l'approprier ; chacun doit respectivement repenser ses méthodes d'enseignement et d'apprentissage. Il s'agit de réaliser un diagnostic des compétences requises et effectives de chacun des acteurs impliqués dans la mise en place de ce type de dispositif technico-pédagogique.

Les plates-formes ont souvent été développées pour répondre à des pédagogies directives et pour satisfaire un apprentissage conçu dans le cadre d'un parcours individuel. Cette perspective n'est cependant pas la seule envisageable. Elles permettent en effet de mettre en place une approche éducative fondée sur des tâches de production, dans des parcours de formation sans document structuré a priori, avec recherche d'informations et dialogues avec des pairs, donc dans une démarche pédagogique de construction de savoirs par l'apprenant. Elles permettent donc la création de parcours d'apprentissage différenciés adaptés à chaque apprenant. L'apprenant pourra consulter et naviguer dans des pages qui auront été publiées. Elles comportent des fonctionnalités pour la création de tests, pour la passation et la gestion de tests, pour la construction d'évaluations, pour le suivi du parcours de chaque apprenant (résultats obtenus, parcours dans la plate-forme...), pour l'élaboration et l'envoi de travaux. Dans le cadre d'un suivi personnalisé des activités d'un apprenant, elles proposent au tuteur des fonctionnalités permettant de tenir un tableau de bord. Elles autorisent des opérations de positionnement, d'évaluation et d'auto-évaluation. Elles sont dotées d'outils répondant aux fonctions de diffusion (mise en ligne du parcours de formation, de notes de cours, de matériel de référence, de modules d'apprentissage interactifs ou non, sous forme textuelle ou multimédia, de travaux d'étudiants...), aux fonctions d'évaluation formative et sommative (quiz et examens en temps réel ou différé), aux fonctions de gestion administrative (contrôle d'inscription et d'accès, liste des étudiants, suivi des dossiers individuels et collectifs, attribution et calcul des notes, production de rapports...), de gestion pédagogique (design du cheminement pédagogique individuel, statistiques individuelles et collectives de fréquentation des contenus). Elles offrent pour certaines un agenda permettant de matérialiser le calendrier des activités et leur échéancier ; cette fonctionnalité est très

importante pour nous car les activités sont toujours cadrées dans le temps avec une date de début et une date de fin. Pour favoriser les échanges entre personnes à distance, dans une démarche de socialisation favorable aux apprentissages, de mutualisation par échange de pratiques, d'entraide entre apprenants (le tuteur n'étant pas la seule personne ressource), elles proposent toutes des fonctionnalités de communication asynchrone (communication en différé) et, moins souvent, synchrone (communication en temps réel), avec ou non des possibilités de sauvegarde des discussions considérées comme source d'information. Parmi ces fonctionnalités : le forum, pour participer à une discussion publique ; la messagerie ou le courrier électronique, pour échanger des messages électroniques individuels ; un outil de bavardage ; plus rarement un tableau blanc (pour le travail en simultané sur un même document en temps réel) ; des possibilités de visioconférence ou d'audioconférence. Ces outils, auxquels nous pourrions ajouter la foire aux questions (FAQ), seront, selon les contextes d'apprentissage, plus ou moins exploités à des fins de travail coopératif ou collaboratif.

1.2.3.2 - Acteurs et fonctionnalités d'un dispositif de téléformation.

Une plate-forme de téléformation est, dans les grandes lignes, un outil qui permet à des enseignants ou formateurs de piloter un enseignement ou un parcours de formation et à un apprenant de suivre ce parcours à distance, en ligne. Il s'agit d'un environnement logiciel, spécifiquement étudié pour répondre aux besoins de la formation à distance, accessible via Internet et qui regroupe les outils nécessaires aux principaux utilisateurs d'un dispositif ayant pour finalités la consultation à distance de contenus pédagogiques, l'individualisation de l'apprentissage et le télétutorat. Comme suggéré précédemment, elles ont été conçues dans le but de servir les besoins des acteurs d'un dispositif de téléformation, notamment le formateur, le créateur de cours, le tuteur, l'administrateur, l'administrateur technique et l'apprenant. Ainsi, une plate-forme offre la possibilité de définir un dispositif de formation et s'articule autour des fonctionnalités suivantes :

- la création de cours et de plans de formation (création de parcours-type, de programmes de cours ; édition de tests)
- la gestion des documents pédagogiques : classification, indexation, administration des matériaux pédagogiques ;
- la gestion administrative de la formation (scolarité) : inscription dans la plate-forme, dans une formation ; gestion de données administratives de la formation, scolarité ;

- l'organisation du tutorat : constitution des groupes d'apprentissage ; choix des enseignants, des groupes d'étudiants et de leurs accès, de types de contenus, de modes de communication, de types de tests ; création et modification de parcours de formation individualisés ; changements ou ajouts de contenus et d'activités d'un cours, gestion d'agenda, planification des parcours de formation ; communication avec les apprenants ou entre eux par messagerie et forum ; suivi des activités d'apprenants sur la plate-forme (temps passé en ligne dans les notes de cours, dialogues, travaux déposés sur la plate-forme...) ; organisation des activités de tutorat en temps réel, synchrones ;
- l'évaluation : gestion et administration des évaluations ;
- l'apprentissage : consultation à distance de contenus pédagogiques, communication entre formateurs et apprenants et entre apprenants, individualisation des apprentissages, télé-tutorat, possibilité de rendre des travaux.

Sur ces fonctionnalités de base se greffent des fonctionnalités relatives aux référentiels de formation et aux catalogues de produits de formation, à la gestion administrative, à la gestion des ressources pédagogiques ; des rôles d'administration des matériaux pédagogiques, des rôles d'administration de la scolarité ou de la formation, etc. Cependant, même si de la structure des rôles d'une plate-forme il est difficile de déduire une architecture exacte de fonctions, et plus encore d'emplois, l'accent doit être mis ici sur le fait que ces fonctionnalités sont au service des usagers de la plate-forme.

Il conviendra donc de décrire les fonctions liées à la mise en place d'un dispositif de téléformation via l'utilisation d'une plate-forme. Il semble important de mentionner ici qu'en matière de description des activités humaines et des métiers ou emplois qui s'y rapportent, nous nous heurtons à une difficulté en ce qui concerne le vocabulaire. Cette difficulté tient à deux paramètres : d'une part le domaine ici décrit est encore récent et les fonctions ne sont pas encore stabilisées ; d'autre part, celles-ci dépendront de l'organisme dans lequel elles s'exercent ou, tout simplement de l'organisation du dispositif mis en place (elles ne se distribueront donc peut-être pas de la même manière qu'indiqué ci-après). Aussi pourront-elles prendre un sens quelque peu différent ou se confondre selon qu'il s'agira d'une entreprise, ou d'un organisme centré sur une démarche purement commerciale, ou bien davantage centré sur une démarche pédagogique, etc. Rappelons enfin que nous nous situons plutôt dans le contexte de dispositifs de formation avec accompagnement pédagogique et non

dans celui de dispositifs qui reposeraient entièrement sur l'accès libre des utilisateurs et l'autoformation. Paquette (2002) identifie cinq acteurs et rôles génériques dans un dispositif de téléformation. Nous les présentons dans le tableau suivant.

Acteur	Rôle principal	Rôles secondaires
Apprenant	Transformer les informations en connaissances	Gérer son scénario et ses activités d'apprentissage. Explorer les sources d'informations. Résoudre des problèmes Réaliser des projets. Réaliser des activités servant à son évaluation. Autoévaluer ses activités. Interagir sur le plan social avec d'autres apprenants. Débattre un sujet lors des télédiscussions. Communiquer et échanger des informations.
Présentateur	Rendre disponibles les informations pour l'apprentissage	Présenter des informations. Clarifier le contenu en réponse à des questions. Gérer des médias donnant de l'information. Analyser et évaluer le contenu des documents. Analyser le contenu de la démarche d'un autre acteur.
Concepteur	Construire, adapter et maintenir un système d'apprentissage	Analyser les besoins de formation. Modéliser les connaissances à des fins pédagogiques. Construire des scénarios pédagogiques. Rédiger les devis de systèmes d'apprentissage. Simuler et valider un devis. Concevoir des matériels pédagogiques. Mettre à l'essai un événement d'apprentissage. Concevoir un plan de réalisation du système. Décrire les processus de diffusion du système d'apprentissage. Concevoir la mise en place du système d'apprentissage.
Formateur	Faciliter l'apprentissage sur le plan pédagogique	Réaliser les diagnostics de l'apprentissage. Évaluer les travaux de l'apprenant. Motiver et orienter les apprenants. Animer des équipes ou un groupe Assister les apprenants dans l'utilisation des matériels et des ressources. Agir comme conseiller pédagogique.
Gestionnaire	Gérer les acteurs et les événements	Planifier le déroulement des activités. Diagnostiquer le déroulement des événements d'apprentissage. Décider de modifier le déroulement des événements. Diriger les opérations de diffusion. Organiser des équipes ou des groupes. Affecter des formateurs et des ressources à des groupes. Organiser la mise à l'essai du système d'apprentissage. Gérer l'évaluation des apprentissages. Gérer l'évaluation du système d'apprentissage. Administer le réseau de communication.

Tableau 1.1 : les acteurs en téléformation et certains de leurs rôles, tiré de (Paquette, 2002)

Notons simplement pour la bonne compréhension du lecteur que Paquette définit l'information comme « toutes les données extérieures aux personnes, communiquées oralement par d'autres ou médiatisées dans des matériels sur divers supports numériques,

imprimés ou analogiques » et la connaissance comme « *le résultat de toute construction mentale effectuée par une personne à partir d'informations ou d'autre stimuli* ». L'apprentissage, selon le même auteur, consistant alors à transformer des informations en connaissances. (Paquette, 2002)

1.3 – Normes et standards de la e-formation

Plusieurs travaux et études portent sur la réalisation des standards, normes et labels au niveau international et français. On peut citer la définition d'une norme ISO (*International Organisation for Standardisation*), les standards techniques tels qu'AICC, SCORM, la labélisation OPQF sur la e-formation etc. Ces efforts de normalisation et de standardisation visent à répondre aux questions relatives à l'interopérabilité des technologies et des contenus entre les plates-formes de e-formation. Les principaux enjeux de la standardisation et de la normalisation sont donc de rendre inter-opérables les techniques entre elles afin de faciliter l'usage des ressources éducatives quels que soient la plate-forme ou l'environnement technologique utilisés.

Notre intérêt pour les questions de normalisation et de standardisation vient de ce que l'un de nos objectifs majeurs est de proposer des outils logiciels suffisamment génériques qui viendraient se greffer aux LMS, le but étant d'offrir un support aux utilisateurs engagés dans des sessions de formation à distance. De ce point de vue, nous avons des préoccupations du même type que les initiateurs du modèle SCORM dans leurs efforts pour définir des standards pour la description d'objets pédagogiques et des environnements d'exécution de ceux-ci.

1.3.1 - Les travaux de normalisation et de standardisation

1.3.1.1 Au niveau international

Il convient d'entrée de procéder à une précision terminologique. En France, une différence est faite entre les normes et les standards, alors que les anglo-saxons utilisent le seul mot « standard » dans le domaine des TIC. Guidon (2002) présente des définitions dans sa note sur les standards pour la formation :

- *norme : un ensemble de règles de conformité qui sont édictées par un organisme de normalisation, comme l'ISO au niveau international.*

- *standard* : un ensemble de recommandations développées et préconisées par un groupe représentatif d'utilisateurs.

Les standards techniques sont construits à partir de spécifications élaborées par des groupes de travail (AICC, ADL, IMS, ARIADNE, Dublin Core ...). Ces spécifications sont ensuite présentées à l'IEEE (*Institute of Electrical and Electronics Engineers*) dans l'optique de les standardiser. Les spécifications sont ensuite étudiées et analysées et obtiennent l'accréditation de standard après qu'un consensus soit obtenu lors d'une réunion annuelle de l'IEEE.

L'enjeu de la normalisation des technologies de l'information pour l'éducation, la formation et l'apprentissage (ou *e-learning*) est international ; elle a été confiée à un sous-comité technique de l'ISO et est également traitée au sein d'un atelier du Comité Européen de Normalisation (CEN). Cette normalisation est faite en liaison avec les travaux menés au sein de consortiums et d'associations, notamment IEEE, AICC, IMS, ADL etc. L'IEEE s'est rapproché de l'organisation internationale de normalisation ISO et travaille dans le groupe JTC1-SC36. Les travaux de l'AICC concernent la manière dont les objets pédagogiques communiquent avec les systèmes de gestion des contenus pédagogiques ou LMS. La production de IMS concerne l'indexation des métadonnées. Le modèle SCORM développé par l'initiative ADL comprend trois éléments majeurs : métadonnées, environnement système et applicatifs, format définissant la structure d'un cours.

En 1999, l'ISO a servi de cadre à une négociation inter-étatique en vue de créer au premier trimestre 2003, les standards pour la e-formation. Les enjeux de la normalisation de la e-formation sont de disposer de normes qui permettent d'assurer l'interopérabilité et la réutilisation des ressources et des outils d'enseignement médiatisé et à distance. En sont exclus la définition des conventions culturelles, des objectifs d'apprentissage ou encore des contenus de formation spécifiques. Pour atteindre ses objectifs en matière de e-formation, l'ISO a créé en mars 2000 un comité intitulé *Information Technology for Learning Education and Training* (JTC1). La mission du JTC1 est de développer, maintenir, promouvoir et faciliter les standards des technologies de l'information requis pour :

- le développement d'outils et de systèmes de technologies de l'information ;
- la performance et la qualité des produits et des systèmes ;
- la sécurité des systèmes ;
- l'interopérabilité des produits et systèmes ;

- la formalisation des environnements et des outils ;
- la portabilité des programmes ;
- l’harmonisation du vocabulaire ;
- l’ergonomie et la facilité de navigation des interfaces.

Le JTC1 comprend 17 sous-comités et 3 groupes de rapporteurs. Parmi ces sous-comités, le SC36 est chargé de développer des standards internationaux dans le champ de l’éducation, de l’apprentissage et de la formation. Les thèmes principaux du SC36 sont les techniques collaboratives, coordonné par le Japon, le vocabulaire, coordonné par l’Ukraine, les métadonnées et le modèle de l’apprenant, coordonnés chacun par la Grande-Bretagne, le système de gestion de la formation et les identificateurs personnels simples, coordonnés chacun par les Etats-Unis, l’architecture. Le JTC1 a participé à la publication de 112 standards internationaux en 1997, 140 en 1998 et 145 en 1999.

Bien que nous n’ayons pas une connaissance exhaustive des résultats atteints par le groupe de travail sur les techniques collaboratives (c’est ce groupe qui nous intéresse le plus, en effet nous sommes davantage focalisés sur une formation à distance basée sur une pédagogie collaborative dans des groupes de petite taille), on peut néanmoins relever quelques points qui font l’objet des travaux de ce groupe (JT1SC36 [http](http://www.jtc1-sc36.org)). Des efforts sont faits pour asseoir une définition consensuelle de l’apprentissage collaboratif. Le groupe travaille sur plusieurs points devant faire l’objet d’une normalisation : traces collaboratives (format et structure de données servant de traces dans un apprentissage collaboratif), création de données (format et structure de données, normalisation des interfaces de programmes d’applications – API), définition des interfaces de logiciels (normalisation des API pour les outils utilisés dans les environnements d’apprentissage collaboratif), interfaces de données, profils des environnements d’apprentissage collaboratif (définition de format de données pour les profils d’environnements d’apprentissage collaboratif et les méthodes de référencement). Les interactions entre apprenants retiennent aussi l’attention de ce groupe de travail en ce qui concerne notamment : la modélisation des activités pédagogiques collaboratives, la modélisation du groupe et de l’apprenant, l’historique des activités du groupe. Les efforts de normalisation se situent aux niveaux conceptuel, logique et physique. Les communications entre agents logiciels sont au cœur des travaux du groupe ; c’est ainsi que l’on s’intéresse à l’identification des fonctions de ces agents logiciels, leurs domaines d’intervention et aux

protocoles de communication inter-agents. La délégation française propose quatre fonctions qu'on assignerait aux agents logiciels dans les environnements d'apprentissage collaboratifs : aide à la recherche d'informations, à la création des forums et de bavardoirs pour les apprenants ; accès à des bases de connaissances ; filtrage des contenus pédagogiques selon les préférences et le profil de l'apprenant ; assistant pédagogique de l'apprenant comme compagnon, conseiller ou expert.

Notons que l'usage d'agents logiciels est envisagé ici, seulement d'un de point de vue apprenant. Nous pensons pour notre part, que tous les acteurs engagés dans la FAD, et singulièrement le tuteur ou le formateur peuvent disposer, grâce au paradigme d'agent, d'outils capables de les assister dans la réalisation de leurs activités didactiques. C'est notre principal centre d'intérêt.

1.3.1.2 En France

La France a mis en place un lieu de concertation réunissant les différentes parties prenantes dans la normalisation de la e-formation. Une commission de normalisation a été mise en place en février 2001, avec pour but de coordonner des contributions nationales qui sont alors portées et défendues au niveau des instances internationales de normalisation. Eu égard aux enjeux du domaine, les travaux de la commission de normalisation bénéficient du soutien des pouvoirs publics, des travaux universitaires, ainsi que de l'appui des entreprises privées.

La commission de normalisation « *technologies pour l'éducation, la formation et l'apprentissage* » est le miroir français du sous-comité international SC36 ainsi que de l'atelier du Comité Européen de Normalisation (CEN), auxquels elle participe et contribue activement. La commission a créé plusieurs groupes de travail sur les métadonnées/vocabulaire ; les technologies collaboratives ; les compétences, qualification et base de cours ; les modèles pédagogiques, stratégies d'apprentissage et bases de données. Cette commission a publié un rapport d'activité en février 2002 (Gemme, 2002). Elle a mené des réflexions sur :

- les intérêts de la normalisation appliquée aux dispositifs d'enseignement ;
- les précautions à prendre dans l'utilisation des métadonnées ;
- le travail collaboratif ;

- le langage de définition des interfaces utilisateur UIML (*User Interface Markup Language*) ;
- le modèle SCORM ;
- des recommandations d'étape.

1.3.2 - Le modèle SCORM

Le Ministère de la Défense des Etats-Unis a lancé l'initiative *Advanced Distributed Learning* (ADL) en 1997. Cette initiative a pour but d'une part, de développer une stratégie mondiale d'utilisation des technologies de l'information et de la communication appliquées à l'éducation, d'autre part, de promouvoir la coopération entre les gouvernements, les universités et les entreprises pour développer des standards dans la e-formation. L'initiative ADL a défini des fonctionnalités de haut niveau pour les contenus pédagogiques comme la réutilisabilité, l'accessibilité, la durabilité et l'interopérabilité pour influencer les pratiques, promouvoir l'usage des technologies éducatives et fournir une base économique viable pour l'investissement. Nous présentons ci-après le modèle SCORM (*Sharable Content Object Reference Model*) qui implémente ces fonctionnalités.

1.3.2.1 - Description et Organisation de SCORM

Le modèle SCORM, qu'on pourrait traduire en français par "modèle de référence pour le partage de contenus", définit un modèle d'agrégation de contenus pour la pédagogie par Internet et un environnement d'exécution des objets pédagogiques. De façon simple, il s'agit d'un modèle référençant un ensemble de spécifications techniques et de directives pour réaliser les fonctionnalités définies par l'initiative ADL. Le but de l'initiative ADL en développant SCORM est aussi de mettre en liaison les opérateurs de FAD et de concilier leurs points de vue et centres d'intérêt. Ce modèle de référence a l'ambition de coordonner les technologies émergentes ainsi que les implémentations publiques ou commerciales en matière de e-formation.

Plusieurs organisations travaillent sur des aspects différents, mais complémentaires et fortement liés, de l'apprentissage via les technologies Internet. Bien que de bons résultats aient été obtenus, on note un cloisonnement des différentes approches. Certaines spécifications émergentes sont assez génériques, anticipant la large variété des implémentations issues de communautés différentes (par exemple, celles utilisant la Toile, les

céderoms, l'apprentissage par le multimédia interactif). D'autres spécifications sont trop liées aux pratiques de l'EAO (Enseignement Assisté par Ordinateur) et exigent une adaptation aux applications basées sur Internet. SCORM applique les technologies de développement de plusieurs organisations comme :

- le IMS Global Learning Consortium,
- le Aviation Industry CBT (*Computer Based Training*) Committee (AICC),
- le Alliance of Remote Instructional Authoring & Distribution Networks for Europe (ARIADNE)
- le IEEE Learning Technology Standards Committee (LTSC),

à un modèle de contenus spécifique, afin de produire des recommandations pour la mise au point d'implémentations consistantes par les industriels.

Les recommandations et spécifications techniques définies dans le modèle de référence SCORM peuvent être regroupées en deux thèmes majeurs :

- le « *Content Aggregation Model* » ou modèle d'agrégation de contenus ;
- le « *Run-time Environment* » ou environnement d'exécution des objets pédagogiques.

SCORM sépare les spécifications en plusieurs livres : on en compte trois à ce jour. Les versions futures de SCORM intégreront de nouveaux livres dans la collection du modèle de référence. La version actuelle de SCORM a été divisée en trois livres :

- Le livre 1, intitulé « *The SCORM overview* » comprend une vue d'ensemble de l'initiative ADL, la raison d'être de SCORM et un résumé des spécifications et des recommandations contenues dans les autres livres ;
- Le livre 2, intitulé « *The SCORM Content Aggregation Model* » contient la démarche à suivre pour identifier et organiser des ressources en contenus pédagogiques structurés. Ce livre fournit une nomenclature d'objets pédagogiques, décrit le modèle d'organisation des contenus (*SCORM Content Packaging*) et référence le modèle de méta-informations pour les ressources pédagogiques mis au point par l'IMS. Ce modèle est basé sur la spécification des métadonnées sur les objets pédagogiques ou LOM (*Learning Object Metadata*). La spécification LOM résulte des efforts conjoints du IMS (à travers son *Global Learning Consortium, Inc*) et de ARIADNE.
- Le livre 3, intitulé « *SCORM Run-time Environment* », contient les directives pour exécuter des objets pédagogiques, communiquer, et garder les traces dans des plates-formes de *e-formation*. Ce livre est basé sur les directives d'interopérabilité contenues

dans le CMI001 mis au point par l'AICC. ADL a collaboré avec l'AICC pour développer des spécifications communes de définition d'une interface de programmation d'applications (API) de lancement et de communication ; et également pour adopter le modèle de données mis au point par l'AICC.

1.3.2.2 - Pourquoi un modèle de référence commun

L'une des exigences clés de l'initiative ADL en matière de contenus pédagogiques est la capacité à réutiliser les composants pédagogiques dans de multiples applications, de multiples environnements sans se préoccuper des outils utilisés pour leur création. Ceci exige, entre autres, que les contenus soient séparés des contraintes d'exécution spécifiques et des systèmes propriétaires. La réutilisabilité exige aussi que les contenus aient des interfaces et des données communes. Le modèle de référence permet de se soustraire aux contraintes d'exécution et définit une interface commune et un modèle de données pour les contenus réutilisables.

Les besoins en compétences

Face à la révolution scientifique et technologique à laquelle on assiste, les organisations doivent disposer d'hommes et de femmes qualifiés pour rester compétitifs. En même temps que la technologie exige des besoins en compétences très importants, elle fournit fort heureusement les moyens d'y arriver. Avec l'émergence des TICE, on a la possibilité d'avoir une formation tout au long de la vie accessible à tous et réellement efficace. La technologie étend la formation et l'apprentissage au-delà des frontières traditionnelles des salles de classe et des campus universitaires pour atteindre les lieux de travail, les domiciles, les musées et les bibliothèques.

La nécessité d'un apprentissage individualisé

Il a été montré que la vitesse d'assimilation des individus varie d'un facteur de trois à sept (même dans une classe où les étudiants ont été soigneusement sélectionnés). En moyenne, un élève dans une salle de classe pose 0,1 question par heure, alors que, dans un enseignement individualisé, il peut lui être demandé de poser ou de répondre à 120 questions en une heure. La mise en œuvre de l'enseignement individualisé se heurte à des contraintes économiques et financières ; il est pratiquement impossible, à de rares exceptions près, de fournir un enseignant à chaque étudiant. La technologie offre alors la possibilité de mettre en œuvre

l'apprentissage individualisé à de moindres coûts avec des contenus qui restent de grande qualité.

Plusieurs études empiriques ont mis en avant l'intérêt pour l'utilisation des TICE. Ces études soulignent que l'efficacité des pédagogies basées sur les TICE ne se fera que dans la mesure où elles pourront s'adapter aux besoins de chaque individu. Contrairement à l'enseignement en présentiel (dans les salles de classe), les TICE peuvent permettre de bâtir des parcours et des contenus pédagogiques adaptés aux objectifs et surtout aux styles d'apprentissage de chaque individu.

L'efficacité de l'enseignement basé sur la technologie

Des études ont montré que l'utilisation de la technologie dans l'enseignement permet de réduire les coûts de réalisation des objectifs d'apprentissage dans une fourchette de 30 à 60 %. Ces études montrent aussi que les temps nécessaires à la réalisation des objectifs pédagogiques sont aussi réduits de l'ordre de 30 % et améliorent d'autant le pourcentage d'acquisition des compétences et habiletés des étudiants. L'impact sur les coûts de formation est immédiat. De même convient-il de garder à l'esprit les gains qu'engendrent une meilleure productivité et la réduction des temps hors du lieu de travail. A titre d'exemple, le Ministère de la Défense américain pourrait réaliser des économies de 500 millions de dollars en réduisant de 30 % les temps de formation de seulement 40 % de ses employés.

De la formation à distance vers l'apprentissage distribué avancé

L'initiative ADL, en prenant aussi bien en compte les technologies basées sur les communications synchrones qu'asynchrones, va au-delà de la conception traditionnelle de la formation à distance. Il n'est plus demandé à l'apprenant bien qu'étant physiquement séparé de ses pairs et des formateurs d'être à un endroit donné à une heure bien indiquée. Les apprenants doivent pouvoir accéder aux contenus et être guidés quels que soient le lieu et l'heure. L'initiative ADL consacre ainsi ce qu'elle appelle l'apprentissage distribué avancé.

1.3.2.3 – L'environnement d'exécution de SCORM

L'environnement d'exécution de SCORM fournit un mécanisme commun de lancement des objets pédagogiques, un dispositif commun de communication entre les objets pédagogiques et la LMS ainsi qu'un vocabulaire constituant la base de cette communication. Ces trois

aspects sont implémentés à travers le lancement, l'interface de programme d'application (API) et le modèle de données. Le modèle SCORM se préoccupe de l'interfaçage entre les LMS et les contenus (ressources pédagogiques). L'environnement d'exécution va donc spécifier la manière dont on peut lancer l'exécution d'objets pédagogiques (il s'agit des SCO et des *assets* dans la terminologie SCORM). L'utilisateur a à sa disposition une API qui permet la communication entre un objet pédagogique et la LMS. C'est au travers de cette API que la LMS sera par exemple informée de l'état de la ressource pédagogique (par exemple : initialisée, terminée, dans une condition d'erreur). Cette disposition n'est pas utile pour les *assets* car ce type de ressources pédagogiques n'a aucune interaction avec la LMS. L'environnement d'exécution fournit un modèle de données communes à la LMS et à l'objet pédagogique (SCO) afin de définir un vocabulaire commun pour la communication entre ces dernières.

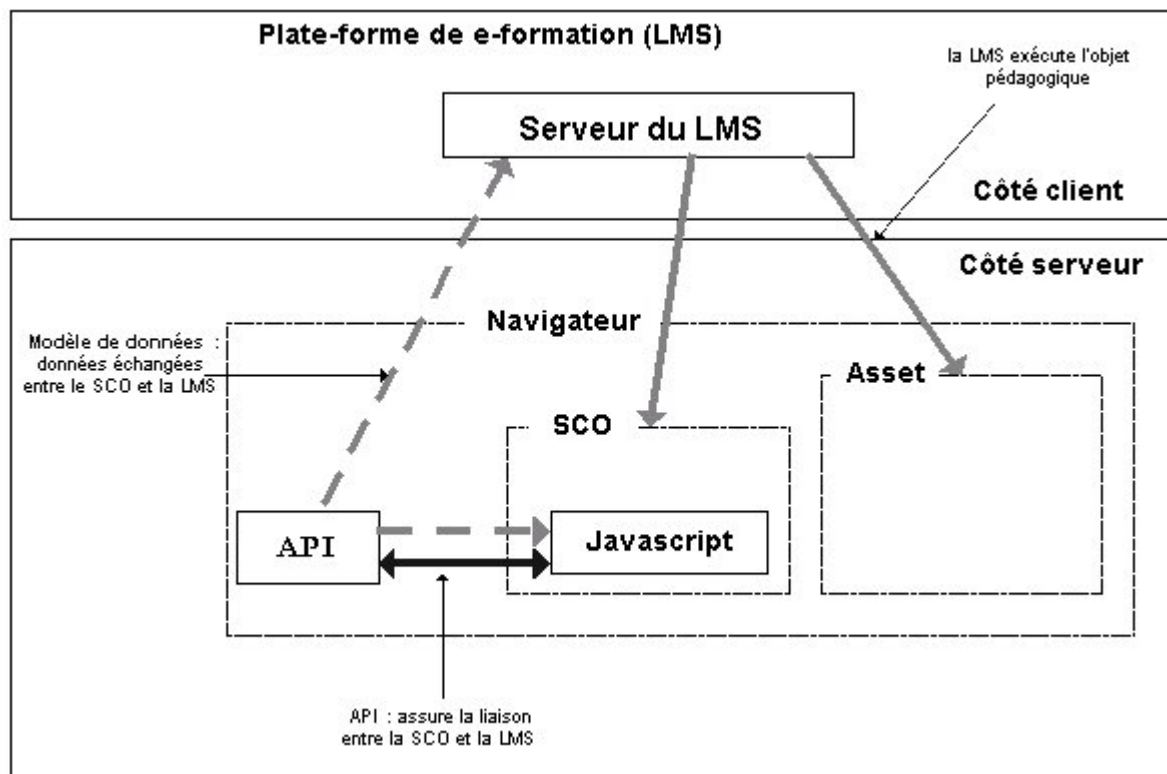


Figure 1.2 : L'environnement d'exécution de SCORM

Le fonctionnement de l'environnement d'exécution de SCORM est illustré sur la figure 1.2. L'interfaçage entre les LMS et les contenus (ressources pédagogiques) est assuré par un mécanisme de lancement des ressources pédagogiques, et pour celles des ressources ayant besoin de communiquer avec la plate-forme (SCO), SCORM définit un modèle de données et une API permettant de faire transiter les informations entre les deux entités. C'est le même

type de questionnement que nous aurons concernant le système multi-agents que nous proposons de greffer aux plates-formes de formation. Puisque nous le voulons suffisamment génériques pour être couplé à tous types de LMS, il s'agira donc de répondre à la question de la définition d'une telle interface.

Chapitre 2 : Fondements théoriques

Nous présentons dans ce chapitre, les principaux fondements théoriques sur lesquels nous nous sommes appuyés pour atteindre nos objectifs de recherche. Au nombre de ces fondements, nous pouvons citer en premier lieu l'apprentissage collaboratif. C'est ce concept qui est à la base du type de pédagogie mis en place au cours de l'expérimentation de formation à distance que notre équipe de recherche a conduite. L'apprentissage collaboratif est intimement lié aux interactions intra-groupes en ce sens qu'il est générateur d'interactions dans le groupe, et que les interactions témoignent précisément de l'effectivité de ce type de pédagogie. Le tuteur étant en charge de la gestion du groupe et devant s'assurer de son existence et de sa vitalité, nous évoquerons le rôle particulier de cet acteur central. Nous présentons ensuite la théorie de l'activité (TA) qui nous aura permis de comprendre la structure d'une activité dans une session de FAD telle que celles faisant l'objet de notre expérimentation. Nous situons également l'échelle d'analyse de la tâche dans nos travaux. Nous avons eu recours au paradigme d'agent pour construire un système capable d'assister les utilisateurs en situation d'apprentissage collaboratif distribué. Nous présentons les raisons d'un tel choix dans ce chapitre et nous nous situons par rapport à l'approche dite par composants. Enfin, nous terminons ce chapitre par la présentation d'une terminologie afin que le lecteur puisse comprendre le sens que nous donnons aux différents termes et notions utilisés dans la suite du présent mémoire.

2.1 - Apprentissage collaboratif

L'apprentissage collaboratif reste une notion très peu consensuelle quant à sa définition. L'ambiguïté qui existe entre les notions de collaboration et de coopération n'est pas pour simplifier les choses. Nous allons dans la présente section caractériser les recherches concernant l'apprentissage collaboratif. Nous espérons que cette caractérisation, au-delà de toute définition, permettra de mieux cerner ce que l'on entend par apprentissage collaboratif.

Le choix délibéré de ne pas partir d'une définition de l'apprentissage collaboratif est dû à l'absence de consensus de la part des différents chercheurs qui y travaillent. Toutefois, il faut reconnaître que l'absence d'une définition poserait quelques problèmes à savoir d'une part, qu'il serait incongru de parler des effets cognitifs des situations d'apprentissage collaboratif si toute situation peut être qualifiée de collaborative ; et d'autre part, qu'il serait alors très difficile de recenser les contributions de différents auteurs si chacun d'eux utilise le terme différemment des autres. Examinons la définition de travail rapportée par Dillenbourg (1999)

selon laquelle l'apprentissage collaboratif désigne « une *situation* dans laquelle *deux* ou *plusieurs* personnes *apprennent* ou essayent d'apprendre quelque chose *ensemble* ». Chaque élément de cette définition peut être interprété de différentes façons :

- « deux ou plusieurs » peut être entendu comme une paire, un petit groupe (3 à 5 sujets), une classe (20 à 30 sujets), une communauté (quelques centaines ou milliers d'individus), une société (quelques milliers ou millions de personnes)... et tous les niveaux intermédiaires ;
- « apprendre quelque chose » peut être entendu comme « suivre un cours », « étudier un élément précis d'un enseignement », « réaliser des activités d'apprentissage comme la résolution de problèmes », « apprendre tout au long de la vie à partir du vécu quotidien » etc.
- « ensemble » peut être interprété comme différentes formes d'interaction : en présentiel ou à travers les ordinateurs, de façon synchrone ou non, fréquemment ou pas, avec un effort commun ou à travers une division systématique du travail.

Ces trois éléments définissent le champ couvert par le terme apprentissage collaboratif.

2.1.1 - La variété des échelles

Les recherches sur l'apprentissage collaboratif se déroulent sur une échelle très variable aussi bien en termes de taille de la population des sujets concernés (de 2 sujets à plusieurs) que de temps (de 20 minutes à plusieurs années). Plusieurs études empiriques (Webb, 1991) (Webb, 1989) sur l'effectivité de l'apprentissage collaboratif concernent des populations de 2 à 5 sujets collaborant pendant une heure. A l'opposé, il existe aussi des expérimentations concernant des groupes de 40 sujets qui suivent un cours pendant une année (Gilly, 1989). Les résultats des premières études ne sont pas généralisables pour prédire ceux des seconds. La généralisation des résultats est rendue impossible non seulement à cause de la variété d'échelle mais aussi par les théories sous-jacentes : la psychologie fournit des outils pour analyser l'apprentissage dans les petits groupes, la psychologie sociale permet d'observer des groupes moyens et les outils émanant de la sociologie, de l'ethnologie et de l'anthropologie s'appliquent à de grands groupes. Il existe plusieurs écoles de pensée au sein même de la psychologie ; chacune se place à une échelle d'observation différente.

La notion d'échelle d'observation aura été le «mur de Berlin» de l'apprentissage collaboratif, en segmentant le domaine. Ce mur est tombé à la fin des années 1980. Perret-Clermont, Perret

& Bell affirment qu'établir une distinction entre le social et le cognitif est une erreur car la causalité entre les processus sociaux et cognitifs est circulaire et certainement très complexe (Dillenbourg, 1999). Cette évolution est illustrée par l'émergence des théories de la cognition distribuée dans lesquelles le groupe est vu comme un système cognitif.

Les concepts théoriques sont utilisés transversalement sur plusieurs échelles. Hutchins (1991) décrit ainsi les processus de groupe avec les termes utilisés par les spécialistes de sciences cognitives pour décrire la cognition individuelle. Réciproquement, la notion de culture utilisée traditionnellement au niveau de la communauté ou de la société est appliquée pour décrire les fondements communs construits par des pairs dans l'interaction. Il convient de signaler que durant le transfert des concepts d'une échelle à l'autre, ces derniers subissent des déformations : dans les recherches en CSCL (*Computer-Supported Collaborative Learning*), la mémoire de groupe est souvent réduite à la mémoire de travail et à la mémoire interactive ; de même, la culture construite par deux sujets au bout d'une heure d'interaction a une acception fonctionnelle au lieu de son acception historique et traditionnelle.

De la même façon que la cognition distribuée traite le groupe comme un système cognitif, on pourrait percevoir un individu comme un système distribué (Minsky, 1987). Ceci amène à percevoir la pensée comme le résultat d'un dialogue avec soi-même. Cette idée n'est pas nouvelle et a été avancée par Piaget (1965, 1928) et Vygotsky (1962, 1978).

2.1.3 - La variété des significations données au mot «apprentissage»

La littérature concernant l'apprentissage collaboratif donne plusieurs significations au terme d'apprentissage. Selon Dillenbourg (1999), certains chercheurs considèrent que l'apprentissage concerne plus ou moins toute activité collaborative dans un contexte d'éducation alors que pour d'autres, l'activité d'apprentissage est perçue comme la résolution commune de problèmes et l'apprentissage est censé apparaître comme un effet de bord de la résolution de problèmes. Toujours d'après le même auteur, nombre de chercheurs examinent l'apprentissage collaboratif sous l'angle de perspective développementale comme un processus biologique et/ou culturel qui apparaît au fil des ans. On a aussi une catégorie d'études qui examinent l'apprentissage à partir du travail collaboratif, cet apprentissage se réfère à l'acquisition des connaissances ou de compétences qui se fait tout au long de la vie dans des communautés professionnelles.

On s'aperçoit que le dénominateur commun à toutes ces situations d'apprentissage est davantage le terme "collaboration" que "apprentissage". Ceci amène à donner deux acceptions à la notion d'apprentissage collaboratif : s'agit-il d'une méthode pédagogique ou d'un processus psychologique ? La première acception est prescriptive : *l'on demande à deux ou plusieurs personnes de collaborer* car on espère qu'ils apprendront plus efficacement. La deuxième acception est descriptive : *l'on observe que deux ou plusieurs personnes ont acquis des connaissances* et la collaboration est vue comme le mécanisme ayant permis l'apprentissage. La confusion entre les points de vue descriptif et prescriptif est génératrice d'exagérations dans l'effectivité de l'apprentissage collaboratif.

Dillenbourg (1999) affirme que l'apprentissage collaboratif n'est ni un mécanisme, ni une méthode. Selon lui, il ne s'agit pas d'un mécanisme sinon il faudrait parler d'apprentissage par solitude comme on parlerait d'apprentissage par collaboration ! En effet, de même qu'un individu isolé n'apprend pas parce qu'il est seul mais parce qu'il réalise des activités (lire, construire, prédire, ...) qui déclenchent des mécanismes d'apprentissage (induction, déduction, compilation, ...), des pairs n'apprennent pas parce qu'ils sont deux mais parce qu'ils réalisent des activités qui déclenchent certains mécanismes d'apprentissage. Les activités collectives incluent des mécanismes individuels, car chacun des pairs reste un système cognitif individuel. Toutefois, l'interaction entre les deux apprenants génère d'autres activités (explications, désagréments, régulations mutuelles) qui déclenchent d'autres mécanismes (contenu des connaissances, internalisation, réduction de la charge cognitive, ...). L'apprentissage collaboratif se préoccupe justement de ces activités et des mécanismes qui apparaissent plus fréquemment dans les situations de collaboration. Toutefois, on n'a aucune garantie que ces mécanismes apparaissent dans des interactions collaboratives ou qu'ils soient circonscrits à ces interactions. L'apprentissage collaboratif n'est pas non plus une méthode du fait qu'il n'est pas garanti que certains types d'interaction apparaissent durant la collaboration. Il s'agit tout au plus d'un contrat didactique ou social entre des pairs ou entre des apprenants et l'enseignant.

Henri et Lundgren-Cayrol (2001) écrivent que l'apprentissage collaboratif propose une démarche active centrée sur l'apprenant, qui se déroule dans un environnement où il peut exprimer ses idées, articuler sa pensée, développer ses propres représentations, élaborer ses structures cognitives et faire une validation sociale de ses nouvelles connaissances. La

définition proposée par ces deux auteurs nous intéresse d'autant plus qu'elle se place dans le contexte de l'apprentissage collaboratif à distance et définit le rôle du formateur (tuteur) et celui du groupe. Nous reviendrons sur cette définition plus loin.

En fin de compte, l'apprentissage collaboratif décrit une situation dans laquelle certaines formes d'interaction (capables de déclencher des mécanismes d'apprentissage) sont attendues sans garantie qu'elles aient lieu. Une des préoccupations est d'ailleurs d'augmenter la probabilité d'apparition de ces formes d'interaction. Les différentes solutions pour y arriver peuvent être regroupées en quatre catégories :

- *La mise en place de conditions initiales* : la taille optimale des groupes, la constitution des groupes selon certains critères ou non, la mixité apparaissant dans les groupes, l'uniformité ou la diversité des points de vue des membres du groupe, le choix des tâches ou activités à effectuer, le choix des outils d'interaction. Plusieurs études (Dillenbourg et al., 1995) ont pris en compte ces différents paramètres et il apparaît qu'il existe une imbrication fort complexe entre ces paramètres ; il est dès lors très difficile de mettre en place des conditions initiales qui garantissent l'apparition des formes d'interaction souhaitées.
- *La mise en place d'un scénario de collaboration basé sur le rôle des différents acteurs* : cette approche tend à transformer l'apprentissage collaboratif en une méthode. Plusieurs méthodes sont ainsi basées, soit sur les différences entre les apprenants pour déclencher des interactions conflictuelles, soit sur la complémentarité des profils qui génèrent des interactions riches : (1) demander aux apprenants de jouer des rôles spécifiques en exprimant des points de vue même s'ils ne sont pas les leurs ; (2) donner des points de vue différents sur un sujet ; (3) contrôler l'accès aux données afin que les différents membres d'un groupe aient accès à des données différentes.
- *Soutenir la production des interactions en incluant les règles d'interaction dans l'environnement* : on peut ainsi avoir des interfaces semi-structurées avec des boutons d'interaction prédéfinis comme « Etes vous d'accord ? », ou encore avec des introducteurs d'actes de langage comme : « Je pense que ... » ; l'enseignant peut aussi établir des règles comme : « Chaque personne devra donner son opinion sur le sujet ».
- *Maîtriser et réguler les interactions* : ici le rôle du tuteur ou du modérateur est très important dans le succès de l'apprentissage collaboratif. Sa charge de travail

augmente avec le groupe. D'aucuns préfèrent parler de facilitateur plutôt que de tuteur car ici, le but n'est pas de donner la bonne réponse ou de porter des jugements sur la connaissance mais plutôt d'animer et de favoriser la production d'interactions dans le groupe ou d'évaluer quelle fraction du groupe est en dehors des interactions. Il est important dans les CSCL de disposer d'outils d'analyse et d'évaluation des interactions survenant à différents lieux et à différentes périodes. La mise au point de ces outils reste un enjeu d'actualité pour les recherches dans le domaine des CSCL (George, 2001), (Després, 2001). Une autre alternative consiste à doter les apprenants d'outils de réification de leur propre participation ou d'aide à la régulation de leurs interactions. Le paradigme d'agent avec entre autres, la métaphore d'assistant personnel qu'il renferme, constitue une piste de recherche prometteuse pour la mise au point de tels outils, c'est l'un de nos objectifs de recherche.

2.1.4 - La variété des significations données au mot : collaboration

La collaboration concerne quatre aspects de l'apprentissage :

- Une situation peut être jugée plus ou moins collaborative : la collaboration est ainsi susceptible d'apparaître davantage entre des personnes ayant le même statut qu'entre un patron et son employé ou entre un formateur et un apprenant.
- Les interactions apparaissant entre différents membres d'un groupe peuvent être plus ou moins collaboratives : une négociation contient davantage de collaboration que l'émission de directives ou d'instructions.
- Certains mécanismes d'apprentissage sont plus ou moins porteurs de collaboration : l'argumentation catalyse davantage de collaboration que la déduction.
- Les effets de l'apprentissage collaboratif permettent, non pas afin de définir la collaboration, mais d'arriver à un consensus sur la façon d'évaluer les effets de l'apprentissage collaboratif. Les divergences observées dans cette évaluation participent à la pauvreté terminologique qui caractérise le domaine.

Intuitivement, une situation est qualifiée de collaborative si elle met en œuvre des personnes (i) ayant plus ou moins le même statut et pouvant accomplir les mêmes actions, (ii) ayant un objectif commun, (iii) travaillant ensemble. Les deux premiers points concernent le degré de

symétrie dans l'interaction : symétrie d'action, symétrie de compétences, symétrie de statut. Il est à noter que chaque type de symétrie peut être objectif ou subjectif et que celle-ci évolue au cours du temps. Le second critère caractéristique de la collaboration est le partage d'objectifs communs alors que la compétition ou la concurrence suppose des objectifs conflictuels. Cette caractérisation n'est toutefois pas partagée par l'ensemble des chercheurs. Le troisième critère caractéristique concerne la division du travail entre les membres du groupe. Les termes collaboration et coopération sont parfois utilisés comme des synonymes ; cependant, d'autres auteurs leur donnent des significations différentes suivant le degré de division du travail. Dans la coopération, les partenaires reçoivent chacun une partie du travail, l'accomplissent individuellement et rassemblent les résultats partiels en un résultat global ; dans la collaboration, tout le travail se fait ensemble. Cependant, on observe parfois une division du travail spontanée même lorsque les partenaires travaillent ensemble, certains par exemple s'occupant des tâches de bas niveau pendant que d'autres se préoccupent des aspects plus stratégiques.

Une autre façon de caractériser la collaboration concerne le degré de collaboration induit dans les interactions. Ici on peut définir plusieurs critères pour le faire : l'interactivité, la synchronicité et la négociation. Il est intuitif de penser qu'une situation collaborative se caractérise par un haut degré d'interactivité. L'interactivité n'est pas définie par rapport à la fréquence des interactions, mais plutôt par la façon dont ces interactions influencent les processus cognitifs des pairs. Il est aussi intuitif de penser que "travailler ensemble" implique une communication synchrone alors que la coopération est souvent associée à une communication asynchrone. Les interactions collaboratives se caractérisent également par la négociation qu'elles induisent entre les pairs : il est clair que contrairement à une situation hiérarchique, aucun des pairs n'imposera son point de vue sur la base unique de son autorité mais devra le justifier, l'expliquer, négocier et essayer de convaincre les autres.

On peut observer deux catégories de mécanismes d'apprentissage, ceux qui sont propres aux systèmes cognitifs individuels et ceux apparaissant dans l'apprentissage collaboratif. Il convient encore une fois de noter que les mécanismes inhérents aux systèmes cognitifs individuels ne disparaissent pas dans l'apprentissage collaboratif car la collaboration ne supprime pas les systèmes cognitifs individuels. Dans la première catégorie, on peut citer : l'induction, la charge cognitive, l'auto-explication, le conflit. Dans la deuxième catégorie, on a : le processus d'internalisation, l'appropriation, la modélisation (mutuelle) du partenaire.

Tous ces mécanismes sont porteurs à des degrés divers de collaboration et plusieurs chercheurs ont mené des études en vue de prendre en compte l'un ou l'autre critère (Schwartz, 1995).

Plusieurs recherches essaient d'évaluer les effets de l'apprentissage collaboratif, très souvent à travers des tests individuels de pré et de post formation (Doise et Mugny, 1984), (Blaye, 1998). Certaines recherches se focalisent sur les changements conceptuels alors que d'autres s'intéressent davantage à l'autorégulation. La prise en compte de l'une ou l'autre approche caractérise aussi des méthodologies différentes. La première approche méthodologique pourrait se caractériser par les effets obtenus. L'apprentissage collaboratif apparaît dans une variété de contextes et d'interactions, c'est pourquoi il convient de préciser les types d'interaction que l'on souhaite étudier ; il serait illusoire de penser pouvoir étudier toutes les interactions qui apparaissent dans les situations d'apprentissage collaboratif. La seconde approche méthodologique concerne le mode d'évaluation, ici les effets sont évalués par la mesure des performances individuelles. D'autres ont avancé qu'un meilleur critère aurait été la mesure des performances du groupe.

2.2 - Le groupe et les interactions

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 point focal de toute 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. Nous pensons que ceci peut contribuer significativement à maintenir un niveau de motivation de l'apprenant suffisant. Il est à ce niveau fort intéressant de citer la définition entière que Henri et Ludgren-Cayrol (2001) proposent de l'apprentissage collaboratif après avoir examiné les apports de différentes

théories (constructivisme, psychocognitivism, sociocognitivism, flexibilité cognitive, cognition répartie, andragogie) :

« L'apprentissage collaboratif est une démarche active par laquelle l'apprenant travaille à la construction de ses connaissances. Le formateur y joue le rôle de facilitateur des apprentissages alors que le groupe y participe comme source d'information, comme agent de motivation, comme moyen d'entraide et de soutien mutuel et comme lieu privilégié d'interaction pour la construction collective des connaissances. La démarche collaborative reconnaît le caractère individuel et réflexif de l'apprentissage de même que son ancrage social en le rattachant aux interactions de groupe. En fait, la démarche collaborative couple deux démarches : celle de l'apprenant et celle du groupe.

L'apprenant s'engage à travailler avec les membres du groupe en vue de la réalisation du but commun tout en conciliant ses intérêts et ses objectifs personnels. Il collabore dans le cadre des interactions de groupe en partageant ses découvertes. Les échanges avec le groupe et la réalisation d'une tâche collective lui permettent de partager ses découvertes, de négocier le sens à donner à son travail et de valider ses connaissances nouvellement construites. Dans sa démarche, il fait preuve d'autonomie et assume la responsabilité de son apprentissage tout en se sentant responsable de l'atteinte du but qu'il partage avec tous.

Le groupe est un catalyseur de l'apprentissage. En se donnant un but, en s'engageant dans la réalisation collective d'une tâche, le groupe, en tant que groupe, apprend et construit des connaissances. Ce faisant, il permet à chaque apprenant de se confronter aux autres et il nourrit les apprentissages de chacun. Dans la démarche collaborative, les apprenants collaborent aux apprentissages du groupe et, en retour, le groupe collabore à ceux des apprenants. » (PP 42 & 43).

Cette définition fait bien ressortir la place de l'apprenant, du formateur, et le rôle du groupe et des interactions dans l'apprentissage collaboratif.

Haythornthwaite (1999) note que dans les environnements informatiques d'apprentissage collaboratif, il est difficile d'évaluer le niveau d'interactions et de communications entre les individus. Ceci est assez surprenant car on sait bien que dans les plates-formes, toutes les

interactions humain-machine sont enregistrées dans des 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 restituer le volume et les types d'interactions dans le groupe sont inexistantes et sont encore à construire.

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.

Plusieurs approches théoriques caractérisent les recherches en apprentissage collaboratif dont l'approche socio-cognitive et l'approche de la cognition distribuée se focalise davantage sur le groupe (Henri et Ludgren-Cayrol, 2001), (Dillenbourg et al., 1996). L'environnement et plus particulièrement le contexte social sont partie intégrante de l'activité d'apprentissage. Il est à présent démontré que les processus cognitifs et sociaux sont intimement liés, 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.

De manière générale, 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.

2.3 - Le rôle du coordonnateur et du tuteur

Nous parlons du coordonnateur et du tuteur eu égard à la spécificité de notre expérimentation de FAD. En effet, il avait été mis dans cette expérimentation une tutelle à deux niveaux. Les tuteurs assuraient la tutelle de leurs apprenants au sein de leurs groupes respectifs. Un coordonnateur de la formation assurait la tutelle des tuteurs et avait une vue d'ensemble sur toute la formation. Ceci ne sera vraisemblablement pas le cas dans toutes les autres situations de FAD où toutes les fonctions tutorielles du groupe sont jouées par une seule personne, le tuteur.

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 coordonnateur 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 coordonnateur de la formation peut être amené à redistribuer des groupes, ou même, lorsque ceci est nécessaire, à supprimer des groupes ou à en 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 en volume suffisant. Plusieurs recherches portent sur les moyens de créer et/ou 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. Nous ne revenons pas ici sur la définition de l'apprentissage collaboratif proposée par Henri et Ludgren (2001) qui met en exergue le rôle du tuteur, du groupe et des interactions. 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, le tuteur peut jouer quatre rôles différents vis-à-vis de ses apprenants (Charlier et al., 1999) :

- 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 partie 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 la disposition de ces derniers des systèmes capables de les assister efficacement dans leurs activités (Labat, 2002).

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 lors de notre expérimentation de FAD :

- 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.

Une étude menée par Ecoutin (2001) précise les fonctions du tutorat. Les fonctions suivantes sont listées :

- 1°) Organiser les apprentissages en pilotant les parcours, en établissant un agenda, en animant des sessions de tutorat, synchrone ou asynchrone, individuel ou collectif, en prenant l'initiative de créer des sessions.
- 2°) Accompagner les apprenants en suivant leur avancement, en répondant à leurs questions, en relançant les apprenants, en abordant des points particuliers dont la nécessité se fait sentir, en les aidant à définir des projets personnels.
- 3°) Réaliser des opérations de bilan, validation des acquis.
- 4°) Evaluer les apprentissages en corrigeant les travaux des apprenants, gérant les outils d'évaluation automatique, donnant une image de l'avancement du groupe quand celui-ci existe.
- 5°) Concevoir des parcours, des éléments de parcours ou des documents complémentaires.
- 6°) Coordonner son travail avec celui d'autres tuteurs ou formateurs en fixant des règles d'organisation, en se faisant le relais des remarques et évaluations que les apprenants formulent.

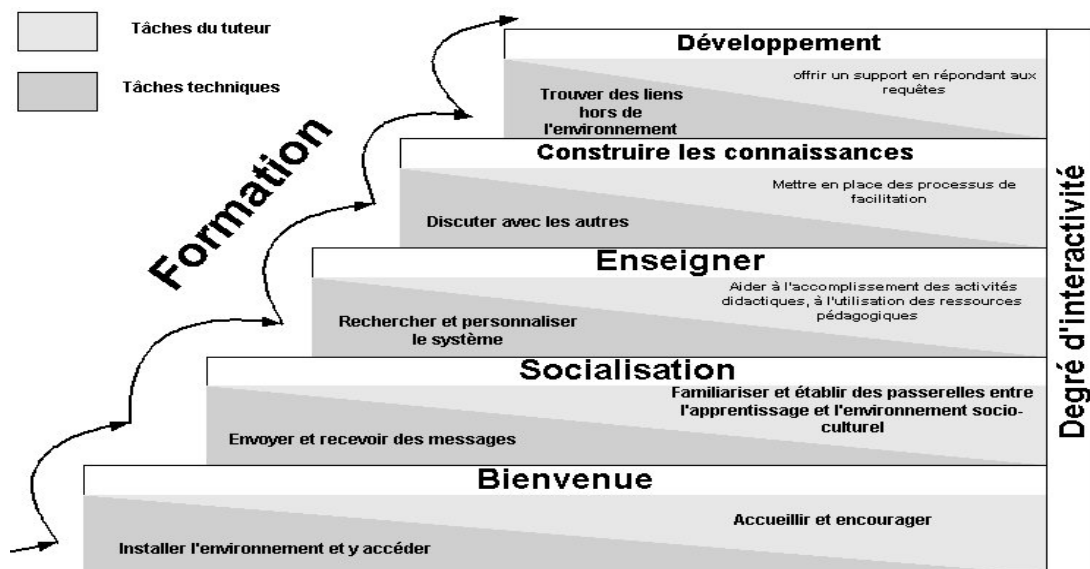


Figure 2.1 : Le modèle de Salmon pour le rôle des tuteurs en ligne, tiré de Salmon (2000)

Les fonctions 1^o), 2^o) et 4^o) ci-dessus sont aussi rapportées par Salmon (2000) dans son ouvrage sur la fonction du tuteur en ligne. Dans le même livre, Salmon propose un modèle d'enseignement et d'apprentissage en ligne à travers les forums. Ce modèle est présenté à la figure 2.1.

Dans le modèle de Salmon, le tuteur a des rôles définis à chaque étape, et ces rôles apparaissent avec l'augmentation du degré d'interactivité dans le groupe. Ces rôles présupposent néanmoins un certain nombre d'habiletés techniques de la part des apprenants. Ainsi, la première étape est celle de l'accessibilité et de la motivation : le tuteur doit alors accueillir les apprenants et les encourager. Ceci suppose que la plate-forme soit correctement installée et que l'apprenant puisse y accéder. La deuxième étape est celle de la socialisation : pour que cette phase puisse se tenir, l'apprenant doit pouvoir envoyer et recevoir des messages, le tuteur étant chargé d'établir des liens entre les aspects liés à l'apprentissage proprement dit et l'environnement socio-culturel. La troisième étape concerne les échanges d'information : le tuteur doit alors aider l'apprenant à utiliser les ressources pédagogiques et à réaliser ses activités didactiques. La quatrième phase est celle de la construction des connaissances : le tuteur met en place les processus qui facilitent cette construction. La dernière phase est celle du développement : le tuteur offre un support aux apprenants en répondant à leurs sollicitations.

Il apparaît évident que le rôle du tuteur 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.

2.4 - La théorie de l'activité

Les éléments de la théorie de l'activité présentés ici sont tirés principalement de la thèse de Bourguin (2000). La TA nous permettra d'examiner les activités pédagogiques de notre expérimentation de FAD et de bâtir dans le chapitre suivant, un modèle conceptuel d'activité.

La TA est un courant des sciences humaines qui profite d'une large audience dans tous les domaines mettant en œuvre les interactions humain-machine, comme c'est le cas dans la FAD médiatisée par ordinateur. Elle possède des fondations dans l'école historique et culturelle

soviétique de la psychologie fondée par L. Vygotsky qui s'est principalement focalisé sur la médiation par le langage, alors que la théorie de l'activité se concentre sur la médiation par l'outil. Le père de la TA est A. Leontiev, qui poursuivit les travaux de Vygotsky d'abord en tant qu'étudiant, puis en tant que collègue. Progressivement, la TA s'est révélée être un corps de concepts dont le but est d'unifier la compréhension de l'activité humaine en fournissant les ponts vers les autres approches provenant des sciences humaines. En effet, les sciences humaines ont toujours souffert d'une dichotomie entre l'individuel et le social. Si les sciences sociales utilisent le système social comme unité d'analyse, elles ont des difficultés à considérer l'individu lui-même. Les sciences du comportement, quant à elles, étudient les actions individuelles et ont des difficultés à en considérer le contexte. La solution offerte par la TA est la définition d'un concept intermédiaire, un contexte minimal pour l'étude des actions individuelles, qui sera considéré comme unité basique d'analyse : le concept d'activité. Elle offre un ensemble de perspectives sur l'activité humaine ainsi qu'un ensemble de concepts pour la décrire. Plus précisément, la TA est un cadre conceptuel pour étudier différentes formes de pratiques humaines en tant que processus développementaux, combinant à la fois et en même temps les niveaux individuels et sociaux. En d'autres termes, l'unité fondamentale d'analyse de la TA est l'activité humaine qui est définie comme un système cohérent de processus mentaux internes, d'un comportement externe et de processus motivationnels qui sont combinés et dirigés pour réaliser des buts conscients.

2.4.1 - Les niveaux d'une activité

L'activité peut être divisée dans une structure hiérarchique comprenant 3 niveaux : l'activité, l'action, et l'opération. Cette structure, représentée dans la Figure 2.2, décrit l'activité (niveau le plus haut) comme étant réalisée au travers de chaînes d'actions (niveau intermédiaire) elles-mêmes réalisées au travers d'opérations (niveau le plus bas) – même si pour certains, d'un point de vue plus proche de la psychologie cognitive, les opérations peuvent elles-mêmes être divisées en blocs de fonctions.

Les actions peuvent être individuelles ou collectives mais sont toujours dirigées vers un but conscient (intermédiaire par rapport à ce qui motive l'activité). Il est difficile, voire impossible, de comprendre une action en la considérant en dehors de son contexte, c'est-à-dire de son activité. Cependant, une action est souvent poly-motivée, c'est-à-dire qu'une seule

et même action peut appartenir à plusieurs activités. De ce fait, l'action effectuée peut être interprétée différemment suivant l'activité dans laquelle on la considère.

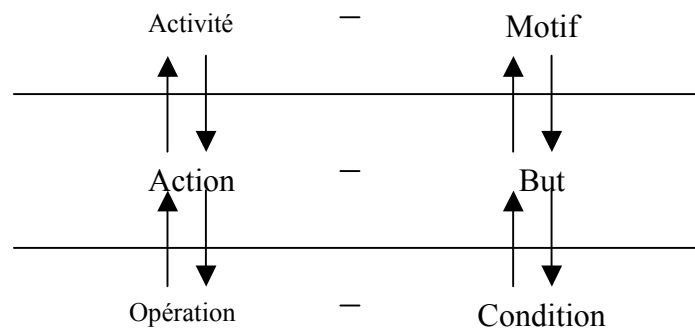


Figure 2.2 : Les niveaux hiérarchiques d'une activité.

L'exécution d'une action est planifiée, en utilisant un modèle, et au cours d'une phase appelée orientation. Sa réalisation peut donc être considérée comme une séquence orientation - exécution. Plus le modèle mis en œuvre est fiable, plus l'action a des chances d'atteindre son but. Dans le cas contraire, le sujet entre dans une phase d'analyse et d'apprentissage au cours de laquelle il sera certainement amené à modifier le modèle défaillant et, ainsi, augmenter son expérience. Ceci fait référence au concept de Plan Interne d'Action (PIA) développé dans la TA. Le PIA est proche des concepts cognitifs de mémoire de travail et de modèle mental, à la différence près qu'il ne renvoie pas à des modèles spécifiques, mais plutôt à la capacité générale de les créer et de les transformer. Le PIA représente la capacité humaine d'exécuter des manipulations sur des représentations internes d'objets externes, avant d'exécuter ces actions, sur ces objets, et dans la réalité. Autrement dit, la fonction du PIA est de prévoir les résultats futurs de possibles événements avant d'exécuter les actions dans la réalité.

Les opérations sont exécutées inconsciemment et sont orientées par une base d'orientation non consciente. Cette base d'orientation est établie au travers de l'expérience apparue au contact des conditions matérielles concrètes de l'opération. Les opérations correspondent à des actions dont le modèle s'est démontré réellement fiable en fonction de certaines conditions. On peut considérer une opération comme une action qui est descendue d'un niveau dans la structure hiérarchique de l'activité, du fait de la quasi-disparition de sa phase d'orientation. Elle est alors exécutée plus rapidement et répond, de façon inconsciente, à des conditions spécifiques. Les actions opérationnalisées sont aptes à participer à la création de nouvelles et plus amples actions. L'exemple le plus souvent cité est celui de l'utilisation

d'une boîte de vitesse. Au départ, elle est difficile et lente car réfléchi et planifié. Peu à peu, elle devient fluide et inconsciente, apte à participer à des actions de plus haut niveau (ex. ralentir).

Les opérations nous permettent d'agir sans penser consciemment à chaque petit pas d'exécution. D'ailleurs, la cause d'une erreur commise dans l'exécution d'une action constituée d'opérations bien maîtrisées est difficile à discerner car ces opérations sont réalisées de façon inconsciente (ce qui entraîne par exemple des questions telles que : « Ai-je bien fermé la porte à clef en partant ? »). Les opérations ne sont pas accessibles à l'auto-réflexion consciente de l'acteur. Il arrive souvent, lorsque les conditions d'exécution d'une opération changent, que l'opération retourne au niveau de l'action, demandant une re-analyse de la situation et une nouvelle phase d'orientation. Ceci arrive par exemple lorsque l'on descend un escalier dont les marches sont trop serrées, ce qui oblige l'opération de descente à revenir au niveau d'une action consciente, demandant plus d'attention, et exécutée plus lentement.

Dans la vie, il n'existe pas une correspondance un à un entre un but à atteindre et la manière d'y arriver. Ceci signifie qu'il n'y a pas unicité entre une action et l'ensemble des opérations qui participent à sa réalisation. Selon la TA, le déclenchement des opérations n'est pas uniquement guidé par les conditions environnantes. Elles sont en fait déterminées par la structure générale de l'action dans laquelle elles ont été incorporées. Ceci permet par exemple de représenter comment un sujet peut apprendre à contrôler ses réactions immédiates face à certaines situations. Enfin, l'exécution d'opérations qui ne réalisent pas d'action orientée vers un but peut être comparée à l'opération d'une machine qui aurait échappé au contrôle de l'humain.

Une action peut elle-même monter d'un niveau et devenir une activité. Réciproquement, une activité peut correspondre à une action dans une activité d'un niveau plus général. Ainsi, dans la TA, les limites entre les différents niveaux de la structure hiérarchique d'une activité sont mouvantes et floues. Ceci est particulièrement remarquable entre les niveaux activité et action, car la classification dépend totalement du point de vue adopté.

La dynamique action- opération et la possibilité de construire, au fil du temps, des actions de plus haut niveau à partir d'actions opérationnalisées constituent une propriété fondamentale

du développement humain. Ces mécanismes nous paraissent importants car ils viennent contribuer à la formation de l'expérience des sujets. Nous verrons plus loin que cette expérience joue un rôle important dans le développement de l'activité et la création des éléments qui la composent.

Finalement, nous pouvons dès à présent remarquer la manifestation d'un phénomène réflexif au sein de l'activité. En effet, nous avons vu que les opérations dépendent fortement du contexte, de la situation d'exécution des actions auxquelles elles participent. Or les actions ont tendance à transformer cette situation. Ainsi, l'activité apparaît comme continuellement influencée par une situation qu'elle ne cesse de modifier.

2.4.2 - Les propriétés de l'activité

La Théorie de l'Activité définit huit propriétés basiques du concept d'activité :

1. Une activité possède un objet (matériel brut ou espace problème) vers lequel elle est dirigée, qui la différencie d'une autre activité, et qui en motive l'existence. Derrière l'objet, se trouve toujours un besoin ou désir auquel l'activité doit répondre. On parle d'ailleurs parfois de motivation 'objectifiée' ou '*objectified motive*'.
2. Une activité possède au moins un sujet (acteur) actif qui comprend son objet et est en accord avec lui. Tous les participants d'une activité ne comprennent pas ou ne reconnaissent pas forcément l'existence de l'objet de l'activité, auquel cas ils sont identifiés comme des sujets passifs. Le sujet peut être individuel ou collectif.
3. Une activité est un phénomène collectif. Comme nous venons de le voir dans le point précédent, le sujet peut être lui-même un collectif. De plus, le sujet fait généralement partie d'une communauté, c'est-à-dire d'un ensemble de sujets partageant le même objet d'activité. Enfin, une activité est toujours influencée par ses conditions environnementales et le sujet impliqué dans une activité, même s'il est seul, est toujours plus ou moins directement influencé par d'autres sujets impliqués dans des activités connexes.
4. Une activité est réalisée par ses participants au travers d'actions orientées vers des buts conscients.
5. Une activité existe dans un environnement matériel qu'elle transforme.

6. Une activité est un phénomène de développement historique. Les éléments qui constituent l'activité possèdent chacun une histoire qui influence l'activité et se construit au cours de sa réalisation.
7. Les contradictions qui y apparaissent en sont la force de développement.
8. Les relations existantes entre les éléments d'une activité sont médiatisées culturellement.

L'énoncé de ces propriétés, dans une première approche, nous permet de considérer des apports intéressants pour la FAD et l'apprentissage collaboratif. En particulier, l'activité est directement considérée comme un phénomène collectif dirigé vers un objet idéal qui la motive et autour duquel est constitué un groupe de sujets interagissant jusqu'à sa réalisation. Notre but étant de fournir un support à la FAD en ligne, certains de ces points ont déjà une résonance particulière pour notre conception : l'activité transforme son environnement, c'est un phénomène dynamique, de développement historique et empli de contradictions. Pour expliquer et comprendre plus en profondeur les concepts et mécanismes sur lesquels reposent ces propriétés, nous allons exposer les travaux d'Engeström et son modèle de la structure basique d'une activité.

2.4.3 - La structure basique d'une activité

Engeström (1987) a défini un modèle structurel simple du concept d'activité exprimant la médiation existante entre le sujet et l'objet de l'activité. Cette médiation est matérialisée par le concept d'outil représentant tout ce qui est utilisé dans le processus de transformation, incluant aussi bien les outils matériels que les outils pour penser. Ce modèle est représenté sur la figure 2.3.

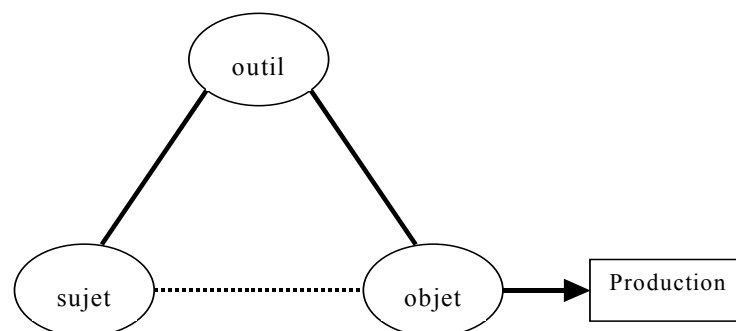


Figure 2.3 La relation médiatisée au niveau individuel. Tirée de (Bourguin, 2000)

L'outil influence fortement l'activité. En effet, l'outil à la fois permet et limite : il permet au sujet de réaliser l'objet de son activité, mais limite en masquant une partie du potentiel de transformation de cet objet. Les outils supportent et complètent les capacités humaines dans la construction de systèmes plus efficaces, permettant des accomplissements plus hauts. En contrepartie, l'outil participe à la formation des objectifs des sujets qui l'utilisent car il porte en lui des buts implicites qui y ont été mis par ses développeurs. L'utilisation d'un outil particulier change la structure de l'activité et peut même résulter en de nouveaux buts à satisfaire. Un outil doit servir le double but de permettre de faire quelque chose et de rappeler que quelque chose doit être fait. D'un autre côté, l'outil est lui-même transformé et construit au cours de l'activité. Il est souvent modifié ou adapté par les sujets, en fonction de leurs besoins émergents, de leurs buts, et de leur expérience. Ainsi, cette expérience se retrouve dans l'outil qui porte en lui l'héritage culturel de la situation. La médiation par l'outil correspond alors à un moyen de transmettre une certaine culture. Il est important de noter que, si l'outil ne possède pas les propriétés adéquates à une adaptation potentielle, il peut être définitivement rejeté de l'activité.

On voit ici réapparaître une certaine réflexivité. D'une part, l'intégration d'un outil dans une activité a pour conséquence d'en influencer le cours, mais, d'autre part, cette activité a tendance à le modifier. En d'autres termes, et comme nous l'avons déjà souligné, les actions des sujets au cours de l'activité ont tendance à créer une situation qui, elle-même, influence l'activité et donc les actions futures. Dans la TA, l'intégration de nouveaux outils est posée en terme de création d'organes fonctionnels ou «*functional organ*». Ces organes fonctionnels correspondent à des outils externes qui sont intégrés et expérimentés comme des propriétés de l'individu. L'exemple souvent cité est celui de l'œil humain qui, équipé de lunettes, compose un organe fonctionnel permettant une meilleure vision.

Le modèle de la relation médiatisée au niveau individuel (cf. figure 2.3), tel qu'il a été décrit, ne permet pas de considérer toutes les relations qui existent entre un individu et le reste de son environnement, en particulier et comme nous l'avons vu, le fait que l'activité soit un phénomène collectif. Il a donc été étendu par Engeström pour représenter la structure basique d'une activité qu'il définit comme le modèle systémique le plus simple en terme d'unité d'analyse. De nouveaux éléments viennent ainsi se greffer sur ceux de l'activité individuelle, de même que de nouvelles relations entre ces éléments. L'analyse est étendue pour montrer que l'individu n'est pas isolé mais fait partie d'une communauté. La communauté représente

l'ensemble des sujets qui partagent le même objet d'activité. Puisque l'activité est médiatisée, les relations qui existent entre ses divers éléments sont elles-mêmes médiatisées par de nouveaux concepts contenant, eux aussi, l'héritage culturel de la situation. Raeithel et Velichkovsky soulignent que, lorsque de nouveaux membres arrivent dans une communauté de pratiques, ils doivent s'approprier une connaissance distribuée et située du pourquoi, du quoi et du comment des actions typiques de celle-ci (Bourguin, 2000). On voit ainsi apparaître deux nouveaux éléments : le concept de règle qui médiatise principalement la relation sujet – communauté et celui de division du travail qui médiatise la relation objet – communauté.

Au cours de l'activité, les règles peuvent être plus ou moins explicites et contrôlent les relations entre le sujet et la communauté. Elles représentent des normes, des conventions, des procédures administratives, des pratiques de travail acceptées, ou encore des relations sociales. En d'autres termes, les règles définissent ce que signifie être membre d'une communauté. Comme tout élément médiateur, les règles cristallisent en elles un certain héritage culturel relatif à la relation qu'elles médiatisent. Par exemple, il est aisé de comprendre comment des lois sont construites et modifiées au cours du temps en tenant compte de l'expérience des sujets qui les ont créées. Cet exemple souligne d'ailleurs l'un des points les plus importants de la TA : la nature de tout artefact médiateur ne peut être comprise que dans le contexte de l'activité humaine. Chacun sait qu'il est souvent difficile de comprendre certaines lois si l'on ne considère pas le contexte dans lequel elles ont été créées. Enfin, en tant qu'élément médiateur, les règles sont ouvertes à de nouveaux développements au sein de la communauté, modifications qui viendront compléter l'héritage culturel qu'elles contiennent.

La division du travail correspond à l'organisation de la communauté dans le but de la réalisation de l'objet. Elle définit ce que chaque sujet d'une communauté doit faire pour réaliser l'activité dans laquelle il est impliqué. La division du travail reflète les responsabilités des individus envers les autres membres de la communauté. Elle peut refléter les différents rôles que jouent les individus. Il faut noter que dans la littérature, les auteurs utilisent la notion de rôle pour ne décrire que la division du travail.

La figure 2.4 représente le modèle présenté ci-dessus. Il faut noter que ce modèle apparaît toujours représenté sous une forme simplifiée pour en augmenter la lisibilité. Cependant, les

auteurs précisent généralement de ne pas oublier que l'activité est un système complet et qu'il existe en réalité des relations entre chacun de ses éléments.

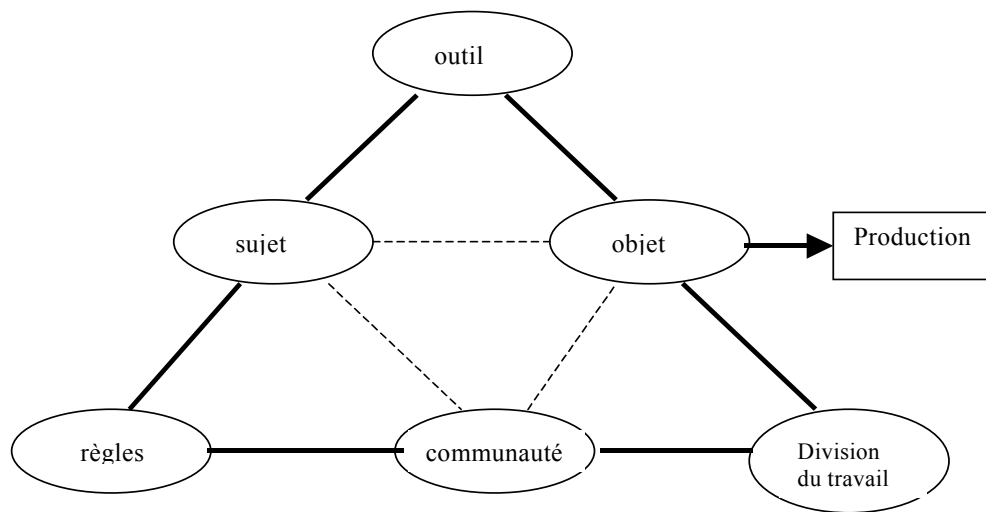


Figure 2.4 : La structure basique d'une activité tirée de (Bourguin, 2000)

Revenons un instant sur la contradiction inhérente à l'activité. Fjuk et Ludvigsen (2001) illustrent cette contradiction en montrant qu'autour d'une activité centrale dont le but est l'apprentissage, on assiste à l'émergence d'autres systèmes d'activités liées d'une part au foisonnement d'outils informatiques et technologiques et d'autre part, à l'ensemble des règles contenues dans les principes didactiques et les traditions institutionnalisées. Ces systèmes d'activités parallèles complexifient davantage l'apprentissage en rendant notamment difficile la délimitation du système d'activités à prendre en compte. L'apprenant est confronté à ces contradictions et en est fortement influencé dans sa collaboration avec les autres membres de la communauté.

Fjuk et Ludvigsen (2001) montrent que ces contradictions doivent être prises en compte dans la conception des environnements d'apprentissage collaboratif distribué (donc typiquement des environnements de supports à la FAD en ligne). Les concepteurs doivent tenir compte des interdépendances et conflits potentiels afin de comprendre, expliquer et contrôler les changements dans ces environnements.

2.5 - L'échelle d'analyse de la tâche dans notre recherche

Plusieurs recherches en apprentissage collaboratif concernent la gestion des interactions dans des groupes. On distingue généralement 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. Comme le soulignent justement Fjùk et Ludvigsen (2001) ces recherches situent souvent 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 isolément. 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.

2.6 - Les raisons du choix du paradigme d'agent

Les objectifs de nos travaux de recherche trouvent leur origine dans la volonté de soutenir l'apprentissage collaboratif distribué. De façon plus spécifique, nous nous fixons comme but la mise en œuvre de systèmes capables de susciter et de soutenir les interactions dans des groupes d'utilisateurs engagés dans une session de FAD. Les systèmes informatiques disposant de telles fonctionnalités sont, de par la réalité même du phénomène de FAD, des systèmes ouverts, complexes, distribués.

Un système ouvert est un système dont la structure peut changer dynamiquement. Ses composants ne sont pas connus à l'avance, changent au cours du temps, et sont essentiellement hétérogènes (en raison du fait qu'ils sont mis au point par différentes personnes, à différentes périodes et utilisant des outils et techniques différentes). Le système ouvert le plus connu est Internet. La FAD médiatisée par ordinateur s'appuie essentiellement sur Internet et exige l'utilisation de technologies appropriées ; il est évident que cette exigence induit de la part des informaticiens la mise en place de systèmes ouverts. En effet, Internet se

présente comme une grande source d'informations distribuée, dont les nœuds sont conçus et implémentés par différentes personnes et organisations selon leurs propres motivations ; tout système informatique s'appuyant sur Internet doit tenir compte de la diversité des organisations et des outils de conception et de développement sans être constamment guidé par les utilisateurs. De telles fonctionnalités exigent des techniques de négociation et de coopération qui relèvent du domaine des agents et des systèmes multi-agents.

La FAD parce qu'elle se trouve au carrefour de plusieurs disciplines des sciences humaines (psychologie, sciences de l'éducation, sociologie, etc.) et des sciences de l'ingénieur (informatique, télécommunications) pose des problèmes d'une grande complexité, diversifiés, interdépendants et difficilement prévisibles. Les outils adaptés pour faire face à la complexité dans le développement logiciel sont la modularité et l'abstraction. Le paradigme d'agent est un bon outil de modularisation, il permet de résoudre les problèmes posés par la FAD en développant des modules spécialisés (en termes de représentation et de type de problème résolu) dans un aspect particulier. Ces modules spécialisés seront vus comme des agents logiciels. L'interdépendance des problèmes de FAD est surmontée grâce à la coopération entre les agents mis en œuvre. Le paradigme d'agent nous permet de résoudre un problème global en le scindant en un nombre de composants simples, de petite taille, faciles à développer et à maintenir, spécialisés dans la résolution d'un aspect particulier du problème. Cette décomposition permet en outre d'utiliser les techniques et outils les plus appropriés pour concevoir et développer chacun des agents, évitant ainsi d'imposer une approche uniforme qui représenterait un compromis pour le développement du système entier sans être optimal pour la mise au point de chacun des modules. L'autonomie présente dans les agents fournit une abstraction comparable à celle offerte par les procédures (des programmes informatiques), les types de données abstraites et les objets. Cette autonomie permet au concepteur de voir son système informatique (support de la FAD) comme une collection d'entités-solutions autonomes et coopérantes. Cette vision de haut niveau est plus appropriée que toute autre alternative.

Les utilisateurs des plates-formes de formation à distance ne sont pas en général des spécialistes d'informatique et sont parfois du reste, peu familiers des environnements informatiques. Malgré les progrès des IHM (interfaces homme-machine) au cours des deux dernières décennies et de la généralisation des interfaces multi-fenêtres, ces utilisateurs trouvent encore l'usage des logiciels difficile. Une des raisons en est que les utilisateurs de

logiciels doivent pour résoudre leurs problèmes décrire toutes les étapes nécessaires allant d'un niveau de description générale au moindre détail. Dans ce contexte, l'exploitation de toute la puissance des logiciels exige de repenser la nature des interactions utilisateur-ordinateur. L'ordinateur doit devenir un partenaire de l'utilisateur en coopérant avec ce dernier pour atteindre ses objectifs. Negroponte (1995) estime ainsi que l'avenir de l'informatique est à 100% dans la délégation de tâches et non dans la manipulation des ordinateurs. Ceci nous semble d'autant plus vrai dans la FAD que celle-ci concerne non seulement, des utilisateurs non informaticiens (apprenants, formateurs, concepteurs de cours, tuteurs, administrateurs), mais, a la prétention de répondre au souci de l'apprentissage tout au long de la vie. Les objectifs de la FAD ne doivent pas se trouver contrariés par la nécessité de maîtriser les techniques informatiques ou la manipulation des ordinateurs et des logiciels. Jennings et Wooldridge (1998) affirment que pour remplir ces objectifs, les logiciels doivent posséder les qualités suivantes :

- autonomie : à partir d'une spécification vague et imprécise, ils doivent déterminer sans une constante intervention humaine le meilleur moyen de résoudre un problème et, partant, le résoudre ;
- dynamisme : ils ne doivent pas attendre qu'on leur indique à chaque fois la prochaine étape mais, au contraire, doivent faire des suggestions aux utilisateurs ;
- réactivité : ils doivent tenir compte des changements des besoins des utilisateurs et des changements intervenant dans leur environnement de travail ;
- adaptabilité : ils doivent apprendre les préférences des utilisateurs et en tenir compte dans leurs interactions avec ces utilisateurs.

Ces caractéristiques sont celles des agents logiciels intelligents.

En résumé, la FAD médiatisée par les ordinateurs exige de prendre en compte :

- *l'hétérogénéité* des sources d'informations, des systèmes d'information et des centres de régulation des activités (au niveau de l'apprenant, du tuteur, du formateur, du concepteur, du gestionnaire de la formation) ;
- *l'ouverture* liée à l'ajout/retrait de sous-systèmes (logiciels, matériels, acteurs) sans stopper ou réinitialiser le système global ;
- *l'évolutivité* pour s'adapter aux changements continuels de l'environnement dans lequel s'inscrit la FAD ;
- *la réactivité* pour se configurer en fonction de l'évolution des activités à mettre en œuvre, en fonction des compétences disponibles et des objectifs d'apprentissage ;

- *la coopération* entre les différents sous-systèmes pour produire des contenus pédagogiques, administrer, assister, évaluer les apprenants ;
- *la coordination* des différentes plates-formes de FAD qui ne manqueront pas de se regrouper en environnement de campus virtuel du fait entre autres de la mondialisation et de l'avènement des autoroutes de l'information.

Dans ce contexte, l'informatique sur laquelle s'appuie la FAD doit intégrer de nouveaux concepts, en s'appuyant sur des technologies fortement liées à Internet, afin de proposer des architectures logicielles qui permettent la coopération de multiples applications hétérogènes et distribuées.

2.7 - L'approche composants dans les EIAH

Notre objectif dans cette section est de fournir quelques éléments permettant de comprendre l'approche «composants» et de la mettre en perspective avec le paradigme agent que nous avons retenu pour la construction d'outils de support pour l'apprentissage collaboratif distribué.

Rappelons que l'idée d'adopter une approche par composants dans l'ingénierie des systèmes d'information vient de la volonté de mettre en œuvre des propriétés de réutilisation, de coopération et d'interopérabilité. La réutilisation permet de réduire les coûts et les délais de conception, d'implémentation, de maintenance lorsqu'une traçabilité est associée (Barbier et al., 2002). La coopération et l'interopérabilité permettraient d'utiliser dans un environnement donné des fonctionnalités complémentaires issues de prototypes différents pour bâtir des activités pédagogiques (Rosselle, 2003). La valeur ajoutée d'une telle démarche est d'éviter de tout redévelopper soi-même mais plutôt, d'intégrer et d'utiliser des composants existants (Futtersack et Labat, 2000).

2.7.1 Définitions

Un composant peut être défini comme une «brique»

- avec une interface bien spécifiée (le service rendu) ;
- respectant un certain nombre de conventions qui permettent de le manipuler sans connaissances préalables.

On pourra ainsi :

- le créer, le détruire, l'activer ;
- l'assembler avec d'autres composants ;
- le configurer, l'adapter ;
- le déployer dans un environnement d'exécution (le conteneur).

Un composant réutilisable traite

- un problème récurrent dans l'ingénierie des systèmes d'information ;
- capitalise un fragment de produit ou de processus ;
- offre une solution conceptuelle et/ou logicielle testée, acceptée et adaptable.

Un modèle de composants fournit des mécanismes de communication standards entre

- les composants : c'est l'appel de méthodes qui s'effectue à travers son interface ;
- un composant et le conteneur : afin que le conteneur puisse prendre en charge un composant et lui permettre d'accéder par ailleurs à différents services.

Il existe différents critères de caractérisation des composants, on peut citer :

- *type de connaissance* : produit ou processus ;
- *couverture* : générale (horizontale), métier (verticale), entreprise ;
- *portée* : étapes de l'ingénierie ;
- *nature de la solution* : conceptuelle ou logicielle ;
- *technique de réutilisation* : spécialisation, composition, instanciation, duplication, ...
- *ouverture* : boîte noire, blanche ou en verre ;
- *granularité* : en nombre de classes, de modules, ...

2.7.2 Quelques types de composants pour les systèmes d'information

La figure 2.5 présente la réutilisation dans l'ingénierie logicielle et on voit apparaître différents types de composants au fil des étapes du cycle de développement logiciel. Nous allons plus particulièrement examiner les composants métier, les composants architecturaux et les patrons d'ingénierie.

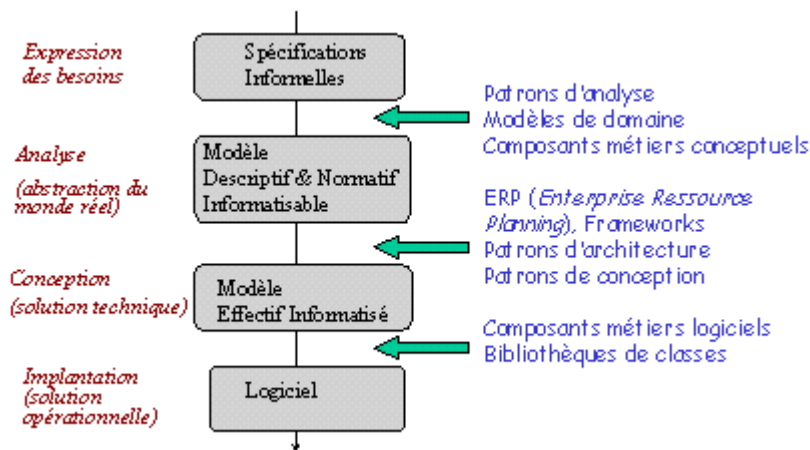


Figure 2.5 : La réutilisation dans l'ingénierie logicielle

Les composants métier répondent à des problèmes d'ingénierie de besoins, de spécification et de modélisation des systèmes. Ce sont des entités logicielles encapsulées et autonomes de déploiement. Ils se différencient des autres composants logiciels en ce sens qu'ils sont spécifiques, c'est-à-dire réutilisables dans un même domaine d'application. On parle souvent de composants verticaux par opposition aux composants horizontaux qui sont eux, réutilisables dans différents domaines. Lors de l'intégration de ces composants, il se pose des problèmes d'interopérabilité sémantique : le demandeur et le fournisseur doivent avoir la même sémantique des services et des données échangées. On observe deux grandes tendances en matière de composants métier :

- le composant métier fournit le processus et les produits de développement nécessaires pour représenter, concevoir, implémenter et déployer un concept de domaine dans un système logiciel. Ce type de composant est le plus souvent implanté au moyen de technologies composants distribués de type CORBA ou EJB (Enterprise Java Bean).
- le composant métier est une unité de réutilisation de connaissance de domaine. L'accent est mis ici sur la connaissance de domaine et sa représentation. Cette approche est suivie par les méthodes d'ingénierie de domaine et de lignes de produits.

Les composants architecturaux sont une nouvelle génération de langages de spécification et de conception des architectures logicielles plus connus sous le sigle anglais ADL (*Architecture Description Language*). Les ADL favorisent le développement de systèmes à base de composants en offrant un certain nombre d'avantages : diminution drastique du temps de développement, facilité d'utilisation des composants produits, diminution du coût de production, augmentation de la fiabilité. Les ADL sont également utilisés comme des méta-

modèles pour guider le développement des systèmes et leur garantir certaines propriétés : raffinement, évolution, réutilisation, consistance (compatibilité entre les composants). Trois dimensions essentielles caractérisent un ADL :

- son niveau d'abstraction : permet d'exprimer les degrés de raffinement d'un produit depuis sa spécification (plus haut niveau d'abstraction) jusqu'au code source (niveau le plus bas) ;
- son point de vue architectural : permet de décrire les différents modèles de représentation d'un composant (textuelle, graphique, flot de données, ...) ;
- son domaine : reflète les différents domaines structurés en couches sur lesquelles repose un système donné. Cette structuration permet à un composant appartenant à une couche particulière de ne pouvoir interagir qu'avec des composants de la même couche ou de la couche adjacente.

Parmi les ADL les plus représentatifs, nous pouvons citer :

- ACME dont le but est de fournir un langage pour faciliter le partage de composants architecturaux issus d'autres ADL et d'intégrer des outils d'analyse.
- RAPIDE qui fournit un ADL exécutable basé sur un modèle d'exécution de règles d'événements. Il est destiné au prototypage, à la simulation et à l'analyse des systèmes logiciels.
- UNICON supporte des constructions architecturales basées sur des « styles » en interconnectant des composants architecturaux prédéfinis.
- METAH spécifie les aspects temps réel d'un système logiciel ou matériel et fournit un support pour l'analyse des propriétés fonctionnelles et non-fonctionnelles d'un système.
- WRIGHT propose un formalisme centré sur différents types de connecteurs explicites.

Les patrons d'ingénierie ou patrons de conception (*design pattern*) capitalisent un savoir-faire permettant de résoudre un problème récurrent dans un domaine. Cette notion a été empruntée à l'architecture des bâtiments (patron d'architecture) et à la couture (patron de couture). En informatique, on peut définir les patrons comme des composants réutilisables :

- alliant problème et solution ;
- offrant des solutions logicielles (spécifications réutilisables) ;
- correspondant à des composants de type « boîte blanche » ;
- dédiés à la réutilisation de micro-architectures (4 à 5 classes).

Trois critères sont généralement utilisés pour la classification des patrons. Il s’agit :

- du type de connaissance : capitaliser des spécifications ou des implémentations pour atteindre un résultat (patron produit) ou une démarche à suivre (patron de processus) ;
- de la couverture : il peut s’agir de patrons généraux, domaine ou entreprise ;
- de la portée : étape dans l’ingénierie logicielle (analyse, conception, implémentation) à laquelle le patron s’adresse.

Avec ces critères de classification, on a ainsi des patrons produits généraux dédiés à une et une seule étape du processus de développement. On a aussi des patrons processus généraux qui couvrent tout l’ensemble du cycle de développement de logiciels.

2.7.3 Démarches d’ingénierie de systèmes à base de composants

Il existe plusieurs technologies supportant les composants : parmi les plus connues, on peut citer CORBA et EJB. Toutefois, les méthodes supportant le développement de systèmes à base de composants ne sont pas encore tout à fait au point.

	Catalysis	RUP	Select Perspective
Processus de développement	Itératif et incrémental	Itératif et incrémental	Itératif et incrémental
Notation	UML + Extensions	UML + Extensions	UML, ERP, CSC Catalyst
Outils Case	CoolSpex, Cool :Gen	Rational Rose, etc.	Select Component Factory
Complexité d’apprentissage	Très complexe	Moyennement complexe	Moyennement complexe
Introduction de composants dans le cycle de développement	Logique, implémentation	Logique, implémentation	Logique, implémentation
Règles d’identification des composants	Pas de règles	Pas de règles	Pas de règles
Niveau de décomposition	Pas d’indications	Pas d’indications	Pas d’indications
Représentation des composants	Logique : type Implémentation : package, composant	Logique : sous-système Implémentation : diagramme composants	Logique : package de services Implémentation : diagramme composants
Rigueur dans la spécification des composants	Oui avec OCL	Non mentionné	Non mentionné
Réutilisation	Framework avec des indications sur la manière de les construire	Package de services sans précision sur l’approche à suivre pour les identifier et construire	Un processus supportant la gestion d’une librairie de composants réutilisables

Tableau 2.1 : Résumé des caractéristiques générales des méthodes Catalysis, RUP, Select

Le tableau 2.1 résume les caractéristiques de trois méthodes connues à savoir : Catalysis (D’Souza et Cameron Wills, 1999), Rational Unified Process (Jacobson et al., 1999) et Select Perspective (Allen et Frost, 1998).

Ces méthodes de développement, bien que largement utilisées, présentent plusieurs faiblesses.

On peut citer :

- l'introduction tardive des composants dans le cycle de développement (seulement aux niveaux logique et implémentation) ;
- le déficit de guidage offert au concepteur dans l'identification des composants. Elles requièrent du concepteur une trop grande expertise ;
- la détermination du niveau de granularité des composants qui reste délicate pour le concepteur ;
- une prise en charge partielle de la réutilisation.

Après avoir examiné le concept de composants et dit quelques mots sur l'approche par composants, nous voudrions à présent expliquer les raisons pour lesquelles nous n'avons pas eu recours à cette approche pour répondre à notre problématique. Une mise en perspective de l'approche par composants et du paradigme d'agent permet d'observer que les deux notions ne sont nullement contradictoires mais ne répondent pas au même souci. En caricaturant à peine, nous dirions que l'approche par composants se préoccupe de faciliter la tâche des autres programmeurs alors que le paradigme agent prend en compte l'utilisateur final du système et la complexité du domaine d'application concerné.

L'approche par composants, en prenant en compte les exigences de réutilisabilité, d'interopérabilité et de coopération des systèmes construits, vise à éviter que les programmeurs essayent de réécrire des outils déjà mis au point par d'autres. Elle permet de s'assurer que d'autres programmeurs (et même des utilisateurs) pourront facilement intégrer ces outils dans un environnement donné et les exécuter. Ceci engendre un gain de temps et de ressources considérables. Le paradigme d'agent pour sa part, convient à des systèmes répondant à des besoins évolutifs, non clairement identifiés au départ ; à des systèmes évoluant dans des environnements ouverts. Bref, l'approche agent permet de construire des systèmes informatiques ouverts, complexes, flexibles avec une informatique dite ubiquitaire. Le paradigme d'agent consacre l'espoir que les utilisateurs auront des systèmes informatiques auxquels ils pourront déléguer des tâches et non plus en faire des entités serviles obéissant aux comportements prescrits par leurs concepteurs (Cerri et Maraschi, 2003). Cette forme d'informatique classique ne convient que pour des applications dont les besoins ne sont guère évolutifs et exigent de la part des utilisateurs des capacités d'analyse, de conception et d'implémentation avérées.

Toutefois, même en recourant au paradigme agent comme nous l'avons fait, il est possible d'intégrer une approche par composants dans la construction de son système. Le but étant de permettre une réutilisation facile du système construit, une intégration dans des environnements d'exécution et une interopérabilité avec d'autres systèmes. Nous n'aurions cependant pas pu adopter une telle démarche car ceci présuppose un minimum de standardisation afin d'être sûr que le système proposé correspond bien aux besoins et aux exigences de la communauté. Ces conditions ne sont pas remplies à la date actuelle. Il convient d'autre part de garder à l'esprit les faiblesses encore actuelles des méthodes de développement des systèmes à base de composants. La notation UML qui est largement utilisée par le formalisme de ces méthodes nous semble constituer aussi un écueil lorsqu'on veut analyser, concevoir, spécifier et implémenter des systèmes à base d'agents.

2.8 - Terminologie

Nous présentons ci-après la terminologie que nous utiliserons tout au long de ce mémoire de thèse. Cette terminologie est construite à partir des concepts utilisés au cours de la formation SimuLigne, des concepts de la méthodologie de développement des systèmes multi-agents que nous avons utilisée pour spécifier notre système multi-agents baptisé SIGFAD et des concepts de la théorie de l'activité. Nous avons recherché une cohérence entre les différents termes et concepts, évitant d'utiliser un même terme dans des situations différentes si ce terme représentait un concept différent. Nous avons gardé, pour les concepts propres à la plate-forme de téléformation, leur désignation en anglais.

2.8.1 - PIFAD, SIGFAD, SimuLigne.

PIFAD (Plate-Forme Informatique pour la Formation A Distance) est le nom donné à la plate-forme de téléformation utilisée dans l'expérimentation SimuLigne. Il s'agit du logiciel WebCT (2000). Quand nous parlerons des propriétés et caractéristiques de PIFAD, il s'agira en fait de celles de ce logiciel. Mais nous avons choisi ce logiciel pour sa généralité en tant que plate-forme de téléformation.

SIGFAD (Système pour le soutien des Interactions dans des Groupes de Formation A Distance) est le nom donné au système multi-agents que nous avons conçu et développé pour

assister les utilisateurs engagés dans une session de formation à distance. Il vient donc compléter PIFAD, voire en remplacer certains éléments ou fonctionnalités.

SimuLigne (SIMULATION en LIGNE) est le nom donné à l'expérimentation de formation à distance en groupe qui s'est déroulée de mars à juillet 2001. Elle a été conçue par les chercheurs du projet ICOGAD. Signalons tout de suite que les choix pédagogiques décidés pour cette formation ont primé sur les considérations techniques mises en avant par les concepteurs du progiciel PIFAD. Pour la bonne compréhension du lecteur (et en particulier des contraintes techniques imposées par ce type de progiciel), nous serons amenés à expliquer comment la philosophie SimuLigne a été implémentée dans PIFAD, et à utiliser principalement la terminologie SimuLigne avant celle, technique, de PIFAD.

2.8.2 - Groupe / Cours:

Dans PIFAD, les personnes sont enregistrées comme faisant partie d'un cours (c'est le cours qui permet d'implémenter la structure de groupe ou de classe). Un cours possède un concepteur et d'autres personnes qui en sont les apprenants ou les membres. L'appartenance à un groupe permet d'évoluer dans un espace circonscrit aux membres du groupe où sont disponibles tous les outils de la plate-forme, tous les documents, toutes les communications. Il convient de préciser qu'un utilisateur donné de PIFAD peut appartenir à différents groupes avec des statuts (apprenant ou concepteur) différents. Chaque utilisateur a un statut unique dans un groupe donné.

On peut néanmoins noter une exception à la relation univoque qui existe entre le groupe et l'espace de travail, il s'agit d'une salle de bavardage bien particulière dans laquelle des personnes n'appartenant pas au même groupe peuvent se retrouver pour discuter de façon synchrone. Cet espace n'ayant pas été utilisé dans SimuLigne comme ressource pédagogique, peu de gens s'en sont servis par eux-mêmes, ne serait-ce que comme espace café.

2.8.3 - Espaces / e-monde / e-univers

Dans la plate-forme PIFAD, l'équipe de conception de SimuLigne a aménagé un ensemble considérable d'espaces de travaux, distincts mais souvent reliés. Ainsi les quatre groupes ont travaillé dans des espaces différents interdisant toute intrusion par un membre extérieur, mais,

d'une part, la coordination générale entre les quatre groupes était assurée dans un espace hors groupes, réservés aux tuteurs (le groupe des Formateurs) et, d'autre part, chaque membre de la formation SimuLigne avait accès à un espace commun général pour accomplir certaines activités interculturelles (le groupe "Monde SimuLigne"). A l'intérieur de chaque groupe, étaient aménagés des espaces strictement individuels, des lieux de rencontres ou d'échanges par binômes, en petits sous-groupes ou pour tout le groupe, lieux limités à l'exécution de tâches spécifiques ou à des discussions sur des thèmes fixés, etc. Pour chaque individu se pose alors le problème de trouver ses repères, d'appréhender l'espace de ces nouveaux mondes. Ce monde en ligne d'un type spécifique associé à chaque groupe sera appelé "e-monde". Au cours de la formation, les individus (apprenants / natifs / tuteurs) agissant dans un groupe lui construisent un vécu, avec des règles de fonctionnement spécifique, d'où l'appellation de "monde" : on change de monde en passant d'un groupe à un autre. L'ensemble des e-mondes de SimuLigne constitue un univers, appelé "e-univers".

2.8.4 - e-personnalité

Le développement d'une e-personnalité (ou *electronic personality*, comme l'appelle certains auteurs américains) se manifeste dans des aptitudes telles que celle consistant à recréer un sentiment d'espace privé dans son espace individuel et dans le respect a priori de celui des autres ; à pouvoir aborder et gérer des aspects émotionnels à travers les contraintes des modes actuels de communication disponibles sur ces plates-formes ; à se créer une image mentale de chaque partenaire avec lequel on communique ; à créer une certaine forme de présence en ligne au long cours en personnalisant ses communications et à appréhender celle des autres. Disposer de telles aptitudes est un point absolument critique pour les tuteurs qui sont au centre de tous les échanges dans le groupe (Chanier, 2001a). Mais la maîtrise de telles aptitudes demande beaucoup de pratique, sans doute aussi une formation spécifique, et ne semble pas pouvoir provenir du transfert de compétences acquises dans la formation en présentiel.

2.8.5 - Utilisateurs et statuts PIFAD

Un utilisateur se comprend comme la réunion de trois choses : une identité d'accès (*login*), un mot de passe et un nom. Le nom est associé à l'identité d'accès dans toutes les actions que fait cet utilisateur dans PIFAD. C'est, au niveau technique, une simple chaîne de caractères qui

peut être modifiée, lorsque l'on en a les droits. Une identité d'accès (login), elle, est un identifiant système non modifiable. Un utilisateur lorsqu'il agit à un instant donné dans PIFAD a l'un des statuts suivants : *administrator*, *learner*, *designer*, *teaching assistant* (ce dernier statut n'a pas été utilisé dans SimuLigne). Un utilisateur PIFAD peut donc avoir des statuts différents suivant les *courses* auxquels il appartient.

Le statut de *administrator* de la plate-forme permet de créer les *courses*, pas les contenus, ainsi que de créer les utilisateurs et de leur donner des droits. Le nom d'utilisateur ayant le statut *administrator* dans SimuLigne est *systeme*. Il n'a pas de nom (champ correspondant resté vide). Pour chacun des *courses*, il existe des *designers* et des *learners*. Un *designer* a les droits de gestion sur le *course*, notamment celui d'inscription d'un *learner*, le *learner* doit par ailleurs avoir été inscrit dans la base de données générale de la plate-forme par *systeme*.

Chaque *course* a un designer principal, désigné (identité d'accès) par *systeme*. Ce *designer* principal est un utilisateur PIFAD pré-existant dans la base des utilisateurs PIFAD. Le *designer* principal d'un *course* est affecté à sa création et ne peut être modifié. A chaque *course* est associé un nom désignant la personne principale en charge du *course*. Ce nom est la valeur d'une chaîne de caractères affectée librement par n'importe quel utilisateur ayant le statut de *designer* dans ce *course*. Ce nom est indépendant (au niveau système, mais non indifférent pour nous) des noms d'utilisateur PIFAD. Le nom de la personne responsable d'un *course* apparaît associé à chaque *course* dans la page d'accueil de PIFAD .

Les *courses* sont des entités pratiquement autonomes. Seul *systeme* a le droit d'effectuer certaines opérations de haut niveau sur plusieurs *courses* (mais ne peut effectuer des opérations de base, comme consulter des contenus, communiquer, etc.). La séparation entre les *courses* est tellement importante, qu'un même utilisateur PIFAD, s'il est inscrit dans deux *courses*, est compté comme deux utilisateurs différents dans les termes de la licence d'exploitation de la plate-forme (donc 50 personnes physiques inscrites chacune dans trois *courses* correspondent à 150 utilisateurs pour une licence d'exploitation ; celle que nous avons achetée autorisait 400 utilisateurs au maximum).

Le *designer*, dans son *course*, a une interface très différente de celui du *learner*, interface qui lui permet d'accéder à ses fonctions spécifiques de gestion du *course*. Il n'a pas non plus la

même vue que le *learner* sur l'organisation et les contenus du *course*. Il existe donc une vue *designer* et une vue *learner*.

Nous utilisons le terme "utilisateur" pour parler d'une personne humaine connectée à un système informatique. L'"utilisateur PIFAD" est un utilisateur dont les droits d'accès sont définis et limités au sens de PIFAD. Cela revêt donc un aspect technique. Nous nous intéressons d'abord aux acteurs dont les fonctions ont été explicitées au sens de SimuLigne.

2.8.6 - Acteur SimuLigne

Un acteur désigne soit une personne physique ayant participé à la formation SimuLigne, soit une classe générique (on pourra parler de type d'acteurs) de personnes physiques se distinguant soit par son statut, soit par ses fonctions dans SimuLigne.

Dans SimuLigne (comme dans les EIAH), nous distinguons les statuts d'acteurs suivants :

- apprenant : personne consciemment et explicitement engagée dans un processus de formation par rapport à un domaine de compétence principal. Par défaut, il s'agit de la compétence en langue française. Dans ce cas, le sujet de la formation SimuLigne étant le français langue étrangère (FLE) pour des anglophones, un apprenant est non francophone. Mais nous avons dû aussi former les tuteurs à l'environnement PIFAD. Ils ont eu le statut d'apprenant pendant leur période de formation (avril 2001).
- formateur : c'est nécessairement un pédagogue, ayant reçu une formation spécifique en formation des adultes et, en principe, en formation à distance. Même si dans la plupart des cas, il est pédagogue dans le domaine d'apprentissage, ce n'est pas une nécessité pour toutes les fonctions qui sont à assurer.
- technicien : a des connaissances spécifiques en informatique, réseaux, administration de plates-formes ou en multimédia.
- expert / natif : le domaine d'expertise dans SimuLigne étant le français, un expert est nécessairement francophone. Il sert de référence pour l'usage du français courant et/ou la culture française, ce qui n'implique ni qu'il ait des connaissances exactes (un formateur peut être amené à rectifier par rapport à des objectifs pédagogiques et il est difficile de parler de connaissances "exactes" dans le domaine de la langue et de la civilisation), ni qu'il soit un "bon" pédagogue. L'expert, comme dans tout EIAH, sert de référence aux apprenants : ceux-ci ont pour objectif très général et à long terme de

savoir parler "comme" les francophones, ou du moins d'être capables de communiquer avec des francophones. L'expert, comme tout acteur d'un EIAH, a pour objectif d'aider les apprenants, ce qui ne veut pas dire qu'il ait reçu une formation pédagogique spécifique. C'est d'ailleurs cette complémentarité avec les pédagogues / formateurs qui est jugée intéressante par les chercheurs en didactique.

- un expert dans SimuLigne étant francophone est nécessairement un natif, au sens linguistique (sa langue maternelle est le français). Mais tous les acteurs natifs ne sont pas experts. Certains formateurs sont des natifs. Pour les distinguer nous parlons alors des "natifs de Besançon" qui sont experts dans SimuLigne.

Dans SimuLigne, nous distinguons les fonctions d'acteurs suivantes :

- coordonnateur de formation : formateur responsable de la coordination entre les différents acteurs et du suivi de la formation. Le coordonnateur a été baptisé "gestionnaire" de formation pendant SimuLigne (Torrent, 2001). Le mot coordonnateur laisse entendre qu'il s'agit d'un pédagogue plutôt que d'un administratif.
- tuteur : formateur responsable du déroulement de la formation à l'intérieur d'un groupe.
- concepteur de formation : formateur chargé de concevoir la formation SimuLigne
- médiatiseur : technicien chargé de créer ou transformer pour les mettre en ligne dans tous les médias appropriés les matériaux pédagogiques, matériaux créés ou conçus par le concepteur de formation.
- administrateur système : technicien, responsable de la gestion de l'installation, de la gestion technique de PIFAD. Il crée les comptes utilisateurs, les groupes et assure l'affichage de certains messages visibles à l'entrée du système par tous les groupes.
- aide technique : technicien chargé de la télé-assistance pendant le déroulement de la formation sur les plans techniques.

Une personne physique peut remplir à des moments différents plusieurs fonctions (une personne a successivement été médiatiseur(e) en mars 2001, conceptrice de formation et tutrice pour les formateurs de l'Open University en avril, tutrice aux côtés des tutrices de l'Open de mai à juillet). De même une fonction peut être assurée simultanément par plusieurs personnes (aide technique, concepteur de formation), au sens où un même travail peut être partagé par plusieurs personnes. Ne pas confondre ceci avec le fait que plusieurs personnes peuvent avoir la même fonction : plusieurs personnes ont été tuteurs, mais ces tuteurs n'étaient pas interchangeables (il aurait été très difficile de remplacer un tuteur à la responsabilité de

son groupe). A noter que la fonction Coordonnateur de formation a été remplie par une seule personne.

2.8.7 - Implémentation des fonctions et statuts SimuLigne dans PIFAD :

- Les experts / natifs ont les statuts PIFAD *learner* dans les groupes de base, dans celui des formateurs, dans le Monde SimuLigne. Ils ont chacun une identité et un nom d'utilisateur propre (*login* à leur nom).
- Un tuteur est *learner* dans le groupe des formateurs, dans le Monde SimuLigne et *designer* dans son groupe de base. Il apparaît toujours sous son identité propre et a son accès propre (*login* à son nom). Le nom du tuteur a été déclaré comme étant le nom de la personne en charge du groupe (cf. partie sur Utilisateurs PIFAD). Le tuteur du groupe des formateurs s'est vu accorder le statut de designer dans ce groupe pendant qu'il assurait cette formation.
- Un apprenant est *learner* dans son groupe de base et dans le Monde SimuLigne. Il apparaît toujours sous son identité propre et a son accès propre (*login* à son nom).
- *systeme* est le nom de l'utilisateur qui est *administrator* de PIFAD. Il n'a pas d'identité propre. Il n'appartient à aucun groupe.
- *concepteur* est le nom d'utilisateur générique qui est designer principal de tous les groupes de PIFAD. Il n'a pas d'identité propre.
- *machin* est le nom d'utilisateur générique qui est déclaré comme *learner* dans tous les groupes de PIFAD. Il n'a pas d'identité propre. Il permet aux techniciens, concepteurs de formation et chercheurs de voir ce que voient les apprenants sans intervenir dans les groupes.
- Le coordonnateur de formation est une personne physique unique qui agit dans PIFAD sous son nom d'utilisateur et avec son identité propre (Stéphanie Torrent). Il a le statut de *designer* dans le groupe des formateurs et de *learner* dans le Monde SimuLigne. Lorsqu'il désire observer (sans agir) ce qui se passe dans un groupe, il se connecte alors sous le nom d'utilisateur *machin*.
- Les différentes personnes qui agissent en tant que médiatiseur ou concepteur de formation dans un groupe doivent le faire en se connectant sous le nom d'utilisateur concepteur. Elles ne doivent pas effectuer de communications dans ces

courses/groupes, sinon leurs messages apparaîtraient avec un nom (ou sans nom) différent de la personne tuteur en charge du groupe.

Les acteurs humains (apprenants, formateurs/tuteurs, administrateur système, coordonnateur de la formation/superviseur académique) et artificiels (rôles, agents) sont identifiés et interviennent dans l'accomplissement des objectifs de SIGFAD. Il convient de préciser que suivant la méthodologie de développement des agents que nous utilisons, les interfaces entre les bases de données et le SMA sont construites de la même façon que les interfaces homme – SMA. Dans la description des différentes interactions entre les entités en présence dans le SMA, on pourrait placer les bases de données au même niveau que les autres acteurs.

2.8.8 - Module de formation / Etape / Activité / Tâche

La formation SimuLigne est composée de deux modules principaux : celui concernant la simulation globale proprement dite et Interculture. Chaque module est organisé en différentes étapes. Chaque étape comporte un certain nombre d'activités bien identifiées. Chaque activité est un ensemble de tâches qu'on peut associer à l'utilisation d'un outil particulier (courrier électronique, Forum, confection d'une page HTML, transfert de fichiers, etc.). A chaque activité sont associées des consignes dont les présentations varient suivant qu'elles sont destinées aux apprenants, aux tuteurs ou aux experts - cf. SimuLigne (2001b, 2001c).

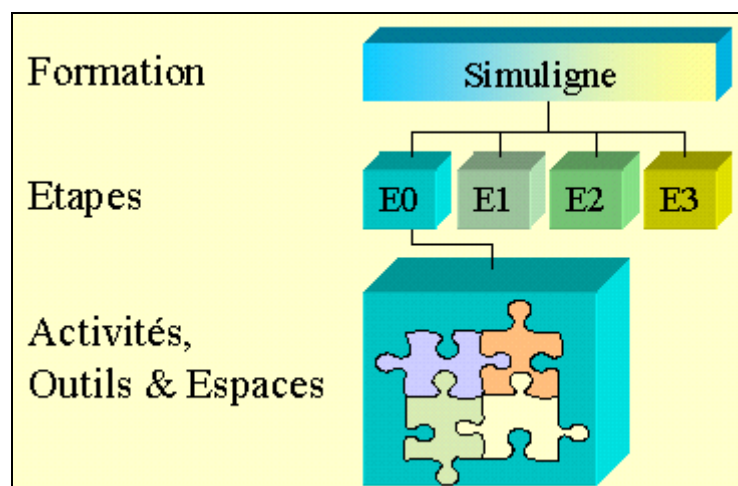


Figure 2.6 : Organisation de SimuLigne

Une étape est l'ensemble des activités permettant d'atteindre un jalon de la vie du groupe ou du projet. Toutes les activités d'une étape doivent être terminées pour valider la fin d'une

étape. Deux étapes sont nécessairement disjointes dans le temps. Une activité est un ensemble cohérent de tâches visant à élaborer un constituant du produit final.

Une tâche est un travail élémentaire concourant à la réalisation d'une activité. Une tâche peut être liée au maximum à un espace (en ligne) ou à un outil (hors ligne). Elle peut être de différentes fréquences (une fois / répétitive), de différentes situations de connexion (en ligne / hors ligne), de différentes natures (individuelle / collaborative) et donner lieu (ou non) à une production. Dans le cas où une production est attendue (ce n'est pas nécessairement un constituant du produit final), on devra préciser la date limite de réception, le lieu précis (groupe, espace, nom) et la nature et le format de la production attendue (image/gif, text/rtf, application/octet-stream...).

La formation SimuLigne est composée des modules suivants³ :

- Prem (première connexion et première activité des apprenants ; espaces : courriel externe et groupe de base, avril 2001) ;
- PreptutE0 (préparation des tuteurs à l'étape E0 ; e-Monde : groupe des formateurs ; avril 2001) ;
- Simulation globale avec les étapes E0⁴ (se présenter et découvrir PIFAD), E1 (Ville et identité), E2 (Faire vivre l'espace), E3 (vote et bilan), les 3 premières étapes se déroulant dans les groupes de base et la dernière dans ces groupes et le Monde SimuLigne, le tout de mai à début juillet sur 10 semaines.
- Interculture (cf. SimuLigne (2001 c)), composé d'activités à caractère interculturel, s'est déroulé dans le Monde SimuLigne. Cependant une activité de préparation à Interculture s'est déroulée dans chaque groupe de base pendant l'étape E0.

2.8.9 - Outil PIFAD

Il existe dans PIFAD plusieurs catégories d'outils, un outil permet de réaliser une tâche. A un outil correspond un espace de travail ou une classe d'espaces de travail (ainsi l'outil bavardage est composé de plusieurs pièces dont les noms sont modifiables). Par convention si le nom

³ Par simplification on omet la question de la formation à PIFAD des natifs qui a été faite en présentiel, début 2001.

⁴ On a préféré mettre l'étape E0 dans le même module que la simulation car : elle constitue une étape inévitable dans une formation (apprentissage de l'outil technologique et présentation entre les membres) ; elle concerne les mêmes personnes et les mêmes groupes que les étapes suivantes (au contraire de PreptutE0 qui se déroule dans le groupe des formateurs et ne concerne que les tuteurs).

d'un outil commence avec une majuscule, alors on fait référence à la forme, aux fonctionnalités que prend un outil dans l'environnement PIFAD (exemple : "courriel" est le mot utilisé pour parler du courrier électronique en général ; "Courriel" désigne l'outil de courriel de PIFAD). On peut distinguer les types d'outils suivants dans PIFAD :

- les outils de communication : courriel, forums de discussion, bavardage, tableau blanc, KiLA, Cumuli ;
- les outils de production : "A rendre" (espace où l'on soumet son travail), Publication (espaces de publications dans le groupe), Tests (utilisé quelquefois pour les votes) ;
- les outils de présentation des contenus et d'organisation du travail : Programme (consignes générales pour SimuLigne, liste des espaces de travail du e-monde, guide méthodologique d'apprentissage), Agenda, Contenus (consignes pour les activités, classées par étapes), Glossaire, "Ma participation" (listage des pages consultées, nombre de messages envoyés, etc.).

2.8.10 - Agent / Système Multi-Agents / rôle MaSE / Tâche informatique concurrente / Objectif MaSE

Un agent est une entité constituée d'un ensemble de processus informatiques communiquant entre eux pour accomplir un objectif donné. Dans ce mémoire le mot "agent" sera utilisé avec cette seule acception informatique. Un agent est donc nécessairement artificiel et non un acteur humain. Un système multi-agents est un ensemble d'agents hétérogènes communiquant entre eux pour accomplir un objectif global identifié.

Un rôle est une terminologie particulière à la méthodologie MaSE (DeLoach et al., 2001) et à plusieurs méthodologies de spécification des agents. Il s'agit d'une entité devant accomplir un certain nombre d'objectifs ou de tâches concourant à l'accomplissement de l'objectif global du système multi-agents. Cette entité peut évoluer vers une classe d'agents lorsque l'on avance dans la spécification du SMA. Chacun des rôles identifiés sera affecté à au moins une classe d'agents, ce n'est qu'ainsi qu'on peut s'assurer que le SMA accomplira toutes les tâches identifiées et concourant à l'accomplissement de son objectif global.

Chaque rôle doit accomplir un ensemble de tâches informatiques (on parlera simplement de "tâches" quand il n'y a pas de risque de confusion avec les tâches à visées pédagogiques telles

que définies ci-avant) pour atteindre les objectifs ou sous-objectifs qui relèvent de lui. C'est pourquoi on parle dans MaSE de tâches concurrentes pour exprimer que ces processus s'exécutent de façon parallèle.

Dans MaSE, le système multi-agents doit avoir un objectif global et cet objectif est divisé en sous-objectifs. Chacun des objectifs identifiés sera affecté à au moins un rôle.

2.8.11 - Interaction / Interactivité / Communication

Le problème avec ces mots est qu'ils sont aussi bien utilisés en informatique qu'en sciences du langage et que certains représentants de ces disciplines déniaient parfois aux collègues de l'autre discipline le "droit" de les utiliser. On peut se passer d'une telle querelle. Pour ne prendre que deux exemples, le lecteur consultera les deux acceptions de l'adjectif "interactif" dans le Petit Robert (édition 1996) et on rappellera que c'est la définition du terme "communication" par les mathématiciens et informaticiens et les travaux afférents qui ont servi de base à l'étude de la communication en sciences du langage (sur ces aspects et, plus généralement, pour une discussion détaillée de ces 3 termes, on se reportera à (Chanier, 2000)).

Or dans ICOGAD, notre problématique nous situe justement dans les deux domaines : nous nous intéressons à la communication entre humains via le système informatique PIFAD (dans SimuLigne la communication est toujours médiatisée par les SIC), à la communication entre agents SIGFAD et à la communication entre agents et acteurs, voire à celle plus générique baptisée de façon standard interaction humain-machine (IHM).

Un des tous premiers termes utilisés pour caractériser le couple action-réaction entre humain et système informatique est interactivité, terme aux interprétations variées comme l'ont bien compris les revendeurs de logiciels désireux d'attirer le client. Dans une acception courante, est interactif un système qui permet des interactions, au sens d'influence réciproque. On parlera de degrés différents d'interactivité suivant la complexité des échanges. Un premier degré d'interactivité est celui où l'utilisateur communique quelque chose au système qui se contente de l'exécuter. Le degré deux de l'interactivité se rencontre lorsque le système ne se contente pas de laisser agir la personne mais offre une réaction, au sens où il informe sur l'opération en cours. On a alors un début de relation bi-directionnelle entre le système et l'utilisateur. Un nouveau degré sera atteint lorsqu'il sera réellement possible à l'apprenant de

transformer et d'agir sur les informations/représentations mises par le système à sa disposition, lorsque le système sera capable d'analyser les échanges en cours afin de mieux s'adapter à l'apprenant. En considérant ce degré d'interactivité, on perd de vue sa connotation d'immédiateté, de court délai entre l'action-décision de l'apprenant et la rétroaction du système. On ne considère plus en fait l'interaction apprenant-système au sens courant d'interdépendance ou action-rétroaction, mais au sens, plus technique, de constitution d'un mode conversationnel impliquant un dialogue apprenant-système. Cette acception technique en informatique du terme "interaction" correspond assez bien à l'acception technique rencontrée dans une partie de la linguistique pour laquelle l'interaction verbale et non verbale entre individus est considérée comme un lieu où se modifient, se manifestent des formes sociales, cognitives, linguistiques d'organisation humaine. Les chercheurs dans une partie des sciences humaines considèrent ainsi l'interaction (conversation, dialogue, etc.) comme un lieu privilégié d'observation de savoir et savoir-faire en construction ... toute chose qui nous intéresse particulièrement ici.

Dans ce mémoire, nous aurons donc tendance à utiliser chacun des termes de façon distincte (sans pouvoir trancher de façon absolue, comme expliqué précédemment), à savoir :

- nous réserverons le terme interactivité à la caractérisation de la communication entre acteur et agent (HM) (donc non pas entre agents (MM), ni entre acteurs (HH)) ;
- le terme communication désignera de façon très générale tout échange d'informations de type HH, MM ou HM, l'échange pouvant être réduit à une simple intervention. Sauf indication contraire ce terme ne présuppose aucune complexité dans ces échanges/interventions, ni dans le contenu des messages (ce peut-être un simple flux d'informations), ni dans la structuration du dialogue (qui peut se limiter à une simple intervention).
- l'expression "une interaction" se référera à un élément constitutif d'une communication plus élaborée de type HH, HM ou MM, c'est-à-dire :
 - dont le contenu peut être interprété sur le plan cognitif (lorsqu'on est dans une situation HM ou HH), même si l'interprétation n'est pas immédiate, ou interprété comme étant rattaché à l'objectif guidant un agent (type HM ou MM)
 - dont la nature est soit un message, soit une action au sens de la théorie de l'activité pour lesquels on peut identifier un émetteur / initiateur et au moins un destinataire.

- cet élément doit pouvoir être mis en relation avec un élément de même niveau créé par le destinataire. Le premier élément peut donc être en position d'initiateur (dont la réponse/réaction s'accomplira ou non) ou de répondeur. Ainsi dire systématiquement qu'un courriel est une interaction, sans le mettre en relation avec un autre élément de même niveau, est commettre un abus de langage.

Comme l'étude des interactions est devenu un véritable domaine de recherche en sciences humaines et en informatique, on pourra parler, par extension, de "l'interaction" pour référer à une communication dans laquelle il est possible d'identifier un ensemble d'interactions.

**Chapitre 3 - L'expérimentation SimuLigne : le
modèle conceptuel d'activité et l'analyse des données
d'interactions dans les groupes**

Ce chapitre est celui dans lequel nous justifierons la construction d'un système multi-agents destiné à être couplé aux plates-formes de télé-formation en vue de mettre à la disposition des utilisateurs un environnement informatique propice au support de l'apprentissage collaboratif distribué. Notre démarche est double. Nous montrerons d'abord que sur le plan conceptuel, les plates-formes existantes ne sont pas adaptées à l'apprentissage collaboratif distribué. Pour cela, nous mettrons en perspective le modèle conceptuel d'activité de SimuLigne et le schéma général de PIFAD. Ceci nous permettra de voir que des concepts centraux dans le modèle conceptuel sont difficilement implémentables dans PIFAD.

Dans un second temps, nous montrerons qu'à partir des données d'interactions intra-groupes on peut d'une part, apprécier l'état du groupe et l'évolution de la session de FAD et d'autre part, faire des prévisions raisonnables sur la durabilité des groupes et le comportement futur des utilisateurs. Cette démarche prospective permettrait au tuteur de prendre des mesures appropriées lorsque le groupe court un risque d'éclatement ou quand des utilisateurs sont en situation d'échec. Nous rappelons que les interactions au sein du groupe restent pour nous la preuve que le groupe existe en tant que tel, évolue normalement et est dynamique. Les interactions sont, rappelons-le, porteuses de changements cognitifs au sein du groupe (Dillenbourg et al., 1996) (Dillenbourg, 1999). Toutes les théories qui étayent l'apprentissage collaboratif s'accordent sur ce qu'*un groupe qui interagit apprend*. Le système multi-agents que nous proposons permet non seulement de résorber les insuffisances conceptuelles des plates-formes existantes mais fournit aussi aux utilisateurs des outils nécessaires à la réification de la participation des utilisateurs et au soutien de leurs activités pédagogiques. Ce système vient compléter les LMS afin d'en faire véritablement des environnements informatiques support de FAD.

3.1 - SimuLigne

3.1.1 - Contexte de départ

Le projet ICOGAD, « Interactions et changements cognitifs dans les groupes d'apprentissage à distance médiatisé par les systèmes d'information et de communication », est un projet de recherche universitaire initié par l'équipe SICAH du Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC) et s'inscrivant dans le cadre du Programme du

Ministère de la Recherche Cognitive 2000⁵, thème « Nouvelles technologies et cognition », sous-thème « Applications pédagogiques des systèmes d'information et de communication ». Le projet ICOGAD concerne à la fois l'équipe SICAH du LIFC, le Département des Langues Vivantes de l'Open University (OU) et le Laboratoire de Psychologie de l'Interaction de l'Université de Nancy 2 ou GRC (Groupe de Recherche sur les Communications).

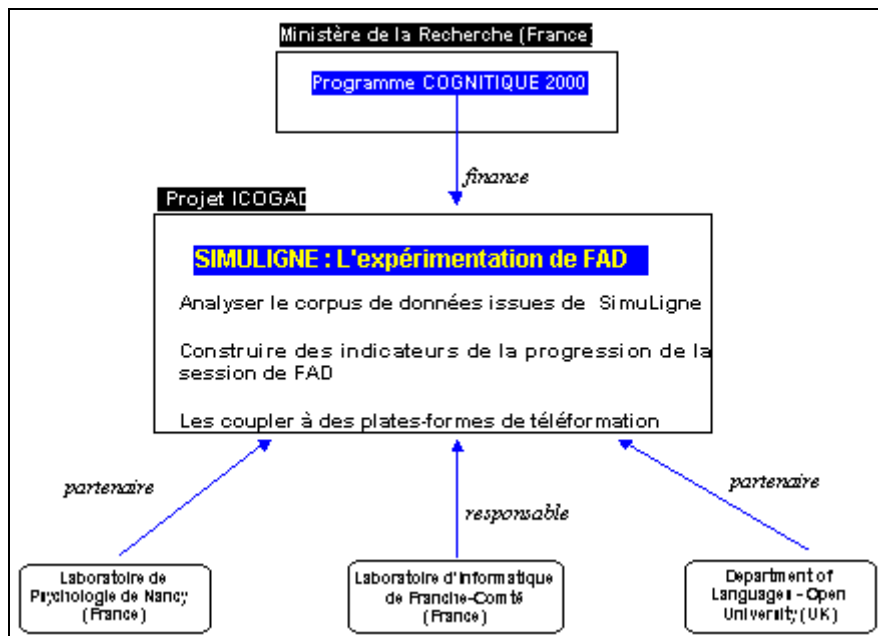


Figure 3.1 : le contexte de départ de SimuLigne

A l'origine de ce projet, avant tout le désir de mener des recherches touchant des situations réelles d'apprentissage totalement à distance déclinées au moyen d'un environnement logiciel de campus virtuel (le choix s'est porté sur la plate-forme WebCT®) et situations concernant des groupes de taille limitée (8 à 15 membres), chacun des partenaires impliqués ayant bien sûr des objectifs spécifiques variables mais complémentaires.

Pour la mise en œuvre de SimuLigne, la répartition du travail entre partenaires a pris la forme suivante :

- première adaptation à la distance d'une méthode de formation réputée en langue en présentiel : la simulation globale : choix du sujet (OU), développement du scénario (OU et LIFC), des ressources pédagogiques (LIFC avec retour OU) ;

⁵ Ce programme visait à encourager les recherches interdisciplinaires sur la cognition en favorisant les collaborations entre, d'une part, les sciences humaines et sociales, et, d'autre part, les sciences de la vie et les sciences de l'information.

- recrutement des acteurs de la formation (sujets de l'expérience) : 400 apprenants de français langue étrangère de l'OU contactés et 40 volontaires (OU) ; recrutement de 4 tuteurs/formateurs de langue (OU) ; recrutement de 10 natifs francophones (LIFC) ;
- mise en place de la plate-forme de téléformation et développement du prototype CUMULI (LIFC) ;
- formation à distance des tuteurs à l'utilisation de la plate-forme (LIFC) ;
- coordination de la formation, suivi, assistance technique (LIFC).

3.1.2 - Principes et objectifs pédagogiques de SimuLigne.

Le scénario de base de la formation devait être en adéquation avec les objectifs de recherche des différents partenaires du projet. Ainsi, la formation fut conçue dans l'esprit de ce que la terminologie de l'enseignement/apprentissage des langues appelle « la simulation globale » (d'où d'ailleurs le nom de la formation « SimuLigne » pour Simulation en Ligne). SimuLigne s'inspire donc des principes de la simulation globale telle que pratiquée en formation présentielle. La simulation globale est connue pour être largement *propice aux interactions* d'une part, pour amener d'autre part les apprenants à *collaborer sur un projet commun* en adéquation avec les principes d'organisation de la mise en place d'un travail collaboratif au sein d'une communauté d'apprentissage : phase de découverte, phase de négociation-distribution des rôles et du travail, réalisation du projet.

3.1.2.1 - La simulation globale

C'est dans les années 1970 que le principe de la simulation globale est apparu dans le cadre de l'enseignement du français langue étrangère (FLE). Partant du principe selon lequel, pour apprendre une langue, il est essentiel de mettre en pratique ses connaissances, de les utiliser en situation, donc d'être en situation de communication, il prôna le recours à des situations de simulation pour apprendre la langue. L'expression même de "simulation globale" met en évidence le souci de dépasser les situations brèves de simulation simple. Plus ambitieuse en effet que le jeu de rôles, la simulation globale consiste à faire inventer par un groupe d'apprenants un univers de référence – un immeuble, un village, une île, un hôtel – à animer cet univers de personnages en interaction et d'événements, et à simuler toutes les fonctions du langage que ce cadre, qui est à la fois un lieu thème et un univers du discours, est susceptible de requérir. Elle permet donc de passer à des simulations plus longues dans le temps et

engageant de façon beaucoup plus complète la pluralité des aspects communicatifs. Debyser écrit que la simulation globale consiste à créer "*progressivement avec le groupe-classe un univers du discours complet, avec son cadre, ses personnages, leur vécu individuel et collectif et leurs relations, leurs interactions ; c'est la construction collective d'un monde vécu, d'une symbolique et donc d'une culture partagée, ... en langue étrangère*" (Torrent, 2001).

Le principe de la simulation est transposable à d'autres contextes que celui de l'enseignement du FLE (enseignement d'autres langues, stages de formation en entreprise...). Le modèle de la simulation globale tel que pratiqué en formation présentielle pouvait désormais s'appliquer à distance grâce aux environnements technologiques qui réunissent et font se rencontrer des apprenants, afin qu'ils puissent travailler et apprendre ensemble, en collaborant. Seraient donc mis à profit tous les outils virtuels de PIFAD qui favorisent la communication de groupe, synchrone ou asynchrone, en misant sur la participation active des apprenants, sur le partage et sur l'entraide.

La simulation globale demande de procéder d'abord à la définition d'un lieu-thème, univers de référence, et à sa construction (création du décor). L'opération consiste à faire "débarquer" l'imaginaire d'un groupe d'apprenants sur un lieu qui fonctionne comme un milieu et comme un thème (une île, un immeuble, un village, un bateau de croisière...). Ce lieu-thème permet de convoquer et de fédérer toutes les activités d'expression écrite et orale traditionnellement menées dans la classe de façon atomisée, de les coordonner les unes aux autres, de faire que ces activités soient des activités de réflexion, de débat, de créativité... L'objectif consiste à donner un sens, une dynamique, à la construction collective d'un univers dans lequel *chacun sera partie prenante*. Dans un deuxième temps, les participants procéderont à l'élaboration d'identités fictives. Chacun mettra entre parenthèses son identité réelle pour se glisser dans la peau d'un personnage qu'il va incarner, et auquel il va donner une âme. En bref, il s'agit de "faire comme si" on était quelqu'un d'autre, "comme si" on vivait autre part. Autrement dit : sortir de son univers par la puissance de l'imagination. Chaque apprenant, après avoir pris une identité fictive, participera à des événements. Ce type d'activité est essentiellement pratique et demande une participation active.

Le caractère ludique de la démarche n'échappera à personne. Elle est une sorte de jeu : on fait semblant d'être en situation réelle pour apprendre la langue en la vivant. Le jeu représente un atout évident dans la mesure où il peut éveiller l'intérêt et la motivation des apprenants.

Cependant, l'attrait de la simulation globale pour la formation SimuLigne était autre : la simulation globale est en fait un projet de création collective qui engage l'ensemble des participants dans l'invention d'un monde imaginaire. Les capacités de chaque membre d'un groupe sont mises en commun pour la réalisation d'un projet collectif ; on partage une expérience de vie en communauté en vue toujours d'enrichir ainsi sa connaissance et sa pratique de la langue. L'enseignant est, au sein du groupe, la personne-ressource ; il est censé intervenir pour donner les consignes, des informations socio-culturelles et faire quelques points méthodologiques. Cette conception du rôle de l'enseignant correspondait globalement à notre vision du tutorat tel que nous souhaitions le voir se mettre en place dans SimuLigne.

3.1.2.2 - Le scénario prétexte de SimuLigne

Pour SimuLigne, le scénario de la simulation s'inspire d'une réalité : chaque année en effet, l'Open University organise pour ses étudiants des stages d'été ou universités d'été de quelques semaines en présence dans une ville universitaire française. L'idée était donc la suivante : imaginer qu'une grande université britannique, en l'occurrence, pourquoi pas, l'Open University, soit à la recherche d'une ville universitaire française qui réponde idéalement à divers critères en vue d'y organiser pour les dix prochaines années la totalité de ses stages linguistiques d'été. Il s'agissait donc, pour chaque apprenant de contribuer à la création d'une ville universitaire (imaginaire mais qui s'inspire de réalités culturelles) qui puisse accueillir un potentiel de deux mille étudiants par an et qui réponde au mieux aux attentes des étudiants et organisateurs tant sur le plan universitaire, commercial, touristique ou culturel, qu'en matière d'infrastructures.

L'apprentissage collaboratif étant au cœur des motivations de départ, la formation devait réunir les conditions favorables au travail en groupe. Aussi, du fait du nombre d'apprenants recrutés, quatre groupes, autrement dit quatre communautés virtuelles d'apprentissage, furent créées. Chaque groupe aurait à construire, en six semaines, sa propre ville avec ses lieux, ses personnages, ses rencontres, et ses événements pour présenter, au terme de ces six semaines, sa candidature au concours "Open city-Ville Ouverte" organisé par l'université britannique supposée à l'origine du projet du même nom. Les diverses productions des groupes seraient rassemblées de telle façon que chaque groupe puisse soumettre une sorte de dossier de candidature se présentant sous la forme d'un poster, c'est-à-dire, en quelque sorte, un ensemble de pages HTML à partir desquelles on aurait accès à des images (plans, photos, par

exemple), des documents écrits (descriptifs des lieux, des personnages qui animent les lieux décrits), ou des enregistrements audio. Les quatre posters devaient donc être soumis, dans le groupe le «Monde SimuLigne», à l'ensemble des personnes concernées par la formation (tuteurs, natifs, administrateurs, concepteurs, chercheurs, gestionnaire de formation) en vue du vote final, pour décider du groupe vainqueur, pour décider quelle ville mériterait d'être retenue par l'université britannique. Les gagnants du concours recevraient un cadeau en récompense de tout leur travail.

3.1.2.3 - Modalités de formation

Dans le dispositif de formation à distance de SimuLigne, apprenants et tuteurs ne se connaissaient pas au préalable, et sont restés géographiquement isolés pendant toute la formation. Les étudiants de l'Open University ont entre 25 et 65 ans, sont en situation de reconversion ou de perfectionnement professionnel, ou ont d'autres motivations plus personnelles. Ils ont des obligations familiales et/ou professionnelles parallèles à celles de la formation. Les tuteurs SimuLigne ont été sélectionnés parmi les enseignants de langue contractuels de l'OU, c'est-à-dire, ceux qui exercent habituellement la fonction de tuteur à distance. Autre situation caractéristique de la FAD, chaque acteur a dû travailler dans son environnement propre, la plupart du temps à la maison, sur son ordinateur personnel, doté d'un traitement de textes, d'un navigateur standard et relié à Internet par une liaison téléphonique ordinaire.

La liaison télématique permettait d'accéder au serveur Internet qui hébergeait la plate-forme de téléformation (ici WebCT™). Dans la plate-forme, l'équipe de conception de SimuLigne a aménagé un ensemble considérable d'espaces de travaux, distincts mais souvent reliés. Ainsi les quatre groupes ont travaillé dans des espaces différents interdisant toute intrusion par un membre extérieur ; mais, d'une part, la coordination générale entre les quatre groupes était assurée dans un espace hors groupes réservé aux tuteurs et, d'autre part, chaque membre de la formation SimuLigne avait accès à un espace commun général pour y accomplir certaines activités interculturelles (InterCulture). A l'intérieur de chaque groupe, étaient aménagés des espaces strictement individuels, des lieux de rencontres ou d'échanges par binômes, en petits sous-groupes ou pour tout le groupe, lieux limités à l'exécution de tâches spécifiques ou à des discussions sur des thèmes fixés, etc.

Étapes et activités SimuLigne

Rappelons que l'environnement de formation était organisé autour de quatre groupes de base, "Aquitania", "Gallia", "Narbonensis", et "Lugdunensis". L'essentiel de la formation SimuLigne s'est d'ailleurs déroulé dans ces groupes de base, en ce qui concerne en particulier la construction de la ville universitaire idéale par chaque groupe. La simulation fut donc jouée dans chacun des groupes de base avant de se conclure dans le «Monde SimuLigne», communauté qui regroupe l'ensemble des participants.

Pour ce qui est de la simulation globale, SimuLigne fut organisée en trois étapes réparties sur dix semaines. Pour les étapes E1 et E2, chaque participant travaillait dans son groupe de base d'appartenance. Pour l'étape E3, une large part des activités s'effectuait dans le «Monde SimuLigne». SimuLigne comportait une étape E0, qui a servi d'étape de socialisation. Sans vouloir nous attarder longuement sur cette étape, rappelons qu'elle est fondamentale dans la formation à distance ; c'est ici que les apprenants et le tuteur vont se connaître et commencer à former le groupe : nous avons vu que cette étape est importante dans le modèle de Salmon (2001). Lors de cette étape, les apprenants devaient se familiariser avec les outils « forum » (E0A1), « A Rendre » (E0A2), « bavardage » (E0A3), puis se présenter aux autres membres du groupe (E0A4 et E0A5). En marge des activités de SimuLigne qui se sont déroulées pour l'essentiel dans les groupes de base, il avait été organisé InterCulture, qui se déroulait dans le Monde SimuLigne. Il s'agissait d'activités interculturelles destinées à appréhender les différences culturelles entre les anglophones et les francophones. InterCulture consistait à réagir à des mots, phrases et situations en remplissant des formulaires. L'ensemble des participants à l'expérimentation devait ensuite prendre connaissance des différentes réponses des uns et des autres et participer à des forums de discussion. Ici les apprenants avaient la possibilité de s'exprimer dans leur langue habituelle, c'est-à-dire l'anglais. Nous présentons ci-après, de façon détaillée, les activités constituant les étapes E1, E2 et E3 de SimuLigne.

L'étape E1

Au cours de ce qui constitua la première étape de la simulation globale, l'étape E1, du 11 mai au 01 juin 2001, les apprenants de SimuLigne étaient invités à construire leur univers de référence, autrement dit à «planter le décor» de leur ville. Il s'agissait de la mise en place du décor - la ville (mairie, services, transports) et l'université - et des personnages qui interviendraient dans la proposition de campus d'été (des collectivités locales, de l'université,

etc.). Le travail se répartissait entre les membres du groupe. Ainsi, chaque apprenant choisissait de décrire une partie de la ville ou de l'université. De plus, chacun choisissait un personnage. Les contributions individuelles étaient affichées au fur et à mesure des activités sur des pages que seuls les membres du groupe pouvaient voir. Elles seraient par la suite présentées dans le poster final, pour le vote, en fin de simulation. Les activités de cette étape pouvaient pour certaines d'entre elles se faire en parallèle ; elles ne s'enchaînaient pas forcément les unes aux autres, d'où la nécessité (à la fois pour les apprenants, les tuteurs et les natifs) de bien prendre connaissance du calendrier de chacune d'entre elles et d'avoir une vue d'ensemble de l'étape.

L'activité «E1A1-Quatre villes françaises».

Les apprenants étaient amenés à découvrir les sites de quatre villes françaises (pages HTML téléchargeables sur PC personnel) dont ils dégagèrent les caractéristiques essentielles. C'était en quelque sorte une « activité déclenchement » pour la suite de la mise en place du décor car elle visait à leur donner des idées pour construire et décrire leur propre ville imaginaire au cours de l'activité E1A2 puis au cours de l'étape E2 où ils complèteraient cette description des lieux. Chaque apprenant remplissait un tableau comparatif fourni, en notant des informations se rapportant aux thèmes indiqués (avantages géographiques, situation économique, population, infrastructure, patrimoine socio-historique, et campus universitaire), le soumettait au moyen de la procédure «A rendre» (travail "e1a1_villes_francaises") et recevait, par courriel, une correction individuelle. Tous les tableaux étaient ensuite publiés (par le tuteur) dans le projet "e1a1_villes_francaises" de l'espace "Publications". Après consultation des publications, les apprenants étaient invités à commenter les réponses et à prendre la défense de la ville qu'ils avaient préférée parmi les quatre étudiées (avantages et qualités, utilisation de formules comparatives) dans le forum "Principal" de leur groupe.

L'activité « E1A2-Imaginer une ville »

Activité de construction et de description de leur ville. Le travail allait être partagé entre les différents membres du groupe. Certains se chargeaient de décrire, à l'aide de documents et ressources fournis, les lieux culturels (cinémas ou musées ou salles de spectacles), ou les services (mairie ou transport et office de tourisme), d'autres le campus universitaire (bâtiments, restaurant universitaire ou cité universitaire). Le choix de chacun devait être exprimé dans le forum avant la date de clôture donnée. Le tuteur publiait les contributions

respectives et préparait une synthèse en vue de l'activité E1A3. Une correction éventuelle était prise en charge, sur demande, par l'un des natifs du groupe (afin de soulager le tuteur).

L'activité « E1A3-Choisir une ville ensemble »

Dans l'activité suivante « E1A3-Choisir une ville ensemble », ils étaient amenés à situer ensemble dans l'espace, lors d'une activité synchrone, les lieux décrits en E1A2 en les plaçant sur une carte de ville et une carte de campus. Pour cela, avant la séance de bavardage/tableau blanc, ils étaient censés avoir lu les contributions de chacun ainsi que la synthèse du tuteur, et avoir examiné les plans de ville et campus proposés. Durant la séance de bavardage, ils se mettaient d'accord sur les plans puis négociaient l'emplacement des divers lieux. Le tuteur se chargeait de gérer le tableau blanc (c'est-à-dire d'afficher les plans) et donc de pointer les lieux sur les plans en fonction des discussions ayant lieu en parallèle dans le bavardage. Les dates de séance de bavardage avaient été auparavant discutées dans le forum prévu à cet effet, après que le tuteur eut fait des propositions de dates. Cinq à six personnes étant un maximum pour une bonne séance de bavardage, deux séances distinctes pouvaient avoir lieu : une pour le campus, une pour le reste des lieux avec le plan de ville (ce qui impliquait chacun des membres du groupe, indépendamment de sa contribution d'origine).

L'activité « E1A4 : Le portrait de votre personnage »

A l'aide de documents tels qu'un article de Libération ("Les quatre origines des patronymes français"-08/02/2001), un lien sur le site de la chaîne de télévision TV5 et une liste de photos de personnages, les membres du groupe se construisaient une nouvelle identité (choix d'un nom, caractéristiques du personnage). Cette activité pouvait être faite en parallèle avec la description des lieux, elle pouvait donc commencer dès le début de l'étape E1. Comme pour l'activité E1A2, ils pouvaient compter sur l'aide des natifs de leur groupe pour corriger leur fiche de présentation avant de la soumettre. Ils indiquaient leur nom et prénom fictifs, l'étymologie du nom, la photo choisie, l'adresse, des caractéristiques physiques et morales, des objets préférés, des événements marquants, leur devise, une rumeur.... Les productions étaient publiées et pourraient être commentées dans le forum.

L'activité « E1A5 : Présenter sa fonction »

Une liste de fonctions qui pouvaient être attribuées à leur personnage, personnage susceptible d'être amené à intervenir au sein du Comité Ville Ouverte, avait été proposée (universitaire local, représentant de l'Open University, animateur culturel / animateur musical, responsable

logement en cité universitaire, chef cuisinier, responsable des menus au restaurant universitaire, directeur de cinéma/théâtre/opéra, conseiller municipal/Maire, représentant(e) de l'association des commerçants). Ils ne se contentaient pas de choisir ; il fallait effectivement motiver son choix, c'est-à-dire expliquer au Comité Ville Ouverte (qui s'occupe du montage de dossier pour l'université d'été) ce que ce personnage pouvait leur apporter. Il s'agissait donc d'envoyer un message (texte ou audio) au Comité Ville Ouverte pour proposer ses services et quelques idées intéressantes et convaincantes. A l'issue de l'activité, les personnages ainsi définis pouvaient alors participer aux échanges et interactions de l'étape E2. Ces productions toujours affichées par le tuteur constituaient également une partie du poster final (visible de tous les apprenants en étape E3).

Première connexion	Etape E0 Socialisation et prise en main technologique	Etape E1 Ville et Identité	Etape E2 Faire vivre l'espace	Etape E3 Vote et Bilan
1/04 – 22/04	30/04 – 13/05	14/05 – 1/06	1/06 – 23/06	25/06 – 6/07
Bienvenue	EOA1 : Discutons dans un forum	E1A1 : Quatre villes françaises	E2A1 : Décrire notre ville	E3A1 : Découvrons puis votons !
Questionnaire Personnel	EOA2 : Ma biographie	E1A2 : Imaginer une ville	E2A2 : Hymne et récit fondateur	E3A2 : Publication des résultats
Questionnaire Informatique	EOA3 : Bavardons quelques instants	E1A3 : Choisir une ville ensemble	E2A3 : Prise de contact	E3A3 : Bilan de la formation
Questionnaire F.A.D.	EOA4 : Présentation personnelle	E1A4 : Le portrait de son personnage	E2A4 : Interactions	
	EOA5 : Présentation orale	E1A5 : Présenter sa fonction	E2A5 : Incidents et événements	

Interculture 0	Interculture 1	Interculture 2
30/04 – 13/05	14/05 – 3/06	11/06 – 1/07
	I1P1 : Réagir à des mots	I2P1 : Réagir à des phrases et situations
Préparation à Interculture	I1P2 : Voir les résultats	I2P2 : Voir les résultats
	I1P3 : Forum sur les mots	I2P3 : Forum sur les phrases et situations

Figure 3.2 : Calendrier des activités de SimuLigne + InterCulture

L'étape E2

Pendant la deuxième étape, l'étape E2 (du 4 au 23 juin 2001), ils allaient animer les lieux et personnages imaginés lors de l'étape E1. Cette étape permettait d'affiner la description des décors de leur ville, et de faire vivre les personnages amenés à se rencontrer, s'écrire,

interagir, résoudre de petits incidents, échanger des points de vue, en un mot à communiquer. Des problèmes inattendus survenaient, le groupe devait alors les résoudre. Les contributions de chacun, une fois soumises, étaient là aussi affichées sur des pages visibles des membres d'un groupe seulement. Dans cette étape, les groupes étaient appelés à prendre des décisions pour faire des choix parmi diverses contributions ou productions (par exemple, choisir le nom de la ville) et ne conserver que les meilleures ou les plus adaptées pour la ville construite, en vue de l'étape 3.

L'activité « E2A1-Description générale de la ville »

Une description synthétique (200 mots) de leur ville (présentation des caractéristiques à la fois du point de vue géographique, culturel, démographique et économique) allait être faite. Pour ce faire, ils s'inspiraient des ressources mises à leur disposition : photos de paysages, des mots clefs - situation/région, relief/hydrographie, climat, végétation, nombre d'habitants, classes d'âge, emploi, formation, agriculture, industries, services, infrastructures, histoire/événements, traditions, monuments, gastronomie - une liste de noms de villes et villages pour comprendre la règle de formation des noms et en inventer un pour leur ville en accord avec une des caractéristiques imaginées. En fin d'activité, après publication, un vote exprimait le choix du groupe parmi toutes les propositions faites.

L'activité « E2A2-Histoire et hymne de la ville »

Elle dépendait de E2A1, puisqu'il allait s'agir de composer l'hymne et le récit fondateur de la ville, en laissant libre cours à son imagination certes, mais en cohérence avec E2A1. Pour cette activité, le travail se faisait en deux sous-groupes. Chaque apprenant d'une première moitié du groupe travaillait sur une proposition de récit fondateur (avec des ressources pour les aider : des exemples de récits, un ensemble de questions : quand la ville fut fondée, comment, pourquoi en ces lieux, qui sont les figures emblématiques, ...). L'autre moitié du groupe composait de manière collaborative un hymne (après avoir lu ou écouté, à titre d'exemple, quelques chansons françaises telles que « Toulouse » de Claude Nougaro) ; il s'étoffait au fil des contributions dans un forum ouvert à cet effet, chacun y allant de ses propositions selon des consignes précises (dates limite des interventions, nombres de vers, son terminant les vers). Le tuteur était en quelque sorte le chef d'orchestre : il annonçait par courriel la tâche que chacun aurait à réaliser. En dépit de cette répartition du travail, le groupe tout entier participait au vote de fin d'activité pour décider du récit retenu parmi les diverses propositions ; le récit choisi est présenté dans leur dossier final. E2A2 se terminait avec la fin

de l'étape E2 (le 23 juin 2001), le travail sur les lieux de la ville se répartissant ainsi sur toute cette étape. En parallèle, il s'agissait de donner vie aux personnages en les faisant interagir.

L'activité « E2A3-Prise de contact »

Dès le début de l'étape 2, les apprenants étaient amenés à travailler par binôme (en fonction de leur identité fictive définie en E1A4 et E2A5) et à entrer en contact pour traiter d'un problème particulier. Cette activité demandait une bonne organisation : l'un des deux était l'initiateur de cette situation de communication par l'envoi d'un premier courriel et était en attente d'une réponse de la part de son partenaire en vue de l'activité E2A4. Le représentant de l'Open par exemple contactait le chef cuisinier, responsable des menus au restaurant universitaire, pour réclamer des menus végétariens. Le chef cuisinier préparait alors une série de suggestions de menus, ainsi qu'une proposition de soirée gastronomique pour les étudiants de l'Open et envoyait le tout par courriel à son correspondant. L'échange se poursuivait et se concluait en E2A4. Le tuteur recevant copie de ces courriels les publiait à la fin de l'activité E2A4. Les natifs pouvaient être ici sollicités pour aider à la rédaction de ce courriel, ou pour toute autre difficulté.

L'activité « E2A4-Interactions »

Cette activité prolongeait la précédente et permettait, lors d'un contact synchrone (séance de bavardage), de prendre les décisions finales, de conclure la négociation entamée au cours de E2A3. Chaque binôme parvenait donc à trouver un compromis. Quelques propositions par binôme dans la fiche-consigne pouvaient servir de trame de départ, source d'inspiration. Un natif de Besançon était présent dans toutes les séances de bavardage. E2A4 se terminait avec la fin de l'étape 2.

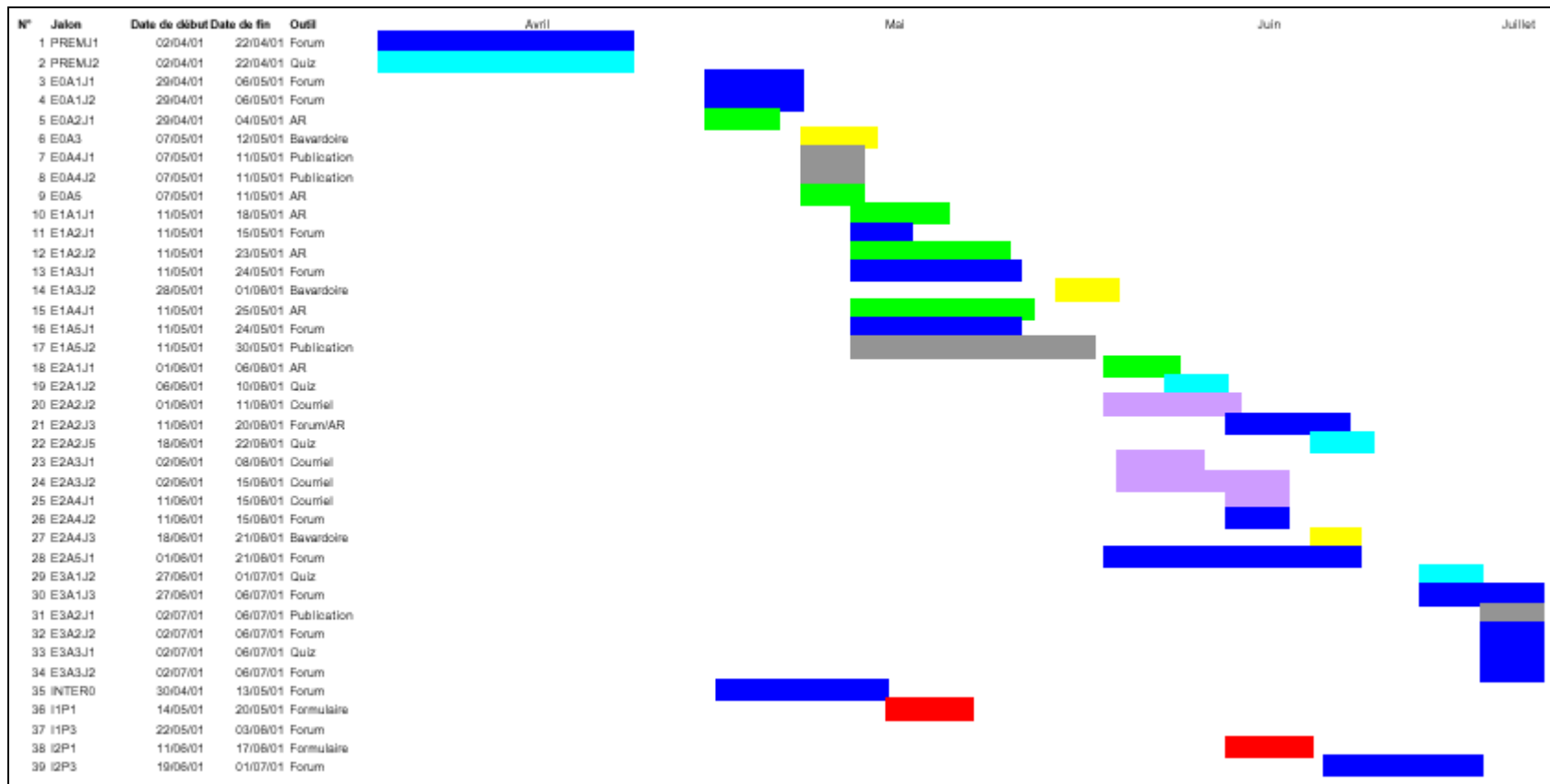


Figure 3.3 : Le calendrier des activités de SimuLigne + InterCulture – Jalons et outils associés (une couleur par outil)

L'activité « E2A5-événements et incidents »

Chaque apprenant était invité à réagir et à s'exprimer, sans affectation de fonction précisément définie, en vue de la résolution de problèmes d'ordres divers (événements imprévus, petites catastrophes : restaurant universitaire hors d'usage, une grève des transports, une double réservation d'amphithéâtre, un bâtiment inondé, des conditions d'accessibilité non prévues pour les étudiants handicapés à adapter) présentés dans le forum "e2a5_incidents" (cinq fils de discussion étaient lancés par le tuteur dans le forum ouvert à cet effet dès le 1er juin 2001, début de l'étape). Deux messages au moins devaient être postés par chaque apprenant dans deux fils de discussion (minimum). La synthèse des échanges les plus riches était affichée dans le groupe puis introduite dans le poster final.

L'étape E3

L'étape finale : du 25 juin au 6 juillet 2001. Une grande partie des activités se déroula dans le « Monde SimuLigne ». L'équipe technique de Besançon rassembla les matériaux produits et sélectionnés dans chaque groupe afin de réaliser les quatre posters et les afficher dans le « Monde SimuLigne ». Tous les participants pouvaient donc découvrir la présentation finale du projet de "Open city-Ville Ouverte" de chaque groupe, et participer au vote pour choisir le meilleur en fonction de critères donnés et commenter les projets et productions dans le forum « Principal » de son groupe ou du « Monde » (Activité « E3A1-Découvrons puis votons ! »). Dans un deuxième temps étaient publiés les résultats ; tout le monde pouvait en discuter là aussi dans le forum « Principal » de son groupe ou du « Monde » (Activité « E3A2-Publication des résultats »). Enfin, un bilan des dix semaines de formation se faisait sous trois formes : dans des forums bilan (un par groupe « Bilan_Aquitania » ou « Bilan_Gallia » ou « Bilan_Lugdunensis » ou « Bilan_Narbonensis ») ouverts à cet effet, et à l'aide d'un questionnaire (questionnaire de recherche confidentiel à renvoyer, par l'adresse externe à WebCT) pour l'activité « E3A4-Bilan formation : questionnaire et forums » ; enfin, dans CUMULI⁶ (Activité « E3A4-Bilan formation avec CUMULI »). Puis ce serait le temps des envois du certificat de participation et du cédérom.

SimuLigne privilégiait une pédagogie de construction des connaissances par les apprenants à travers diverses tâches, et dans l'interaction ; les technologies étaient au service d'un modèle de formation qui visait l'implantation d'une pédagogie de construction des savoirs par

⁶ CUMULI est un prototype développé par Reffay et Chanier (2001) en vue de construire une mémoire collective de groupe dans des échanges asynchrones à l'aide des forums.

l'apprenant, où les démarches et le travail collaboratifs constituaient le pivot de l'apprentissage. Les activités décrites ci-dessus laissaient à l'apprenant une large part de responsabilité face à son apprentissage, mais également face aux autres membres du groupe, puisque de ses contributions dépendait pour partie, voire pour beaucoup, la construction du projet commun du groupe, c'est-à-dire le poster final.

3.1.2.4 - Modalité d'évaluation à l'issue de la formation, bilan.

Cette formation ne devait déboucher sur aucune évaluation formelle ni même sur aucune certification diplômante. Il s'agissait d'une participation sur la base d'un simple volontariat dans le cadre d'un projet pilote pour l'Open University d'une part, d'une expérimentation universitaire, d'autre part. En d'autres termes, aucune activité destinée à vérifier les compétences acquises par les apprenants à la fin de SimuLigne ne fut organisée.

On peut cependant considérer le vote final, dans le cadre du concours « Open city-ville ouverte », comme une forme d'évaluation par les pairs. Il s'agit en effet de porter un jugement sur l'ensemble des productions ; dans le contexte d'une co-évaluation, on peut également mentionner le fait que, dans chacun des groupes, des choix devaient être faits pour la présentation de tel ou tel document (choisis en fonction d'une certaine qualité linguistique et culturelle). Rappelons que chaque apprenant a remis diverses productions qui pouvaient être corrigées par la tutrice ou les natifs. Les posters pouvant, enfin, être téléchargés par les apprenants (fichiers ZIP fournis), ils devenaient l'élément final d'un portfolio individuel. Enfin, cette formation fut en quelque sorte validée par l'envoi d'un certificat de participation, officiel, individuel et indiquant les modalités de formation.

En termes de bilan de la formation, on peut noter que des quatre groupes de base définis au début de SimuLigne, trois sont allés jusqu'au bout et ont effectivement produit un poster final (cf. Annexe A). Le groupe Lugdunensis a été fermé le 30 mai 2001, soit un mois après le début effectif de la formation, le 30 avril 2001; nous ne prenons pas en compte les activités préparatoires aussi bien pour les tuteurs, les natifs que les apprenants. Un tel taux d'achèvement est excellent dans le monde de la FAD où les taux d'abandon sont particulièrement sévères, et ce d'autant plus que notre public n'était pas captif (simple engagement moral, pas de notation, ni d'intégration possibles dans le cursus de l'OU).

En marge de la participation à la compétition virtuelle "Open City - Ville ouverte", l'ensemble du "Monde SimuLigne" a participé à une activité interculturelle baptisée "InterCulture". InterCulture consistait, sur le modèle de l'activité "Cultura" mise au point par Funsterberg, à échanger sur les perceptions culturelles des anglais et des français. InterCulture constituait une espèce de café virtuel dans lequel tout le monde pouvait s'exprimer. Plusieurs tâches de InterCulture se déroulaient en dehors de PIFAD dans un environnement bâti à dessein mais les différents forums de discussion se tenaient à l'intérieur de PIFAD dans le "Monde SimuLigne".

Sur un plan purement quantitatif, SimuLigne a donné lieu à un volume d'interactions relativement important. On a ainsi 12.356 heures de connexion (temps passé en ligne), 13.568 entrées dans PIFAD (nombre de fois auquel un utilisateur s'est connecté à PIFAD en utilisant son nom de connexion et son mot de passe). 2.686 messages ont été postés dans les forums de discussion, ces messages contenaient 879.015 caractères représentant 45% des flux de communication (en termes de nombre de caractères). 4.062 courriels ont été échangés, ils contenaient en tout 834.753 caractères, soit environ 43% des flux de communication. 5.680 tours de parole ont été pris dans les bavardages, pour un nombre de caractères de 234.694, soit environ 12% des flux de communication.

	Groupe base Aquitania	Groupe base Gallia	Groupe base Lugdunensis	Groupe base Narbonensis	Groupe Monde	Groupe Formateur
Productions	269 documents (liés aux activités) rendus par les apprenants				NA	NA
Messages dans forums	542	370	77	262	890	534
Caractères dans forums	163511	82389	23888	74465	312901	221073
Nombre de Courriels	289	328	116	350	76	642
Caractères des courriels	100377	89340	48465	127240	22288	388899
Bavardage	5679 tours de paroles (217657 caractères)					

Figure 3.4 : Bilan quantitatif de SimuLigne

3.2 - Les insuffisances conceptuelles des plates-formes de télé-formation

La théorie de l'activité (TA) proposée par Leontiev (cf. Chapitre 2) nous semble la plus appropriée pour comprendre la structure d'une activité dans SimuLigne. En effet, 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.

3.2.1 - Le modèle conceptuel d'activité dans SimuLigne

3.2.1.1 - Introduction de concepts complémentaires

Nous avons ajouté au modèle originel d'Engeström des concepts supplémentaires afin de tenir compte des spécificités de notre contexte. Il s'agit des concepts de statut, de temps et de groupe.

Le concept de statut

Pour notre conception, nous avons choisi d'introduire le concept de statut, similaire au concept de rôle défini dans Bourguin dans ses travaux de thèse (Bourguin, 2000). Le statut, comme le rôle défini par Bourguin dans ses travaux précités, synthétise les concepts de règles et de division du travail. Bourguin justifie l'introduction de ce concept par plusieurs considérations : les utilisateurs de systèmes informatiques et donc d'environnements de travail ou d'apprentissage collaboratifs sont habitués à la notion de rôle ; il est plus naturel et plus facile pour *un sujet de comprendre son rôle que de découvrir un ensemble de règles générales ainsi qu'une division du travail*. Bourguin élargit le concept de rôle utilisé dans la TA pour médiatiser la relation communauté – objet pour désormais lui permettre en plus de médiatiser la relation sujet – communauté. En réalité, Bourguin pense que le rôle tel qu'utilisé très souvent dans la TA n'introduit que des devoirs pour le sujet qui le possède, à son sens les devoirs sont à relier directement aux droits ; ce sont les droits nécessaires sur les éléments de l'activité qui permettent au sujet de tenir sa place dans la communauté. Or dans la TA, les droits sont définis à partir des règles en vigueur. C'est le souci de rapprocher les devoirs (le concept de division du travail dans la TA) des droits nécessaires (le concept de règles dans la TA) qui justifie l'introduction de la notion de rôle. Ces explications restent parfaitement valables dans notre démarche et le changement d'appellation est uniquement motivé par un souci de cohérence par rapport à notre terminologie. Pour chacune des activités de SimuLigne, chaque utilisateur a un statut particulier qui peut varier d'une activité à une autre. On peut donc assister à une redistribution des statuts par activité.

Le concept de groupe

Toujours dans un souci de cohérence par rapport à notre terminologie, nous parlerons de groupe pour désigner le concept de communauté apparaissant dans le modèle d'Engeström. Il convient de rappeler que les activités de SimuLigne sont organisées en modules collaboratifs, les sujets travaillent dans des groupes qui définissent les espaces auxquels ils peuvent accéder ainsi que l'ensemble d'outils et de ressources disponibles. Le statut de chaque sujet est défini dans le groupe et ne change pas quelle que soit la nature des activités définies dans le groupe.

Le concept de temps

Le temps est pour nous ce qui nous différencie des démarches jusqu'ici entreprises pour modéliser les activités dans la FAD. Bourguin (2000), dans les bilans et perspectives de ses travaux de thèse, inscrit l'introduction du temps comme étant une évolution attendue de son environnement DARE. Il pense que la prise en compte du temps est importante dans un environnement de TCAO pour cristalliser et réutiliser l'expérience des sujets.

Pour notre part, la notion de temps est encore plus fondamentale. En effet, nous avons indiqué que les activités dans les formations de type SimuLigne se passent de façon collaborative et coopérative, et conduisent à des productions de la part des sujets. Ceci induit de la part des uns et des autres, une synchronisation nécessaire par rapport au temps. La FAD, d'une manière générale, suppose une autonomie des sujets ; ceux-ci, du fait des distances géographiques, des décalages horaires et des disponibilités décalées, doivent mener leurs activités en tenant compte des dates de début au plus tôt et de fin au plus tard et des durées de ces activités. Contrairement à la formation dite présentielle, il est impossible d'avoir un emploi de temps qui cadre toutes les activités dans un calendrier rigoureux. Les sujets ont ainsi un souci évident d'évaluation de leur charge de travail pour ne pas se laisser déborder et se montrer défaillants, ce souci est encore plus évident dans des formations basées sur des activités collaboratives marquées par des productions. Pour en revenir de façon plus précise à SimuLigne, les activités sont organisées dans des étapes, chaque activité est un ensemble de tâches visant à élaborer un constituant de la production finale. L'apprentissage collaboratif visé dans SimuLigne induit à la fois collaboration et coopération, activités synchrones et asynchrones ; tout ceci suppose une importance particulière du concept de temps.

3.2.1.2 - Structure d'une activité dans SimuLigne

La prise en compte des concepts que nous venons de présenter nous permet de définir la structure d'une activité de SimuLigne ; la figure 3.5 présente cette structure. Cette figure montre que dans SimuLigne, un *sujet* ayant un *statut* donné dans un *groupe* utilise un *outil* pour réaliser l'*objet* d'une activité pendant un *temps* donné (en définissant une date de début au plus tôt et une date de fin au plus tard).

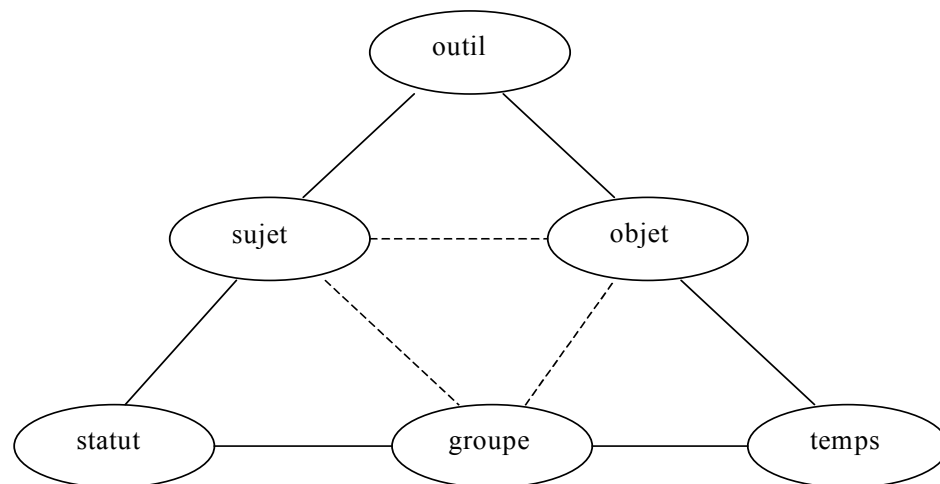


Figure 3.5: Structure d'une activité dans SimuLigne

3.2.1.3 - Le modèle conceptuel d'activité

Nous présentons à la figure 3.6 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 (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 original d'Engeström, il est donc inapplicable tel quel en FAD. En résumé, le modèle montre que chaque sujet possède un statut dans un groupe et utilise des outils (chaque tâche est liée à un outil) pour réaliser l'objet d'une activité dans un délai.

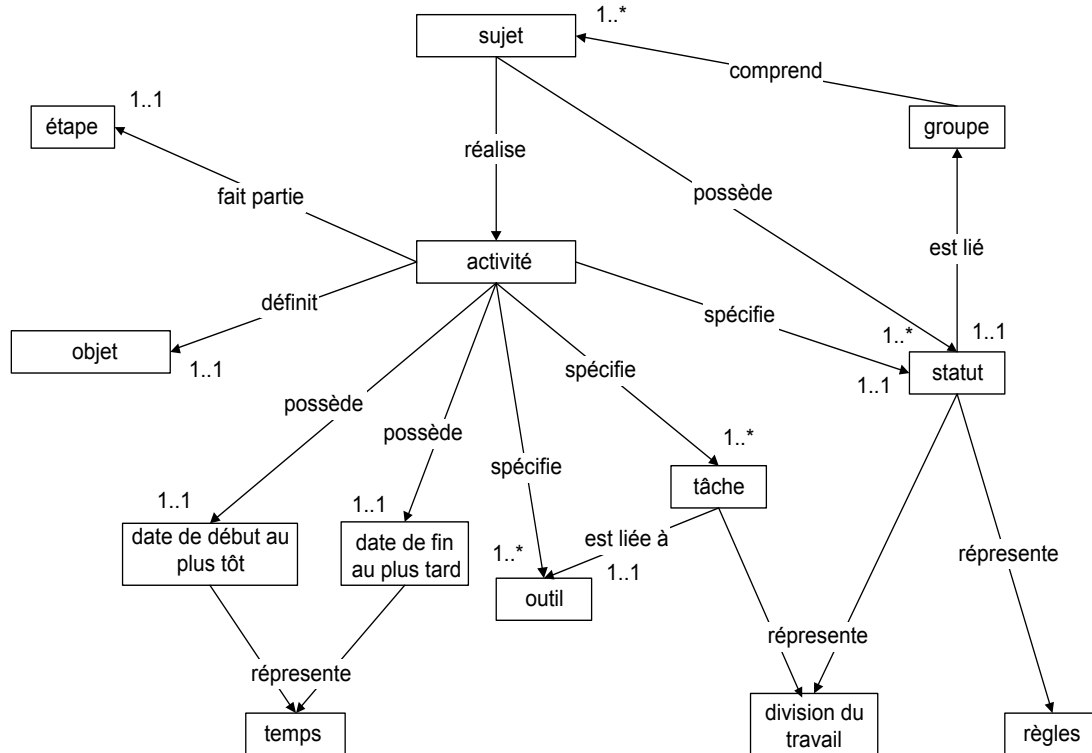


Figure 3.6 : Le modèle conceptuel d'activité de SimuLigne

Il est intéressant de noter que, bien que les tâches contenues dans une activité n'aient pas de dates de fin au plus tard ou de début au plus tôt en général, il peut arriver que certaines tâches aient des dates butoirs parfaitement identifiables comme une date de début au plus tôt et d'une date de fin au plus tard. C'est ainsi que des tâches consistant à utiliser les outils de forum ou de bavardage auront très probablement des dates de début et de fin.

Nous allons illustrer notre modèle conceptuel d'activité en l'instanciant sur une activité particulière de SimuLigne, c'est l'activité « *EIAI Quatre villes françaises* ». Le but de cette activité était de faire découvrir aux apprenants quelques villes françaises et d'en découvrir les caractéristiques en tant que villes universitaires. Les apprenants du groupe doivent, pour cela, prendre connaissance de quatre sites de villes françaises (Besançon, Caen, Nancy et Pau) en ligne ou hors connexion (à charge pour eux de télécharger les sites auparavant), remplir un tableau comparatif, le soumettre à l'ensemble des membres du groupe, modifier éventuellement leur tableau en tenant compte des remarques du tuteur et commenter les

réponses des autres membres du groupe dans un forum. Le tuteur, quant à lui, doit corriger les soumissions des apprenants, publier les éventuelles versions corrigées des apprenants qu'il aura reçues par courriel et participer au forum de discussion sur les réponses fournies par les membres du groupe. Cette activité est préparatrice à l'activité E1A2 servant à décrire une ville imaginaire.

Le *sujet* est ici soit un apprenant soit un tuteur, le *groupe* est l'un des quatre groupes de base à savoir : Lugdunensis, Aquitania, Gallia ou Narbonensis. L'*objet* de l'activité est celui décrit plus haut : faire découvrir aux apprenants quelques villes françaises et en dégager les caractéristiques en tant que villes universitaires. Cette activité appartient à l'*étape* E1. Le sujet considéré ici a un *statut* d'apprenant s'il s'agit d'un apprenant et de formateur s'il s'agit du tuteur. L'apprenant utilise plusieurs *outils* au cours de cette activité : "A rendre", "Courriel", "Publications", "Forum". Cette activité comporte plusieurs tâches :

- Télécharger les sites Internet des quatre villes choisies : l'outil utilisé ici est « Publication » ;
- Consulter les sites des quatre villes : l'outil utilisé ici est le navigateur de l'apprenant ;
- Enregistrer le tableau comparatif, le remplir : l'outil utilisé ici est l'éditeur de texte de l'apprenant ;
- Soumettre le tableau comparatif au tuteur : l'outil utilisé ici est « A rendre » ;
- Prendre connaissance de la correction du tuteur et envoyer éventuellement un tableau comparatif modifié : l'outil utilisé ici est « Courriel » ;
- Consulter les publications des autres membres du groupe et commenter les réponses : l'outil utilisé ici est le « Forum Principal ».

On observe ici que l'activité est découpée en plusieurs tâches liées chacune à un outil particulier. L'activité comporte une *date de début au plus tôt* qui est la date de mise en ligne, soit le 11 mai 2001, une *date de fin au plus tard* pour la tâche de première publication du tableau comparatif de l'apprenant qui est la date limite de soumission, soit le 18 mai 2001. On a aussi une date de correction et de soumission qui est une *date de fin au plus tard* pour la tâche de correction des premières versions et de publication des deuxièmes versions des soumissions, c'est le 21 mai 2001; le forum sur les réponses est clôturé à la fin de l'étape E1, le 3 juin 2001 : c'est la *date de fin au plus tard* de l'activité. On voit bien ici que toutes ces dates matérialisent le *temps* pendant lequel doit se dérouler l'activité. On voit aussi très bien

ici comment le statut du sujet dans le groupe représente une *division du travail* : l'apprenant publie lui-même dans l'espace approprié une première version, modifie éventuellement cette première version qu'il envoie au tuteur par courriel et qui se charge cette fois-là de publier la version corrigée dans le bon emplacement. Le statut spécifie aussi très bien les *règles* qui président au déroulement de l'activité : l'apprenant publie une première version, le tuteur en publie la deuxième.

Il convient aussi de noter que durant cette activité, il existe a priori un système d'activités parallèles qui apparaît entre les natifs et les apprenants : rappelons que dans SimuLigne, les natifs avaient pour rôle d'assister les apprenants, d'en être des compagnons d'apprentissage. Les natifs sollicités par les apprenants ou ayant pris connaissance des soumissions des apprenants ont aussi la possibilité d'intervenir en suggérant des corrections à l'apprenant. Ces activités ne sont pas représentées dans notre instanciation, elles font partie des activités qui émergent de l'ensemble des règles contenues dans les principes didactiques et les traditions institutionnalisées rapportées par Fjuk et Ludvigsen (2001).

La mise en perspective 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.

PIFAD propose une architecture de statuts⁷ (Torrent, 2001) ; les statuts sont des fonctionnalités spécifiques, affectées aux formateurs, aux tuteurs, aux apprenants dans leurs relations avec les autres statuts. Chaque utilisateur (ensemble constitué d'un nom de connexion et d'un mot de passe) est membre d'un cours - *course* (que nous avons utilisé comme espace caractérisant le groupe, il est impossible d'implémenter autrement le concept de groupe) et a un statut permanent. C'est la caractéristique d'une plate-forme que de rendre possible et d'instrumenter l'intervention de personnes différentes dans une même formation. PIFAD prévoit quatre statuts :

- l'apprenant : l'ergonomie de l'environnement de travail de l'apprenant correspond à l'interface utilisateur ; un accès sur la plate-forme en tant qu'apprenant offre bien entendu la possibilité d'un accès aux supports de cours ; un espace privé (sécurité de l'identification et du mot de passe, espace de présentation personnelle, calendrier, annotations personnelles dans les supports de cours) ; des outils de communication

⁷ Nous parlons ici de statut pour rester cohérent avec notre terminologie mais Torrent (2001) parle elle de rôle.

synchrone (bavardage, tableau blanc) et asynchrone (messagerie électronique, forums de discussion) ; des outils pédagogiques (tests, auto-évaluation, suivi des résultats aux tests) ;

- le concepteur qui sera en fait l'enseignant (avec accès à l'interface « *course designer* » ou « concepteur » en français) : il produit et introduit sur la plate-forme les supports de cours (nécessite une connaissance préalable de base du langage HTML), des ressources et documents, crée l'index, le glossaire, il développe, édite, gère les tests et QCM ; il ouvre des espaces de travail (forums par exemple). Il administre un « cours ». Il a les mêmes outils de communication que l'apprenant ; il édite/gère l'agenda visible des étudiants de son groupe ; il a accès à des rapports statistiques et graphiques sur l'utilisation du cours par ses étudiants, il peut avoir une trace du passage des étudiants dans le cours, il gère les résultats des évaluations ;
- l'assistant pédagogique (« *Teaching Assistant* »), qui se charge de noter et commenter les travaux d'apprenants (statut qui ne sera pas exploité pour SimuLigne) ;
- l'administrateur : l'environnement dans lequel il intervient lui permet d'inscrire, sur la plate-forme (donc de les identifier) et dans des « cours » spécifiques, de gérer des dossiers d'apprenants, et de donner ou supprimer des droits d'accès ; il est celui qui décide des identifiants et mots de passe individuels pour accéder sur la plate-forme.

On peut schématiser ces statuts avec d'une part leurs interdépendances et d'autre part, leur relation aux parcours technologiques (activités de SimuLigne) et aux matériels pédagogiques (consignes, etc.). Ceci nous conduit au schéma général de PIFAD que nous avons présenté à la figure 3.7. Un examen du schéma général de PIFAD nous permet d'observer plusieurs insuffisances. On peut observer que la notion de temps n'est pas suffisamment prise en compte. Ceci nous empêche de dissocier les différents temps qu'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. Les apprenants réalisent leurs activités. Les statuts présents dans PIFAD ne sont pas toujours compatibles avec les différents corps de métier qu'on retrouve dans la FAD. La réalité de SimuLigne ne nous a pas permis d'utiliser par exemple le statut d'assistant pédagogique et plusieurs personnes ont exercé des tâches essentielles qui ne se retrouvent pas dans les statuts de PIFAD. Les personnes ayant exercé les tâches de médiatiseur (mise en ligne des ressources pédagogiques), de techniciens (pour l'assistance des utilisateurs), de conception de

SimuLigne (matériels et parcours pédagogiques), de compagnons d'apprentissage (natifs de Besançon), de gestionnaire de la formation (sur le plan pédagogique et didactique) n'ont pas de statut au sens de PIFAD.

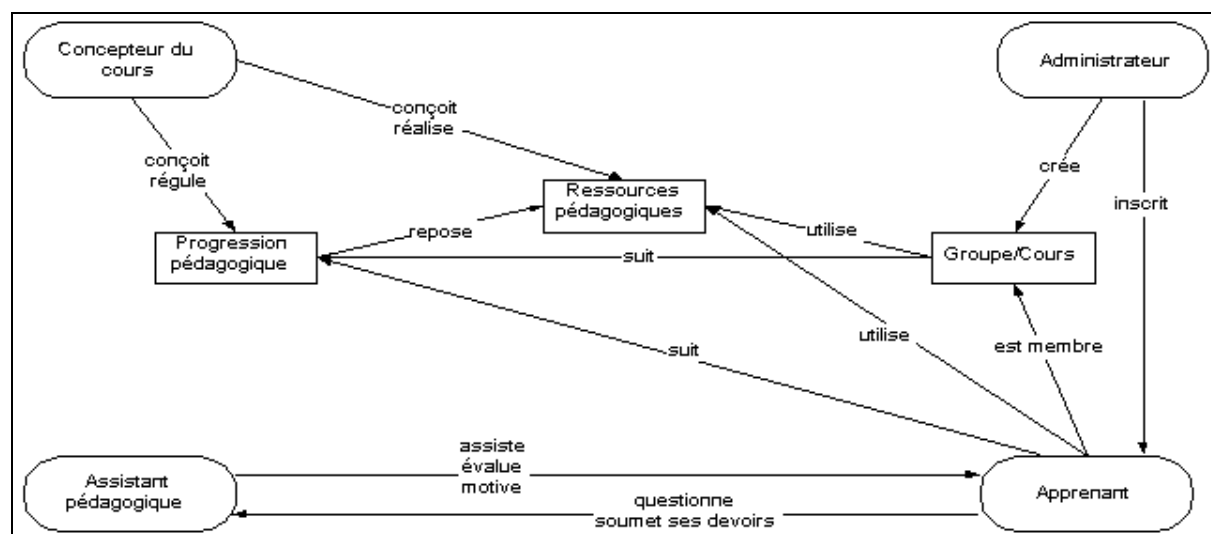


Figure 3.7 : Le schéma général de PIFAD

Ceci montre que PIFAD n'a pas été conçu au départ pour supporter le type de formation qui nous intéresse (entièrement à distance et basée sur une approche collaborative). On peut aussi noter qu'alors que le concept d'activité occupe une place centrale dans SimuLigne, il a 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.

3.3 - L'analyse des données d'interaction obtenues

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'activité 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 cette section, nous allons montrer que les données d'interaction enregistrées constituaient des indicateurs pertinents pour éclairer le tuteur et le coordonnateur en temps réel, sur le

comportement des apprenants et l'évolution des différents groupes de base. Notre souci est de montrer que la fermeture du groupe *Lugdunensis* était prévisible à partir de l'observation des statistiques d'interaction.

Nous allons faire une étude comparative du volume des interactions intra-groupes entre les quatre groupes de base dans une première partie. Pour cela, nous prenons comme base théorique qu'une activité normale dans chacun des groupes situerait les pourcentages de volume d'interactions du groupe autour de 25%. Dans un deuxième temps, nous allons examiner les données d'interaction à l'intérieur du groupe *Lugdunensis* et montrer que là aussi, les interactions pouvaient fournir une bonne indication des apprenants en situation de décrochage. Ces mêmes statistiques permettent aussi au tuteur d'avoir une indication acceptable des activités et des tâches en cours de réalisation ; en d'autres termes, des apprenants en train de bien réaliser les activités pédagogiques. L'idée ici est d'attirer l'attention sur une situation quelque peu paradoxale : les LMS comportent des données permettant de bâtir des indicateurs sur l'activité des utilisateurs ; cependant, on note une absence de ces indicateurs bien qu'ils soient assez simples d'un point de vue conceptuel. En effet, ils sont uniquement basés sur le type et le nombre d'interactions : nombre de courriels émis ou lus, nombre de messages écrits et lus dans les forums, nombre de tours de parole pris dans les bavardoirs, nombre et temps de connexions à la plate-forme.

3.3.1 - Les données 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 3.1 et 3.2 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 une faible motivation. Le dynamisme, l'engouement et la motivation des membres du groupe se traduisent aussi par le temps pendant lequel ils restent 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 3.1 : *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 3.2 : *Pourcentage cumulé des temps de connexion à PIFAD*

Le tableau 3.2 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.

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 3.3 : *Pourcentage cumulé du nombre de courriels lus à l'intérieur du groupe*

Les tableaux 3.3 et 3.4 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 pertinente 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	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 3.4 : 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 3.5 et 3.6 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.

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 3.5 : Pourcentage cumulé du nombre de messages lus dans les forums

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	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 3.6 : *Pourcentage cumulé du nombre de messages postés dans les forums*

L'observation des données contenues dans les deux tableaux ci-dessus montre 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 indique probablement des tentatives de remédiation de la part du tuteur du groupe et du coordonnateur de la formation, restées manifestement infructueuses. Ceci nous amène à penser qu'une action plus volontariste aurait probablement dû être entreprise dès la première semaine si ces données d'interaction avaient été portées à la connaissance du tuteur et du coordonnateur. Le coordonnateur aurait pu entreprendre soit de réorganiser les groupes, de manière à les rééquilibrer, soit 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 HTTP 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 3.7, 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.

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.

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 3.7 : *Durées de connexion aux pages de consignes*

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 afférentes dans les plates-formes de formation à distance. Les plates-formes existantes doivent être couplées à des systèmes

permettant de collecter les données d'interaction dans les endroits appropriés, de les calculer automatiquement et de fournir des indicateurs de l'état et de la durabilité des groupes concernés.

3.3.2 - Les données d'interaction relatives aux utilisateurs du groupe Lugdunensis

Dans ce paragraphe, 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 essayer de le montrer dans cette section.

Le tableau 3.8 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 3.8 : Pourcentages cumulés du nombre de messages lus par les apprenants du groupe Lugdunensis

L'analyse des informations contenues dans le tableau 3.8 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 a néanmoins été actif durant toute sa présence dans le groupe. 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 - Ville ouverte") 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 3.9 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'au maximum, 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 probablement pas 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 correctement la progression pédagogique envisagée au départ.

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 rémédiation dès le 13 ou le 14 mai par exemple. Il est aussi évident qu'une action de la part du tuteur vis-à-vis de ces apprenants leur montrerait qu'ils sont suivis et accompagnés et leur ôterait tout éventuel sentiment d'isolement, 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 affaiblissent le groupe et en tous cas, ne manquent pas d'être remarqués.

	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	

Tableau 3.9 : Nombre d'entrées dans le groupe Lugdunensis entre le 11 mai et le 25 mai

Les tableaux 3.10 et 3.11 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 3.10 : 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 3.11 : Nombre de messages postés par les apprenants dans les forums du Groupe entre le 11 et le 15 mai 2001

Le tableau 3.12 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 consignes entre le 11 et le 23 mai : ceci nous amène à dire qu'au maximum, 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 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 3.12 : 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 de déclencher des actions de rémédiation 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 SimuLigne. 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'isolement sociologique qui est un des principaux écueils en FAD et combattrait la perte de motivation qu'on observe souvent chez les personnes impliquées dans une session de FAD. Ceci renforcerait aussi l'idée d'appartenance au groupe, idée centrale en apprentissage collaboratif distribué.

Dans l'analyse des données d'interactions présentée ci dessus, nous nous sommes focalisés d'une part sur le groupe, d'autre part 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'interactions relatives aux tuteurs donnerait, là aussi, des éléments révélateurs du comportement et de la progression de nos groupes.

Nous avons montré dans cette section qu'en analysant les données d'interaction enregistrées dans les fichiers de traces, on peut fournir aux personnes en charge de 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 porteuses 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 (Mbala et al., 2002a).

Nous devons cependant attirer l'attention du lecteur sur le fait que les indicateurs statistiques présentés dans les différents tableaux, bien que relativement simples sur le plan conceptuel, ont nécessité des traitements informatiques assez fastidieux. Les données brutes étaient principalement contenues dans deux sources, le fichier de traces du serveur HTTP et la base de données interne de PIFAD. Il a fallu dans un premier temps, comprendre la signification des données du fichier des traces et ensuite, écrire des programmes informatiques nécessaires à leur extraction et à leur sauvegarde dans une base de données. La compréhension et l'exploitation de la base de données interne furent une tâche autrement plus ardue. En raison de l'absence de normalisation et de standardisation des LMS, il s'agit en réalité d'une base de données propriétaire dans laquelle les données sont extrêmement éparpillées. Elle a été certainement conçue suivant les motivations propres à l'éditeur de PIFAD, sans commune relation avec les préoccupations qui peuvent être les nôtres. Le travail de compréhension, d'organisation et d'extraction des informations contenues dans la base de données interne de PIFAD est présenté dans (Reffay, 2001) (Simudonnées, 2001).

Nous voulons souligner avec force que l'obtention des indicateurs statistiques présentés est loin d'être immédiate. Ces données statistiques ne peuvent non plus être obtenues à partir des logiciels commerciaux d'exploitation des fichiers de traces que l'on peut rencontrer sur Internet. Leur mise au point nécessitait l'ensemble des traitements informatiques auxquels nous avons dû recourir.

3.4 - Limites de PIFAD et proposition d'un SMA : SIGFAD

3.4.1 - Les limites des plates-formes existantes

Au cours de la formation SimuLigne, nous avons utilisé PIFAD pour constituer les groupes et organiser les activités des différents utilisateurs. Nous avons relevé un certain nombre de limites dans la mise en œuvre de SimuLigne sur la plate-forme choisie. La bonne audience dont bénéficie PIFAD dans le comparatif des plates-formes de formation ouverte et à distance

(Ecoutin, 2000)⁸ nous font dire que les plates-formes actuelles présentent un certain nombre d'insuffisances pour la mise en œuvre des formations à distance basées sur l'apprentissage collaboratif dans des groupes. Ces insuffisances se trouvent premièrement à un niveau très général telles que résumées dans (George, 2001) :

- peu d'exploitation d'objets pédagogiques interactifs (simulations, exercices) ;
- pas d'outil de suivi « sémantique » des activités de l'apprenant ;
- limitation des possibilités de communication synchrone ;
- surcharge cognitive pour le poste enseignant ;
- relatif isolement de l'apprenant, absence de conscience de groupe.

Pour notre part, pendant la durée de SimuLigne, nous avons noté un certain nombre d'insuffisances au niveau de PIFAD, insuffisances illustratives à notre avis de l'ensemble des plates-formes :

- Au niveau de l'apprenant :
 - Absence d'un outil pour l'appréciation de sa charge de travail sur une période donnée ;
 - Absence d'un outil d'aide pour renforcer son autonomisation ;
 - Absence d'un outil de réification de sa participation à la vie du groupe et aux différentes activités ;
- Au niveau du tuteur :
 - Absence d'un outil pour suivre l'évolution d'un apprenant ;
 - Absence d'un outil de visualisation de l'avancement du groupe : nombre de personnes actives sur les plans de la production et des interactions sociales ;
 - Absence d'un outil pour l'appréciation de la charge de travail sur une période donnée ;
 - Absence d'un outil de réification de sa propre participation.
- Au niveau du coordonnateur de la formation :
 - Absence d'un outil d'aide pour suivre l'évolution d'un apprenant ou d'un groupe ;
 - Absence d'un outil d'analyse des activités/interactions afin éventuellement de recadrer le calendrier de la formation, de suggérer l'abandon de certaines activités

⁸ L'auteur écrit dans cette étude : « WebCT est la plate-forme analysée qui propose le niveau de fonctionnalités le plus élevé pour la création de cours avec des documents en format HTML, pour la conduite de l'enseignement et le support de l'apprentissage. L'espace de travail étudiant est élaboré. » (Page 14)

- ou tâches par tout ou partie d'un groupe, de procéder à la réorganisation de certaines activités et de certains groupes qui ne fonctionneraient pas normalement ;
- Absence d'un outil de réification de sa propre participation.
 - Au niveau du groupe :
 - Absence d'un outil d'appréciation de l'état et de la durabilité du groupe ;
 - Absence d'un outil d'aide à la planification des activités synchrones ;
 - Absence d'un outil d'aide à la mise en œuvre des activités collectives en général.

Il convient néanmoins de préciser que dans PIFAD, le tuteur (avec son statut de concepteur qu'il avait dans SimuLigne par rapport à son groupe de base) dispose de deux outils lui permettant de voir la participation des apprenants de son groupe. Le premier représenté à la figure 3.8 lui permet d'avoir sur un seul tableau et pour chacun des membres du groupe : la date de sa première connexion, la date de la dernière connexion, le nombre de pages HTML visitées, le nombre de messages lus et le nombre de messages postés dans les forums.











Personal Information		Access Information		Articles		
Full Name	User ID	First Access	Last Access	Hits	Items Read	Posted
Sort	Sort	Sort	Sort	Sort	Sort	Sort
 ,	machine	October 8, 2002 12:51	April 11, 2003 09:15	612	292	1
 ██████████	██████████	January 23, 2003 10:12	March 6, 2003 16:00	393	224	31
 ██████████	██████████	January 23, 2003 13:50	March 21, 2003 08:33	442	257	16
 ██████████	██████████	January 30, 2003 19:09	March 17, 2003 19:26	85	47	10
 ██████████	██████████	January 23, 2003 12:42	January 23, 2003 12:46	20	16	0
 ██████████	██████████	January 22, 2003 20:26	March 22, 2003 18:22	154	85	6
 ██████████	██████████	January 22, 2003 21:34	April 4, 2003 12:03	636	390	29
 ██████████	██████████	January 25, 2003 16:34	April 6, 2003 07:57	782	369	42
 ██████████	██████████	January 23, 2003 10:18	April 10, 2003 17:14	420	199	26
 ██████████	██████████	January 22, 2003 19:05	April 9, 2003 20:53	654	387	52

Figure 3.8 : Informations fournies par PIFAD au tuteur sur son groupe

Le tuteur peut, pour un utilisateur donné, avoir des informations plus précises. Cette vue est représentée sur la figure 3.9. En plus des informations concernant l'ensemble du groupe, le tuteur peut ainsi visualiser le nombre d'accès à la page personnelle, au calendrier, aux pages de contenus (il s'agit des pages de consignes dans SimuLigne), au glossaire. L'apprenant concerné dispose de la même vue, en cliquant sur le lien « Ma Participation ».

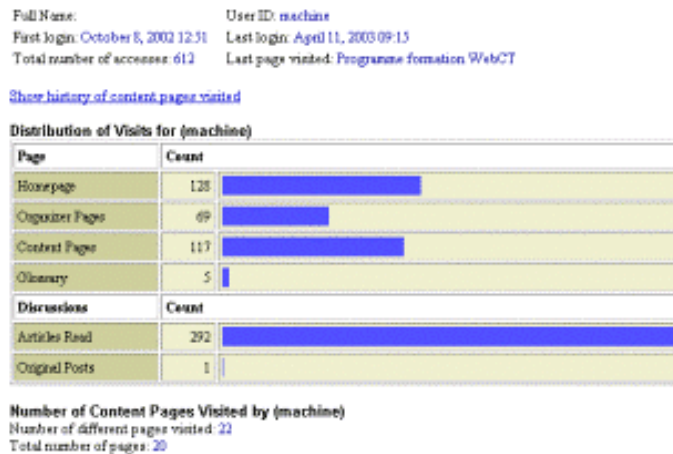


Figure 3.9 : Le tuteur peut disposer de plus d'informations sur un apprenant donné.

On voit donc que PIFAD dispose bien d'outils de réification, mais on peut noter deux inconvénients. Ces outils affichent les informations au jour le jour, les informations de la veille sont à chaque fois écrasées et ceci empêche d'avoir une vue étalée dans le temps. Autre inconvénient majeur de notre point de vue, il n'existe pas d'information concernant le groupe pris comme tel. On pourrait dire que PIFAD affiche des informations brutes, c'est, de ce point de vue, un «*mirroring system*» suivant la terminologie de Jerman et al. (2001).

Toutes les insuffisances et les limites relevées ci-dessus nous ont amené à proposer l'ajout d'un certain nombre de fonctionnalités aux plates-formes de téléformation existantes. Nous proposons d'ajouter ces fonctionnalités sous la forme d'un système multi-agents qui viendrait se greffer aux LMS. Le choix de cette option est motivé par le fait que nous pensons que les plates-formes existantes, bien que présentant certaines insuffisances, ont des fonctionnalités très intéressantes et élaborées : il serait illusoire et fastidieux pour nous d'essayer de les développer de nouveau. Il nous semble par contre plus efficace et réaliste de proposer l'ajout d'un certain nombre d'agents pour assister les utilisateurs des PIFAD dans leurs activités et de favoriser les interactions sociales, gage de la conscience de groupe et élément favorisant l'autonomisation des différents utilisateurs et principalement des apprenants. Nous avons baptisé notre système multi-agents SIGFAD pour **S**ystème pour les **I**nteractions dans des **G**roupes de **F**ormation **A** Distance.

3.4.2 - Les fonctionnalités de SIGFAD

Afin de pouvoir suivre correctement l'apprentissage collaboratif distribué, nous avons dégagé quatre groupes de fonctionnalités à mettre en œuvre dans notre système multi-agents. Nous

reviendrons sur ces fonctionnalités dans le chapitre 5 afin de dégager un diagramme hiérarchisé des objectifs à partir duquel, les autres étapes de la méthodologie d'analyse et de conception du SMA seront construites.

3.4.2.1 - Surveiller l'activité du groupe et des apprenants

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.

Nous déterminons plus précisément le nombre d'apprenants présents, absents, dormants durant une période donnée. Nous entendons par 'dormant', un apprenant qui se connecte et ne fait rien d'autre que lire des informations à différents endroits de la plate-forme ; il ne laisse aucune trace visible pour les autres membres de son groupe. SIGFAD indiquera l'état de productivité du groupe en fonction des activités prévues durant une période et en fonction du nombre d'apprenants ayant effectué toutes ces activités.

Ces appréciations se feront aussi au niveau des individus, le système indiquera pour les apprenants, leur état de productivité et leur sociabilité. Nous entendons par sociabilité, les informations telles que le nombre de connexions, le temps de connexion en ligne, les messages lus et émis dans les courriels et les forums, le nombre d'accès aux pages de consignes et au calendrier. Ce sont des informations qui témoignent de la vivacité de l'apprenant sans être directement liées au bon déroulement des activités pédagogiques. Indiquer les temps de connexion des apprenants et comparer ces temps avec ce que prévoit le contrat pédagogique permettrait d'apprécier la charge de travail des apprenants afin de déterminer si elle est trop importante ou pas.

3.4.2.2 - Indiquer l'état de réalisation d'une activité

Le système fournira à tout instant l'information sur une activité donnée. Il pourra ainsi dire si à la date de la requête, l'activité a débuté ou si elle est en cours. Si l'activité a débuté et qu'elle a des jalons révolus, alors SIGFAD fournira le niveau de déroulement de l'activité

dans chacun des groupes de base (ce sont les groupes d'apprenants). Cette information est précieuse car elle donnerait une base objective pour modifier le calendrier de telle ou telle activité.

3.4.2.3 – Gestion des forums par les tuteurs

Une troisième catégorie de fonctionnalités concernera la gestion des forums. Le forum apparaît comme l'un des outils les plus importants en apprentissage collaboratif distribué (Henri et Ludgren-Cayrol, 2001). C'est pour cela que nous implémenterons une fonctionnalité au sein de SIGFAD permettant d'indiquer la proximité des interventions de début et de fin du tuteur avec les dates d'ouverture et de clôture des forums. L'idée étant que le tuteur doit être le premier dans le forum pour accueillir et orienter les apprenants et le dernier à intervenir afin de bien clôturer le forum.

Chapitre 4 : Agents, systèmes multi-agents et méthodologies

Nous avons indiqué que les LMS existantes doivent être dotées de fonctionnalités supplémentaires pour leur permettre de suivre effectivement l'apprentissage collaboratif distribué. Eu égard aux caractéristiques spécifiques de la FAD, nous avons choisi de recourir au paradigme d'agent pour bâtir un système regroupant les fonctionnalités qui manquent aux LMS pour principalement soutenir et susciter des interactions dans des groupes lorsque celles-ci sont jugées insuffisantes.

Parler des notions d'agent et de système multi-agents n'est pas un sujet aisé. Ceci est certainement dû à diverses raisons. La première en est que plusieurs significations sont données aux notions d'agent et de système multi-agents. La deuxième raison, dont la première est la conséquence de notre point de vue, est que fort peu d'implémentations des systèmes à base d'agents existent. Ceci laisse naturellement la porte ouverte à toutes sortes de spéculation puisque finalement, chacun peut en parler sans être confronté à une réalisation concrète. Le paradigme d'agent n'est pas exclusivement réservé à l'informatique et ceci naturellement vient compliquer davantage le sujet. Nwana et Ndumu (1999), deux auteurs ayant beaucoup travaillé dans ce domaine, ont souligné une fâcheuse tendance des chercheurs de ce champ disciplinaire à « réinventer la roue » sans véritablement faire avancer les choses. Ils écrivent ceci :

« ...Seen from the viewpoints of the sorts of problems above which truly plague the deployment of multi-agent systems, we are forced to truly question what progress we have made beyond 1994... We therefore sadly conclude that in place of real progress in multi-agent systems, there has been much reinvention of wheels, and worse still too much 're-labelling' of old concepts. »

La difficulté à cerner le concept d'agent apparaît dès que l'on essaie de savoir ce qu'est un agent. Pour illustrer notre propos, nous donnons un aperçu de ce qui a souvent été considéré comme agents logiciels par différents chercheurs.

Certains logiciels sont appelés agents simplement parce qu'ils peuvent être paramétrés à l'avance pour faire des actions sur des machines distantes ; d'autres parce qu'ils accomplissent des tâches de bas niveau à partir des langages de programmation de haut niveau ou de scripts (Apple Computer, 1993). Knoblock et Ambite (1997) considèrent les agents comme des logiciels qui rendent transparentes les différences entre les sources

d'informations ou les services offerts comme des agents. Pour Minsky et Riecken (1994), les agents sont des logiciels qui implémentent une primitive ou qui agrègent une fonction cognitive. Pour Moulin et Chaib-Draa (1996), les agents sont des logiciels qui offrent des caractéristiques d'intelligence distribuée. Coutaz (1990) et Wiederhold (1989 ; 1992) considèrent que les logiciels servant d'interface entre les humains et les autres logiciels sont des agents. Boy (1991) et Maes (1997) classent comme agents les logiciels jouant le rôle d' « assistant intelligent ». White (1997) considèrent que les agents sont des logiciels capables de migrer d'un ordinateur à un autre sans intervention externe. Ball et al. (1996), Bates (1994), Hayes-Roth et al. (1995) voient les agents comme des logiciels ayant des croyances. Pour Finin et al. (1997), Genesereth (1997), les agents sont des logiciels utilisant un langage de communication entre agents. Shoham (1997) classe des logiciels manifestant un caractère intentionnel et d'autres aspects d' « états mentaux » comme des agents.

On perçoit à présent la nécessité de définir précisément ce que nous entendons par agent et par système multi-agents. C'est ce que nous allons essayer de faire dans ce chapitre après avoir rapporté différentes définitions proposées par des auteurs. En attendant, de la multitude des caractérisations de l'agent, il se dégage deux approches distinctes mais complémentaires. L'une considère un agent comme le mandataire d'une personne ; l'autre est basée sur la description de caractéristiques que doivent posséder les agents logiciels. La deuxième approche propose une définition qui nous convient davantage, c'est une entité logicielle qui possède des fonctions de continuité et d'autonomie dans un environnement particulier, contenant souvent d'autres agents ou d'autres processus (Shoham, 1997). Les exigences de continuité et d'autonomie viennent de notre désir de voir un agent capable de mener des activités intelligemment, en répondant aux changements de son environnement sans intervention extérieure ou humaine.

4.1 - Agents

4.1.1 - Définitions

La notion d'agent n'est pas simple à définir. Il existe en effet plusieurs définitions ou significations données à cette notion. C'est la raison pour laquelle plusieurs auteurs essayent

d'en donner une définition avant de se pencher sur l'utilisation de ce paradigme dans tel ou tel contexte.

Jennings et Wooldridge (1998), Jennings et al. (1998) définissent un *agent* comme un système informatique *situé* dans un certain *environnement*, capable d'exercer de façon *autonome* des *actions* sur cet environnement en vue d'atteindre ses objectifs. La figure 4.1 proposée par Wooldridge (1999) offre une abstraction de haut niveau de la notion d'agent. On voit dans ce diagramme l'action de l'agent qui affecte son environnement. Il est important de noter que l'agent n'aura en général pas un contrôle total de son environnement et ne l'affectera que partiellement ; ceci signifie en particulier qu'une même action effectuée deux fois dans des conditions apparemment identiques n'aura pas le même effet ou conduira à des effets inattendus. Wooldridge résume cet état de choses en disant que l'environnement de l'agent est en général non déterministe. D'après Jennings et Wooldridge (1998), un agent intelligent se caractérise par les propriétés suivantes :

- autonomie : un agent possède un état interne (non accessible aux autres) en fonction duquel il entreprend des actions sans intervention d'humains ou d'autres agents ;
- réactivité : un agent perçoit des stimuli provenant de son environnement et réagit en fonction de ceux-ci ;
- capacité à agir : un agent est mû par un certain nombre d'objectifs qui guident ses actions, il ne répond pas simplement aux sollicitations de son environnement ;
- sociabilité : un agent communique avec d'autres agents ou des humains et peut se trouver engagé dans des transactions sociales (négocier ou coopérer pour résoudre un problème) afin de remplir ses objectifs.

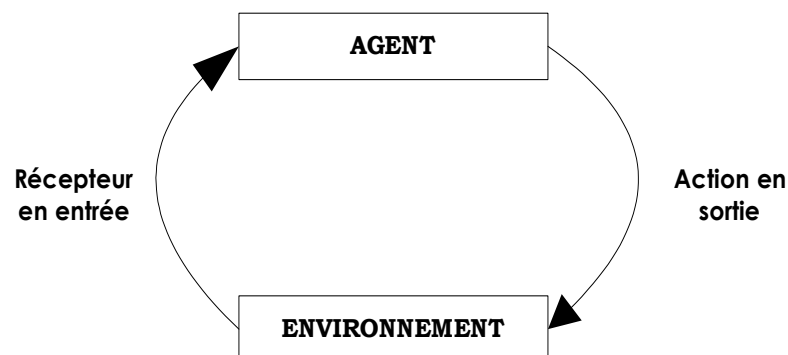


Figure 4.1 : Un agent dans son environnement (Wooldridge, 1999).

Ferber (1995) définit un agent comme une entité physique ou virtuelle

- a. qui est capable d'agir dans un environnement,
- b. qui peut communiquer directement avec d'autres agents,
- c. qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
- d. qui possède des ressources propres,
- e. qui est capable de percevoir (mais de manière limitée) son environnement,
- f. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- g. qui possède des compétences et offre des services,
- h. qui peut éventuellement se reproduire,
- i. dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

Ferber (1995) opère une différence entre les agents logiciels et les agents purement situés (qui sont davantage des entités physiques). Un agent logiciel ou agent purement communiquant est une entité informatique qui :

- a. se trouve dans un système ouvert,
- b. peut communiquer directement avec d'autres agents,
- c. est mue par un ensemble d'objectifs propres,
- d. possède des ressources propres,
- f. ne dispose que d'une représentation partielle des autres agents,
- g. possède des compétences (services) qu'elle peut offrir aux autres agents,
- i. a un comportement tendant à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de ses représentations et des communications qu'elle reçoit.

Plusieurs chercheurs définissent la notion d'agent par rapport à celle d'objet, arguant du fait que la maîtrise d'une nouvelle technologie ou d'un nouveau concept est facilitée par une référence à des technologies ou concepts voisins et d'ores et déjà maîtrisés (Odell et al., 2000). Les objets se définissent comme des entités informatiques qui *encapsulent* un état, sont capables d'accomplir des actions, ou *méthodes* dans cet état, et communiquent par échange de messages. Bien qu'il existe plusieurs similitudes entre les agents et les objets, il n'en existe

pas moins d'importantes différences. La première concerne le degré d'autonomie. La propriété la plus caractéristique des objets est le principe d'encapsulation, l'idée selon laquelle les objets contrôlent leur propre état interne. Dans ce sens, un objet peut être considéré comme ayant une autonomie sur son état. Mais un objet n'a pas de contrôle sur son comportement : si une méthode m est rendue accessible aux autres objets, alors ceux-ci peuvent les invoquer chaque fois qu'ils le veulent. En d'autres termes, une fois qu'un objet a rendu une méthode publique, il n'a plus de contrôle sur l'exécution ou la non exécution de cette méthode. Les agents, quant à eux, n'invoquent pas des méthodes mais adressent des requêtes pour l'exécution des actions. Si un agent i demande à un autre agent j d'exécuter une action a , j exécutera l'action ou ne l'exécutera pas. On voit donc que dans le cas des objets, la décision d'exécuter l'action appartient à l'objet qui invoque la méthode alors que pour les agents, cette décision revient à celui qui reçoit la requête. Wooldridge (1999) rapporte un slogan très original qui résume de ce point de vue la différence entre les objets et les agents : « *Objects do it for free ; agents do it for money.* » La seconde différence entre les objets et les agents réside au niveau des caractéristiques de réactivité, de dynamisme et de sociabilité. Bien qu'on puisse essayer de construire des objets disposant de ces propriétés, celles-ci ne sont pas des standards de la programmation orientée-objet (POO). La troisième différence entre les objets et les agents vient du fait que ces derniers sont censés avoir chacun leur fil d'exécution (*thread*). Dans le modèle objet standard, il existe un fil d'exécution unique pour tout le système, bien que l'on voie apparaître du parallélisme dans la POO avec des langages comme Java, qui offre la possibilité de lancer plusieurs fils d'exécution en même temps. Il faut noter une évolution dans la communauté des adeptes de la POO vers la notion d'objet actif. Un objet actif est un objet possédant son propre fil d'exécution. Les objets actifs sont généralement autonomes, c'est-à-dire qu'ils peuvent exhiber un comportement sans avoir été sollicités par un autre objet. Odell et al. (2000) présentent les agents comme une extension d'objets actifs possédant une prise d'initiative (la capacité à initier des actions sans intervention externe) et un libre arbitre (la capacité à refuser ou à modifier une requête externe). Odell et ses collègues définissent un agent comme un objet qui peut prendre des initiatives (*an objet that can say 'go'...*) et refuser de répondre à une sollicitation (*... and 'no'*). Finalement ces auteurs définissent un agent comme toute entité autonome capable de communiquer avec son environnement. Il faut noter que l'auteur utilise cette mise en perspective des agents par rapport aux objets pour proposer une extension au langage UML pour spécifier des agents. Le résultat en est le AUML pour *Agent UML*.

Il est tout à fait intéressant de mettre en perspective les agents par rapport aux systèmes experts. Ceux-ci ont été la technologie la plus importante de l'intelligence artificielle des années 1980. Un système expert est un système capable de résoudre des problèmes ou de donner des conseils dans un domaine de connaissances fort élaboré. La différence la plus importante entre les agents et les systèmes experts est qu'un système expert est fondamentalement désincarné (Wooldridge, 1999) (Jennings et al., 1998). Ceci signifie qu'ils n'interagissent pas directement avec leur environnement, ils ne prennent pas leurs informations à travers des capteurs, mais à travers un utilisateur qui agit comme intermédiaire. De la même façon, ils n'agissent pas directement sur leur environnement mais donnent un *feedback* ou des conseils à une tierce personne. D'autre part, on ne demande généralement pas à un système expert de coopérer avec d'autres agents.

DeLoach et al. (2001) pour leur part définissent un agent, et c'est cette définition que nous retiendrons comme base de travail, comme un ***ensemble de processus qui communiquent entre eux pour atteindre un objectif donné***. Ils écrivent :

« In our research, we do not take the typical artificial intelligence view of agents where agents are required to be autonomous, proactive, reactive, and social. For our purposes, agents are simple software processes that interact with each other to meet an overall system goal... We view agents merely as a convenient abstraction, which may or may not possess intelligence. »

Toutes les propriétés que les uns et les autres utilisent pour caractériser un agent doivent être implémentées par le concepteur suivant la pertinence de son analyse.

4.1.2 - Typologie des agents

La prolifération des variétés d'agents non seulement nécessite de rechercher une définition consensuelle servant de base de travail, c'est ce que nous avons essayé de faire dans le paragraphe précédent ; mais aussi conduit à procéder à une typologie des agents afin de mieux nous positionner par rapport aux types d'agents qui nous intéressent. Ici aussi, il existe plusieurs typologies selon les auteurs. Nous en présentons quelques-unes avant de retenir la typologie proposée par Nwana (1996).

Les spécialistes d'intelligence artificielle font souvent une distinction entre des agents dits faibles et des agents forts. Conceptuellement, les agents de cette dernière catégorie possèdent des propriétés mentales et émotionnelles (Shoham, 1997), (Wooldridge et Jennings, 1995). Des chercheurs issus de l'intelligence artificielle distribuée comme Moulin et Chaib-Draa (1996) caractérisent les agents par leur capacité à résoudre les problèmes :

- un agent réactif réagit aux changements de son environnement ou aux messages provenant des autres agents ;
- un agent intentionnel est capable de raisonner à propos de ses intentions, ses croyances et peut concevoir et exécuter des plans d'actions ;
- un agent social possède en plus des propriétés d'un agent intentionnel, des modèles explicites des autres agents.

Franklin et Graesser (1996) proposent une typologie des agents qui est présentée dans la figure ci-après :

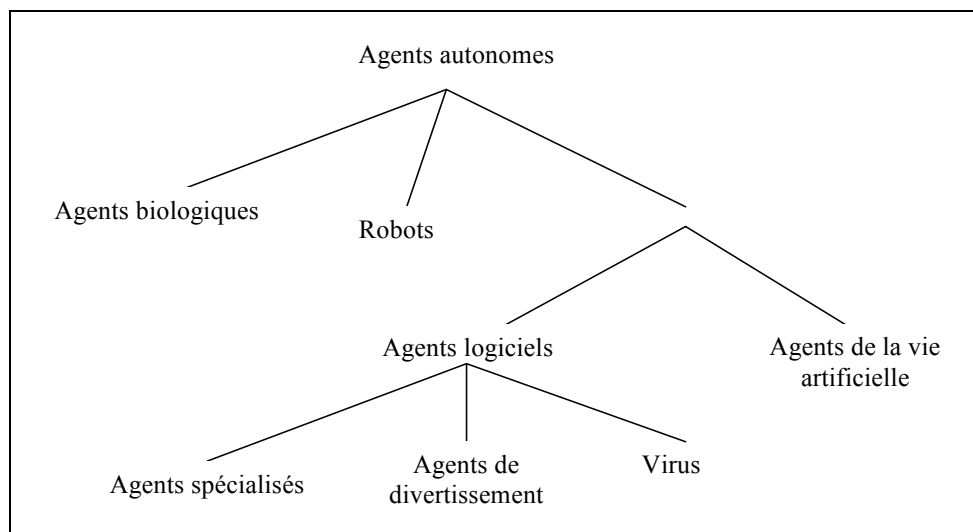


Figure 4.2 : Taxonomie des agents de Franklin et Graesser (1996)

Ferber (1995) classe les agents en fonction de leur caractère cognitif (chaque agent dispose alors d'une base de connaissances) ou réactif (chaque agent dispose de mécanismes de réaction aux événements) et en fonction du comportement téléonomique (dirigé vers des buts explicites) ou réflexe (régé par des perceptions).

	RELATION AU MONDE	
CONDUITES	<i>Agents cognitifs</i>	<i>Agents réactifs</i>
<i>Téleonomiques</i>	Agents intentionnels	Agents pulsionnels
<i>Réflexes</i>	Agents « modules »	Agents

Tableau 4.1 : Classification des agents proposée par Ferber (1995)

Les agents cognitifs sont capables d'anticipation et peuvent planifier leur comportement, les agents réactifs en sont incapables.

S'il s'agit d'un agent cognitif , il pourra construire un plan « dans sa tête » tel que :	
<i>Plan ouvrirPorte</i>	
<i>aller jusqu'à l'endroit où se trouve la clé</i>	
<i>prendre la clé</i>	
<i>aller jusqu'à la porte</i>	
<i>ouvrir la porte avec la clé</i>	
S'agissant d'un agent réactif , il faudra le doter du comportement suivant :	
<i>R1</i>	<i>: si je suis devant la porte et que j'ai une clé, alors l'ouvrir</i>
<i>R2</i>	<i>: si je suis devant la porte et sans clé, alors essayer de l'ouvrir,</i>
<i>R3</i>	<i>: si la porte ne s'ouvre pas et que je n'ai pas la clé, alors aller chercher la clé</i>
<i>R4</i>	<i>: si je cherche une clé et qu'il y a une clé devant moi, alors prendre la clé et aller vers la porte</i>

Tableau 4.2 : Exemple d'un robot voulant franchir une porte (Ferber , 1995)

Nwana (1996) propose une typologie des agents à partir de plusieurs critères de classification :

- Mobilité : statique ou mobile ;
- Présence d'un modèle de raisonnement symbolique : délibératif ou réactif ;
- Existence d'un objectif et de propriétés initiales comme l'autonomie, la coopération et l'apprentissage. A partir de ces propriétés, Nwana déduit quatre types d'agents : collaboratifs, collaboratifs et apprenants, d'interface et intelligents ;
- Rôles : recherche d'informations ou travail sur Internet ;

- Philosophies hybrides : combinaison deux ou plusieurs approches dans un seul agent.

De cette typologie, Nwana propose sept catégories d'agents : agents collaboratifs, agents d'interface, agents mobiles, agents d'information ou d'Internet, agents réactifs, agents hybrides, agents intelligents. La typologie des agents proposés par Nwana nous semble être la plus complète et la mieux expliquée, c'est pourquoi nous présentons cette taxonomie un peu plus en détail dans les lignes qui suivent. On ne retrouvera toutefois pas la dernière catégorie car Nwana pense que les agents logiciels intelligents n'existent réellement pas encore !

4.1.2.1 - Les agents collaboratifs

Les agents collaboratifs ont davantage des caractéristiques d'autonomie et de coopération avec les autres agents dans la réalisation de leurs objectifs. Ils doivent pouvoir négocier afin d'arriver à des compromis acceptables. La faculté d'apprendre peut exister mais n'est pas une caractéristique fondamentale requise pour ce type d'agents. Plusieurs chercheurs en intelligence artificielle leur prêtent des états mentaux comme les croyances, les désirs et les intentions. Ils sont alors qualifiés d'agents collaboratifs de type BDI (*beliefs, desires, intentions*). Les propriétés caractéristiques des agents collaboratifs sont : l'autonomie, la réactivité, le dynamisme (capacité à initier des actions) et la sociabilité. Les principales raisons pour lesquelles l'on peut être amené à implémenter les agents collaboratifs relèvent de l'intelligence artificielle distribuée ; elles sont les suivantes :

- résoudre des problèmes trop importants pour un seul agent à cause des limitations des ressources et des risques de systèmes centralisés ;
- permettre l'interconnexion et l'interopérabilité de plusieurs *legacy systems* comme les systèmes experts, les systèmes décisionnels etc. ;
- résoudre des problèmes fondamentalement distribués comme le contrôle du trafic aérien ou des problèmes inhérents aux systèmes d'information distribués ;
- résoudre des problèmes pour lesquelles l'expertise disponible est distribuée ;
- encourager la modularité, la vitesse d'exécution, la flexibilité.

4.1.2.2 - Les agents d'interface

Les agents d'interface disposent d'une autonomie et d'une capacité à apprendre. Maes (1997) qui a beaucoup travaillé sur cette classe d'agents, les caractérise par la métaphore *d'assistant personnel collaborant avec l'utilisateur* dans le même environnement. Alors que les agents

collaboratifs collaborent avec les autres agents, les agents d'interface collaborent avec l'utilisateur. La collaboration avec l'utilisateur ne nécessite pas l'existence explicite d'un langage de communication entre agents comme dans le premier cas. Les agents d'interface apprennent les préférences des utilisateurs afin de mieux les assister. Ils apprennent :

- en observant et en imitant leurs utilisateurs ;
- à travers le feed-back des utilisateurs suite à leurs actions ;
- en recevant des instructions de l'utilisateur ;
- en demandant les conseils des autres agents.

La collaboration avec les autres agents lorsqu'elle existe se limite à la demande de conseils, et diffère ainsi du type de collaboration qui a cours entre les agents collaboratifs. Un agent d'interface est doté d'un minimum de connaissances au départ et acquiert lui-même, par la suite, la connaissance dont il a besoin pour assister l'utilisateur. La principale raison qui amène à considérer de tels agents est le souci de déléguer certaines tâches répétitives et fastidieuses (il faut d'ailleurs signaler que la répétitivité de certaines actions ou attitudes dans le comportement de l'utilisateur est le gage de la réussite de tels agents, autrement ils ne pourraient rien apprendre de ce comportement).

4.1.2.3 - Les agents mobiles

Les agents mobiles sont des processus capables de se déployer à travers de grands réseaux d'information comme Internet, interagissant avec différents hôtes, recueillant des informations pour leurs propriétaires et accomplissant des tâches qui leur sont confiées. Ces tâches vont de la réservation de vols d'avions à l'administration de réseaux de télécommunications. Il faut dire que la mobilité n'est pas une caractéristique des agents, les agents mobiles sont des agents du fait qu'ils sont autonomes et coopèrent (certes différemment des agents collaboratifs). Un agent mobile peut ainsi communiquer ou coopérer avec un autre agent chargé d'informer les autres agents de la localisation de ses attributs et méthodes ; ceci dispense cet agent de rendre publiques toutes les informations et données le concernant. Les raisons qui poussent à recourir aux agents mobiles sont les suivantes :

- réduction des coûts de communication : en permettant à l'agent d'aller s'exécuter là où se trouvent les informations brutes, on évite ainsi de les rapatrier sur son système local pour n'en utiliser qu'une petite partie ;

- l'insuffisance des ressources locales : les capacités de traitement et de stockage peuvent être limitées au point de justifier l'usage des agents mobiles qui iraient s'exécuter sur des systèmes distants plus performants ;
- une coordination plus facile : il est plus simple de coordonner un nombre de requêtes distantes et d'en collecter simplement les résultats ;
- un traitement asynchrone : on peut initialiser ses agents mobiles et faire autre chose pendant que ceux-ci s'exécutent ;
- une architecture distribuée plus flexible : les agents mobiles offre une architecture distribuée unique.

4.1.2.4 - Les agents d'information

Les agents d'information ou agents d'Internet sont apparus du fait de l'explosion de l'information et du besoin de disposer d'outils de manipulation de ces informations. Les agents d'information ont pour rôle d'administrer, manipuler ou collecter les informations à partir de plusieurs sources d'informations distribuées. Du fait de l'avènement et du succès d'Internet, la distinction entre les autres types d'agents, tels que les agents d'interface ou de collaboration, et les agents d'information est subtile : en effet, tous manipulent des informations. Il faut néanmoins garder à l'esprit que les agents d'information sont définis par ce qu'ils font, alors que les autres sont définis par ce qu'ils sont, c'est-à-dire par leurs attributs et propriétés caractéristiques. Les raisons qui président à la mise en œuvre des agents d'information sont de deux ordres. D'une part, il y a la nécessité de faire face à l'explosion des sources d'informations et donc de répondre à un besoin similaire à celui rempli par les moteurs de recherche. D'autre part, sur le plan financier, de tels agents rapporteraient à leur éditeur un succès comparable à celui de Netscape avec son navigateur sur Internet.

4.1.2.5 - Les agents logiciels réactifs

Les agents réactifs sont une catégorie spéciale d'agents ne disposant pas de modèles internes et symboliques de leur environnement : ils ne font que réagir aux stimuli provenant de l'environnement. Les agents réactifs sont relativement simples et interagissent avec les autres agents de façon basique. La mise au point de tels agents n'exige pas une spécification complète de leur comportement. Les agents réactifs peuvent en outre être considérés comme une collection de modules autonomes chargés chacun d'une tâche spécifique. La communication entre les différents modules est minimale. Les agents réactifs opèrent sur des

données brutes provenant des capteurs contrairement aux précédents types d'agents qui utilisent des représentations symboliques de haut niveau. L'existence et la mise en œuvre d'agents réactifs se justifient dans les situations où l'on ne veut prendre en compte que des hypothèses basées sur l'environnement physique. Ces agents réactifs sont supposés aussi être plus robustes et plus tolérants aux pannes que les autres types d'agents, la perte d'un tel agent n'entraîne pas de conséquences catastrophiques. Ce sont des agents situés qui n'ont pas besoin d'avoir une représentation de leur environnement, ils ne réagissent qu'aux sollicitations du moment.

4.1.2.6 - Les agents hybrides

Nous avons passé en revue cinq types d'agents : collaboratifs, d'interface, mobiles, d'information et réactifs. Le débat sur l'opportunité d'avoir tel ou tel type d'agent est davantage un débat théorique et plutôt stérile. Chaque type d'agent possède ses points forts et faibles, ses avantages et ses inconvénients. Une approche consiste donc souvent dans la réalité à bâtir des agents hybrides. Un agent hybride consiste en la combinaison de plusieurs caractéristiques au sein d'un même agent ; ces caractéristiques concernent la mobilité, la collaboration, l'autonomie, la capacité à apprendre, etc. L'utilisation d'agents hybrides reste toujours guidée par le souci de minimiser les faiblesses et d'augmenter les forces de tel ou tel type d'agents que l'on mettrait en œuvre dans une application.

4.2 - Les systèmes multi-agents

Traditionnellement, les recherches dans les systèmes composés de plusieurs agents sont conduites sous la bannière de l'IAD (Intelligence Artificielle Distribuée). Historiquement, elles ont été divisées en deux grands champs : la résolution de problèmes distribués (connue en anglais sous le sigle de DPS pour *Distributed Problem Solving*) et les systèmes multi-agents (SMA). Depuis peu, l'acception de SMA a été étendue à tous les systèmes constitués de composants (semi-) autonomes. La résolution des problèmes distribués se préoccupe de la façon dont un problème donné peut être résolu par plusieurs modules (appelées nœuds) qui coopèrent en divisant et en partageant la connaissance à propos du problème et des solutions développées. Les SMA implémentent des stratégies de résolution basées sur le comportement d'un ensemble d'agents autonomes qui, éventuellement, peuvent déjà exister. Un SMA peut être vu comme un ensemble faiblement interconnecté d'entités-solutions qui travaillent ensemble pour résoudre un problème en s'appuyant sur les capacités et les connaissances

individuelles de chaque entité. Les entités-solutions - agents – sont autonomes et sont de nature hétérogène. Les SMA se caractérisent dès lors par :

- une perception partielle de l’environnement par chaque agent ;
- des compétences limitées pour chaque agent, c’est-à-dire insuffisantes pour résoudre entièrement le problème ;
- une absence de contrôle du système global ;
- une décentralisation des données ;
- des traitements en mode asynchrone.

L’intérêt pour l’utilisation des SMA se fonde sur leur capacité : à fournir robustesse et efficacité ; à permettre l’interopérabilité de systèmes existants ; à résoudre des problèmes pour lesquels les données, l’expertise (la connaissance) et le contrôle sont distribués.

Il convient toutefois de souligner que malgré les avantages potentiels, les SMA font face à plusieurs difficultés. Au nombre de ces problèmes, on peut citer :

- Comment formuler, décrire, décomposer et allouer les problèmes, synthétiser les résultats parmi un ensemble d’agents intelligents ?
- Comment implémenter la communication et l’interaction entre les agents ? Quels langages et protocoles de communication utiliser ?
- Comment assurer la cohérence dans les actions et les décisions des agents sachant que chaque agent n’a qu’une perception partielle de l’environnement et du problème ?
- Comment permettre à chaque agent de représenter et de raisonner à propos des actions, plans et connaissances des autres agents ?
- Comment reconnaître et réconcilier les points de vue disparates ou des intentions conflictuelles entre des agents essayant de coordonner leurs actions ?
- Comment assurer l’équilibre entre les traitements et la communication ? Plus généralement, comment gérer l’allocation des ressources (limitées par nature) ?
- Comment éviter ou atténuer un comportement nuisible du système global, comme un comportement chaotique ou oscillatoire ?
- Comment implémenter, concevoir des plates-formes et des méthodologies de développement des SMA ?

4.2.1 - Définitions

Ferber (1995) définit un système multi-agents (SMA) comme étant un système composé des éléments suivants :

- un environnement E , c'est-à-dire un espace disposant généralement d'une métrique ;
- un ensemble d'objets O . Ces objets sont situés, c'est-à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans E . Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents ;
- un ensemble A d'agents, qui sont des objets particuliers ($A \subseteq O$), lesquels représentent les entités actives du système ;
- un ensemble de relations R qui unissent les objets (et donc les agents) entre eux ;
- un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler les objets de O ;
- des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

Un SMA communiquant est un SMA dont les agents communiquent entre eux par le biais de messages. Un SMA purement communicant est un SMA ne disposant pas d'environnement et dans lequel les agents ne font que communiquer entre eux ($O = A$ et E est l'ensemble vide). Un SMA situé est un SMA dont les agents sont positionnés dans un environnement. Un SMA purement situé est un SMA situé dans lequel les agents ne communiquent pas par l'envoi de messages, mais par propagation de signaux.

Comme définition de travail, nous dirons qu'*un système multi-agents (SMA) est un ensemble d'agents travaillant ensemble (coopérativement ou parallèlement) pour remplir les objectifs du système.*

4.2.2 - Raisonnement et communication dans les SMA

Le raisonnement au sein d'un groupe d'agents consiste en des décisions concernant la sélection des agents et la délégation de tâches, ainsi que des décisions concernant les transmissions et le traitement des informations. Il dépend :

- de la distribution des tâches et des compétences : soit une compétence ou une tâche n'est affectée qu'à un seul agent, soit une même tâche ou compétence est partagée

entre deux ou plusieurs agents, qui sont alors en concurrence ; il faut ajouter un mécanisme de résolution de conflit à ce niveau ;

- du type de protocole de communication : soit les échanges d'informations et de solutions partielles se font au travers d'une mémoire commune ou tableau noir (le destinataire n'est pas choisi, les informations sont toutes écrites sur le tableau noir, chaque agent peut les consulter à loisir), soit les communications s'effectuent par envoi de messages d'un agent à un autre (dans ce cas les raisonnements sont plus complexes, car le choix du destinataire du message doit être fait de manière pertinente) ;
- du mode de coopération qui définit la manière dont les agents vont collaborer à la résolution du problème ; cette collaboration dépend des compétences de chaque agent et de l'importance des échanges qu'ils effectuent entre eux.

Le raisonnement global d'un groupe d'agents pose donc des problèmes d'affectation et de partage des tâches, des compétences et des connaissances entre les différents agents, ainsi que des problèmes de synchronisation et de cohérence entre les activités de chacun d'entre eux.

4.2.2.1 - Distribution des tâches et des compétences

La manière dont sont distribuées les tâches et les compétences dans un SMA est très importante. Deux aspects de cette distribution vont en fait déterminer le type de contrôle et le fonctionnement du système :

- la quantité de compétences donnée à chaque agent ;
- l'unicité de chaque compétence.

La quantité de compétences donnée à chaque agent va déterminer leur granularité. En effet, un SMA peut être construit avec des agents capables d'effectuer plusieurs tâches complexes, ou avec des agents très spécialisés et n'ayant qu'une seule compétence très précise. Le comportement des agents ainsi que le raisonnement issu de leur collaboration dépend de leur granularité. Ensuite les compétences peuvent être distribuées soit de manière unique (une compétence n'est attribuée qu'à un seul agent), soit entre plusieurs agents (une même compétence est offerte à plusieurs agents). Dans le second cas, les agents possédant la même compétence sont aptes à réaliser les mêmes tâches. Ils peuvent donc collaborer en se partageant une tâche si elle est trop lourde pour un seul agent. Par contre, dans certains cas, ils

peuvent se retrouver dans une situation de concurrence. Cela nécessite alors de prévoir des mécanismes pour la résolution des conflits.

4.2.2.2 - Protocoles de communication

Il existe principalement deux modes de communication dans un SMA (Labidi et Lejouad, 1993) :

- par partage de ressources ;
- par envoi de messages.

Dans les systèmes fonctionnant par partage de ressources, les différents composants (souvent appelés sources de connaissances) ne sont pas directement liés entre eux. Ils communiquent au travers d'une zone de données commune, le tableau noir, dans laquelle sont stockées les connaissances du système. En revanche, dans les systèmes où la communication se fait par envoi de messages, les connaissances sont distribuées entre les différents agents. Chacun d'eux communique directement avec les autres par envoi de messages.

Nous allons nous intéresser ici seulement à la communication par envoi de messages et nous présentons les trois modèles de communication à base de messages les plus couramment utilisés dans les SMA. Il s'agit du modèle acteur, d'une de ses extensions, le langage *Abcl*, et du protocole *Contract net protocol*.

Le modèle acteur

Le modèle acteur a été proposé par Hewitt (1971). L'idée est de définir plusieurs agents, appelés acteurs, communiquant entre eux par envoi de messages afin de résoudre un problème. Les seuls moyens de communication et de contrôle d'un système acteur sont les messages transmis entre les différents acteurs (agents) de manière asynchrone.

Les connaissances d'un acteur sont limitées aux autres acteurs qu'il connaît directement (ses accointances) et à son comportement défini par un script. Le script décrit l'ensemble des réactions de l'acteur face à l'arrivée d'un message, c'est-à-dire la manière de répondre à ces messages, ainsi que les méthodes à activer suivant le contenu du message. Les acteurs sont en fait des agents réactifs de type automate.

Le comportement de l'acteur est quant à lui très simple. L'acteur peut se trouver dans deux états différents : actif ou inactif. Il est actif lorsqu'il reçoit un message et le traite, et inactif lorsqu'il attend des messages. Lorsqu'il reçoit un message de l'extérieur, il modifie son état, puis en fonction de son script émet d'autres messages ou crée d'autres acteurs. Il repasse ensuite en état inactif.

Abcl

Abcl est une extension du modèle acteur. C'est un protocole utilisé pour la résolution distribuée de problèmes, la planification, la modélisation de processus cognitifs humains et la conception de systèmes temps réel ou d'informations (Boissier, 1990). Ce langage a été défini dans le but d'obtenir deux qualités bien précises. La première est d'avoir une sémantique très proche des modèles de calculs sous-jacents. La seconde est que toutes les caractéristiques et les propriétés de LISP doivent être utilisables.

Dans le langage *Abcl*, on n'impose pas de tout représenter à l'aide d'objets (agents, acteurs) et de messages comme dans le modèle acteur. De plus, chaque objet peut avoir une mémoire locale servant à décrire son état. En ce qui concerne les communications, le premier apport est la préservation de l'ordre temporel d'envoi de messages. Si deux messages sont envoyés à un même objet, et par un même objet, ils arrivent en conservant l'ordre temporel qu'ils avaient au départ. Le second est que les messages peuvent être de trois types différents : synchrone, asynchrone ou faiblement synchrone. Un objet peut donc désormais être dans un des trois modes suivants :

- endormi, lorsqu'il n'attend rien et n'a rien à envoyer ;
- actif, lorsqu'il envoie un message à un autre ;
- en attente, lorsqu'il attend une réponse ou un acquittement à un message qu'il a envoyé.

Contract net protocol

Le *Contract net protocol* a été développé par Smith (1980). Dans ce protocole, le partage de tâches se fait sur un modèle d'offre et de demande. Les agents (appelés des nœuds) sont tour à tour soit *commanditaire*, soit *contractant*. Lorsqu'un agent ne parvient pas à résoudre lui-même un problème, il le décompose en plusieurs sous-problèmes et demande alors aux autres agents d'essayer de les résoudre. L'agent qui demande la résolution au groupe est appelé

commanditaire. Les agents susceptibles de résoudre tout ou partie du problème posé proposent une solution au commanditaire. Ce dernier examine l'ensemble des propositions et détermine l'agent le plus apte à répondre à son problème. Cet agent est appelé le contractant, et c'est lui qui est alors chargé d'effectuer la tâche demandée.

Plusieurs auteurs comme Odell et al. (2000), ont proposé de représenter le processus de négociation entre le commanditaire et les autres nœuds en adaptant les diagrammes de séquence de UML.

4.2.2.3 - Modes de coopération

Ferber (1995) définit la coopération entre plusieurs agents comme une situation dans laquelle, soit l'ajout d'un nouvel agent permet d'accroître les performances du groupe, soit l'action des agents sert à éviter ou à résoudre des conflits potentiels ou actuels. Ferber identifie six modes de coopération entre les agents :

- Le regroupement et la multiplication : ces aspects sont issus de l'observation du monde animal. Le rapprochement physique permet de former un bloc homogène, contenant des spécialistes de tous les domaines, et une communication plus facile entre les spécialistes. La multiplication permet de prévenir les défaillances individuelles possibles. Si le groupe ne dispose que d'un spécialiste pour un domaine particulier, et que celui-ci venait à disparaître, la survie du groupe serait en péril.
- La communication : la communication entre les différents agents est indispensable à leur coopération. Elle permet d'élargir leur perception de l'environnement, puisqu'elle permet aux agents de bénéficier de celles des autres agents et de leurs connaissances.
- La spécialisation : ce mécanisme permet aux agents de devenir plus aptes à la tâche qui leur est confiée. Plutôt que d'essayer de créer des agents spécialisés dans plusieurs tâches, il est, au contraire, bénéfique pour le groupe d'essayer d'avoir des agents très spécialisés dans une tâche précise.
- La collaboration par partage de tâches et de ressources : la collaboration consiste en la répartition des tâches, des connaissances et des ressources entre les agents afin de résoudre un problème commun. Cette répartition s'effectue par des mécanismes d'offre et de demande. Elle est réalisée, soit par un agent coordinateur qui centralise les offres et les demandes, soit de manière distribuée (les agents peuvent directement offrir ou demander des services aux autres agents). On procède alors, soit par appel

d'offre (*Contract net protocol*), soit par des réseaux d'acointances : chacun connaît les capacités des autres.

- La coordination d'actions : dès lors que des agents autonomes essaient de réaliser de manière individuelle leurs buts propres, il est nécessaire de coordonner leurs actions afin de veiller à ce que le but global soit aussi réalisé. Cela nécessite d'effectuer des actions supplémentaires permettant de déterminer par exemple l'ordre d'exécution des tâches des agents.
- La résolution de conflits par arbitrage et négociation : la collaboration entre différents agents poursuivant des buts individuels peut mener à la création de conflits. Il est donc nécessaire de veiller à ce qu'aucun conflit ou désaccord ne survienne entre les différents individus du groupe, afin de ne pas altérer les performances. Cela se fait par arbitrage ou négociation. L'arbitrage consiste en la rédaction de lois et règles de comportement auxquels les agents devront se plier. Dans le cas de sociétés d'agents cognitifs, la résolution d'un conflit entre deux agents peut aussi s'effectuer localement par une négociation directe entre les deux antagonistes.

4.2.3 - Méthodologies et plates-formes de développement des systèmes à base d'agents

On peut définir un système à base d'agents comme un système qui est bâti sur l'abstraction du paradigme agent. Un système à base d'agents peut contenir un seul agent (c'est principalement le cas dans les agents d'interface) mais contient potentiellement plusieurs agents comme dans les SMA.

Les méthodologies de développement des systèmes à base d'agents peuvent être classées en deux catégories : la première catégorie regroupe les méthodologies s'inspirant des méthodologies de développement des objets, la deuxième regroupe des méthodologies s'inspirant des systèmes à base de connaissances. Dans la première, nous passerons en revue la méthodologie proposée par Kinny et al. (1996) dite méthodologie AAI, la méthodologie GAIA proposée par Wooldridge et al. (1999), et nous examinerons le AUMI inspirée de UML et proposée par Odell et al. (2000). Dans la seconde catégorie, nous examinerons la méthodologie CASSIOPEA proposée par Collinot et al. (1996), et l'utilisation du langage Z (Spivey, 1992) pour spécifier les agents (Luck et d'Iverno, 1995). Nous examinerons ensuite

la méthodologie Prometheus (Padgham et Winikoff, 2002) ainsi que MadKit, une plate-forme de développement et d'exécution de systèmes multi-agents (MadKit, [http](http://)).

La méthodologie MaSE proposée par DeLoach et al. (2001) est celle que nous avons retenue pour le développement de notre SMA, elle sera davantage détaillée et sera présentée dans la dernière partie de ce chapitre. Elle fait partie de la catégorie de méthodologies basées sur la programmation orientée objet.

4.2.3.1 - La méthodologie AAI

Cette méthodologie est un mélange de concepts empruntés à la POO et de ceux propres aux agents, elle est basée sur l'architecture BDI (*beliefs, desires, intentions*) et sur un système de raisonnement procédural (PRS pour *Procedural Reasoning System*). Elle propose la construction d'un ensemble de modèles qui permettent de spécifier les différents agents du système.

Le SMA dispose d'un modèle interne et d'un modèle externe. Le modèle externe renferme comme composants principaux les différents agents et les relations entre eux ; on n'y retrouve ni la façon dont les agents sont construits, ni ce qu'ils font. A l'opposé, le modèle interne présente les différents états internes des agents, c'est-à-dire leurs croyances, désirs et intentions.

Le modèle externe a pour but de définir les relations entre les classes d'agent, d'identifier les instances des classes qui apparaîtront lors de l'exécution. Le modèle externe est divisé en deux sous-modèles : le modèle d'agent et le modèle d'interaction. Le modèle d'agent se divise encore en deux modèles : le modèle de classes d'agent et le modèle d'instanciation de l'agent. Ces deux classes permettent de définir les agents, les classes d'agents et permet de relier ces classes les unes aux autres par des relations d'héritage, d'agrégation et d'instanciation. Chaque classe d'agent a au moins trois attributs représentant les croyances, les désirs et les intentions. Le concepteur peut indiquer la façon dont les propriétés seront attribuées lors d'une relation d'héritage par exemple.

Les auteurs ne donnent pas les détails sur la manière de concevoir le modèle interne ; toutefois, il est clair que ceci correspond à l'implémentation du système de raisonnement

procédural de l'agent, c'est-à-dire à la définition des structures de croyances, désirs et intentions de l'agent. On peut résumer la méthodologie AAIL par les différentes étapes suivantes :

- identifier les rôles de l'application et, à partir de là, construire une hiérarchie des classes d'agent ;
- associer des responsabilités à chacun des rôles identifiés, les services demandés et fournis par chaque rôle et déterminer les objectifs associés à chaque service ;
- pour chaque objectif, déterminer les plans associés suivant les conditions d'exécution ;
- déterminer la structure des croyances du système (ceci est important pour chacun des plans et objectifs identifiés).

4.2.3.2 - La méthodologie GAIA

Cette méthodologie a pour objectif de permettre au concepteur de passer d'un cahier des charges aux détails de l'implémentation du système. Le déroulement de la méthodologie permet de passer graduellement de concepts abstraits à des concepts plus concrets, elle peut être vue comme un processus de développement de modèles de plus en plus détaillés du système à construire. Dans les concepts abstraits, on note : les rôles, les permissions, les responsabilités, les protocoles, les activités, les propriétés de vivacité, les propriétés de sûreté. Dans la catégorie des concepts concrets, on retrouve : les types d'agent, les services et les connaissances. Cette catégorisation des concepts correspond aussi aux deux phases de GAIA, une phase d'analyse (plus abstraite) et une phase de conception (plus concrète).

L'objectif de la phase d'analyse est de comprendre le système et sa structure sans se préoccuper des détails d'implémentation. L'organisation du système est perçue à travers un ensemble de rôles et des interactions entre ces rôles. Chaque rôle est défini par quatre attributs : les responsabilités, les permissions, les activités et les protocoles. Les responsabilités sont probablement l'attribut clé du rôle et permettent d'implémenter les fonctionnalités du système. Elles se divisent encore en deux catégories : les fonctions d'exécution et les fonctions de maintenance. Afin de remplir ses responsabilités, le rôle dispose d'un ensemble de permissions ; ce sont des droits associés au rôle et elles identifient des ressources accessibles au rôle. Les activités du rôle sont les traitements que ce dernier peut effectuer sans devoir communiquer avec d'autres agents.

4.2.3.3 - Agent UML

L'unification des différentes méthodologies de développement des objets a abouti à l'émergence du langage de modélisation des systèmes à base d'objets UML (*Unified Modeling Language*). Plusieurs chercheurs ont vu en UML un point de départ pour développer des outils et méthodes des systèmes à base d'agents. C'est ainsi que Odell et d'autres chercheurs ont proposé des extensions à apporter à UML afin de modéliser et de spécifier des agents (Odell et al., 2000). Ces extensions permettent :

- de représenter des protocoles d'interactions entre agents comme le *Contrat net protocol* ;
- de définir un concept de rôle comme une extension des acteurs dans les diagrammes d'interaction de UML, qui permet ainsi de représenter un agent remplissant plusieurs rôles.

L'organisme de normalisation OMG (*Object Management Group*) et le FIPA (*Foundation for Intelligent Physical Agents*) soutiennent les développements à base d'UML pour la modélisation et la spécification des agents.

4.2.3.4 - Cassiopea

Cette méthodologie a une vision par le bas, contrairement aux méthodologies GAIA et AAIL. Ici, on commence par spécifier les comportements nécessaires pour réaliser les tâches. Ceci correspond à une vision comportementaliste des agents mis en avant par Brooks (1999). Elle peut se résumer en trois étapes :

- identification des comportements élémentaires de l'ensemble du système ;
- identification des relations entre les comportements élémentaires ;
- identification des comportements organisationnels de l'ensemble du système comme par exemple la façon dont les agents se répartissent en groupes.

Collinot et ses collègues ont illustré cette méthodologie dans la conception de RoboCup soccer team (RoboCup, [http](http://www.robocup.org)).

4.2.3.5 - Utilisation du langage Z

Luck et d'Inverno (1995) ont développé une plate-forme de spécification des agents à partir du langage Z. Il faut signaler toutefois que les agents pris en compte par les auteurs ici sont d'un type un peu particulier et ne correspondent pas à la définition des agents couramment

admise. Les auteurs définissent leurs agents comme étant des objets ayant des objectifs et les agents autonomes comme des agents ayant des motivations. Une chaise serait donc un agent puisque son « objectif » est de permettre à un humain de s'asseoir mais ne serait pas autonome car n'ayant aucune motivation !

Ce formalisme permet de saisir les relations qui existent entre les différents agents. L'accent est mis sur les agents agissant au nom d'une autre entité plutôt que comme un système rationnel. Ceci est principalement dû au langage Z qui est davantage adapté à la spécification de systèmes fonctionnels.

4.2.3.6 - La méthodologie Prometheus

Cette méthodologie a été développée par Lin Padgham et Michael Winikoff (Padgham et Winikoff, 2002). Le but de ses auteurs était de mettre au point une méthodologie qui déroule l'ensemble du cycle logiciel et qui soit facile à utiliser pour des industriels et des personnes peu familières du concept d'agent. Elle permet de construire des agents dont l'architecture interne est basée sur le modèle BDI. A titre de comparaison avec les autres méthodologies, les auteurs affirment que la plupart des autres méthodologies considèrent les agents comme des boîtes noires (Padgham et Winikoff, 2002). Prometheus a la prétention d'être une méthodologie complète, suffisamment détaillée et facile d'utilisation.

Prometheus comporte trois phases :

- La phase de spécification consiste en l'identification des fonctionnalités du système avec les données en entrée (capteurs), les actions sur l'environnement (effecteurs) et toutes les sources importantes de données partagées. Le concepteur utilisera ici les cas d'utilisation et les diagrammes de séquence (comme dans UML) pour dégager et spécifier les fonctionnalités du système.
- La phase de conception architecturale utilise les effecteurs dégagés lors de la précédente phase pour déterminer le nombre et les types d'agents du système et la manière dont ils vont interagir entre eux. Le concepteur regroupe l'ensemble des fonctionnalités utilisant les mêmes données, celles-ci seront probablement affectées à un agent unique. Les interactions entre les agents sont représentées dans un diagramme d'accointances. Un descripteur d'agent est utilisé pour indiquer le nom et la description de l'agent, les fonctionnalités de l'agent et une description des moyens

d'interaction avec les autres agents. Le concepteur identifie à ce stade toutes les données partagées ; les agents qui accèdent aux mêmes données partagées nécessitent d'être synchronisés. Une façon de contourner cette difficulté peut consister à définir un agent unique qui accède aux données et les fournit aux autres. Lors de cette phase, un diagramme global du système permet de représenter les liens entre les agents, les événements et les données partagées. Les interactions entre les agents sont indiquées dans des diagrammes d'interaction (semblables aux diagrammes d'interaction entre objets). Les protocoles d'interaction indiquent quelles interactions sont autorisées dans le système, ils utilisent une notation empruntée à AUML (Odell et al., 2000).

- La phase de conception détaillée s'intéresse à la structure interne des agents et à la façon dont ils vont accomplir leurs tâches dans l'ensemble du système. Cette partie de la méthodologie est spécifique au modèle BDI. Ici le concepteur va définir les modules de chaque agent, les événements internes, les plans et les structures de données. L'application de cette phase est incrémentale ; le concepteur définit les composants de l'agent, puis entre dans la structure interne de chaque composant. Les fonctionnalités dégagées lors de la phase de spécification constituent un bon point de départ pour déterminer les composants de l'agent. Chaque composant est défini par un descripteur de composant qui indique son interface externe (les événements en entrée et les événements en sortie qui constituent les entrées des autres composants). Les relations entre les composants sont représentées dans un diagramme d'agent, il est similaire au diagramme global du système si ce n'est qu'à la place des agents, on a les composants d'un agent. Les plans de l'agent sont décrits dans des descripteurs de plans. Un descripteur de plan contient un identifiant, le type d'événement déclencheur, les étapes du plan, une brève description du plan en langage naturel, un contexte spécifiant quand le plan doit être utilisé et une liste de données en entrée et en sortie. Le concepteur doit en outre définir dès le début du projet un dictionnaire de données qu'il enrichit et met à jour au fil du déroulement de la méthodologie.

4.2.3.7 - La méthodologie MadKit

MadKit est une architecture de plate-forme multi-agents générique. C'est une plate-forme de conception et d'exécution de systèmes multi-agents. Ses auteurs lui trouvent une originalité dans le fait qu'elle ne s'appuie sur aucune architecture interne d'agent particulière et aucun modèle d'interaction spécifique.

MadKit s'appuie sur un modèle organisationnel dénommé Aaladin (Ferber et Gutknecht, 1998). Ce modèle définit les notions d'agent, de rôles et de groupes. MadKit voit un agent comme une entité autonome communicante, qui joue des rôles au sein de différents groupes (Gutknecht et Ferber, 1998). MadKit est basée sur une architecture à micro-noyau, une agentification systématique des services et un découplage applicatif entre noyau, agents et application d'accueil. Très concrètement, MadKit est un ensemble de packages Java qui implémentent le noyau agent, diverses bibliothèques de base de messages, d'agents et de sondes. MadKit offre au concepteur un éditeur graphique de formalismes représentables sous forme de graphes (UML, réseaux de pétri, diagrammes d'interaction...) que ce dernier peut utiliser pour spécifier son SMA. Malheureusement, la documentation disponible sur le site de MadKit (MadKit [http](http://)) n'est pas très explicite sur l'utilisation concrète de la plate-forme pour analyser, spécifier et implémenter un système multi-agents.

4.3 - Apprentissage machine et systèmes multi-agents

Il est impossible lorsqu'on conçoit des systèmes à base d'agents de prévoir toutes les situations que les agents vont rencontrer et de spécifier leur comportement à l'avance. C'est précisément cette possibilité, que d'aucuns appellent flexibilité, qui constituent l'un des attraits du paradigme d'agent. Les agents doivent dès lors apprendre et s'adapter à leur environnement, surtout un environnement de système multi-agents. Nous présentons dans cette section quelques considérations sur l'apprentissage dans les SMA. Bien que nous n'ayons pas implémenté l'apprentissage automatique au sein des agents de SIGFAD, nous pensons que ceci constitue l'une des perspectives de notre recherche. Notre souci est aussi de montrer que nous sommes sensibles à la nécessité de doter nos agents des capacités d'apprentissage.

4.3.1 Généralités

Pendant longtemps toutefois, les recherches en apprentissage machine se sont déroulées sans référence aux travaux sur les agents logiciels intelligents. Ceci est assez surprenant car la capacité à apprendre et à s'adapter est l'un des aspects les plus importants de l'intelligence. L'intelligence implique un certain degré d'autonomie ; laquelle autonomie implique la capacité à prendre des décisions. Les agents doivent disposer d'outils leur permettant de prendre de telles décisions. En effet, dans la plupart des environnements dynamiques, le

concepteur ne peut prévoir toutes les situations. En particulier dans les systèmes multi-agents, le comportement global émerge souvent des interactions et ne peut être défini à l'avance. L'apprentissage est donc ainsi un élément crucial de l'autonomie et devrait retenir une attention majeure dans les recherches sur les agents et les systèmes multi-agents.

Les agents et les systèmes multi-agents peuvent être vus comme un domaine d'application en apprentissage machine, avec bien sûr des enjeux spécifiques. Les recherches se plaçant dans cette optique consistent très souvent à appliquer des algorithmes d'apprentissage existants (pour un agent unique) plus ou moins directement à des systèmes à base d'agents. Vu ainsi, l'apprentissage multi-agents est perçu comme une propriété émergente. Malgré l'intérêt que ceci revêt pour les recherches dans les SMA, ceci est moins intéressant pour les spécialistes de l'apprentissage machine. A l'opposé, les SMA posent le problème de l'apprentissage distribué, c'est-à-dire plusieurs agents apprenant séparément pour acquérir un point de vue commun. Plusieurs algorithmes d'apprentissage ont été développés pour des agents apprenant séparément sur des hypothèses indépendantes. A partir du moment où le processus d'apprentissage est distribué sur plusieurs agents, il est nécessaire de modifier significativement ces algorithmes et même d'en écrire de nouveaux. En apprentissage distribué, les agents ont besoin de coopérer et de communiquer pour apprendre. Il apparaît très important que les communautés de chercheurs en SMA et en apprentissage machine joignent leurs efforts car chacune des communautés a certainement beaucoup à apprendre de l'autre. Ceci fort heureusement commence à être compris à en juger par l'intérêt que les spécialistes des systèmes à base d'agents manifestent pour l'apprentissage machine et vice versa. Dans la suite de cette section, nous allons examiner certains algorithmes d'apprentissage machine existants et applicables aux systèmes multi-agents.

4.3.2 - Définition et principes

Un programme informatique apprend une classe de tâches (de même nature) T à partir d'une expérience E (ensemble d'expériences passées) mesurable par une fonction d'évaluation P , si ses performances dans l'accomplissement des tâches de T s'améliorent avec l'expérience E .

Avec un programme informatique, la classe de tâches T doit être relativement restreinte compte tenu de la capacité des ordinateurs modernes, de la qualité de données à traiter (bruit, données manquantes), de la complexité de la tâche. La fonction d'évaluation P doit être

quantitative ; sinon il faut recourir à une fonction qui permet de passer du qualitatif au quantitatif comme avec la logique floue par exemple. L'expérience E doit être une sélection parmi les événements passés : données, stratégies, résultats, tests, abstractions sous un format manipulable par un ordinateur.

La conception d'un algorithme d'apprentissage comporte plusieurs étapes. En premier lieu, il faut définir précisément la tâche t à accomplir, la fonction d'évaluation v associée à la tâche, E l'expertise passée utile à l'apprentissage. Il faut se soucier de la représentation et de la manipulation de ces entités : quels sont les opérateurs nécessaires pour la génération d'hypothèses (évaluer, comparer, inférer, planifier) et la mise à jour de E (générer, spécialiser, exclure, joindre). La mise au point de l'algorithme exige de savoir d'où viendra la rétroaction, comment elle sera représentée. Est-elle crédible ? Sera-t-elle compatible avec la fonction d'évaluation v ou son approximation v^* ? Comment représenter et modifier la fonction v^* ? Quelles sont les stratégies d'exploration de E ?

L'apprentissage automatique traite de plusieurs problématiques :

- apprentissage de concepts : selon des généralisations/spécialisations ou selon des mesures de similarité statistique ;
- classification conceptuelle : apprentissage de multiples concepts, structuration de la mémoire, arbres de décision (génération des règles de classification), analyse formelle des concepts (liens entre classes), *data mining* (découverte de connaissances dans des corpus de données immenses) ;
- apprentissage à base d'explications : expliquer la différence ou le lien entre v et v^* , apprendre à partir de cette différence ;
- réseaux de neurones : simulation du comportement du cerveau, bases en physiologie humaine (renforcement naturel) ;
- algorithmes génétiques : recherche dans un espace de recherche immense, bases en génétique (amélioration ciblée) ;
- apprentissage par renforcement : l'approximation de la fonction d'évaluation v^* est comprise entre les valeurs -1 et 1 , sans autre explication.

4.3.3 - L'apprentissage par les agents

4.3.3.1 - Apprentissage mono-agent et apprentissage multi-agents

Jusqu'à cette date, plusieurs algorithmes d'apprentissage ont été développés dans une perspective mono-agent. Comment de tels algorithmes peuvent-ils être utilisés dans un environnement multi-agents ? D'après Stone et Veloso (2000), l'apprentissage mono-agent se focalise sur la façon dont un agent améliore ses compétences individuelles sans prendre en compte l'environnement dans lequel il se trouve. L'on ne peut parler d'apprentissage multi-agents si ce qu'un agent apprend n'affecte pas (et n'est pas influencé par) ce que font les agents voisins. Un agent placé dans un environnement multi-agents peut-il apprendre sans affecter (ou être affecté par) les autres agents ? Même si l'agent n'a pas une connaissance explicite des autres agents, ceux-ci seront perçus comme faisant partie de l'environnement. Ces agents influencent donc (et sont affectés par) les hypothèses que le premier agent apprend. Certains auteurs (Sugawara et Lesser, 1993) affirment qu'il est possible d'obtenir un comportement de groupe à partir de l'apprentissage mono-agent. Des chercheurs ont montré que les performances en apprentissage mono-agent peuvent diminuer lorsque l'agent prend conscience de la présence des autres agents (Mundhe et Sen, 2000). D'un autre côté, l'apprentissage mono-agent ne conduit pas toujours à un niveau de performance optimal dans les environnements multi-agents. Il existe des domaines où l'apprentissage multi-agents coordonné est une métaphore naturelle et conduit à de meilleures performances. Le choix entre l'apprentissage mono-agent et l'apprentissage multi-agents reste une question ouverte.

4.3.3.2 - Les méthodes d'apprentissage hors connexion et en ligne

Les algorithmes d'apprentissage en ligne (ou incrémentales) comme les réseaux de neurones et (dans une certaine mesure) l'apprentissage par renforcement ont été utilisés pour établir de façon incrémentale de nouvelles hypothèses dès qu'on dispose de nouveaux exemples. De l'autre côté, les méthodes d'apprentissage hors connexion induisent des hypothèses à partir d'un ensemble d'exemples disponibles à une date précise. Les algorithmes en ligne semblent mieux convenir aux systèmes multi-agents ; ici, les agents doivent constamment réactualiser leurs connaissances. Toutefois, il serait souhaitable d'utiliser aussi des algorithmes hors connexion. Dans cette optique, l'agent doit rassembler un ensemble d'exemples et décider à une date donnée de générer de nouvelles hypothèses. Le problème majeur consisterait alors à

déterminer les exemples à prendre en compte et le moment auquel il faut exécuter l'algorithme d'apprentissage.

4.3.4 - Les techniques d'apprentissage

Nous examinons ici différentes techniques d'apprentissage applicables aux systèmes multi-agents.

4.3.4.1 - Apprentissage logique et apprentissage réactif

Dans les systèmes réactifs, le comportement global émerge de l'interaction entre les comportements des composants. Plutôt que de concevoir des protocoles de coordination ou de doter les agents de modèles de reconnaissance complexes, les entités travaillent sur des informations quantitatives (comme la distance minimale de séparation avec ses voisins) pour produire un comportement social. En l'absence de traitement interne, ces techniques permettent aux agents de réagir pratiquement en temps réel. Dans le *Q-learning*, les agents réactifs disposent d'une description de leur état actuel et doivent choisir leur prochaine action de façon à maximiser la valeur de renforcement reçue après chaque action.

Si l'on veut simuler des interactions sociales complexes (marchés, conflits) ou ne pas réduire le comportement de l'agent à des règles situation-action, il faut des agents qui ne soient pas simplement réactifs. Pour doter les agents d'un comportement plus sophistiqué, il faut leur permettre de conceptualiser leur expérience. Bien qu'ils soient principalement issus des techniques hors connexion, l'apprentissage à base d'explications et la programmation logique inductive constituent de bons outils pour remédier aux systèmes d'apprentissage réactifs. L'apprentissage à base d'explications peut être utilisé par les agents pour améliorer l'efficacité dans la solution d'un problème plutôt que pour augmenter leurs connaissances. L'apprentissage à base d'explications permet d'extraire des règles générales à partir d'exemples en expliquant pourquoi le système arrive ou non à résoudre un problème. A l'opposé de l'apprentissage à base d'explications, la programmation logique inductive pose une hypothèse à partir de circonstances externes non connues à l'avance. La programmation logique inductive déduit des hypothèses à partir des exemples et des connaissances acquises. Les agents déterminent les exemples à partir des actions exécutées, des plans et des résultats obtenus. Estlin (1998) a montré comment l'apprentissage à base d'explications et la programmation logique inductive peuvent être combinés pour implémenter l'apprentissage

mono-agent. Plusieurs chercheurs (Alonso et Kudenko, 1999) étudient la possibilité de combiner ces deux techniques pour implémenter l'apprentissage dans des systèmes multi-agents, complexes et dynamiques.

4.3.4.2 - L'apprentissage social

L'apprentissage social est influencé à la fois par l'intelligence artificielle et la biologie. Elle constitue une alternative aux approches logiques présentées plus haut.

Considérons un système multi-agents dans lequel de nouveaux agents arrivent dans un environnement hébergeant des agents plus expérimentés. Dans un tel contexte, un agent peut ne pas tout apprendre par lui-même, mais bénéficier de l'accumulation des connaissances d'agents plus expérimentés.

L'apprentissage social semble convenir davantage aux systèmes multi-agents simulant la vie artificielle. Les mécanismes d'apprentissage social sont fortement influencés par les travaux en biologie. Le lecteur intéressé peut utilement consulter (Weiss, 1999) et (Alonso et Kudenko, 1999).

4.4 - Les outils de développement de SIGFAD

4.4.1 - La méthodologie Multiagent System Engineering (MaSE)

Nous allons dans cette partie présenter la méthodologie MaSE que nous avons retenue pour le développement et la spécification de SIGFAD, notre système multi-agents. MaSE a été développée par Scott DeLoach et ses collaborateurs au sein du Laboratoire d'Intelligence Artificielle du *Air Force Institute of Technology* (AFIT). Cette méthodologie appartient à la catégorie des méthodologies qui s'inspirent des méthodes de spécification et de développement des systèmes à base d'objets. Nous avons choisi MaSE pour plusieurs raisons. Elle bénéficie d'une abondante littérature disponible sur le site de ses auteurs (MaSE [http](http://)), littérature qui a le mérite de décrire dans le détail les différentes étapes de la méthodologie. Elle part d'une définition de l'agent et des SMA très modeste et très pragmatique : ici, un agent est considéré comme un ensemble de processus informatiques qui communiquent entre eux pour atteindre un objectif global donné. Elle s'appuie sur AgentTool, un environnement qui permet de déployer la méthodologie et de procéder à un certain nombre de vérifications.

La méthodologie MaSE met l'accent sur la façon de coordonner dans les SMA le comportement de chaque agent afin d'obtenir un comportement du système tout entier. Pour cela, MaSE utilise un certain nombre de modèles graphiques pour décrire le type d'agents, les interfaces inter-agents, une architecture indépendante de la structure interne des agents. Dans MaSE, un agent est considéré comme une abstraction de plus haut niveau qu'un objet. Alors qu'un objet peut voir ses méthodes invoquées par n'importe quel autre objet, les agents interagissent entre eux à travers des interactions (*conversations*) et prennent des initiatives pour accomplir leurs buts personnels et les objectifs du système tout entier. Un agent n'est pas caractérisé ici par le fait qu'il possède ou non une intelligence : à ce titre MaSE permet de développer des agents intelligents ou pas. En revanche, l'agent étant considéré comme une spécialisation d'un objet, MaSE utilise des techniques propres au développement des objets et les applique au développement des SMA. L'objectif de MaSE est d'aider le concepteur à analyser, concevoir et implémenter un SMA à partir d'un cahier des charges initial. MaSE comporte sept étapes réparties en deux phases. Chacune des étapes a pour résultat, un ou plusieurs diagrammes. La phase d'analyse comprend trois étapes :

- l'*identification des objectifs*, dont le résultat est le *diagramme hiérarchisé des objectifs* ;
- l'*identification des cas d'utilisation*, dont le résultat est un *ensemble de cas de d'utilisation* exprimés par un ou plusieurs *diagrammes de séquence* (à la UML) ;
- l'*affinage des rôles*, dont le résultat est le *modèle de rôles de MaSE* et un ensemble de *diagrammes de tâches concurrentes*.

La phase de conception comporte quatre étapes :

- *la création des classes d'agent*, dont le résultat est le *diagramme de classes d'agent* ;
- *la construction des interactions entre agents*, dont le résultat est un *ensemble de diagrammes d'interactions* ;
- *l'assemblage des classes d'agents*, dont le résultat est *l'architecture interne des agents* ;
- *la conception du système*, dont le résultat est le *diagramme de déploiement* (à la UML).

Bien que la méthodologie s'énonce en une succession d'étapes où les résultats d'une étape servent à l'étape suivante, dans la pratique, MaSE est itérative et demande souvent des allers et retours entre les différentes étapes. MaSE offre l'avantage de repérer toute entité apparaissant dans une étape dans les autres étapes. Un objectif identifié dans la première étape

se retrouve dans un rôle spécifique, une tâche et une classe d'agent ; de même, à partir d'une classe d'agent, on peut remonter aux tâches et rôles qu'elle doit accomplir.

4.4.1.1 - La phase d'analyse

Cette phase a pour but de produire un ensemble de rôles dont les tâches décrivent ce que doit faire le système pour atteindre ses objectifs ou les exigences de son cahier des charges. Un rôle décrit une entité qui remplit une fonction dans le système, il est responsable ou aide à l'accomplissement d'un objectif ou sous-objectif. Les spécifications du cahier des charges sont transformées en un ensemble d'objectifs. Un objectif résume un ensemble de fonctionnalités du SMA.

Le SMA a un objectif global décomposé en un ensemble de sous-objectifs. La phase d'analyse permet de définir ces objectifs et les rôles nécessaires pour leur accomplissement. Plutôt que d'essayer de déduire les rôles à partir des objectifs, MaSE suggère l'utilisation des cas d'utilisation afin de définir et de valider les rôles. Les cas d'utilisation exprimés à l'aide de diagrammes de séquence permettent aussi d'appréhender les interactions entre les différents rôles.

Identification des objectifs

La première étape de MaSE consiste à transformer les spécifications initiales du cahier des charges du SMA en un ensemble structuré d'objectifs. Cette étape correspond à la phase d'analyse dans le cycle du développement du logiciel classique. Les fonctionnalités attendues du système sont supposées être facilement obtenues à partir des spécificités du cahier des charges. Les fonctionnalités indiquent à l'analyste les services que le système doit fournir et le comportement du système suivant les informations en entrée et son état courant. La première étape consiste à transformer ces fonctionnalités en objectifs du système, ceci signifie que les objectifs ne sont pas regardés du point de vue de l'utilisateur mais bien du système. MaSE est basée sur les objectifs du système car ces derniers sont plus stables que les fonctions, procédures et structures de données qui elles, peuvent évoluer au fil du temps.

L'identification des objectifs du système se fait en deux temps, à savoir le recensement des objectifs d'une part et leur structuration d'autre part. Le recensement des objectifs s'opère en extrayant de l'ensemble des fonctionnalités contenues dans le cahier des charges initial l'ensemble des scénarios ; on décrit ensuite l'objectif de chaque scénario. Les objectifs

doivent rester à un bon niveau d'abstraction et s'affranchir de détails ; on ne dira rien ici de la façon de les atteindre : ce sont les fondements du modèle de rôles de MaSE (*MaSE Role Model*). Les rôles et les tâches définis dans les étapes suivantes doivent accomplir l'un des objectifs identifiés au cours de cette étape. La deuxième sous-étape consiste à structurer les objectifs de façon hiérarchique ; on produit ici un diagramme hiérarchisé d'objectifs. Les objectifs placés au même niveau sont des « pairs ». Le sommet du diagramme est constitué de l'objectif global du système. Il arrive souvent qu'on ne puisse pas déduire un objectif global de l'ensemble des fonctionnalités ; dans ce cas, les objectifs les plus élevés sont résumés en un objectif global et les premiers deviennent les sous-objectifs du dernier.

Identification des cas d'utilisation

L'objectif de cette étape est de déduire un ensemble de cas d'utilisation à partir du cahier des charges initial et de créer un ensemble de diagrammes de séquence. Les diagrammes de séquence aident à identifier un ensemble initial de rôles et les différentes interactions présentes dans le système. Les cas d'utilisation définissent des scénarios de base que le système peut exécuter. Les diagrammes de séquence décrivent les cas d'utilisation comme un ensemble d'événements survenant entre les rôles constituant le système. Les séquences d'événements sont utilisées plus loin pour définir les tâches qu'un rôle donné doit accomplir. Les tâches seront éventuellement implémentées sous forme d'interactions entre les agents, traduisant ainsi la mise en œuvre du cas d'utilisation dans le SMA.

Les cas d'utilisation reflètent la vision du comportement du SMA offerte à l'utilisateur. Ils sont extraits à partir des fonctionnalités initiales, des propos des utilisateurs ou de toute autre source existante. Bien qu'un nombre élevé de cas d'utilisation soit d'un usage pratique pour la compréhension du système, il convient de ne pas trop en exhiber ; le but des cas d'utilisation est d'explicitier les interactions dans le système et non de définir toutes les combinaisons possibles d'événements et de données pouvant subvenir dans le système. Le concepteur s'efforcera de montrer à travers les cas d'utilisation comment chaque objectif sera atteint, il devra aussi considérer les cas d'utilisation en fonctionnement normal et les cas d'utilisation illustrant un fonctionnement anormal du système. Les auteurs envisagent pour illustrer ce dernier cas de figure, d'utiliser des obstacles comme un concept dual des objectifs.

Les diagrammes de séquence utilisés dans MaSE sont les mêmes que ceux contenus dans UML par exemple. Un diagramme de séquence représente une séquence d'événements transmis entre les différents rôles identifiés à partir des cas d'utilisation. En général, un diagramme de séquence est créé pour un cas d'utilisation, mais on peut aussi avoir plusieurs diagrammes de séquence pour illustrer un cas d'utilisation.

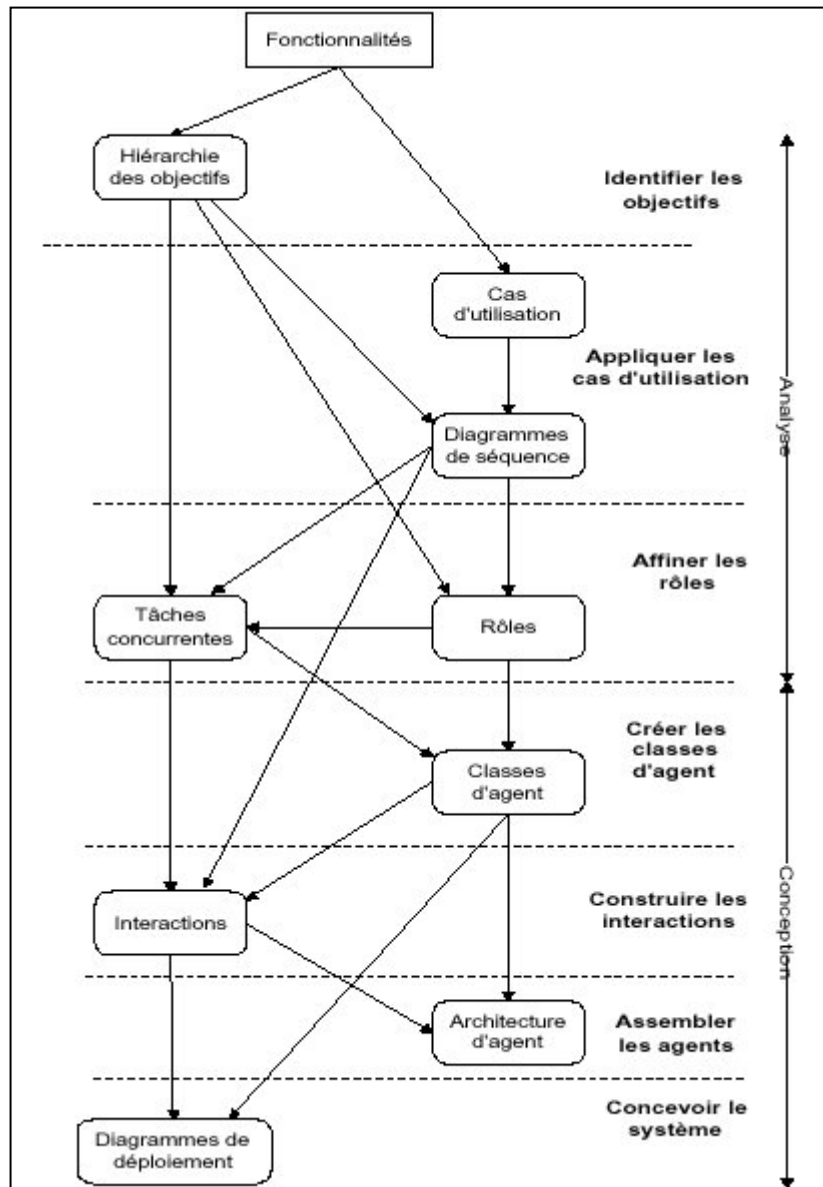


Figure 4.3 : Les étapes et les phases de la méthodologie MaSE

Affinage des rôles

La troisième étape de la phase d'analyse consiste à affiner les rôles, c'est-à-dire à transformer les objectifs hiérarchisés et les diagrammes de séquence en un diagramme de rôles et de tâches associées. Les rôles constituent la fondation de la définition des classes d'agents et

représentent les objectifs du système durant la phase de conception. MaSE se fonde sur l'assertion selon laquelle tous les objectifs du système seront atteints si chacun d'entre eux est associé à un rôle et si chaque rôle est rempli par une classe d'agents. En général, chaque objectif est associé à un rôle, mais il existe des situations où un même rôle peut remplir plusieurs objectifs. Dans l'affinage des rôles, on a plusieurs considérations. Des objectifs similaires ou voisins doivent être combinés en un rôle unique. On peut aussi découvrir à ce stade des objectifs qui n'avaient pas été identifiés plus tôt ; si c'est le cas, ceux-ci seront inclus dans le diagramme hiérarchisé des objectifs comme s'ils appartenaient aux fonctionnalités initiales. Des objectifs voisins peuvent aussi être associés à un rôle unique. L'interfaçage avec des ressources internes ou externes requiert en général un rôle séparé ; l'utilisateur humain est généralement considéré comme une ressource externe. Dans MaSE, l'interaction homme-machine n'est pas modélisée de façon explicite, il est suggéré de créer un rôle spécifique pour encapsuler l'interface utilisateur ; on peut alors définir les interactions avec le système sans devoir définir l'interface elle-même. Les autres ressources telles que les bases de données, les fichiers ou des systèmes existants requièrent un rôle pour leur interfaçage.

La définition des rôles se fait dans le *MaSE Role Model* qui contient des informations sur les interactions entre les tâches. Les rôles sont représentés dans des rectangles et les tâches associées dans des ellipses reliées aux rectangles. Chaque tâche est détaillée dans un diagramme de tâches. Les lignes entre les tâches représentent les protocoles de communication entre les tâches, les flèches indiquent l'initiateur et le destinataire. Les lignes continues indiquent des communications entre différents rôles alors que les lignes interrompues représentent des communications entre des tâches appartenant à un même rôle. Les rôles ne doivent pas partager ou dupliquer des tâches. Après avoir créé les rôles, des tâches associées définissent le comportement du rôle en vue de remplir avec succès ses objectifs. En général, un rôle donné a plusieurs tâches concurrentes qui définissent son comportement. Chaque tâche spécifie un fil de contrôle (*thread*) unique définissant un comportement particulier du rôle dans ses interactions. Les tâches concurrentes sont spécifiées graphiquement au moyen des automates à états finis.

4.4.1.2 - Phase de conception

La phase de conception comporte quatre étapes. Dans la première étape qui est la création des classes d'agents, on associe les rôles à des types d'agent spécifiques. La deuxième étape permet de définir les interactions entre les différentes classes d'agents. La troisième étape consiste à définir l'architecture interne et les processus de raisonnement des agents. Dans la dernière étape, le concepteur définit le nombre et la localisation des agents une fois que le système sera déployé.

Création des classes d'agents

Les classes d'agent sont créées à partir des rôles définis lors de la phase d'analyse. Le résultat de cette étape est le diagramme de classes d'agent qui montre l'organisation du système en classes d'agent et les interactions entre elles. Une classe d'agent est une *template* (comme dans UML) pour un type d'agent du système et est similaire aux classes d'objets dans la POO. Les classes d'agents sont définies à partir des rôles qu'ils vont jouer et des interactions les concernant. Chaque rôle identifié plus haut doit être rempli par au moins une classe d'agents. Une classe d'agents peut jouer plusieurs rôles, ceux-ci changeant au cours du temps et plusieurs agents appartenant à une même classe peuvent jouer plusieurs rôles au même instant. Un flux d'informations trop important entre deux rôles montre que ceux-ci doivent être joués par une seule et même classe d'agents. De même, deux rôles exigeant de grandes capacités de traitement doivent être assignés à des classes d'agents différentes afin qu'ils puissent être exécutés sur des processeurs différents. Les interactions dans lesquelles sont engagées les classes d'agents sont déduites des communications externes des rôles qui leur sont associés. Les classes d'agents et les interactions associées sont représentées dans un diagramme de classes d'agents comparable aux diagrammes de classes d'objets. Il existe néanmoins deux différences entre ces deux diagrammes : la première est que les classes d'agents ne sont pas définies par les attributs et les méthodes mais plutôt par les rôles qu'elles jouent ; la deuxième différence est que dans les classes d'agents, toutes les relations représentent des interactions entre classes d'agents. Le diagramme de classes d'agents est la première entité de MaSE décrivant le SMA tout entier et la façon dont il sera éventuellement implémenté.

Construction des interactions

Cette étape a pour but de définir le détail des interactions entre agents. On se sert ici des informations contenues dans les diagrammes de tâches concurrentes. Une interaction MaSE

définit un protocole de coordination entre deux agents, elle est représentée par deux diagrammes d'interaction : l'un pour l'émetteur et l'autre pour le destinataire. Chaque diagramme consiste en un automate à états finis. Une communication commence toujours par un message de l'émetteur ; lorsque le destinataire reçoit le message, il le compare avec ses communications actives. S'il trouve une correspondance, il opère la transition nécessaire et migre vers un nouvel état, il effectue les actions nécessaires contenues dans la transition ou le nouvel état dans lequel il se trouve. S'il ne trouve pas de correspondance avec le message reçu, c'est qu'il s'agit d'une nouvelle communication et il la compare avec toutes les interactions pouvant apparaître avec l'agent émetteur. S'il trouve une correspondance, il engage une nouvelle communication. Les détails des interactions entre agents sont définis à partir des informations contenues dans les diagrammes de tâches concurrentes.

Assemblage des classes d'agents

Durant cette étape, l'architecture interne des agents est définie. Cette étape comporte deux sous-étapes : définition de l'architecture et définition des composants de cette architecture. Les concepteurs ont le choix entre définir leur propre architecture ou utiliser des architectures prédéfinies comme l'architecture BDI. On peut aussi utiliser des composants prédéfinis ou en créer soi-même. Un composant consiste en un ensemble d'attributs et de méthodes et peut lui-même avoir une sous-architecture s'il est trop complexe. Les auteurs de MaSE envisagent la possibilité de définir une architecture directement à partir des tâches identifiées dans la phase d'analyse ; l'idée serait que chaque tâche d'un rôle devienne un composant dans la classe d'agents correspondante, la tâche concurrente se transformant en une combinaison de composants du diagramme des états internes et de communications. Les activités identifiées dans la tâche deviendraient les méthodes du composant.

Conception du système

La dernière étape de MaSE consiste à instancier les classes d'agents définies préalablement. On utilise un diagramme de déploiement pour montrer le nombre, le type et la localisation des agents dans le système.

4.4.2 - L'architecture JAM

La méthodologie MaSE ne définit et n'impose aucune architecture interne des agents mis en œuvre. Cette tâche est laissée au concepteur. Pour notre part, nous avons choisi le modèle

d'architecture JAM pour définir la structure interne et implémenter le comportement de nos agents.

Le modèle JAM a été élaboré par Huber et ses collègues (Huber, 2001). JAM est une architecture hybride qui se fonde sur les idées et les théories du système de raisonnement procédural (PRS pour *Procedural Reasoning System*), de la sémantique des circuits structurés (SCS pour *Structured Circuit Semantic*) et du Act plan interlingua. JAM offre des représentations procédurales riches et étendues, un raisonnement rationnel et de niveau méta prenant en compte plusieurs objectifs en même temps, un comportement basé à la fois sur les objectifs poursuivis et la réaction aux événements. L'architecture JAM permet aussi d'implémenter des capacités de mobilité ; elle 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 4.4 : 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 de l'agent dans la réalisation de ses 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 (par exemple, 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.

4.4.2.1 - L'interpréteur

L'interpréteur sélectionne et exécute les plans sur la base des intentions, plans, objectifs et croyances de l'agent. Il 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.

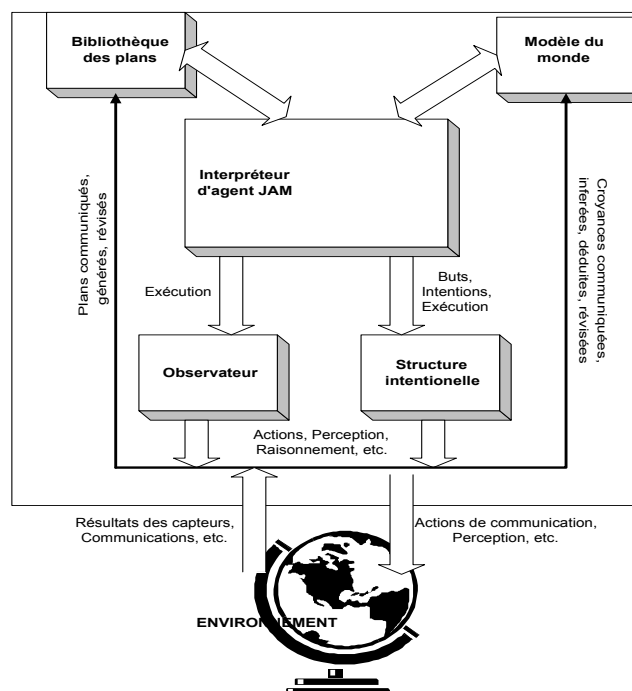


Figure 4.4 : L'architecture JAM tirée de (Huber, 2001)

4.4.2.2 - 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 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).

4.4.2.3 - 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 ; 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).

4.4.2.4 - Le modèle 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.

4.4.2.5 - 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.

4.4.3 – La communication entre les agents

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 pour *Remote Method Invocation*) et la technologie CORBA (*Common Request Broker Architecture*).

Plusieurs travaux concernent la mise en œuvre d'infrastructures de communication qui dispensent le programmeur d'implémenter lui-même la communication entre les agents. Avec ce type d'outils, le fonctionnement typique consiste à fournir une adresse à chaque agent. Cette adresse lui permettra d'émettre et de recevoir des messages de la part des autres agents. L'avantage est la possibilité d'avoir des communications asynchrones et d'être dispensé de la nécessité de localiser un agent avant de lui envoyer un message. JATLite est une infrastructure de ce type, elle permet aux agents de communiquer via Internet.

4.4.3.1 - 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 réseau 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 des fonctionnalités telles que l'intégration de bases de données existantes ou l'accès aux services d'un objet serveur sans se soucier de sa localisation physique, 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 ORB, mais permet une compatibilité entre les différents ORB des fournisseurs. Ainsi, l'OMG définit un ensemble de services qui sont accessibles par l'intermédiaire d'interfaces standard. 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* utilisés 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 intégrée à Java à travers les RMI ; toutefois, là où le RMI rend possible les RPC entre des objets Java, CORBA le fait entre des objets implémentés dans n'importe quel langage.

4.4.3.2 - JATLite

JATLite (*Java Agent Template, Lite*) est 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 ainsi que d'autres langages de requêtes comme SQL et FIPA ACL.

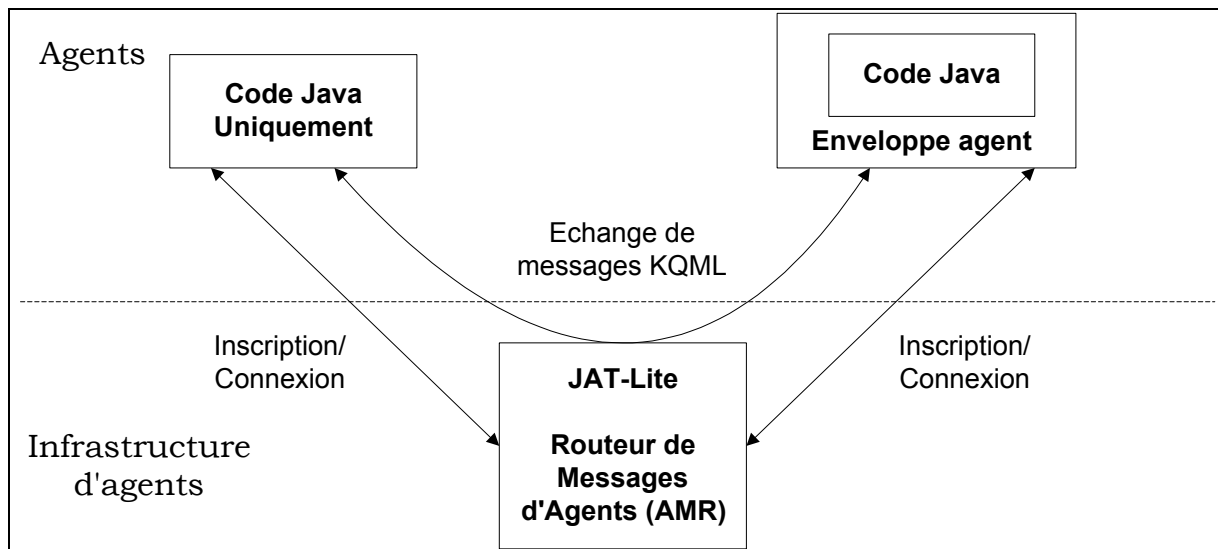


Figure 4.5 : Le Routeur de messages de JATLite

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 de Mark Cutkosky et 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 existe entre chaque agent connecté et le 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 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.

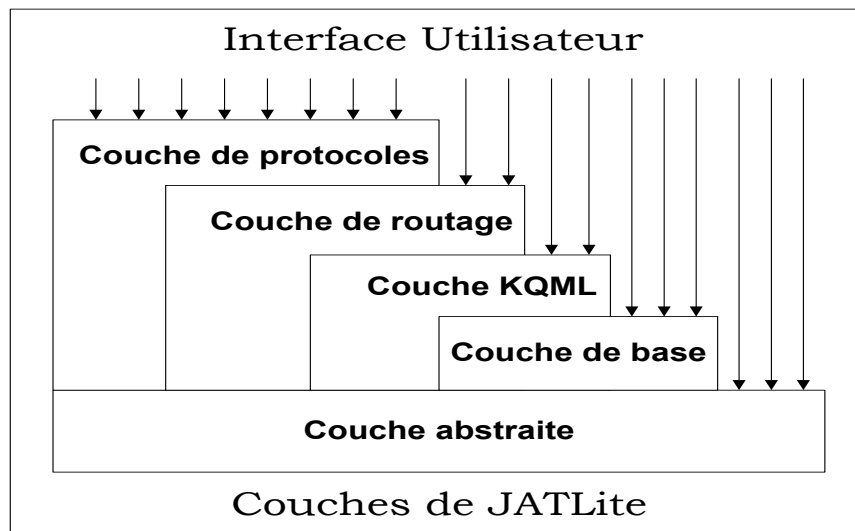


Figure 4.6 : L'architecture de JATLite est basée sur un ensemble de couches spécialisées et hiérarchiques

- 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 au programmeur la possibilité 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 (Pietrie, 2003) 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 d'autres protocoles.

Chapitre 5 : Analyse, conception, spécification et implémentation de SIGFAD

Dans ce chapitre, nous présentons le système multi-agents que nous avons développé en vue de soutenir et de susciter les interactions dans des groupes de formation à distance. Les fonctionnalités de SIGFAD, notre système multi-agents, sont déduites des enseignements que nous avons tirés de l'expérimentation SimuLigne. Nous avons évoqué au chapitre 2 les raisons pour lesquelles nous avons eu recours au paradigme d'agent pour implémenter les fonctionnalités attendues.

Nous présentons dans ce chapitre l'application de la méthodologie d'analyse, de conception et de spécification du SMA. Il s'agit de la méthodologie MaSE et nous utilisons pour cela l'environnement AgentTool qui lui est associé. Ensuite nous sommes passés à la programmation concrète de nos agents en utilisant l'architecture JAM fondée sur le modèle BDI. Nos agents ont été écrits en Java et communiquent à travers des *sockets*. L'utilisateur interagit avec son agent à travers une interface graphique. L'architecture actuelle de SIGFAD repose sur une base de données des interactions qui sert d'interface entre le SMA et la LMS. Elle peut donner à juste titre, la vision d'une centralisation excessive des données, justement contraire à l'approche agent. Cette architecture est le fruit du vécu de SimuLigne. Durant l'expérimentation, le système multi-agents n'existait pas et il a fallu néanmoins extraire les données d'interaction, les conserver et les rendre accessibles facilement : le choix d'une base de données pour le faire s'imposait de façon naturelle. La mise en œuvre de la partie de SIGFAD consistant à extraire et à sauvegarder les données d'interactions nous permettra de nous affranchir de cette architecture et d'évoluer vers une décentralisation des données, plus conforme à l'approche agent. Après avoir identifié l'ensemble des fonctionnalités que doit posséder SIGFAD pour répondre entièrement aux différents problèmes identifiés, nous avons pris le parti de dégager un sous-ensemble d'objectifs pilotes. C'est ce sous-ensemble que nous avons effectivement implémenté dans le cadre de nos travaux de thèse. C'est un choix guidé par les contraintes de temps et par le souci d'arriver à une implémentation réelle d'un système multi-agents. Nous pensons, ainsi que nous l'avons rappelé à différents endroits de ce mémoire, que la mise en œuvre concrète des idées et promesses contenues dans la littérature afférente aux systèmes à base d'agents constitue un enjeu de recherche important. Le développement de systèmes mettant effectivement en œuvre des agents contribuera à convaincre les autres chercheurs de la pertinence du paradigme agent. Dans une dernière partie, nous fournissons quelques résultats issus de l'application de SIGFAD au corpus de données de SimuLigne. Il s'agit d'illustrer l'utilisation concrète qui pourrait être faite de notre prototype au cours d'une session de formation à distance. En guise de conclusion à ce

chapitre, nous présentons les difficultés rencontrées dans l'implémentation de SIGFAD. Nous souhaitons ainsi illustrer le déficit actuel d'outils de développement dédiés à la programmation orientée agent.

5.1 - SIGFAD

5.1.1 - Analyse, conception et spécification de SIGFAD

Afin de permettre au lecteur de bien comprendre notre système, nous présentons dans la figure 5.1 une vue d'ensemble de SIGFAD avec les types d'agents présents dans le système. Dans les plates-formes de télé-formation, chaque utilisateur est connecté à son ordinateur et accède aux programmes et à toutes les ressources pédagogiques à travers le protocole HTTP et les autres protocoles Internet (TCP/IP, SMTP, IRC, FTP, etc.). L'utilisateur a donc besoin d'un navigateur Internet pour accéder à la plate-forme et réaliser ses activités d'apprentissage. SIGFAD s'appuie sur la même structure, nous distinguons trois types d'utilisateurs : les apprenants, les tuteurs, le coordonnateur. Les natifs servent de compagnons d'apprentissage aux apprenants et sont classés comme apprenants. Chaque utilisateur a un agent, localisé sur son poste de travail. L'agent devra avoir été installé auparavant. Nous avons finalement retenu cette solution (agent programmé comme une application Java) alors que nous avons envisagé au départ de programmer nos agents (du moins les interfaces agent-utilisateur) sous forme d'*applets*. Ce choix a été dicté par les restrictions imposées aux *applets* (interdiction d'accéder au système de fichiers local, interdiction d'établir des connexions réseau avec des ordinateurs autres que celui à partir duquel l'*applet* a été chargé).

Un agent superviseur des interactions est défini, il réside sur le serveur et est chargé des transactions avec la base de données des interactions. L'agent constructeur de la base de données réside lui aussi sur le serveur et est chargé de construire la base de données des interactions au cours de la formation. Cet agent est celui qui assure l'interfaçage entre la plate-forme de téléformation et notre système multi-agents. Les classes d'agent de SIGFAD sont les suivantes :

- un agent apprenant (un pour chaque apprenant) ;
- un agent tuteur (un pour chaque tuteur) ;
- un agent coordonnateur ;
- un agent constructeur de la base de données des interactions ;

- un agent superviseur des interactions.

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*).

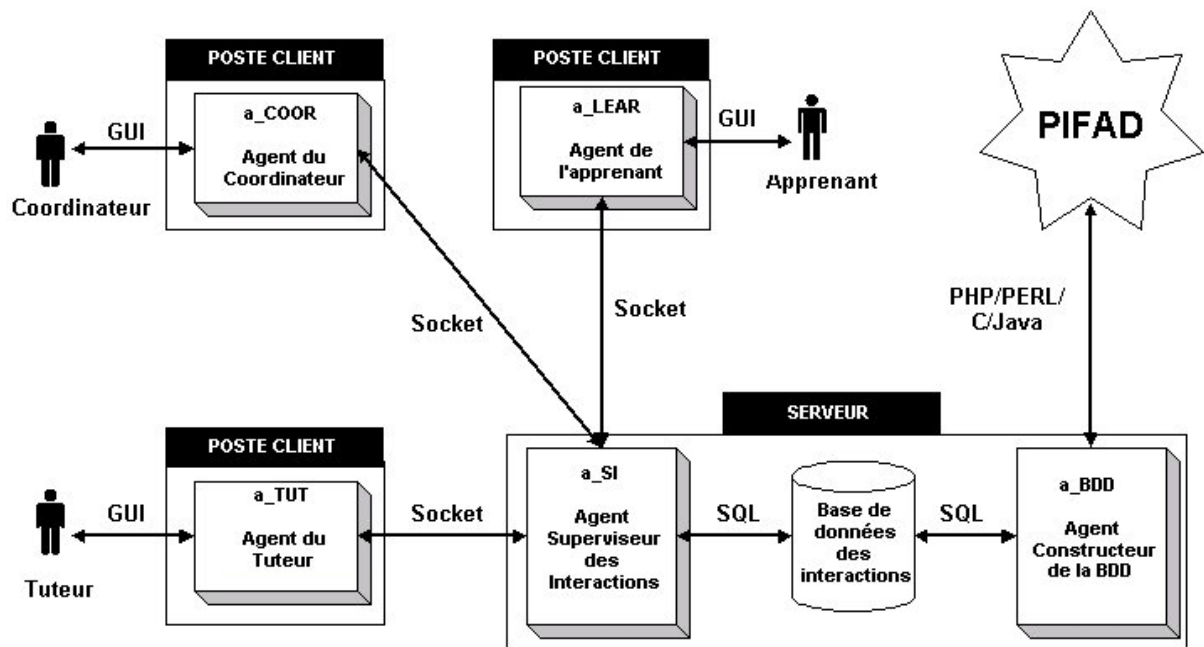


Figure 5.1 : Une vue d'ensemble de SIGFAD

5.1.1.1 - Le diagramme hiérarchisé des objectifs de SIGFAD

Le diagramme hiérarchisé des objectifs a pour principal but 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 disposent de très peu de fonctionnalités de réification de la participation des acteurs aux activités pédagogiques. 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 rémédiation et d'assistance nécessaires. SIGFAD a pour but de remédier aux insuffisances des plates-formes existantes dans la mise en œuvre de l'apprentissage collaboratif distribué. Ses objectifs peuvent être scindés en deux catégories : les objectifs liés à l'assistance aux utilisateurs et les objectifs dédiés à la sauvegarde et à l'extraction des données d'interaction. Il s'agit, pour la deuxième catégorie d'objectifs, d'examiner automatiquement les données d'interaction dans le but d'alerter les utilisateurs lorsque la progression de la session de FAD n'est pas satisfaisante ou que le groupe court un risque d'éclatement. 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.

Dans le but de s'assurer de la faisabilité de notre démarche conceptuelle et des choix techniques opérés, nous avons dégagé un ensemble de fonctionnalités à implémenter concrètement. La base de données des interactions ayant été construite tout au long du déroulement de SimuLigne, nous ne nous sommes pas attardés sur la construction de cette dernière par l'agent qui en est chargé dans SIGFAD. 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 ci-dessus ont été organisées en objectifs :

1. *Déterminer les apprenants présents/absents sur une période, dans un groupe donné.*
Cette fonctionnalité permettrait par exemple au tuteur de connaître à tout moment et sur une période spécifiée la liste des apprenants de son groupe qui se sont connectés. Il convient de préciser que 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, en charge du bon déroulement des activités de son groupe,

pourra déterminer les apprenants encore présents dans la formation. Cette fonctionnalité est évidemment très importante dans le contexte de la FAD, qui connaît des taux d'abandon.

2. *Déterminer les apprenants actifs (qui communiquent et réalisent les activités)*. Nous entendons par apprenant actif, un apprenant qui est présent dans son groupe et réalise les activités didactiques prévues. N'accédant pas au contenu des productions et n'en faisant dans tous les cas pas 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 de ses pairs, les 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 apprécier 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. *Repérer les apprenants dormants (qui 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 de nuancer 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. En revanche, 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 permet 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 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 en 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 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 comparée à d'autres outils de communication comme le bavardage, le tableau blanc, le courriel, le transfert de fichiers, les pages HTML.
7. *Fournir pour chaque utilisateur le ratio temps de connexion effectif/temps 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.

Nous avons organisé les objectifs que nous venons d'énumérer dans un diagramme hiérarchique d'objectifs. Nous y avons inclus les objectifs concernant la sauvegarde des données d'interactions dans la base de données, ainsi que leur extraction. Dans un souci de simplification, nous n'avons pas inclus les objectifs concernant la notification de différents phénomènes ou informations aux utilisateurs, mais ceci se conçoit de façon implicite.

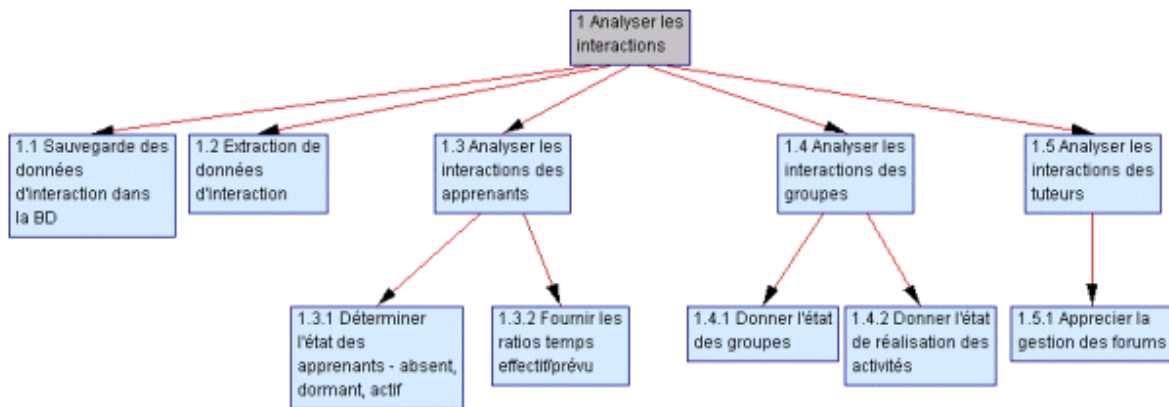


Figure 5.2 : Le diagramme hiérarchisé des objectifs de SIGFAD

5.1.1.2 - L'application des cas d'utilisation

L'objectif de cette étape de MaSE est d'extraire les différents cas d'utilisation du cahier des charges initial du système. Les cas d'utilisation sont explicités 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 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. Nous présentons dans la suite la description de tous les cas d'utilisation de SIGFAD en précisant les données à partir desquelles le SMA réalisera les objectifs associés.

Les diagrammes de séquence dans MaSE indiquent les interactions entre les rôles. Pour l'ensemble de nos cas d'utilisation, nous avons défini les rôles 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 (BDD est la base de données d'interaction) ;
- r_COOR encapsule les tâches effectuées par l'agent du coordonnateur, à savoir : fournir les méta-informations sur les activités et les forums ;
- r_SI encapsule les tâches de l'agent superviseur des interactions, qui consistent principalement à répondre aux requêtes venant des agents tuteur ; il est le seul à faire des transactions avec la base de données des interactions ;
- r_TUT encapsule les tâches de l'agent du tuteur, qui consiste à adresser des requêtes relatives au déroulement de la session de la formation et à l'état du groupe.

Fournir la liste des apprenants présents/dormants/absents durant une période

Ce cas d'utilisation décrit la façon dont le système fournit au tuteur la liste des apprenants absents. La demande, initiée par l'agent du tuteur, est adressée à l'agent superviseur des interactions. Ce dernier va reformuler cette demande sous forme de requête SQL qu'il soumet à la base de données. Il fournit l'information demandée sous la forme d'une liste de noms, à l'agent du tuteur qui peut la transmettre au tuteur. Les rôles concernés sont : r_SI, r_TUT, H_TUT, DataBase

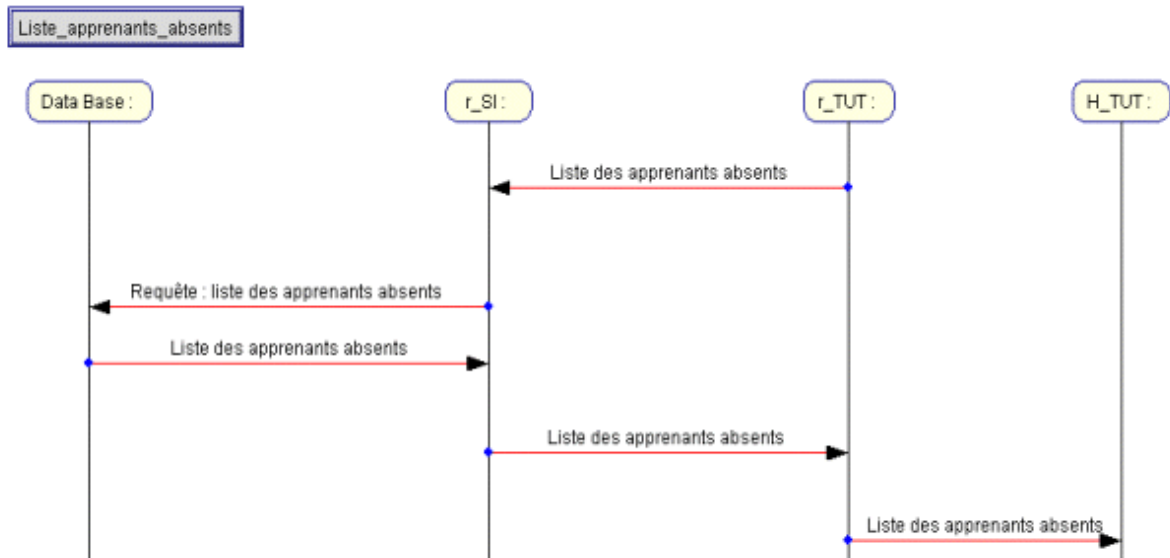


Figure 5.3 : Le diagramme de séquence fournissant la liste des apprenants absents

Ici nous n'avons représenté que le diagramme de séquences concernant la liste des utilisateurs absents. Les messages circulent exactement de la même façon et entre les mêmes entités pour la liste des apprenants présents et dormants. Un apprenant présent est considéré comme présent s'il s'est connecté au moins une fois à PIFAD durant la période spécifiée. Un apprenant ne s'étant jamais connecté durant cette période est classé absent. Pour déterminer les apprenants dits dormants sur une période, SIGFAD répertorie la liste de toutes les actions des utilisateurs à partir des outils utilisés (connexion, accès aux pages de consignes, consultation du calendrier, lecture et émission de messages dans les forums, lecture et écriture de courriels, utilisation du Quizz, remise de fichiers à travers l'outil «A Rendre»). Un apprenant est dit dormant s'il n'a effectué au surplus que des actions consistant à lire des informations. C'est dire que son passage ne laisse aucune trace perceptible par les autres membres de son groupe.

Fournir les ratios temps effectif/temps prévu par semaine

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.

Donner l'état du groupe

Ce cas d'utilisation décrit la façon dont le système détermine et indique l'état du groupe. La demande initiée par l'agent du tuteur est adressée à l'agent superviseur des interactions. La requête est reformatée sous forme SQL et adressée à la base de données des interactions. L'agent superviseur des interactions détermine le pourcentage d'apprenants actifs durant la période spécifiée. 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.

Nous présentons sur la figure 5.4 l'algorithme de détermination de l'état du groupe. L'algorithme comprend trois grandes parties. Déterminer la liste de jalons (il s'agit des tâches constituant les activités) révolus et possibles (un jalon possible est un jalon dont les dates limites font qu'il est possible qu'un apprenant l'ait déjà accompli) pour la période concernée. Ensuite, on établit la liste des apprenants dits 'actifs' en fonction des jalons accomplis et enfin, on calcule le pourcentage d'apprenants actifs dans le groupe. C'est ce dernier qui permettra de donner l'état du groupe. La liste et le nombre d'apprenants actifs sont aussi fournis.

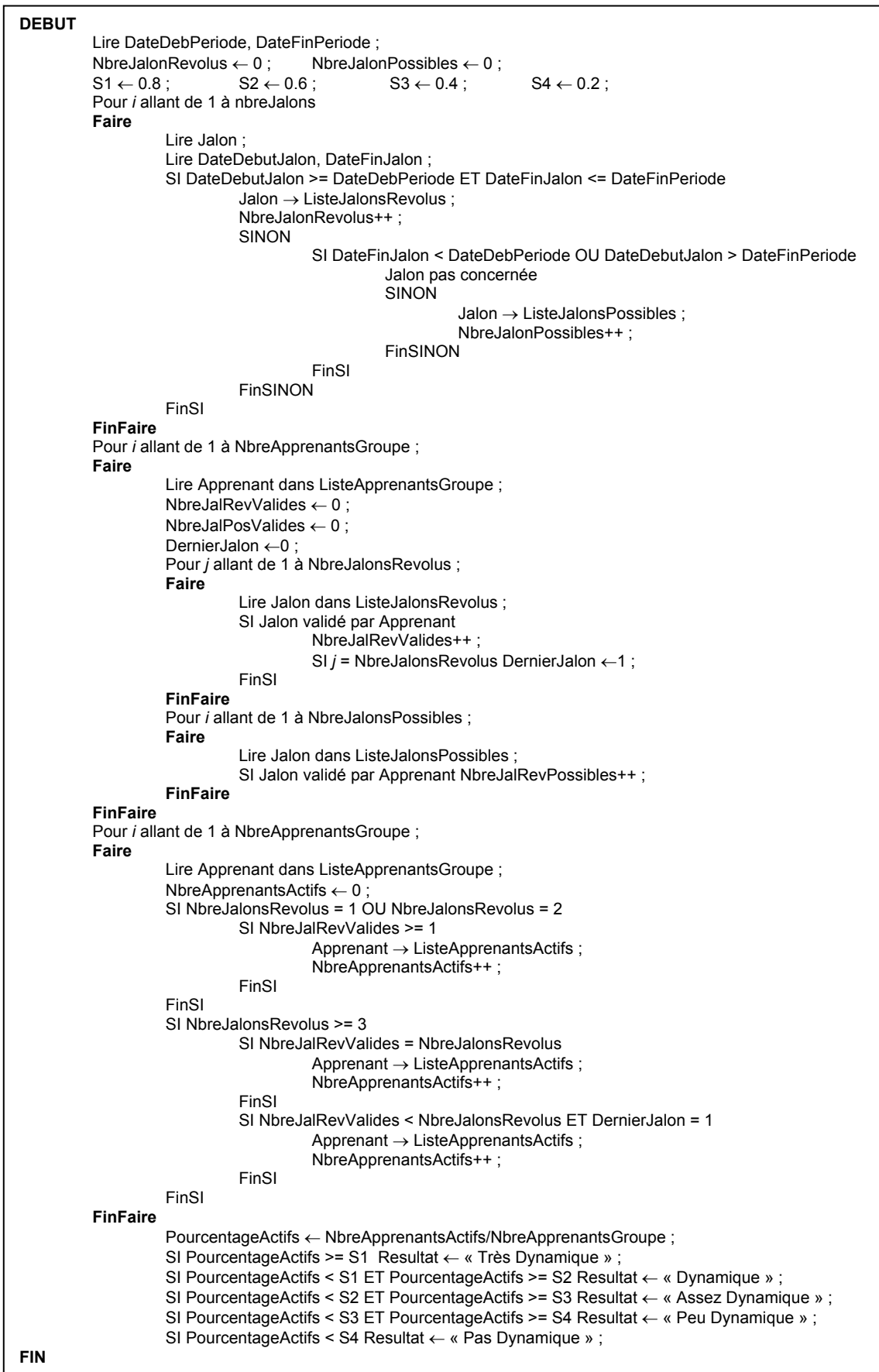


Figure 5.4 : Algorithme donnant l'état du groupe

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 ;
- publier un document ;
- participer à une séance de bavardage ;
- remettre un document par l'outil « A rendre » ;
- utiliser le tableau blanc ;
- télécharger un fichier ;
- télécharger les consignes/cours ;
- accéder à la page des consignes en ligne ;
- etc.

A 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 cours de réalisation. Il fournira à l'agent du tuteur la liste des apprenants effectuant 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 validé l'activité.

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 limites 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.

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 va à travers des requêtes SQL adressées à la base de données indiquer 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). Ce qui vient d'être dit et qui est représenté en annexe dans un diagramme de séquence implémente les caractéristiques de pro-activité de notre SMA.

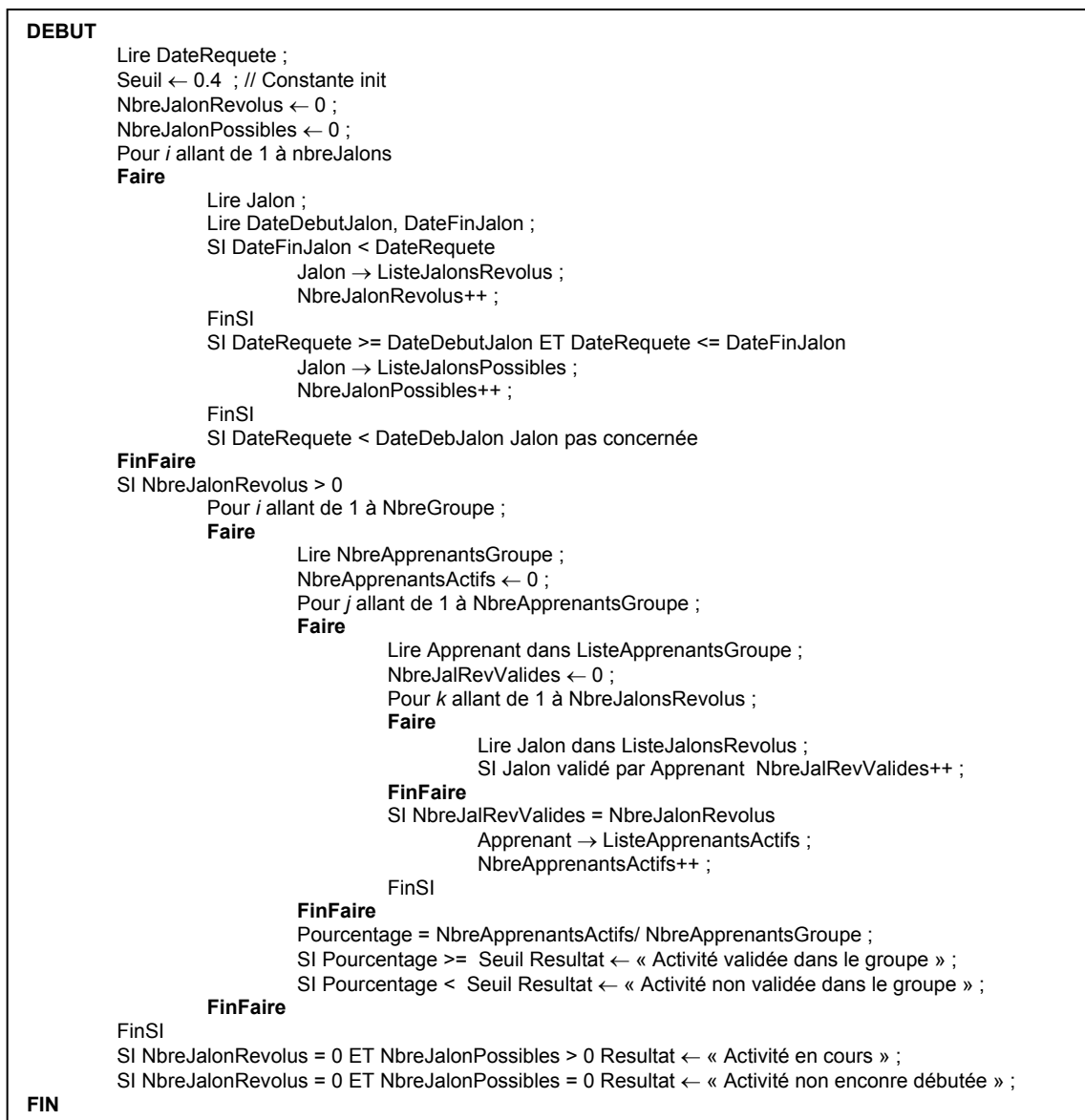


Figure 5.5 : Algorithme de détermination de l'état de réalisation d'une activité

Toutefois, SIGFAD détermine aussi l'état de réalisation d'une activité à la demande de l'agent du tuteur. Dans ce cas, ce dernier indique la date à laquelle, il souhaite connaître l'état de l'activité. L'agent superviseur des interactions en formulant une requête appropriée à la base de données, va indiquer d'abord si l'activité a débuté à cette date ou pas, si certains de ses jalons sont révolus ou si aucun n'est encore révolu. Dans ce dernier cas, l'activité sera dite en cours d'utilisation. Si tous les jalons de l'activité sont révolus à la date indiquée dans la demande de l'agent du tuteur, SIGFAD pourra indiquer si l'activité a été validée ou non. Nous présentons l'algorithme de détermination de l'état d'une activité à la figure 5.5.

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 avec la date 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 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 non, avec des mentions :

- 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 ouvert : si le premier message du forum ne respecte pas les conditions définies ci-dessus ou est inexistant ;
- mal suivi : si le tuteur ne lit pas les messages dans le forum ;
- bien suivi : si le tuteur lit régulièrement (le tuteur lit le message au plus tard 24 heures après la date d'émission dans le forum) 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 (24 heures au plus tôt avant) la date de clôture prévue du forum ;
- mal clôturé : dans tous les autres cas.

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.

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 T du forum. L'évaluation de la gestion du forum est faite à l'ouverture du forum (date de début + T/10), à 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 non 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.

L'ensemble des données pour la satisfaction des requêtes adressées à l'agent superviseur des interactions se trouve dans la base de données des interactions. Elles proviennent principalement du fichier de traces du serveur HTTP et de la base de données interne de PIFAD.

5.1.1.3 - Le diagramme de rôles de MaSE et les tâches concurrentes

L'objectif de cette étape est de transformer le diagramme hiérarchisé des objectifs (figure 5.2) et les diagrammes de séquence (figure 5.3 et figures B.1 à B.6 en annexes) 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. Etant 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.

Le diagramme des rôles de SIGFAD

Dans la figure 5.6, nous présentons le diagramme des rôles de SIGFAD. Le diagramme des rôles de SIGFAD comprend six rôles auxquels sont associés dix-huit tâches concurrentes et 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 faut comprendre par là un chemin de communication entre les rôles. Plus simplement, on pourrait dire que le diagramme nous a permis d'exhiber neuf communications entre rôles. On peut s'étonner de ce que tous les objectifs de notre SMA soient exclusivement pris en charge par le r_SI, ceci est simplement dû au fait que nous avons omis d'explicitier les objectifs de notification aux utilisateurs (pris en charge par r_TUT, r_COOR) et n'avons pas pris en compte ici les agents Constructeur de la base de données et les agents des apprenants. Ceci n'affecte en rien la compréhension de notre système. Comme nous l'avons dit plus haut, les rôles exhibés dans cette étape proviennent des différents diagrammes de séquence obtenus lors de l'étape précédente. Il en est de même pour les communications entre tâches. L'assignation des objectifs aux rôles permet de définir les tâches associées aux différents rôles. Il convient encore une fois de

préciser que l'utilisation de l'environnement AgentTool pour appliquer la méthodologie MaSE contraint le concepteur d'assigner chacun des objectifs identifiés lors de la première étape à un rôle. De cette façon, on s'assure bien que tous les objectifs seront remplis par le système multi-agents.

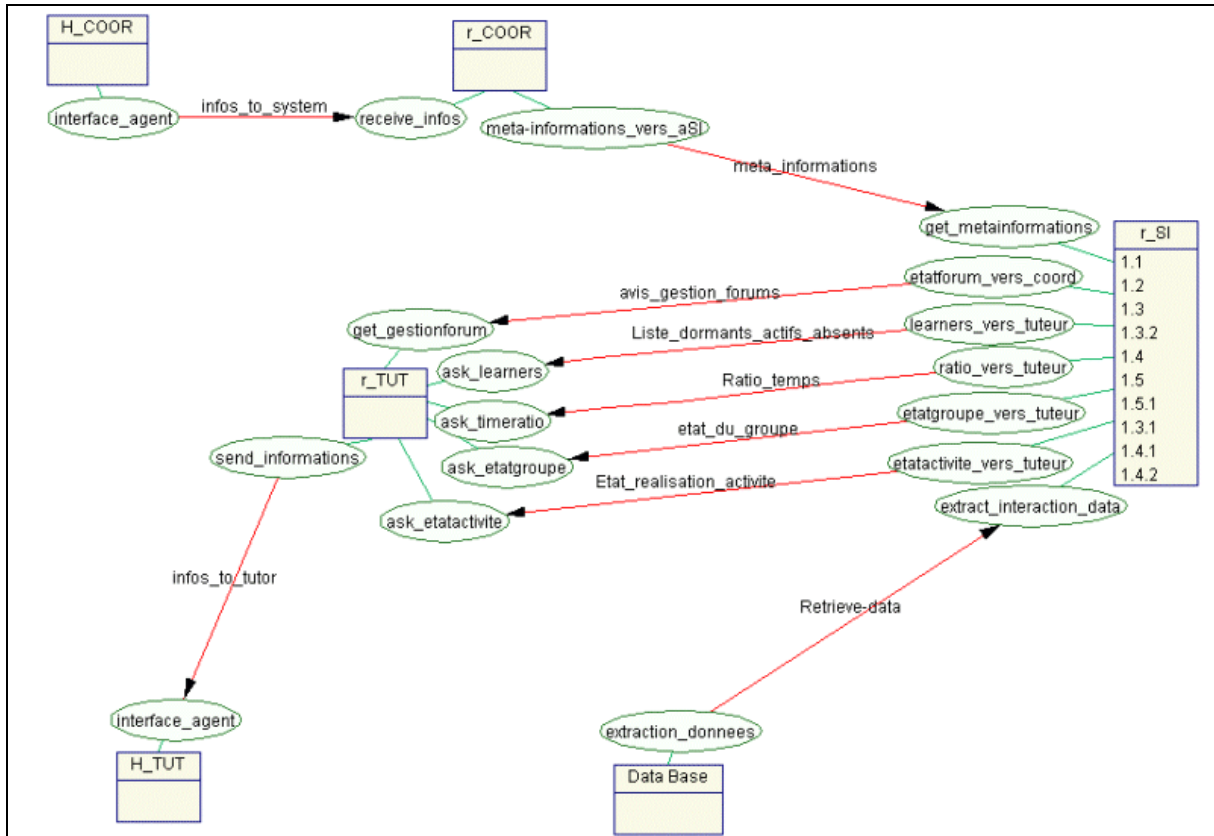


Figure 5.6 : Le diagramme de rôles de SIGFAD

La description des tâches concurrentes

La description des tâches concurrentes vient compléter la définition du diagramme des rôles et l'ensemble forme la troisième étape de la méthodologie MaSE. Ici l'on décrit chacune des tâches concurrentes que l'on voit apparaître dans les ellipses du diagramme de rôles. Le but est de spécifier les différents états et les transitions des agents dans la réalisation d'une tâche. Ceci se fait à travers des automates à états finis et n'est valable naturellement que pour les entités logicielles (rôles qui deviendront les agents lors de la phase de conception). La conception de SIGFAD offre un fonctionnement assez simple et répétitif des différents rôles. Il s'agit très souvent de l'agent de l'utilisateur (tuteur, coordonnateur, apprenant) qui envoie une requête à l'agent superviseur des interactions. Ce dernier, s'il n'est pas très occupé, reformule la requête au format SQL et extrait l'information recherchée de la base de données et l'envoie au requérant. Nous illustrons ce fonctionnement sur les figures 5.7 et 5.8.

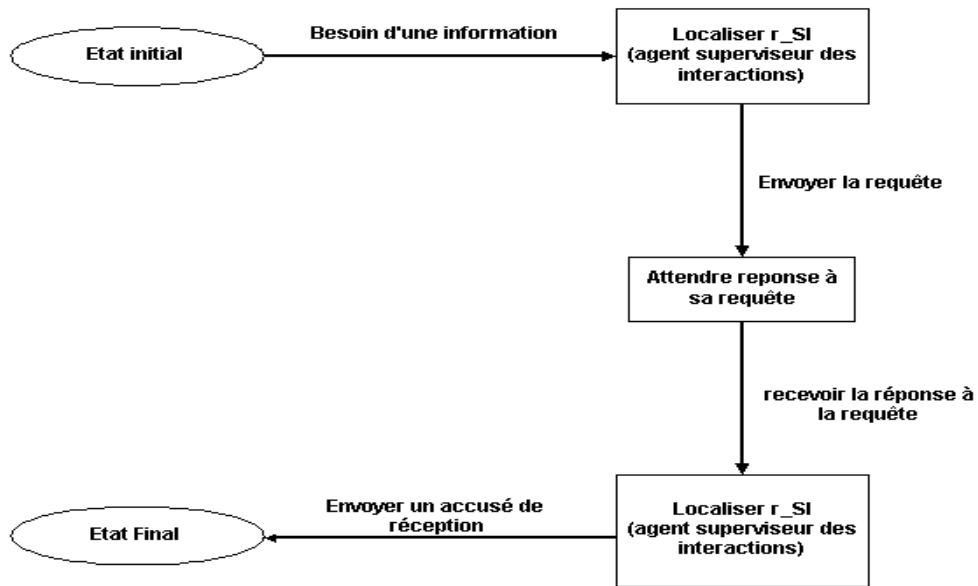


Figure 5.7 : Etats et transitions du rôle émetteur (agent tuteur, coordonnateur, apprenant) d'une demande

Dans la suite, nous présentons la description des tâches concurrentes pour la fourniture de la liste des apprenants présents, dormants et absents. Elle est illustrée par deux automates à états finis représentées sur les figure 5.9 et 5.10 qui sont des captures d'écran de AgentTool. Ces automates sont exactement calqués sur les modèles génériques présentés sur les figures 5.7 et 5.8. Ils disposent simplement d'informations supplémentaires données sous forme de paramètres. Les automates à états finis décrivant les autres tâches concurrentes sont présentés en annexe (figures B.7 à B.16).

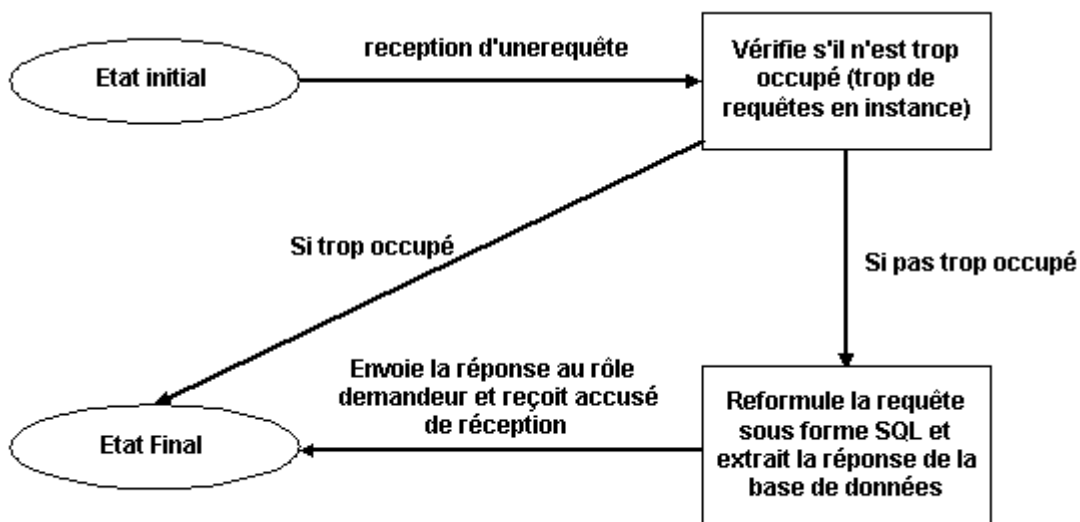


Figure 5.8 : Etats et transitions de r_SI (agent superviseur des interactions)

La fourniture de la liste des apprenants présents, 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*. L'automate à états finis représentant les états et les transitions de r_SI est présenté sur la figure 5.9, celui de r_TUT est présenté sur la figure 5.10.

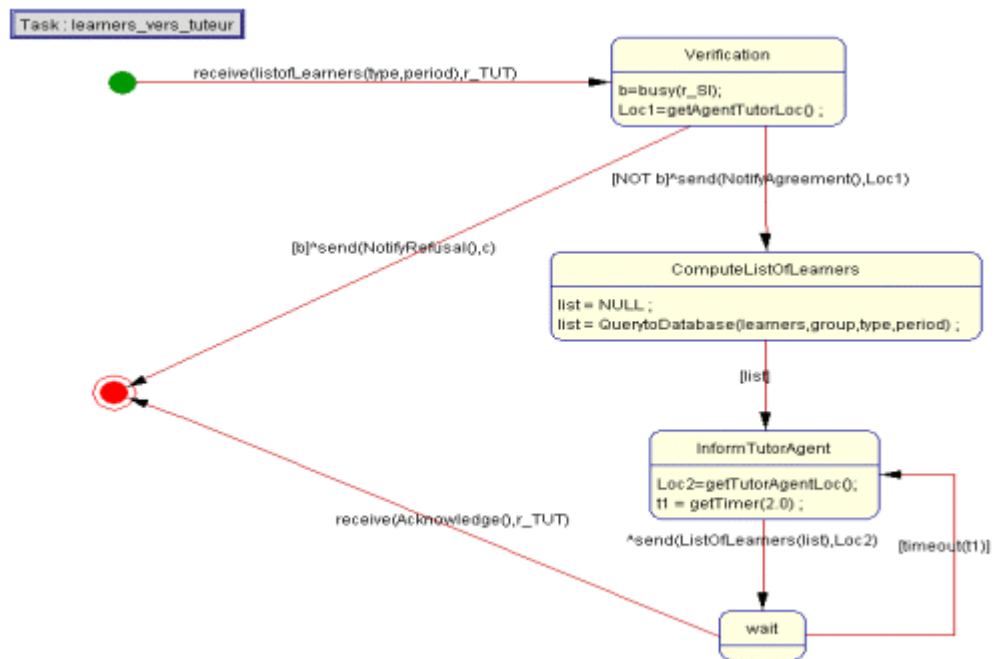


Figure 5.9 : Etats et transitions de r_SI dans la fourniture de la liste des apprenants à r_TUT

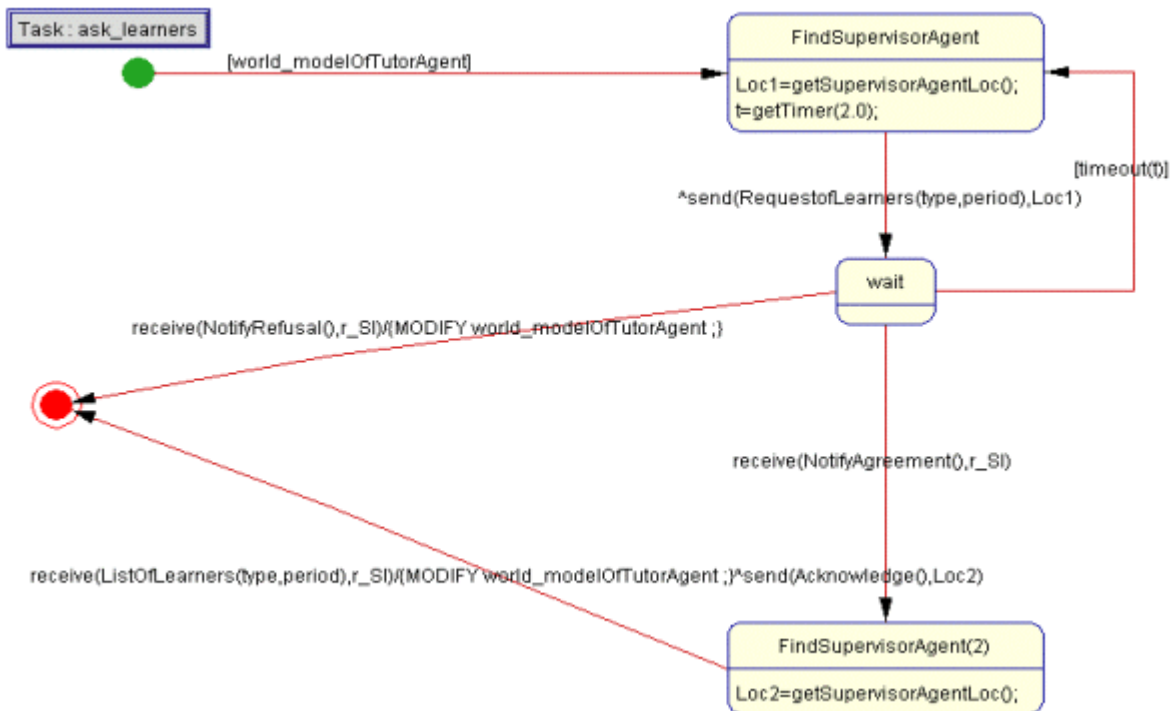


Figure 5.10 : Etats et transitions de r_TUT dans la demande de la liste d'apprenants à r_SI

Nous présentons en annexes la description des autres tâches concurrentes à savoir :

- La fourniture des meta-informations concernant les activités et les forums ;
- La fourniture des ratios de temps du groupe sur la semaine ;
- La fourniture de l'état du groupe ;
- La fourniture de l'état de réalisation d'une activité ;
- La fourniture de l'avis de gestion d'un forum ;
- L'interface agent – tuteur ;
- L'interface agent – coordonnateur.

Le concept de « busy state »

En plus des concepts de *move activity* (pour indiquer qu'un agent a changé de localisation 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 répondre 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* (figure

5.9). Des situations où r_SI est très sollicité peuvent arriver, dans l'état *Verification*, r_SI évalue l'expression :

$$b = \text{busy}(r_SI) ;$$

Si b contient la valeur TRUE, alors r_SI notifie à r_TUT son refus d'exécuter sa requête.

Le concept de *busy state* permet d'illustrer, par une implémentation concrète, une caractéristique des agents : la capacité de refuser d'exécuter une requête.

5.1.1.4 - Le diagramme des classes d'agents de SIGFAD

Durant cette étape, première étape de la phase de conception, les classes d'agents 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 et assigne chaque rôle à au moins une classe d'agents. 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'agents. 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. La méthodologie recommande de noter dans chaque rectangle représentant une classe d'agents les rôles qu'elle remplit.

Dans SIGFAD, nous avons neuf classes d'agents. 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). Il 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. a_BDD (qui remplit le rôle r_BDD) est chargé d'extraire les données d'interaction et de les rendre accessibles au sein de SIGFAD. a_BDD est l'interface entre SIGFAD et la LMS. *CoordinatorInterface*, *TutorInterface*, *LearnerInterface* et *DatabaseInterface* encapsulent les interfaces du système

avec respectivement : le coordonnateur, les tuteurs, les apprenants et la base de données. Le diagramme de classes d'agents de SIGFAD est représenté sur la figure 5.11.

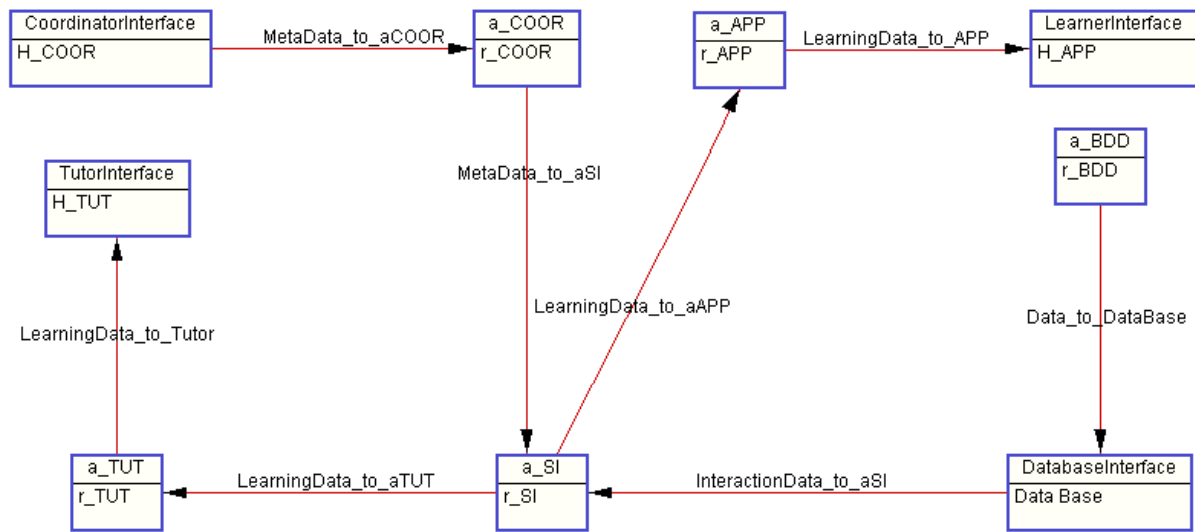


Figure 5.11 : Le diagramme de classes d'agent de SIGFAD

5.1.1.5 - 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 assigné 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'agents tel que demandé par la méthodologie.

5.1.1.6 - La conception du système

La dernière étape de MaSE consiste à instancier les classes d'agents 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. La figure 5.12 montre le diagramme de déploiement de SIGFAD. Nous avons au total 5 classes d'agents. Si p est le nombre de groupes et si on a au total n apprenants, nous aurons p agents de tuteurs et n agents d'apprenants. Le nombre d'agents total de SIGFAD est donc de $n+p+3$. Le nombre 3 correspond à l'agent du coordonnateur, l'agent superviseur des interactions et l'agent constructeur de la base de données.

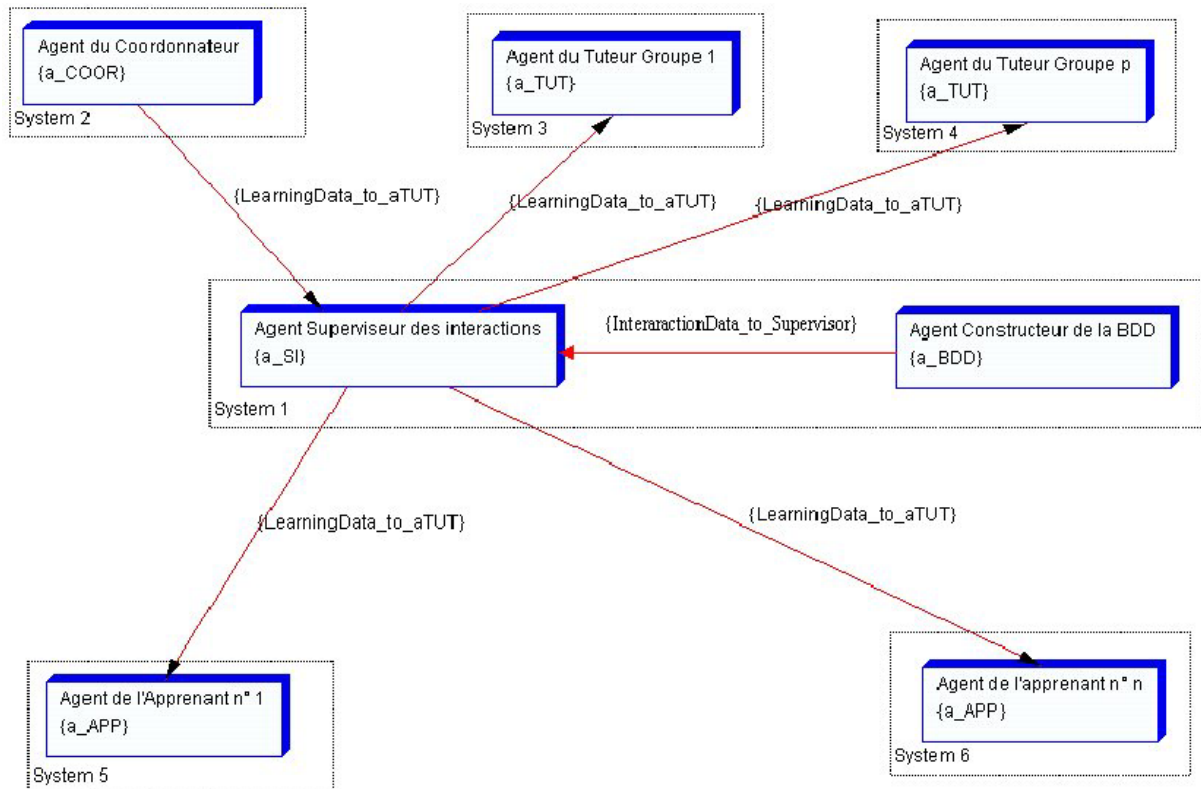


Figure 5.12 : Le diagramme de déploiement de SIGFAD

Les agents Superviseur des interactions et Constructeur de la base de données s'exécutent sur le serveur (system1 sur la figure). Les autres agents s'exécutent sur les postes clients du coordonnateur (system 2 sur la figure), les postes clients des tuteurs (system 3 system 4 sur la figure), les postes clients des apprenants (system 5 et system 6 sur la figure).

5.1.2 - Implémentation de SIGFAD

Nous avons eu recours à l'architecture JAM pour développer notre SMA. Ainsi que nous l'avons indiqué au chapitre 4, le modèle JAM offre un squelette qu'il appartient au programmeur de compléter par des procédures écrites en Java. La base de données contenant les données des interactions survenues durant le déroulement de SimuLigne est sous MySQL. Le développement de SIGFAD nous a permis d'identifier plusieurs points importants pour la programmation d'un tel système :

- le développement des agents suivant le modèle JAM ;
- l'interface agent - base de données (principalement donc entre l'agent superviseur des interactions et la base de données) ;
- l'interface agent – utilisateur ;
- la communication entre les agents.

Nous allons examiner chacun de ces points dans les paragraphes qui suivent et faire ressortir les principales difficultés rencontrées dans la programmation de SIGFAD.

5.1.2.1 - Le développement des agents suivant le modèle JAM

Cette première étape, cruciale on s'en doute, est celle où pour la première fois on touche concrètement à la notion d'agent. C'est ici en effet que l'on implémente cette entité en espérant voir en œuvre toutes les caractéristiques et propriétés des agents. Pour notre part, nous avons eu recours au modèle JAM pour définir l'architecture interne de nos agents. Nous avons utilisé la version *0.65+0.76i* de novembre 2001 (JAM [http](http://)).

Un agent JAM possède cinq parties : un interpréteur, une bibliothèque de plans, une structure intentionnelle, un modèle du monde et un observateur. L'interpréteur JAM et la structure intentionnelle ont déjà été écrites sous forme d'une librairie Java ; bien qu'il soit possible de définir soi-même d'autres interpréteurs, on se contentera en général de l'interpréteur existant. Le programmeur JAM devra donc simplement définir, dans des fichiers texte, le modèle du monde de chaque agent (modèle initial puisque ce dernier peut évoluer au cours de l'existence de l'agent), les objectifs de l'agent, les plans associés et éventuellement l'observateur. L'agent sera exécuté en chargeant à partir de la ligne de commandes DOS (notre environnement de

développement est l'environnement DOS/Windows) tous les fichiers texte définissant cet agent.

```
GOALS:
  ACHIEVE group_state;
FACTS:
  FACT initialized "False";
OBSERVER:
{
  RETRIEVE initialized $VALUE;
  WHEN : TEST (== $VALUE "False") {
    UPDATE (initialized) (initialized "True");
  };
}
*/
Plan: {
NAME: "Top-level plan"
GOAL: ACHIEVE group_state;
CONTEXT:
BODY:
  ASSIGN $port_server 8080;
  ASSIGN $host_server "localhost";
  ASSIGN $port_client 8090;
  ASSIGN $host_client "localhost";
  ASSIGN $Group "Aquitania";
  ASSIGN $Date1 "2001-05-10";
  ASSIGN $Date2 "2001-05-31";
  ASSIGN $type1 "Etat du groupe";
  EXECUTE print "Envoi de la requete au Supervisor.\n";
  EXECUTE com.irs.jam.primitives.SendQuerySupervisor.execute $port_server $host_server $host_client $Group $Date1 $Date2
$type1;
  EXECUTE com.irs.jam.primitives.ReceiveListSupervisor.execute $port_client $liste $type;
  EXECUTE print $type " " $Group " entre le " $Date1 " et le " $Date2 " : ";
  EXECUTE print $liste "\n";
}
```

Figure 5.13 : L'agent du tuteur du groupe Aquitania : tutor_gpe.jam

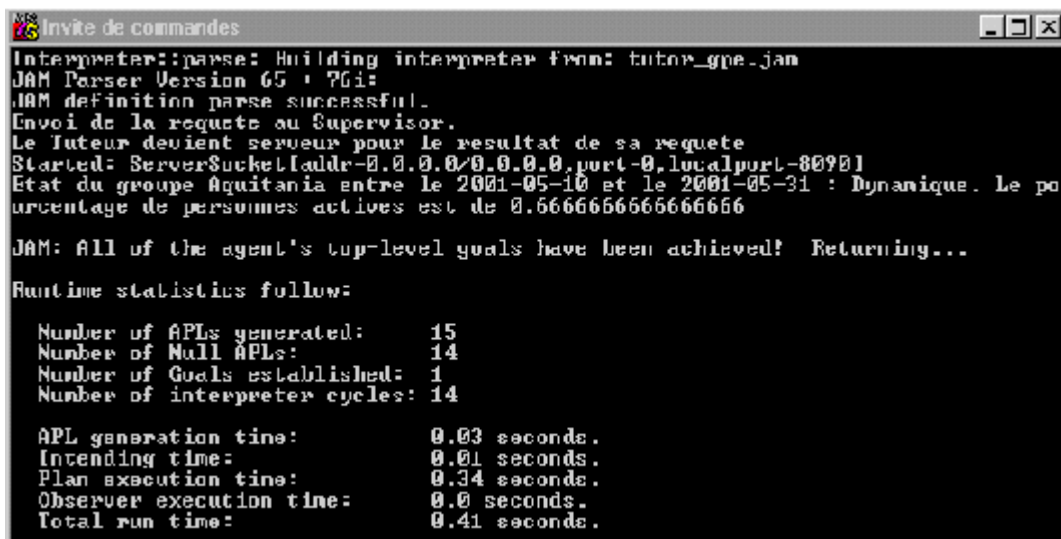
La figure 5.13 présente un agent tuteur comprenant un seul objectif, un seul fait dans son modèle du monde et un seul plan. Cet agent demande à l'agent superviseur des interactions l'état de son groupe (Aquitania) entre le 10 mai et le 31 mai 2001. Il précise la machine hôte sur laquelle est hébergée l'agent superviseur des interactions ainsi que le numéro de port. La ligne de commande utilisée pour exécuter cet agent est la suivante :

```
java com.irs.jam.JAM tutor_aquitania.jam
```

Le résultat de l'exécution de cet agent est montré dans la capture d'écran présentée à la figure 5.14. Cette exécution nous indique non seulement le résultat de la requête envoyée par l'agent du tuteur à l'agent superviseur des interactions, mais aussi des statistiques d'exécution, à savoir, le nombre de bibliothèques de plans générées, le nombre d'objectifs et les durées d'exécution entre autres. Après exécution, tutor_aquitania reçoit de l'agent superviseur des interactions, supervisor_v13nov.jam l'information selon laquelle le groupe Aquitania a eu un pourcentage d'apprenants actifs de 66 % et qu'il aura ainsi été dynamique entre le 10 mai et le 31 mai 2001. Naturellement, la bonne exécution de l'agent du tuteur est subordonnée au fait

que l'agent superviseur des interactions tourne effectivement sur la machine et le numéro de port annoncés. De même, il convient de s'assurer que la machine et le numéro de port sur lesquels l'agent du tuteur s'exécute sont bien spécifiés au sein de l'agent superviseur des interactions.

L'agent superviseur des interactions `supervisor_v13nov.jam` est écrit sur le même modèle. Il ne comporte pas d'observateur et contient un seul objectif, exécuter une requête venant d'un agent tuteur et un plan associé. Son modèle du monde est plus élaboré car il contient, au stade actuel de nos développements, la liste de membres des groupes de base, les méta-informations sur les activités et les tâches (jalons) de chaque activité. L'agent superviseur des interactions tourne en permanence sur sa machine hôte qui constitue de fait notre serveur. Nous présentons en annexe le code source de l'agent superviseur des interactions (Annexe D.1).



```

Interpreter::parse: Building interpreter from: tutor_gpe.jam
JAM Parser Version 65 + 76i:
JAM definition parse successful.
Envoi de la requete au Supervisor.
Le tuteur devient serveur pour le resultat de sa requete
Started: ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=8080]
Etat du groupe Aquitania entre le 2001-05-10 et le 2001-05-31 : Dynamique. Le po
urcentage de personnes actives est de 0.6666666666666666

JAM: All of the agent's top-level goals have been achieved! Returning...

Runtime statistics follow:

Number of APLs generated:      15
Number of Null APLs:          14
Number of Goals established:    1
Number of interpreter cycles:  14

APL generation time:           0.03 seconds.
Intending time:                0.01 seconds.
Plan execution time:           0.34 seconds.
Observer execution time:       0.0 seconds.
Total run time:                0.41 seconds.

```

Figure 5.14 : Capture d'écran montrant l'exécution de l'agent `tutor_gpe.jam`

La figure 5.15 présente le lancement de l'exécution de `supervisor_v13nov.jam` sur le serveur et le résultat obtenu. L'agent `supervisor_v13nov.jam` attend les requêtes de la part des agents des tuteurs, il utilise le port 8080.



```

C:\JAM\com\irs\jam\sigfad>java com.irs.jam.JAM supervisor_v13nov.jam
Interpreter::parse: Building interpreter from: supervisor_v13nov.jam
JAM Parser Version 65 + 76i:
JAM definition parse successful.
Waiting as Server for tutors' connections.
Creation du Serversocket
Adresse du ServeSocket est : ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=8080]
Started: ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=8080]

```

Figure 5.15 : Lancement de `supervisor_v13nov.jam` et résultat de la commande

L'agent superviseur des interactions est chargé principalement des transactions avec la base de données des interactions. Il se contente au stade actuel de traiter les requêtes venant des

différents agents tuteurs. Il est possible de faire évoluer sa représentation du monde en modifiant ou en ajoutant de nouveaux faits dans son modèle du monde. Signalons que JAM permet d'implémenter facilement cette possibilité grâce aux points de reprise. L'on peut ainsi exécuter un agent et sauvegarder son état dans un fichier grâce à la primitive *CheckpointAgentToFile* et relancer l'exécution de cet agent plus tard avec la primitive *RestoreAgentCheckpoint*. C'est une possibilité très intéressante dans la mise en œuvre de SIGFAD. Un tuteur sauvegardera toujours ainsi l'état de son agent dans un fichier, il relancera l'exécution de l'agent à sa prochaine connexion à partir du fichier de sauvegarde ; cela permettra de tirer profit des caractéristiques d'apprentissage des agents quand elles seront implémentées.

5.1.2.2 - L'interface agent – BDD

L'interfaçage de nos agents (principalement, l'agent *supervisor_v13nov.jam*) avec la base de données des interactions constituait un autre point important du développement de SIGFAD. Signalons que les données d'interaction (certaines) ont été au préalable enregistrées dans une base de données MySQL. Ceci s'est fait pratiquement pendant le déroulement de SimuLigne même lorsque l'architecture de SIGFAD n'était pas encore définie. Les données d'interaction provenaient du fichier des traces du serveur HTTP (il s'agissait de Apache dans notre cas) et de la base de données interne de PIFAD. Des applications spécifiques ont été développées pour accéder à ces sources, extraire les informations et les ranger dans la base de données. SIGFAD dispose d'un agent qui sera chargé de ce travail dorénavant : c'est l'agent constructeur de la base de données. En l'absence de l'agent constructeur de la base de données, l'agent superviseur des interactions est le seul qui accède actuellement à la base de données des interactions. C'est une tâche essentielle, car toutes les données nécessaires au traitement des requêtes issues des autres agents se trouvent dans la base de données. Il était donc essentiel de pouvoir accéder correctement à la base de données. Le problème était le suivant : comment accéder à une base de données MySQL à partir d'applications écrites en Java ?

Une solution à ce problème consiste à recourir au JDBC (*Java DataBase Connectivity*). Il faut installer une interface entre Java et la base de données, il existe quatre types d'interface pour le faire. Nous avons choisi d'implémenter une interface de type « *Native-protocol pure Java driver* » qui assure une connexion directe entre Java et la base de données à travers le réseau.

Il existe différents pilotes pour assurer l'accès à une base de données MySQL ; le travail consiste à installer un tel pilote. Pour notre part, nous avons choisi le pilote JDBC *org.git.mm.mysql*, et les étapes concrètes de l'interfaçage entre nos programmes et la base de données sont les suivants :

- Chargement du pilote JDBC

```
Class.forName("org.git.mm.mysql.Driver");
```

- Définition d'une URL de connexion :

```
String url="jdbc:mysql://localhost/mysql/data";
```

- Etablissement de la connexion

```
Connection con = DriverManager.getConnection(url,"","");
```

- Création d'une instruction

```
Statement statement = con.createStatement();
```

- Exécution d'une requête (exemple)

```
String query = "SELECT DISTINCT login, groupe FROM logs WHERE groupe= '"+  
monGroupe+ "' && action_WebCT='connexion' && date_action >= '" + maDate1+"' &&  
date_action <= '" + maDate2 +"' "+ "ORDER BY login ;";
```

- Récupération de l'exécution de la requête

```
ResultSet resultset = statement.executeQuery (query);
```

Les agents JAM peuvent contenir des procédures en Java, qu'ils exécutent comme des primitives JAM (il s'agit d'une classe JAM prédéfinie). C'est à l'intérieur de telles procédures que l'on accède à la base de données des interactions. C'est ainsi, par exemple, que l'agent *supervisor_v13nov.jam*, pour fournir la liste des personnes absentes dans un groupe durant une période donnée va exécuter l'instruction suivante :

```
EXECUTE com.irs.jam.primitives.AbsentsInGroup.execute $Group $Date1 $Date2 $resultat;
```

La procédure *AbsentsInGroup* est une primitive que nous avons écrite ; elle possède quatre arguments : le nom du groupe contenu dans la variable *\$Group*, les dates précisant la période voulue dans les variables *\$Date1* et *\$Date2* ; le résultat de la requête contenu dans la variable *\$resultat*. La primitive *AbsentsInGroup* permet alors à l'agent de se connecter à la base de données pour extraire l'information recherchée par une simple requête SQL.

5.1.2.3 – Les interfaces SIGFAD – utilisateur

L'interface agent-utilisateur est un problème important dans l'implémentation de SIGFAD. Ceci est principalement dû au fait que les agents JAM sont initialement conçus pour être exécutés en mode ligne de commande (DOS dans notre cas). Ceci est très peu pratique dans un SMA comme le nôtre où les agents ont un rôle d'assistant personnel des utilisateurs, ce qui

suppose l'existence de multiples interactions entre l'utilisateur et son agent. Le tuteur qui utilise SIGFAD est amené à formuler régulièrement des requêtes à son agent, qui les transmet à l'agent superviseur des interactions. Il n'est pas possible de passer des paramètres en argument des agents JAM ; ceci signifie que l'utilisateur devrait modifier le code de l'agent à chaque fois qu'il formule une requête. Naturellement, ceci n'est pas du tout envisageable.

Il est immédiat que l'utilisation de SIGFAD exigeait de définir des interfaces graphiques à partir desquelles l'utilisateur peut formuler des requêtes et obtenir des réponses. La première possibilité envisagée était naturellement de définir une interface de type formulaire HTML. Cette possibilité est malheureusement irréalisable car il n'est pas possible de lancer l'exécution d'un agent à partir d'un tel formulaire. L'impossibilité de lancer une commande DOS à partir d'une fonction écrite en Javascript est venue supprimer définitivement cette possibilité. La seule possibilité qui nous restait était de recourir à une interface utilisateur écrite en Java.

Il faut tout de suite dire qu'autant la mise en place d'interface est immédiate en HTML grâce à l'utilisation des formulaires, autant celle-ci est ardue et quelque peu laborieuse en Java. Peut-être nous arguera-t-on que c'est le prix à payer pour avoir des interfaces robustes et sophistiquées ! L'existence de plusieurs bibliothèques comme Swing, AWT et l'évolution des versions du langage Java (avec notamment des méthodes qui tombent en désuétude d'une version à l'autre) ne sont pas pour simplifier le problème. Nous avons eu recours à la bibliothèque Swing pour la création de nos interfaces agent-utilisateur. Nous avons suivi les étapes classiques de la création d'une interface graphique à savoir : créer un cadre, des composants dans ce cadre, une mise en page pour ces composants, des méthodes en réponse aux actions des utilisateurs.

L'interface tuteur – SIGFAD

Compte tenu des fonctionnalités de SIGFAD, nous avons identifié trois groupes de requêtes pour l'agent du tuteur. Le premier groupe comprend les requêtes portant sur le groupe, le second concerne l'apprenant et le dernier concerne l'état de réalisation d'une activité. Nous avons implémenté, au niveau de l'interface agent-tuteur, un bouton pour chaque type de requête. Le tuteur dispose d'une zone de texte dans laquelle il voit le résultat de sa requête s'afficher après exécution par l'agent superviseur des interactions. La figure 5.16 présente

l'interface entre le tuteur et son agent, c'est de fait la seule interface entre le tuteur et SIGFAD. C'est cet écran qui apparaît lorsque le tuteur lance l'exécution de son agent. Il lui est demandé de choisir la nature de la requête qu'il souhaite formuler. S'il s'agit d'une requête sur le groupe, il clique sur le bouton radio correspondant et les zones de saisie appropriées deviennent actives dans la figure 5.17. Il en est de même si la requête porte plutôt sur l'apprenant et l'activité.

Figure 5.16 : L'écran d'entrée de l'interface agent – tuteur

Figure 5.17 : Ecran disponible si le tuteur veut des informations sur le groupe

L'interface coordonnateur – SIGFAD

L'agent coordonnateur est celui qui fournit à SIGFAD toutes les informations de niveau méta sur les activités, la composition des groupes, les forums. Le coordonnateur pourra, à travers l'interface graphique avec son agent, consulter les méta-informations enregistrées par l'agent Superviseur des interactions. Il pourra aussi modifier ces informations, soit en ajoutant de nouvelles informations, soit en supprimant des informations qui existent déjà. De façon exhaustive, il pourra :

- consulter la liste des activités prévues durant une période spécifiée : cette requête lui permettra d'obtenir la liste des jalons des activités, leurs dates limites ainsi que les outils devant être utilisés ;
- consulter les dates limites d'un jalon bien précis ;
- consulter la liste des membres d'un groupe ;
- ajouter une activité dans la formation : il précisera la liste des jalons faisant partie de l'activité ;
- ajouter un jalon, en fournissant le nom du jalon, les personnes devant l'accomplir, l'outil à utiliser, les dates limite et en indiquant si le tuteur doit être alerté lorsque le jalon n'est pas validé à la date révolue ;
- ajouter un membre à un groupe, en précisant son statut ;
- supprimer une activité entièrement ou en supprimer seulement des jalons ;
- supprimer un jalon ;
- supprimer un membre dans un groupe.

La figure 5.18 représente la vue offerte au coordonnateur lorsqu'il exécute son agent. Il peut alors cliquer sur l'un des boutons pour indiquer le type d'opération qu'il désire effectuer sur les méta-informations. Les champs concernés sont alors activés. Le lecteur intéressé pourra consulter en annexes, les vues offertes au coordonnateur lorsqu'il clique sur l'un des boutons.

L'interface apprenant – SIGFAD

L'interface de l'apprenant avec son agent lui permettra de demander à SIGFAD de lui fournir son niveau de participation dans la session de formation. Il pourra ainsi consulter son niveau de productivité et sa sociabilité. La sociabilité et la productivité recouvrent les mêmes notions que celles définies pour le tuteur. L'apprenant peut aussi obtenir la liste des activités prévues durant une période et les dates limites d'une activité donnée. La vue d'entrée de l'interface

apprenant – SIGFAD est représentée sur la figure 5.19. L'apprenant, selon l'objet de sa requête, aura une vue où seuls les champs nécessaires sont accessibles à la saisie.

Figure 5.18 : La vue d'entrée offerte au Coordonnateur par son agent

Figure 5.19 : La vue d'entrée de l'interface apprenant-SIGFAD

5.1.2.4 – La communication entre les agents

Ainsi que nous l'avons dit plus haut il existe plusieurs possibilités pour assurer la communication entre les agents. Il est recommandé de recourir à des moyens de communication qui ne nécessitent pas de programmer à un niveau très bas. C'est dans ce sens que nous avons envisagé d'utiliser une infrastructure de communication comme JATLite. Toutefois, en raison des caractéristiques de l'architecture JAM et pour une meilleure maîtrise du fonctionnement de notre SMA, nous avons choisi de faire communiquer nos agents par les *sockets* écrits en Java. L'essentiel des communications entre agents se déroule entre l'agent superviseur des interactions et les agents des utilisateurs. Nous allons expliquer ce qui se passe par exemple entre l'agent Superviseur des interactions et l'agent d'un tuteur. La communication se passe en deux temps :

- L'agent superviseur des interactions est lancé et se place comme serveur pour attendre les requêtes venant des agents des utilisateurs.
- Une fois la requête envoyée par l'agent de l'utilisateur, celui-ci établit une nouvelle communication pour laquelle il devient le serveur. L'agent superviseur des interactions exécute la requête en recherchant les informations dans la base de données et envoie le résultat à l'agent demandeur.

Les deux lignes suivantes du code de *supervisor_v13nov.jam* montrent que ce dernier s'exécute comme un serveur attendant les demandes des agents des tuteurs :

```
EXECUTE print "Waiting as Server for tutors' connections.\n";
EXECUTE      com.irs.jam.primitives.ReceiveQueryTutor.execute      $port_server
$host_client $Group $Date1 $Date2 $type;
```

La deuxième indique que nous avons défini une primitive *ReceiveQueryTutor* qui permet à l'agent *supervisor_v13nov.jam* de recevoir une requête venant d'un agent tuteur. Cette requête comporte le numéro du port et le nom de la machine de l'agent du tuteur, le nom du groupe (de l'activité ou de la personne) sur lequel porte la requête, les dates de début et de fin de la période et l'objet de la requête. C'est au sein de la primitive *ReceiveQueryTutor* qu'un *socket* est défini et un échange d'informations *half-duplex* a lieu entre les deux agents.

La ligne suivante du code de *supervisor_v13nov.jam* nous indique qu'il se connecte à l'agent tuteur qui lui a adressé la requête pour lui communiquer le résultat :

```
EXECUTE      com.irs.jam.primitives.SendResultToTutor.execute      $port_groupe
$host_client $resultat $type;
```

SendResultToTutor est une primitive JAM qui permet de communiquer le résultat (*\$resultat*) à l'agent qui tourne sur le port *\$port_groupe*, la machine *\$host_client* en lui rappelant la nature de sa requête (*\$type*). Précisons que l'agent superviseur des interactions tourne en permanence sur le serveur grâce à une boucle infinie définie au moyen de la structure de contrôle JAM : DO...WHILE. Du point de vue de la communication, l'algorithme exécuté par *supervisor_v13nov.jam* est résumé sur la figure 5.20 suivante.

```

DO
{
  Etablir une connexion en tant que serveur ;
  Recevoir une requête d'un agent tuteur ;
  Couper la connexion ;
  Exécuter la requête en se connectant à la base de données des interactions ;
  Se connecter comme client à la connexion établie par l'agent tuteur ;
  Lui fournir la réponse à la requête ;
  Couper la connexion ;
} WHILE : TEST (= = 1 1);

```

Figure 5.20 : L'algorithme exécuté par *supervisor_v13nov.jam*

L'agent du tuteur exécute un algorithme symétrique qui est expliqué sur la figure 5.21 qui suit. La différence est dans l'absence de boucle infinie.

```

{
  Se connecter comme client à la connexion établie par supervisor_v13nov.jam
  Envoyer sa requête ;
  Couper la connexion ;
  Etablir une connexion en tant que serveur ;
  Recevoir le résultat de la requête ;
  Couper la connexion ;
}

```

Figure 5.21 : L'algorithme exécuté par un agent du tuteur.

5.1.2.5 – Les difficultés rencontrées dans l'implémentation de SIGFAD

Il nous semble que c'est en recensant les difficultés rencontrées dans l'implémentation de SIGFAD que les chercheurs orienteront mieux leurs actions de mise au point d'outils de développement propres à la programmation orientée agent. Il convient de dire d'emblée que les difficultés dans la programmation des systèmes à base d'agents comme SIGFAD viennent précisément du fait qu'il n'existe pas encore d'outils spécifiques dédiés à cet usage.

Nous avons pu expérimenter les inconvénients qui existent lorsqu'on utilise des outils non dédiés. Nous avons indiqué que la méthodologie MaSE avait été mise au point pour développer des systèmes multi-agents ; dans sa mise en application, on voit bien que les interactions et communications entre les agents ont été prises en compte à leur juste titre par ses auteurs. Malheureusement, elle pêche par une absence d'architecture interne des différents agents ; elle considère implicitement les agents comme des boîtes noires (Padgham et Winikoff, 2002). Face à cela, nous avons eu recours à une architecture interne basée sur le modèle BDI, c'est l'architecture JAM. Malgré toutes les qualités du modèle JAM, on s'aperçoit vite que les agents JAM n'ont pas été conçus dans l'objectif premier de communiquer entre eux. Les exemples d'agent fournis par les auteurs de l'architecture JAM sont révélateurs de ceci, aucun ne met en oeuvre des agents communiquant entre eux ! Il apparaît donc difficile d'utiliser une architecture qui n'a pas été pensée au départ pour permettre la communication entre des agents dans un système multi-agents. Rappelons que les interactions entre agents caractérisent au mieux les systèmes multi-agents.

On rencontre le même type de difficulté lorsqu'on essaie d'utiliser des infrastructures de communication comme JATLite. Là aussi, on se rend rapidement compte que la philosophie inhérente à leur mise au point n'est pas toujours en adéquation avec nos besoins. C'est pourquoi nous avons dû nous résoudre à programmer des *sockets* Java pour assurer la communication entre nos agents dans SIGFAD. Il nous semble néanmoins envisageable d'adapter JATLite afin de pouvoir l'utiliser dans SIGFAD dans un avenir très proche.

5.2 – L'application de SIGFAD aux données de SimuLigne

Dans cette section, nous voulons présenter les résultats que l'on peut obtenir si l'on applique notre système multi-agents aux données d'interaction de SimuLigne. Nous nous livrons à cette étude a posteriori sur une formation qui s'est déjà déroulée. Rappelons cependant que le but de SIGFAD est d'être couplé à une LMS pour aider à mettre en oeuvre l'apprentissage collaboratif distribué.

La partie actuellement implémentée permet d'accéder à la base de données des interactions et de fournir aux utilisateurs des informations utiles pour le bon déroulement de la session de formation, l'existence et la durabilité des groupes en tant que tels.

Dans la présente partie, nous nous focaliserons sur l'utilisation que le tuteur peut faire de notre prototype. Nous allons montrer sous forme de tableau les résultats de requêtes qu'il peut soumettre à SIGFAD. Nous présentons ici les résultats sur des requêtes de deux catégories : les quatre groupes de base (Aquitania, Gallia, Lugdunensis, Narbonensis), les apprenants (nous avons sélectionné trois apprenants). Les requêtes doivent toujours spécifier les dates de début et de fin sur lesquelles elles portent. Ceci sera de la responsabilité du tuteur qui, à tout moment et en fonction de ses besoins, indiquera lesdites dates. Nous avons indiqué que Lugdunensis, l'un des groupes de base a dû être fermé au bout de quatre semaines de formation (le 30 mai 2001) ; nous présentons dans les tableaux 5.1 et 5.2 des résultats de requêtes pour les groupes Aquitania et Lugdunensis. Nous présentons en annexe les mêmes tableaux pour les deux autres groupes. Les informations contenues dans ces tableaux sont celles effectivement affichées aux tuteurs par SIGFAD. La liste exhaustive des apprenants présents, absents et dormants est fournie. L'état du groupe est fourni avec les éléments permettant au système de le calculer, à savoir : le nombre d'apprenants actifs (ce sont les apprenants qui ont validé tous les jalons obligatoires) et leur liste exhaustive. En effet, nous pensons que le tuteur est le mieux placé pour se faire une opinion exacte de son groupe à la lumière des éléments d'information disponibles. Les contraintes de mise en forme nous ont amené à présenter ces résultats sur des tableaux différents. Nous avons choisi des périodes qui partent de l'étape de socialisation E0 à la fermeture du groupe Lugdunensis et avons essayé de choisir des plages de dates correspondant à des dates de début et de fin d'activités ; ces activités sont indiquées à chaque fois dans la première colonne. De même nous avons choisi des plages de date qui ne se recouvrent pas trop mais qui couvrent de bout en bout toute la période allant du 02 avril au 03 juin.

Période	Aquitania			Etat du groupe
	Présents	Absents	Dormants	
02 avr. – 22 avr. PREMJ1 PREMJ2	Linda Douglas George Jessica Anita Elsa Alan Jim Peter Orly	<i>Pas d'absents sur cette période</i>	<i>Pas de dormants sur cette période</i>	<i>Très Dynamique</i> 10 apprenants actifs Linda Douglas George Jessica Anita Elsa Alan Jim Peter Orly

29 avr. – 06 mai E0A1J1 E0A1J2 E0A2J1	Douglas Jessica Anita Elsa Alan Jim Peter Orly	Linda George	<i>Pas de dormants sur cette période</i>	<i>Dynamique</i> <u>6 apprenants actifs</u> Anita Elsa Orly Jim Peter
30 avril-13 mai E0A3 E0A4J1 E0A4J2 E0A5 INTER0	Linda Douglas Jessica Anita Elsa Alan Jim Peter Orly	George	<i>Pas de dormants sur cette période</i>	<i>Assez dynamique</i> <u>5 apprenants actifs</u> Douglas Anita Alan Jim Orly
14 mai - 20 mai I1P1	Douglas George Jessica Anita Elsa Peter Orly	Linda Jim Alan	<i>Pas de dormants sur cette période</i>	<i>Peu dynamique</i> <u>3 apprenants actifs</u> Douglas Anita Orly
22 mai - 03 juin I1P3 E1A3J2	Douglas Anita Elsa Orly	Linda George Jessica Alan Jim Peter	<i>Pas de dormants sur cette période</i>	<i>Assez dynamique</i> <u>4 apprenants actifs</u> Douglas Anita Elsa Orly

Tableau 5.1 : Quelques résultats fournies par l'agent tuteur du groupe Aquitania

Les requêtes sont présentées sur trois apprenants : un apprenant ayant très tôt abandonné la formation, un apprenant ayant abandonné vers le milieu de la formation et un apprenant étant allé jusqu'au bout de la formation. Aucun de ces apprenants n'a été choisi dans le groupe défaillant. Les plages de dates vont du 02 avril au 06 juillet 2001 (fin de la formation) avec les mêmes considérations que pour les requêtes portant sur les groupes. Les résultats des requêtes portant sur les apprenants sont présentés dans le tableau 5.3. SimuLigne nous a donné l'occasion de comprendre la difficulté de la tâche des tuteurs et du coordonnateur de la formation. C'est certainement la raison pour laquelle nous avons privilégié les fonctionnalités de SIGFAD utiles à cette catégorie d'utilisateurs. Ce qui est valable pour les tuteurs est extensible à d'autres catégories d'utilisateurs comme les apprenants.

Période	Lugdunensis			Etat du groupe
	Présents	Absents	Dormants	
02 avr. – 22 avr. PREMJ1 PREMJ2	Lindsay Terry Ben Kareen Barry David Eddy Michelle William	<i>Pas d'absents sur cette période</i>	<i>Pas de dormants sur cette période</i>	<i>Très Dynamique.</i> <u>9 apprenants actifs</u> Lindsay Terry Ben Kareen Barry David Eddy Michelle William

29 avr. – 06 mai E0A1J1 E0A1J2 E0A2J1	Lindsay Terry Kareen Barry Eddy Michelle William	Ben David	Terry	<i>Dynamique</i> <u>6 apprenants actifs</u> Lindsay Terry Barry Eddy Michelle William
30 avril-13 mai E0A3 E0A4J1 E0A4J2 E0A5 INTER0	Lindsay Terry Kareen Ben Barry David Eddy Michelle William	<i>Pas d'absents sur cette période</i>	Terry	<i>Peu dynamique</i> <u>1 apprenant actif</u> Ben
14 mai - 20 mai I1P1	Lindsay Ben Barry David Eddy Michelle William	Kareen Terry	Lindsay Barry	<i>Pas dynamique</i> <u>1 apprenant actif</u> David
22 mai - 03 juin I1P3 E1A3J2	Lindsay Ben Barry David Eddy Michelle William	Terry	<i>Pas de dormants sur cette période</i>	<i>Pas dynamique</i> <u>1 apprenant actif</u> Ben

Tableau 5.2 : Quelques résultats fournis par l'agent du tuteur de Lugdunensis

Période	Bill		Alan		John	
	Productivité	Sociabilité	Productivité	Sociabilité	Productivité	Sociabilité
02 avr. – 22 avr. PREMJ1 PREMJ2	<i>Actif</i> 2 jalons révolus (PREMJ1 PREMJ2) sur 2	Connexion : 10 Calendrier : 4 Consignes : 22 Courriels lus : 0 Courriels écrits : 0 Forums lus : 16 Forums postés : 1	<i>Actif.</i> 1 jalon (PREMJ1) sur 2	Connexion : 6 Calendrier : 1 Consignes : 6 Courriels lus : 0 Courriels écrits : 0 Forums lus : 1 Forums postés : 1	<i>Actif</i> 2 jalons révolus (PREMJ1 PREMJ2) sur 2	Connexion : 4 Calendrier : 1 Consignes : 1 Courriels lus : 0 Courriels écrits : 0 Forums lus : 11 Forums postés : 1
29 avr. – 06 mai E0A1J1 E0A1J2 E0A2J1	<i>Très Actif</i> 3 jalons révolus (E0A1J1 E0A1J2 E0A2J1)	Connexion : 6 Calendrier : 0 Consignes : 8 Courriels lus : 14 Courriels écrits : 3 Forums lus : 15 Forums postés : 1	<i>Peu actif</i> 2 jalons (E0A1J1 E0A1J2)	Connexion : 5 Calendrier : 2 Consignes : 8 Courriels lus : 0 Courriels écrits : 0 Forums lus : 36 Forums postés : 2	<i>Actif</i> 2 jalons (E0A1J2 E0A2J1), Dernier jalon effectué	Connexion : 4 Calendrier : 1 Consignes : 2 Courriels lus : 1 Courriels écrits : 0 Forums lus : 14 Forums postés : 1
30 avril-13 mai E0A3 E0A4J1 E0A4J2 E0A5 INTER0	<i>Actif</i> 4 jalons révolus (E0A3 E0A4J2 E0A5 INTER0). Dernier jalon effectué	Connexion : 26 Calendrier : 4 Consignes : 30 Courriels lus : 73 Courriels écrits : 16 Forums lus : 92 Forums postés : 5	<i>Dormant</i> <i>Aucun jalon effectué</i>	Connexion : 5 Calendrier : 2 Consignes : 8 Courriels lus : 0 Courriels écrits : 0 Forums lus : 36 Forums postés : 2	<i>Actif</i> 4 jalons révolus (E0A3 E0A4J1 E0A5 INTER0)	Connexion : 8 Calendrier : 1 Consignes : 9 Courriels lus : 1 Courriels écrits : 0 Forums lus : 77 Forums postés : 3
14 mai - 20 mai I1P1	<i>Actif</i> 1 jalon révolu (I1P1) et 5 jalons possibles (E1A2J2 E1A3J1 E1A4J1 E1A5J1 E1A5J2)	Connexion : 17 Calendrier : 3 Consignes : 12 Courriels lus : 7 Courriels écrits : 5 Forums lus : 74 Forums postés : 3	<i>Absent.</i> <i>Aucun jalon révolu.</i>	<i>Absent</i>	<i>Peu actif</i> 1 jalon possible (E1A2J2), mais aucun jalon révolu effectué.	Connexion : 4 Calendrier : 1 Consignes : 6 Courriels lus : 0 Courriels écrits : 0 Forums lus : 16 Forums postés : 1
22 mai - 03 juin I1P3 E1A3J2	<i>Actif</i> 1 jalon révolu (I1P3) et 5 jalons possibles (E2A1J1 E2A2J2 E2A3J1 E2A3J2 E2A5J1).	Connexion : 19 Calendrier : 8 Consignes : 22 Courriels lus : 11 Courriels écrits : 6 Forums lus : 133 Forums postés : 13	<i>Absent.</i> <i>Aucun jalon révolu.</i>	<i>Absent</i>	<i>Dormant</i> <i>Aucun jalon effectué</i>	Connexion : 3 Calendrier : 1 Consignes : 1 Courriels lus : 8 Courriels écrits : 2 Forums lus : 0 Forums postés : 0

01 juin – 06 juin E2A1J1	<i>Actif</i> 1 jalon révolu (E2A1J1) et 4 jalons possibles (E2A2J2 E2A3J1 E2A3J2 E2A5J1)	Connexion : 9 Calendrier : 2 Consignes : 24 Courriels lus : 19 Courriels écrits : 7 Forums lus : 27 Forums postés : 2	<i>Absent.</i> <i>Aucun jalon révolu.</i>	<i>Absent</i>	<i>Absent</i> <i>Aucun jalon révolu</i>	<i>Absent</i>
06 juin – 10 juin E2A1J2	<i>Peu actif</i> 3 jalons possibles (E2A2J2 E2A3J2 E2A5J1) mais aucun jalon révolu	Connexion : 11 Calendrier : 1 Consignes : 2 Courriels lus : 20 Courriels écrits : 7 Forums lus : 11 Forums postés : 4	<i>Absent.</i> <i>Aucun jalon révolu.</i>	<i>Absent</i>	<i>Absent</i> <i>Aucun jalon révolu</i>	<i>Absent</i>
11 juin – 17 juin E2A4J1 E2A4J2 I2P1	<i>Très actif</i> 3 jalons révolus (E2A4J1 E2A4J2 I2P1)	Connexion : 16 Calendrier : 0 Consignes : 2 Courriels lus : 19 Courriels écrits : 8 Forums lus : 44 Forums postés : 4	<i>Absent.</i> <i>Aucun jalon révolu.</i>	<i>Absent</i>	<i>Absent</i> <i>Aucun jalon révolu</i>	<i>Absent</i>
18 juin – 21 juin E2A4J3	<i>Actif</i> 1 jalon révolu (E2A4J3)	Connexion : 10 Calendrier : 0 Consignes : 0 Courriels lus : 17 Courriels écrits : 10 Forums lus : 10 Forums postés : 3	<i>Absent.</i> <i>Aucun jalon révolu.</i>	<i>Absent</i>	<i>Absent</i> <i>Aucun jalon révolu</i>	<i>Absent</i>
19 juin – 01 juill. E3A1J2 I2P3	<i>Actif</i> A effectué 1 jalon révolu (E3A1J2) et 1 jalon possible (E3A1J3)	Connexion : 49 Calendrier : 4 Consignes : 8 Courriels lus : 34 Courriels écrits : 25 Forums lus : 93 Forums postés : 7	<i>Absent.</i> <i>Aucun jalon révolu.</i>	<i>Absent</i>	<i>Absent</i> <i>Aucun jalon révolu</i>	<i>Absent</i>
02 juill. – 06 juill. E3A2J1 E3A2J2 E3A3J1 E3A3J2	<i>Dormant</i> Aucun jalon révolu effectué	Connexion : 5 Calendrier : 0 Consignes : 0 Courriels lus : 11 Courriels écrits : 0 Forums lus : 65 Forums postés : 7	<i>Absent.</i> <i>Aucun jalon révolu.</i>	<i>Absent</i>	<i>Absent</i> <i>Aucun jalon révolu</i>	<i>Absent</i>

Tableau 5.3 : Résultats de requêtes fournies par SIGFAD pour trois apprenants

5.2.1 – Les informations fournies sur les apprenants

Dans le tableau 5.3, nous avons présenté les résultats concernant les apprenants que le tuteur peut obtenir de SIGFAD. Le tuteur peut ainsi obtenir le niveau de productivité d'un apprenant de son groupe. Il lui sera indiqué si un apprenant a été absent sur une période donnée. Si l'apprenant n'a validé aucun des jalons obligatoires ou possibles dans la période concernée, le système lui indiquera que cet apprenant est dormant. Ceci diffère légèrement de la notion d'apprenant dormant lorsque la requête porte sur le groupe tout entier. Dans ce cas, est considéré comme dormant, un apprenant qui s'est connecté et a, au plus, lu des messages sans jamais en écrire ou laisser une trace de son passage perceptible par les autres membres du groupe.

L'état d'apprenant dormant est généralement celui qui précède l'abandon complet de la formation. En fonction du nombre de jalons effectivement accomplis par l'apprenant, le système indiquera si l'apprenant (présent et non dormant) est peu actif, actif ou très actif. Là aussi, il est naturel de penser qu'un apprenant actif au départ va progressivement passer à un état peu actif, dormant puis absent (lorsqu'il abandonne complètement la formation). On observe bien ceci pour les deux apprenants Alan et John (signalons qu'il s'agit ici de prénoms d'emprunt afin de préserver l'anonymat des personnes ayant participé à SimuLigne) : ils passent progressivement d'un état actif puis peu actif, puis dormant, avant d'abandonner complètement. Le système signale alors qu'ils sont absents. On peut donc conclure que le passage à un état peu actif doit éveiller l'attention du tuteur et des autres responsables de la formation. Ceux-ci peuvent se montrer dès ce moment plus attentifs et mettre en place des mécanismes d'aide au bénéfice des apprenants concernés.

SIGFAD fournit ensuite pour chaque apprenant, des informations sur ce que nous avons appelé sa sociabilité ou sa propension à communiquer avec les autres. Ici, il nous a été très difficile de mettre en place des critères d'émission de jugement de valeur par le système. SIGFAD indique simplement le nombre de fois où l'apprenant s'est connecté durant la période indiquée, le nombre de fois où il a accédé au calendrier de la formation, aux pages de consignes. Le système indique aussi le nombre de courriels lus et écrits, le nombre de messages lus et écrits dans les forums. Nous avons montré au chapitre 3 que ces données d'interaction permettent d'apprécier l'état du groupe et la progression de la formation. Il est évident qu'un apprenant qui se connecte moins souvent dans des périodes de même durée est probablement en train de perdre sa motivation pour la formation. De même, un apprenant qui communique peu avec le reste du groupe connaît certainement des difficultés d'intégration. Il faut rappeler que les forums constituent un outil d'intégration privilégié en FAD. Au cours de SimuLigne, le forum était l'outil le plus utilisé. C'est dire que l'indication du nombre de messages lus et écrits dans les forums constitue un bon élément d'appréciation de la participation et de l'engagement d'un apprenant.

5.2.2 - Les informations portant sur le groupe tout entier

Nous avons présenté dans les tableaux 5.1 et 5.2 les résultats des requêtes de groupe pour Aquitania et Lugdunensis. Outre la liste des apprenants présents, absents et dormants dans la période indiquée, l'on a aussi une indication de l'état de productivité du groupe. Cet état est

calculé en fonction du nombre d'apprenants ayant accompli tous les jalons prévus dans la période. Ces apprenants sont dits actifs, leur nombre et leur liste sont indiqués au tuteur. Ceci permet au tuteur de se faire une opinion plus exacte et personnelle de l'état de son groupe.

Nous représentons dans la figure 5.22 un graphique représentant l'évolution de l'état de productivité des quatre groupes de base. Les données des deux autres groupes sont fournies en annexes. Nous avons choisi de représenter l'état «Très dynamique» par la valeur 4, l'état «Dynamique» par 3, «Assez dynamique» par 2, «Peu dynamique» par 1 et «Pas dynamique» par 0. On observe ici que l'état du groupe Lugdunensis chute très rapidement. Ces informations auraient permis de réaliser que le groupe Lugdunensis se portait déjà très mal vers le 15 mai. A partir de cette date, seul un apprenant de ce groupe accomplissait correctement les activités prévues.

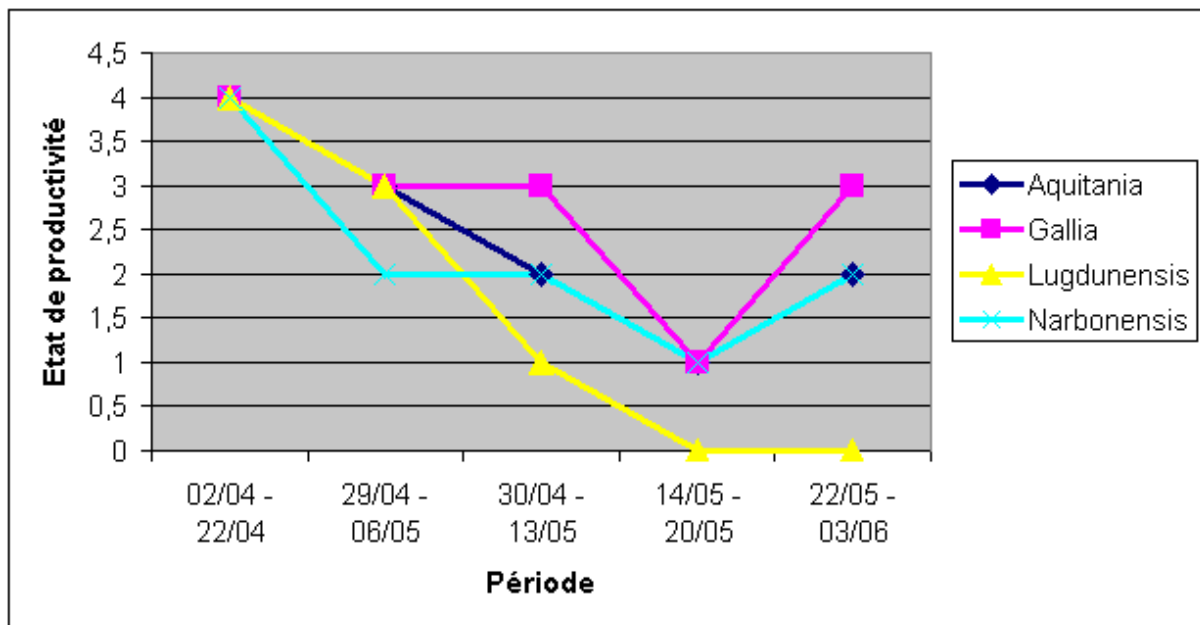


Figure 5.22 : Evolution de l'état de productivité des groupes entre le 02 avril et le 03 juin

La liste des présents permet en outre d'observer que les apprenants de ce groupe ont été toujours présents. C'est dire que des procédures d'aide et d'accompagnement auraient pu permettre d'éviter la fermeture de ce groupe à la fin du mois de mai. On observe en tous cas que SIGFAD permet au tuteur d'apprécier l'état du groupe à tout moment avec la liste des personnes absentes, présentes, dormantes, actives.

5.2.3 - Les informations portant sur les activités

SIGFAD peut répondre à des requêtes du tuteur portant sur le déroulement des activités. C'est une fonctionnalité de SIGFAD très importante, elle permet en effet de savoir si une activité donnée se déroule correctement dans le groupe. Des jalons ont été identifiés pour chaque activité. Il est possible de repérer ainsi les jalons qui ont été correctement franchis à la date indiquée dans la requête. Suivant le pourcentage des membres du groupe ayant correctement effectué une activité, SIGFAD détermine si l'on peut estimer l'activité validée dans le groupe. Actuellement, le pourcentage des personnes actives est déterminé par rapport à l'effectif initial du groupe ; nous pensons qu'il serait plus réaliste de modifier cela pour ne retenir que les personnes encore impliquées dans le groupe. Signalons que la difficulté vient ici de ce qu'un membre a toujours la possibilité d'accéder à la plate-forme même s'il s'est absenté pendant une longue période. Il est donc crucial de déterminer l'effectif à prendre en compte dans le calcul du pourcentage de personnes actives.

Nous présentons dans le tableau 5.4 les résultats fournis par SIGFAD au tuteur du groupe Aquitania s'il demande le niveau de réalisation des activités E1A1, E2A2 et I1P3 respectivement aux trois dates suivantes : 05 mai, 30 mai et 15 juin 2001.

	05 mai 2001	30 mai 2001	15 juin 2001
E1A1	<i>L'activité E1A1 n'a pas commencé</i>	<i>L'activité E1A1 n'est pas validée dans Aquitania. 3 apprenants sur 10 l'ont validée : Anita Elsa Orly</i>	<i>L'activité E1A1 n'est pas validée dans Aquitania. 3 apprenants sur 10 l'ont validée : Anita Elsa Orly</i>
E2A2	<i>L'activité E2A2 n'a pas commencé</i>	<i>L'activité E2A2 n'a pas commencé</i>	<i>L'activité E2A2 est validée dans Aquitania 4 apprenants sur 10 l'ont validé : Douglas Anita Elsa Orly</i>
I1P1	<i>L'activité I1P3 n'a pas commencé</i>	<i>L'activité est en cours. Aucun jalon de I3P3 n'a encore atteint sa date butoir</i>	<i>L'activité I1P3 n'est pas validée dans Aquitania. 2 apprenants sur 10 l'ont validée : Anita Elsa</i>

Tableau 5.4 : Résultats de requêtes portant sur l'exécution des activités dans le groupe Aquitania

Les résultats de requêtes sur le niveau d'avancement d'une activité peuvent justifier la modification du calendrier des activités. Les délais de réalisation d'une activité peuvent ainsi être modifiés selon qu'à la date de la requête, le pourcentage de validation d'une activité est satisfaisant ou non.

5.3 – Vers une prise en compte des données d’interaction dans la normalisation et la standardisation de la e-formation

La conduite de l’expérimentation SimuLigne et le souci de proposer et de construire des outils qu’on pourrait mettre à la disposition des utilisateurs pour soutenir l’apprentissage collaboratif distribué nous ont amené à nous intéresser aux questions de normalisation et de standardisation de la téléformation. Nous avons évoqué ces questions dans le chapitre 1 du présent mémoire. Nous avons évoqué les différents travaux qui se déroulent au sein des organismes de normalisation de la e-formation. De même avons-nous fait état des groupes de travail existants au niveau français. Parmi ces groupes de travail, celui portant sur l’apprentissage collaboratif nous intéresse particulièrement pour des raisons évidentes. Dans la suite de cette section, nous répertorions un certain nombre d’informations manquantes au sein des plates-formes existantes et suggérons la mise en place de telles données. Ces données permettront de mettre plus facilement et plus efficacement l’apprentissage collaboratif en œuvre dans un contexte de FAD.

Le point le plus crucial dans l’apprentissage collaboratif à distance (nous nous plaçons de ce point de vue compte tenu de notre expérimentation tout en sachant que ceci resterait fortement valable dans toute forme d’apprentissage tuteuré) concerne l’appréciation de l’état du groupe dans l’accomplissement des activités collaboratives. Il est crucial pour le tuteur (ou les autres personnes en charge de la gestion des groupes : dans notre cas nous avons aussi un coordonnateur de la formation implémentant par ailleurs une tutelle à plusieurs niveaux – tutelle du tuteur par le gestionnaire de la formation, tutelle des apprenants par le concepteur, tutelle des apprenants par le tuteur) de savoir :

- quelles sont les personnes présentes dans la plate-forme (ou dans le groupe)
- qui est en ligne lorsqu’on se connecte à la LMS (ceci favorise les interactions en encourageant notamment les communications en mode synchrone)
- quelles sont les personnes actives (par opposition aux utilisateurs dits «dormants» qui se contenteraient de se connecter sans mener des tâches ou actions rentrant dans la réalisation ou l’avancement de l’activité collaborative)
- quel est l’état de réalisation d’une activité donnée à un instant donné pour l’ensemble du groupe ou des personnes impliquées dans l’activité (en effet, il arrive qu’une activité ne concerne qu’une partie du groupe, c’est le cas dans SimuLigne avec une

partie des membres du groupe qui doivent composer l'hymne de la ville alors qu'une autre partie du groupe élaborait le récit fondateur de la ville)

- pour un utilisateur donné, quel est le niveau de sa charge de travail, s'il arrive à répondre à toute sa charge de travail ou s'il convient de le décharger d'une partie du travail en le dispensant de certaines tâches ou activités
- quel est l'état du groupe total, s'il s'agit toujours d'un groupe (cohésion, durabilité). Si le groupe reste engagé en tant que tel dans la formation ou le travail collaboratif ou s'il a disparu avec une cassure nette entre des membres déjà éjectés ou ayant abandonné et d'autres membres toujours impliqués dans les activités.

La réponse à ces différentes questions peut être construite dans les plates-formes informatiques de formation à distance existantes dans la mesure où celles-ci enregistrent toutes les traces des actions menées par les différents utilisateurs. Toutefois, les plates-formes actuelles (si nous nous référons à WebCT que nous avons utilisé au cours de SimuLigne) sont néanmoins limitées et ces limitations se retrouvent notamment :

- au niveau de l'absence de prise en compte du temps : absence de *flowchart* pour représenter l'enchaînement des activités ;
- au niveau de l'absence de la notion de groupe ou de sous-groupe engagé dans la réalisation d'une tâche ou d'une activité donnée ;
- au niveau de la difficulté à modéliser une activité donnée : il est impossible de rattacher une activité à un ensemble (de tâches et) d'outils à utiliser et ceci dans des délais fixés.
- au niveau de l'éparpillement important des données d'interaction. La plate-forme utilisée durant SimuLigne exige de recouper des données disséminées dans plusieurs fichiers texte différents pour obtenir des informations de suivi d'un forum par exemple. Ceci exige du reste un travail fastidieux et loin d'être évident pour celui qui a la hardiesse de s'engager dans une telle tâche !

Bref, le modèle conceptuel d'activité de SimuLigne (figure 3.5) est pratiquement impossible à mettre en œuvre dans les LMS existantes. Il faut noter que certaines LMS proposent des bases de données, mais celles-ci, on s'en doute, ne peuvent contenir que des données partielles correspondant à la vision de leurs constructeurs. Il serait bien plus efficace de définir la nature et le type de telles données dans le cadre d'une normalisation des LMS.

Pour répondre donc aux exigences que nous pose l'apprentissage collaboratif distribué, il faut pouvoir, dans les LMS :

- définir les membres des groupes (login et statut). ;
- définir les membres de sous-groupes concernés par une activité donnée si celle-ci ne concerne pas l'ensemble du groupe de départ ;
- définir pour chaque activité (dans un *flowchart*) les différentes tâches qui la composent et surtout les outils à utiliser (ce n'est que l'utilisation de ces outils enregistrée sous forme de traces dans des fichiers de traces du serveur HTTP – Apache dans notre cas – ou d'autres fichiers spécifiques de la PIFAD qui permettra de dire si oui ou non la tâche a été réalisée par un individu donné), les dates de réalisation des différentes tâches, et les différentes personnes impliquées dans l'activité (ceci rejoint la notion de sous-groupe évoquée au point précédent) ;
- définir des informations de session : date et heure d'ouverture d'une session, durée de connexion en ligne, outils utilisés ;
- définir des informations de niveau méta permettant de suivre une activité donnée ou la gestion des forums de discussions ouverts. C'est évidemment une tâche très importante car elle traduit l'accompagnement offert aux apprenants et aux autres utilisateurs par le dispositif humain et technique de la formation ;
- organiser les données d'interaction dans de véritables bases de données, rendant ainsi ces données plus accessibles à un système comme SIGFAD.

Bilan, Perspectives et Conclusion

Dans le cadre de cette thèse, nous avons étudié l'apprentissage collaboratif distribué et proposé quelques outils informatiques pour le soutenir. Nous présentons dans ce dernier chapitre un bilan de nos travaux en les mettant en perspective avec des travaux similaires existants ; ceci nous permettra de dégager les apports de nos travaux de thèse. Nous dégagerons les perspectives futures et nous finirons par une conclusion.

Nous avons montré au chapitre 3 que les données d'interaction enregistrées dans les LMS peuvent être utilisées pour bâtir des indicateurs (très simples sur un plan conceptuel) de l'état des groupes et de la progression d'une session de FAD. Nous avons montré que la baisse des activités dans un groupe ou par un utilisateur donné se reflète dans les données d'interaction afférentes. Ceci constituait pour nous la justification majeure pour la mise au point d'un système informatique intégrant de tels outils et qu'on pourrait ajouter aux plates-formes existantes. Pour diverses autres considérations, nous avons eu recours au paradigme d'agent pour proposer les fonctionnalités annoncées dans un système multi-agents. Ce système multi-agents, que nous avons baptisé SIGFAD, fournit principalement aux tuteurs une appréciation de l'état de leurs groupes (personnes présentes, absentes, dormantes, actives), le niveau de réalisation d'une activité. SIGFAD permet aussi de se focaliser sur un utilisateur donné et d'obtenir, d'une part, son niveau de productivité, d'autre part, sa téléprésence (nombre de connexions, nombre de courriels lus et postés, nombre de messages lus et postés dans les forums, etc.). Ce dernier facteur est très important dans le contexte de la formation à distance, et plus encore lorsque la pédagogie est basée sur une approche collaborative. Le coordonnateur de la formation peut consulter les méta-informations (informations permettant le suivi des groupes, des activités), les modifier, en ajouter ou en supprimer.

A l'état actuel, SIGFAD n'intègre pas encore les propriétés d'apprentissage machine, qui lui permettraient de notre point de vue, de tirer pleinement profit des caractéristiques annoncées des systèmes à base d'agents. De même, les contraintes de l'architecture interne de nos agents ne nous ont pas encore permis de tirer pleinement profit du degré de sophistication que peuvent nous procurer les états mentaux du modèle BDI. Cet état de choses ne nous choque pas, car il était d'abord important de cerner et d'implémenter les fonctionnalités de SIGFAD. Ceci a été fait, il s'agira dans la suite de permettre à nos agents d'apprendre à partir du comportement des utilisateurs. C'est lors de cette phase que les états mentaux des agents BDI permettront d'implémenter au sein de ces derniers l'apprentissage machine. De même, l'existence actuelle de la base de données des interactions donne l'impression d'une

centralisation excessive des données. Les données d'interaction seront réparties entre les différents agents des utilisateurs : l'agent du tuteur conservera les données relatives à son groupe alors que l'agent de l'apprenant aura pour sa part, les données spécifiques à son propriétaire. En attendant, le tuteur dispose déjà d'un outil capable de lui donner par un simple clic un état du groupe ou le niveau de réalisation d'une activité donnée. Ceci n'existait pas avant SIGFAD. Nous avons, à ce titre, rapporté plusieurs remarques de chercheurs indiquant la difficulté à savoir ce que font les apprenants dans un contexte de formation à distance (Haythornworthe, 1999). D'autres chercheurs ont essayé de mettre au point de tels outils ; si la démarche conceptuelle a souvent été faite, on a rarement vu un début d'implémentation. Fort de ce constat, notre souci aura été de partir de l'analyse du système à son implémentation concrète. Nous n'avons pas hésité à utiliser des méthodologies et outils identifiés comme faisant partie de la programmation orientée agent. Un autre enjeu majeur de nos travaux consistait à développer un système en restant dans la démarche agent. S'il est vrai aujourd'hui, en programmation orientée agents, que les déclarations d'intention sont bien plus nombreuses que leur mise en œuvre concrète, nous pensons que ce paradigme, loin d'être démodé, doit compléter sa panoplie d'outils pour que les espoirs qu'il recèle puissent voir le jour dans des implémentations.

Bilan des travaux

Dans cette thèse, nous sommes partis d'une expérience de formation à distance grandeur réelle pour dégager un certain nombre de problèmes spécifiques à ce mode de formation. Partant d'une approche pédagogique basée sur l'apprentissage collaboratif, les problèmes que nous avons identifiés sont davantage inhérents à l'apprentissage collaboratif distribué. Finalement, ces problèmes recourent à la fois ceux de la formation à distance et ceux de l'apprentissage collaboratif. Nous avons perçu dès le départ que les plates-formes de télé-formation existantes ne possèdent pas toutes les fonctionnalités nécessaires au suivi de l'apprentissage collaboratif distribué. Grâce aux concepts de la théorie de l'activité, nous avons construit un modèle représentant une activité dans une session de formation à distance comme celle qui nous concernait. En mettant ce modèle en relief avec le schéma général de PIFAD, nous avons pu identifier les insuffisances de telles plates-formes. Nous avons aussi mis en exergue l'insuffisance de réification de la participation des utilisateurs dans les plates-formes existantes malgré le fait que les actions et interactions de tous les utilisateurs laissent des traces dans la plate-forme. Nous avons montré que les données d'interaction enregistrées dans

les fichiers de traces peuvent permettre de construire des indicateurs (relativement peu sophistiqués sur le plan conceptuel, puisque basés uniquement sur le volume et le type d'interactions) reflétant l'état des groupes engagés et l'évolution de la session de formation à distance (Mbala et al., 2002). L'absence d'indicateurs permettant d'apprécier l'état du groupe, l'évolution de la session de formation et les insuffisances conceptuelles des plates-formes de télé-formation nous ont amené à proposer l'ajout de nouvelles fonctionnalités axées principalement sur le soutien des interactions dans les groupes. En raison des spécificités de la FAD, nous avons dû recourir au paradigme d'agent pour proposer la construction de ces nouvelles fonctionnalités. Plusieurs recherches dans les EIAH portent sur l'utilisation d'une approche par composants dans le développement des prototypes afin de tirer profit des propriétés de réutilisabilité pour en faciliter la coopération et l'interopérabilité. Il a fallu nous positionner par rapport à cette approche et montrer que le paradigme d'agent était le mieux à même de répondre à nos préoccupations.

Le recours au paradigme d'agent nous a permis d'ouvrir un vaste chantier qui a consisté en l'identification des outils de développement et à leur mise en œuvre. Nous nous sommes en effet bien vite rendu compte que la littérature scientifique relative aux agents est pleine de bonnes intentions et de promesses alléchantes, mais reste assez discrète sur les outils et méthodologies de mise en œuvre des systèmes à base d'agents. Durant nos travaux, nous avons identifié une méthodologie d'analyse, de conception et de spécification de systèmes multi-agents ; nous l'avons mise en œuvre. Cette méthodologie a comme point fort, l'identification et la spécification des interactions entre les agents du SMA ; elle n'impose et pour ainsi dire, ne dit rien sur le contenu de chaque agent, qu'elle considère dès lors comme une boîte noire. Afin de construire l'intérieur de nos agents, nous avons identifié un modèle d'architecture interne basée sur le modèle d'états mentaux BDI. L'implémentation concrète des mécanismes de communication entre nos agents s'est faite en ayant recours à la programmation des *sockets* en Java.

Nous avons implémenté un système multi-agents, SIGFAD, destiné à être couplé aux plates-formes de télé-enseignement pour les faire progresser vers de véritables environnements informatiques de formation à distance. L'expérimentation de formation à distance que nous avons menée nous a permis de cerner le rôle crucial du tuteur dans ce mode de formation. SIGFAD met à la disposition du tuteur des outils simples capables de lui permettre d'avoir l'état de son groupe (personnes présentes, absentes, dormantes, actives ou dynamiques), le

niveau de réalisation d'une activité donnée (à partir d'un certain nombre de jalons identifiés au départ), l'état d'un apprenant quelconque. SIGFAD propose de la même façon des outils à l'apprenant et au coordonnateur : SIGFAD permet au coordonnateur de consulter des informations, mais aussi de modifier le calendrier des activités ou de réorganiser les groupes. Les apprenants peuvent obtenir une réification de leur participation à la session de FAD et aussi évaluer leur charge de travail sur une période donnée.

Résultats obtenus

Nos travaux de thèse avaient pour but de proposer des systèmes informatiques pour soutenir la formation à distance. Pour cela, nous avons participé à la réalisation d'une expérimentation de formation à distance grandeur réelle, c'est-à-dire, une formation à distance étalée sur plusieurs semaines et mettant en œuvre de vrais apprenants avec des objectifs pédagogiques bien circonscrits. En recourant à la théorie de l'activité, nous avons proposé un modèle conceptuel d'une activité en formation à distance. Nous avons pris en compte le facteur temps dans ce modèle et avons ramené l'échelle d'analyse de l'activité au niveau du groupe. La modélisation d'une activité dans le contexte de l'apprentissage collaboratif distribué constitue un de nos résultats.

La mise en perspective de notre modèle conceptuel d'activité et du schéma général des LMS nous a permis de mettre en exergue les insuffisances de ces dernières dans le suivi des sessions de formation à distance en général. Ces insuffisances, de notre point de vue, se manifestent surtout dans l'absence d'outils permettant au tuteur d'apprécier l'état de son groupe et de faire le point sur la réalisation des activités didactiques par les apprenants. Nous avons développé un système multi-agents qui reprend toutes les fonctionnalités qui permettraient, lorsqu'il est couplé à une LMS, au tuteur de disposer d'un tableau de bord permettant de suivre l'activité des apprenants et d'apprécier l'état du groupe tout entier. C'est le résultat majeur de nos travaux de thèse.

Les questions de recherche en FAD imposent une approche pluridisciplinaire. Nous avons ainsi dû manipuler plusieurs termes qui n'ont pas toujours la même acception suivant la discipline scientifique. Nous avons été amenés à mettre en place une terminologie qui recensait tous les termes rencontrés et à leur donner une signification bien précise. Ceci peut être inscrit au nombre de nos résultats de recherche.

Le souci de rendre SIGFAD générique nous a conduit à nous intéresser aux problèmes de normalisation et de standardisation de la e-formation. Nous avons proposé un certain nombre de données d'interaction qui devraient être facilement accessibles au sein des LMS pour faciliter la mise en place d'outils de support à l'apprentissage collaboratif distribué.

Apports de nos travaux de thèse

Plusieurs travaux de recherche portent sur la mise au point d'environnements informatiques de support pour la formation à distance. Ces recherches partent toujours du principe qu'il faut encourager les interactions entre les membres pour juguler les problèmes inhérents à ce type de formation. C'est la raison pour laquelle nombre de chercheurs essayent de mettre en place un type de pédagogie qui puissent encourager les interactions entre les personnes concernées. C'est ainsi que Sébastien George (George, 2001), dans ses travaux de thèse, travaille sur la mise en place d'un environnement support de projet. Christophe Després (Després, 2001) travaille sur la mise en place d'un environnement de suivi synchrone par le tuteur dans une pédagogie fondée sur l'apprentissage par l'action. Nous souhaitons plus particulièrement nous positionner par rapport à ces deux thèses qui ont été soutenues au sein du Laboratoire d'Informatique de l'Université du Maine (LIUM). Ceci nous permettra de mieux expliciter l'apport de nos travaux de thèse. Comme objectifs communs de nos travaux et de ceux développés au LIUM, nous pouvons citer le soutien des interactions dans un contexte de FAD, l'instrumentation de l'activité du tuteur/formateur et de l'apprenant. Les travaux de George et Després se focalisent davantage sur l'enrichissement des interactions et la simplification de la tâche du tuteur. Nos travaux de thèse quant à eux sont orientés vers le développement d'outils informatiques capables de combattre l'isolement sociologique des apprenants et les forts taux d'abandon qui en résultent. Les travaux menés au sein du LIUM concernent un contexte de formation présentielle pour les élèves des classes de collège alors que nous nous intéressons à un public d'adultes suivant une formation chacun chez soi, entièrement distants les uns des autres. Ces contextes différents induisent aussi des outils et des modes de communication distincts : mode asynchrone dans notre cas et synchrone au LIUM. Tout ceci vient souligner la complémentarité de nos travaux avec ceux menés au LIUM et montre encore l'extrême complexité et variété des situations de FAD. Sur le plan purement informatique, il faut dire que nous avons utilisé entre autres, le paradigme d'agent pour construire nos environnements. Toutefois, les travaux de George et de Després avaient une démarche centrée sur le produit dans ce sens qu'ils ont cherché des outils permettant de construire des architectures multi-agents pour leurs environnements sans se poser des

questions plus fondamentales liées au cycle de développement logiciel des systèmes à base d'agents. Quant à nous, notre souci aura davantage été d'identifier et d'appliquer des méthodes et des outils appropriés pour analyser, concevoir, spécifier et développer un système multi-agents : une démarche beaucoup plus génie logiciel. Ayant en effet identifié le manque d'outils de développement, nous avons davantage souligné ce manque en insistant sur les besoins.

L'autre singularité de nos travaux de thèse réside dans la démarche expérimentale qui aura été la nôtre. Nous sommes partis d'une expérimentation de formation à distance grande nature, mettant en œuvre de véritables apprenants, avec des objectifs didactiques précis et travaillant réellement à distance. Cette expérimentation nous a permis d'identifier différents problèmes liés à la formation à distance comme la qualité de service des réseaux de communication et d'information, les contraintes induites par la technologie aussi bien pour les apprenants, que pour les tuteurs et les autres acteurs. Nous sommes aujourd'hui familiers de plusieurs phénomènes qu'on ne saurait percevoir dans des expérimentations concernant un nombre réduit de sujets et fortement limitées dans le temps. Il convient de noter aussi que notre démarche aura été pluridisciplinaire, même si nous rapportons peu ici les contributions issues de la conduite du projet de recherche ICOGAD avec les chercheurs d'autres disciplines comme les collègues de l'Open University et ceux du Laboratoire de Psychologie de Nancy.

D'un point de vue purement informatique, divers travaux de recherche évoquent le paradigme agent comme un moyen d'implémenter les systèmes informatiques destinés au soutien de la FAD. Cette évocation ne va pas souvent jusqu'aux préoccupations concrètes liées au développement de systèmes à base d'agents. Le paradigme agent et les outils de développement des systèmes informatiques fondés sur ce dernier sont cardinaux dans notre démarche. En effet, les concepts liés aux agents étant relativement récents, les méthodes et les langages adaptés à cette forme de programmation sont encore en cours de développement. Plusieurs applications industrielles et notamment liées à la robotique ou à la vie artificielle viennent nourrir les réflexions des chercheurs qui mettent au point les outils de développement. La FAD et la e-formation en général, ont des spécificités qui doivent aussi être prises en compte dans les préoccupations des personnes qui mettent au point les méthodologies, langages et autres outils de développement des systèmes à base d'agents. Nos travaux de thèse viennent souligner ce besoin. On voit ici que le paradigme d'agent est aussi porteur d'espérances pour la FAD et la e-formation qui en sont aussi des domaines

d'application, au même titre que le commerce électronique, le *data mining* ou la robotique, pour ne citer que ceux-là.

Dans la conduite de nos études bibliographiques, nous avons laissé toujours transparaître notre souci d'aller de l'abstraction à l'implémentation concrète d'un système multi-agents. C'est la raison pour laquelle nous évoquons les préoccupations liées à l'apprentissage machine dans les systèmes multi-agents. Les recherches sur les systèmes multi-agents et l'apprentissage machine sont très souvent menées de façon parallèle, bien que nombre de chercheurs commencent à proclamer l'interconnexion des deux domaines de recherche. Nos travaux, par leur constance à aller jusqu'à des implémentations concrètes, viennent souligner la proximité qui existe entre l'apprentissage machine et les systèmes multi-agents.

Perspectives

Le concept d'agent nous a intéressé dans la mesure où il donne la possibilité de construire des artefacts logiciels capables d'apprendre à partir des actions de l'utilisateur. Au cours de nos travaux de thèse, nous nous sommes attachés à identifier quelles actions le tuteur avait besoin de mener pour bien conduire sa session de formation à distance. Ceci constituait déjà un vaste chantier. La suite logique sera d'implémenter au sein des différents agents les mécanismes d'observation des actions de l'utilisateur, en d'autres termes, les mécanismes d'apprentissage automatique.

Les perspectives de nos travaux consistent à mettre au point les outils plus spécifiquement destinés au coordonnateur de la formation (possibilité d'avoir une vue synthétique et comparative de la progression de l'ensemble des groupes ; possibilité de modifier très rapidement le calendrier de la formation ou de réorganiser les groupes lorsque ceci s'impose) ; à l'apprenant, et surtout la construction de la base de données des interactions tout au long de la session de formation. A ce titre, nous tenons à faire évoluer l'architecture de SIGFAD afin qu'il s'affranchisse de la base de données des interactions qui constitue une trop forte centralisation des données.

Une perspective imminente de nos travaux de thèse est l'utilisation d'une infrastructure de communication de plus haut niveau que les *sockets* que nous avons utilisés pour l'implémentation de la communication entre nos agents. Il convient aussi d'intégrer très

rapidement à notre SMA un langage de communication comme KQML pour permettre un réel partage de connaissances entre nos agents.

L'élaboration de modèles mathématiques qui permettraient à partir du nombre et du type d'interactions apparues, de déclencher des mécanismes d'alerte en direction des tuteurs et/ou du Coordonnateur est un point important de la suite de nos travaux de thèse. L'exploration d'outils issus de l'analyse des réseaux sociaux, comme la cohésion du groupe constitue une voie prometteuse pour l'élaboration de tels modèles (Reffay et Chanier, 2003a, 2003b, 2002).

Conclusion

Le présent mémoire est le résultat de quatre années de recherche. Partant d'une problématique assez laconique : « Comment soutenir les interactions dans la formation à distance ? », nous avons identifié le paradigme d'agent comme pouvant servir de socle pour la solution informatique à une telle problématique (Mbala et al., 2002), (Mbala, 2002). Nous avons participé à la mise en œuvre et à l'administration d'une expérimentation de formation à distance, c'était l'expérimentation SimuLigne. Il a fallu procéder à un recueil et à une organisation des données d'interactions ; nous avons participé à la construction de la base de données contenant ces informations. Il a fallu montrer que les données d'interaction enregistrées pouvaient permettre de construire des indicateurs de l'état des groupes et de la progression de la session de formation. L'absence de tels indicateurs dans les plates-formes de télé-enseignement existantes nous a permis de justifier de devoir ajouter à celles-ci de nouvelles fonctionnalités. Nous avons choisi d'implémenter ces fonctionnalités sous la forme d'un système multi-agents qui viendrait se greffer aux plates-formes existantes pour en faire de véritables systèmes supports de la formation à distance. Nous avons choisi d'aller jusqu'au bout de l'implémentation de notre système multi-agents. Très vite, nous nous sommes aperçus de la rareté des outils de développement appropriés. Il nous a fallu donc identifier une méthodologie pour l'analyse, la conception et la spécification des systèmes multi-agents. Nous l'avons fait et avons appliqué toutes les étapes de cette méthodologie. L'objectif de construction concrète des agents nous a amené à nous intéresser à un modèle d'architecture interne des agents. Cette architecture interne nous conduit vers le langage Java, dont il a fallu saisir toutes les subtilités pour programmer notre système multi-agents. Le souci de généralité de notre système multi-agents nous a naturellement conduit vers les questions de normalisation du *e-learning* en général, et de la standardisation des plates-formes en particulier.

Est-il utile de rappeler que, la formation à distance étant dès le départ un champ pluridisciplinaire, ces quatre années de recherche nous auront permis de nous intéresser à plusieurs domaines de recherche allant des sciences humaines et sociales à l'informatique. L'expérimentation SimuLigne nous a conduit à travailler avec des personnes d'horizons divers ; ceci s'est révélé fort enrichissant pour nous, y compris sur un plan purement humain.

Annexes

A - Les productions de SimuLigne

Nous présentons ci-après quelques productions auxquelles sont parvenues les trois groupes de base engagés dans la simulation globale. Rappelons que l'un des groupes, Lugdunensis, n'est pas allé jusqu'au terme de la formation.

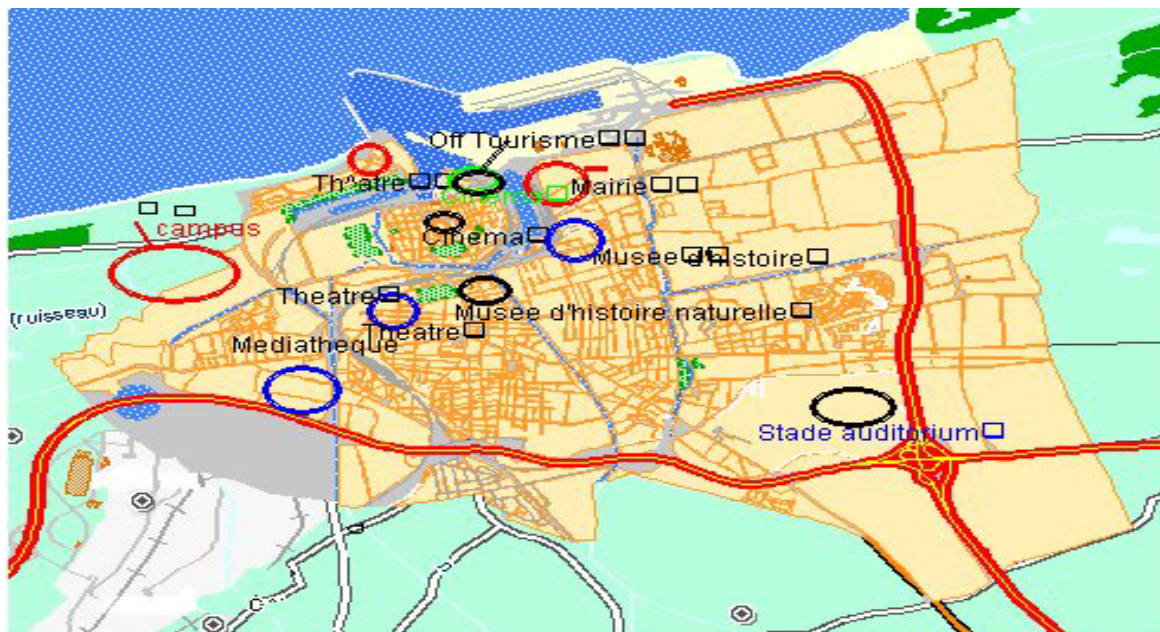


Figure A.1 : Plan de la ville proposé par le groupe Aquitania



Figure A.2 : Plan du campus proposé par le groupe Gallia

Hymne de St Jacques

O vieux port aimé d'où je m'étais enfui
La ville où j'ai erré cette dernière heure
Où sont les rues où elle avait souri
La rue Saint Jacques avec la coquille et le cœur.

Si seulement j'étais à Saint Jacques-sur-Mer pour l'heure
Au bord d'une mer turquoise où on est bien accueilli
Les petites maisons rouges, les arbres verts, une ville en fleur
Mon rêve, une coquille Saint Jacques et un grand cru Chablis.

Quel paysage ! Ses champs et ses ruisseaux,
Ses vergers pleins de fruits si mûrs et beaux !
Courons aux baies où, tous enfants, jouions-nous –
A la mer la plus belle qu'on ait jamais connue!

Cette petite ville de mes rêves où tous les jours faisaient beau
Les mémoires durent de la beauté, dont le nom je n'ai jamais connu
Des cheveux d'or, des yeux bleus, la peau de pêche, quel tableau !
Une soirée mémorable, hélas, mon idole du temps jadis, elle a disparu

Figure A.3 : Hymne de la ville proposé par le groupe Aquitania

Mr N Russell
Summers Schools
The Open University
Milton Keynes
MK7 6AA
Grande Bretagne

Monsieur

Je vous remercie de votre lettre de date récente, ainsi que de votre demande de renseignements concernant les menus pour l'école d'été. Nous avons à notre disposition un grand nombre de menus végétarien mais je ne vous envoie que quelques uns comme spécimens ainsi que un menu bien particulier.

Entré

Briks farcies aux poivrons
Bouchées aux noix
Champignons farcis à la crème de ciboulette
Frisée aux noix
Fromage farci aux fruits secs
Salade à ciboulette

Plats Principaux
Courge sphaghetto à la provencale
Croque végétarien
Lasagnes aux aubergines
Sandwich crudité
Tarte aux poireaux

Desserts
Charlotte aux amandes et aux noix
Crème brûlée
Gâteau léger à l'ananas
Tarte aux pommes

Je vous tiens à mon entière disposition pour tout demande de renseignements supplémentaires et vous prie d'agréer, Monsieur, l'assurance de mes sentiments dévoués.

Michel Martinez
Chef Cuisinier

Figure A.4 : Une proposition de menus par le Chef Cuisinier du groupe Narbonensis

B – Application de la méthodologie MaSE à l’analyse, la conception et la spécification de SIGFAD

B-1 Les diagrammes de séquence décrivant les cas d’utilisation

Nous présentons dans les figures qui suivent les différents diagrammes de séquence décrivant tous les cas d’utilisation que nous avons identifié. Nous ne présentons pas le digramme de séquence qui fournit la liste des utilisateurs présents/absents/dormants, ce dernier se trouve dans le corps du présent mémoire.

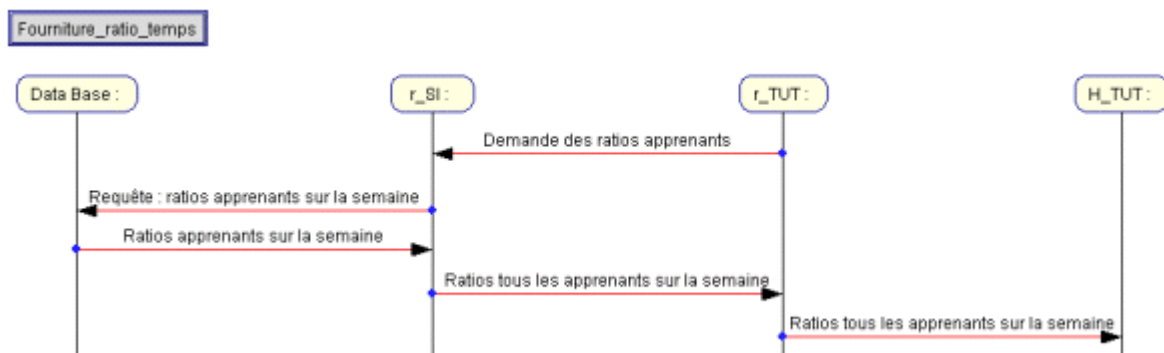


Figure B.1 : Le diagramme de séquence pour fournir les ratios de temps par semaine

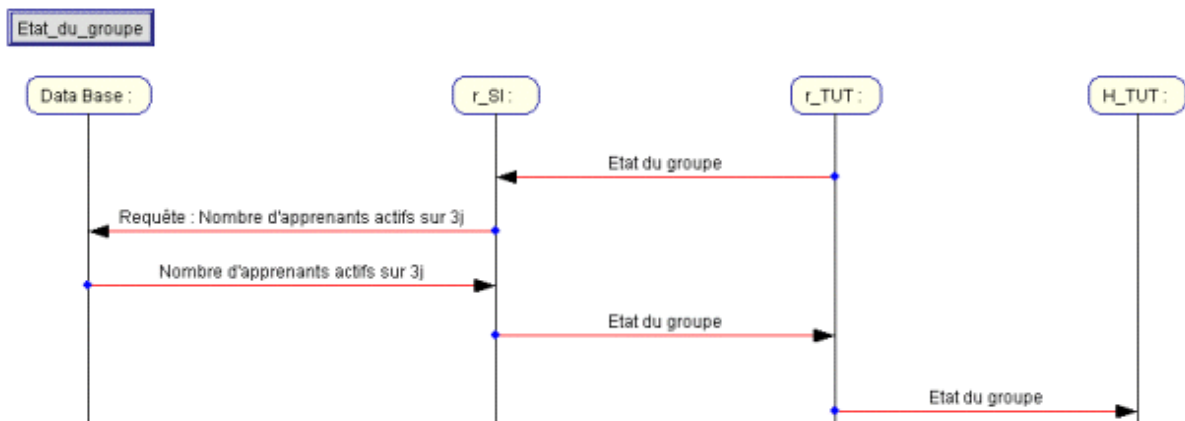


Figure B.2 : Le diagramme de séquence pour fournir l’état du groupe

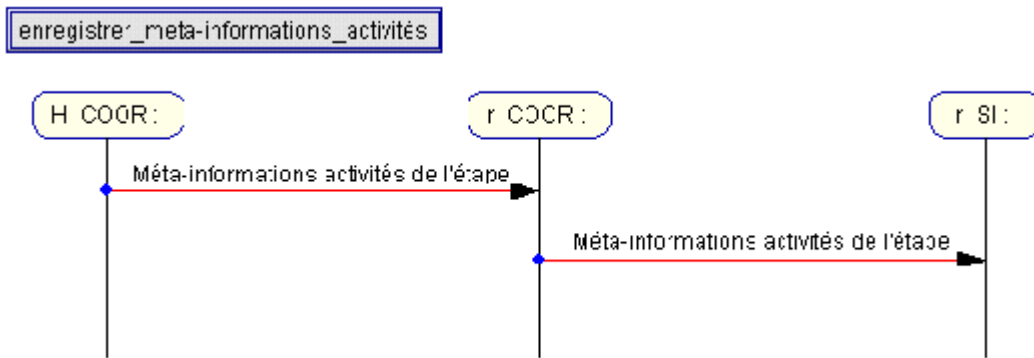


Figure B.3 : Le diagramme de séquence pour recevoir la liste des méta-informations sur les activités

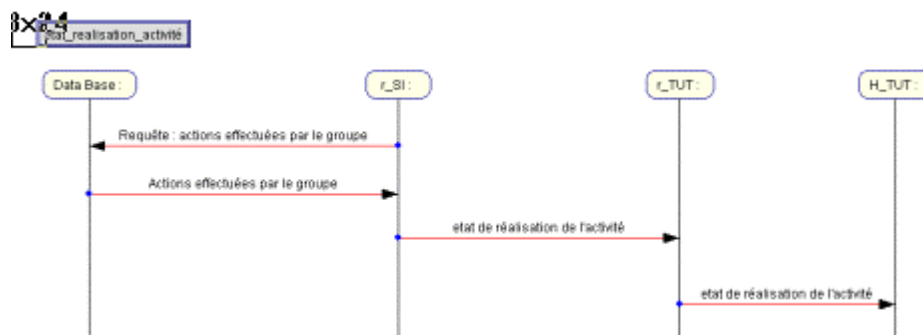


Figure B.4 : Le diagramme de séquence pour déterminer l'état de réalisation d'une activité

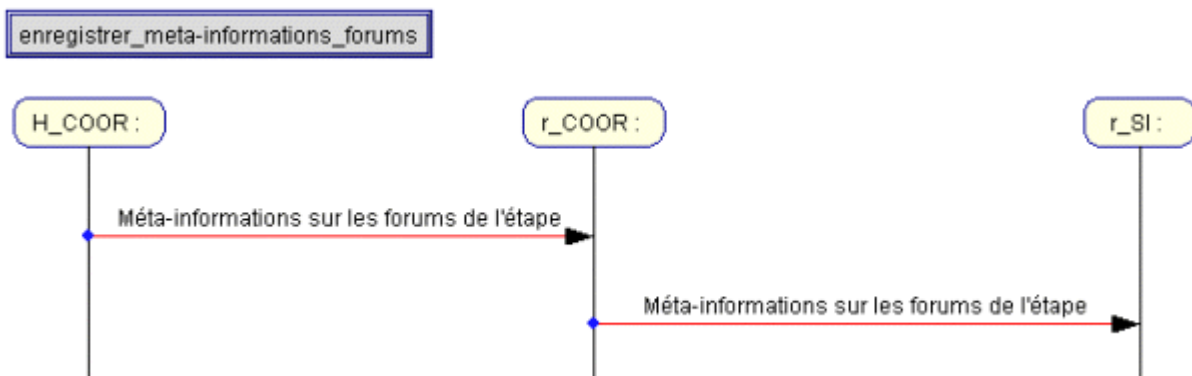


Figure B.5 : Le diagramme de séquence pour recevoir les méta-informations sur les forums

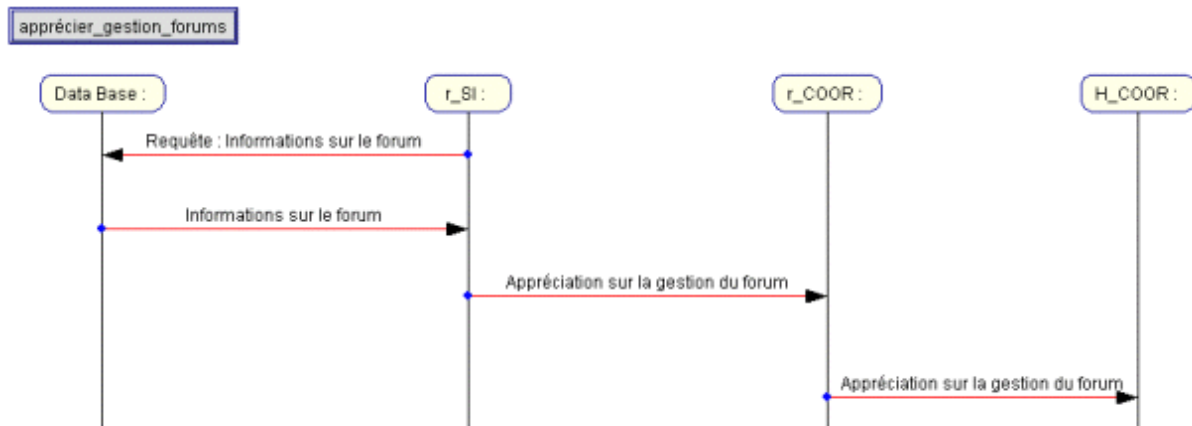


Figure B.6 : Le diagramme de séquence pour déterminer si un forum est bien suivi

B.2 Les automates à états finis décrivant les tâches concurrentes

Le fonctionnement des tâches concurrentes associées aux rôles est décrit dans des automates à états finis. Nous avons présenté en figures 5.9 et 5.10 les automates décrivant la manière dont l'agent superviseur des interactions fournit la liste des apprenants présents/dormants/absents à l'agent du tuteur. Les figures présentées dans cette section décrivent les autres tâches concurrentes :

- la fourniture des meta-informations concernant les activités et les forums : sont mis en œuvre les rôles r_COOR et r_SI avec les tâches *meta-informations_vers_aSI* et *get_metainformations* ;
- la fourniture des ratios de temps du groupe sur la semaine : les rôles r_TUT et r_SI sont concernés avec les tâches *ask_timeratio* et *ratio_vers_tuteur* ;
- la fourniture de l'état du groupe : les rôles r_TUT et r_SI avec les tâches *ask_etatgroupe* et *etatgroupe_vers_tuteur* ;
- la fourniture de l'état de réalisation d'une activité : les rôles que précédemment avec les tâches *ask_etatactivite* et *etatactivite_vers_tuteur* ;
- la fourniture de l'avis de gestion d'un forum : les rôles que précédemment avec les tâches *get_gestionforum* et *etatforum_vers_tuteur*.

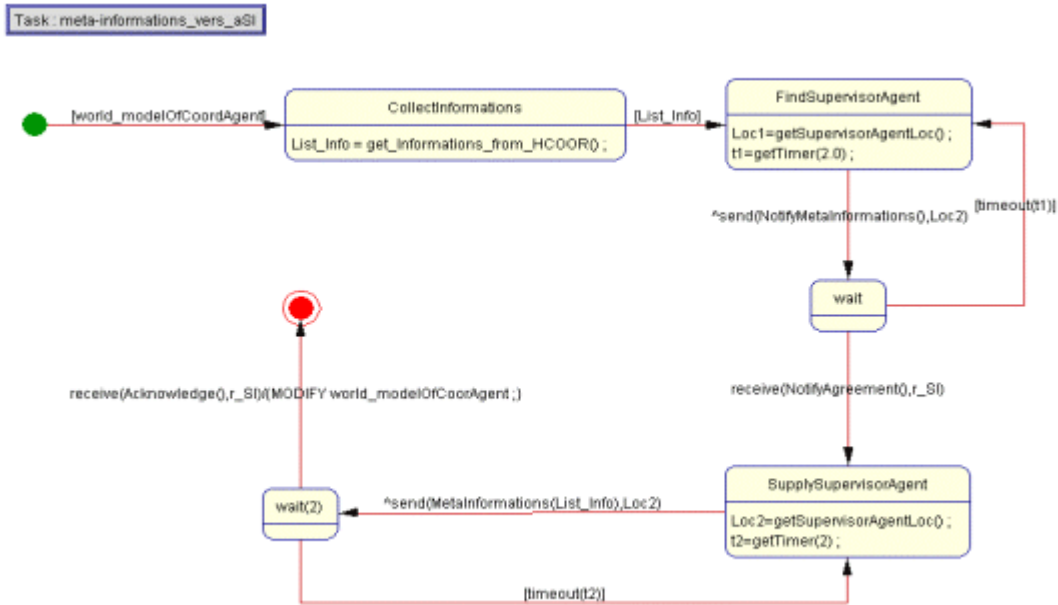


Figure B.7 : Etats et transitions de r_COORD dans la fourniture des méta-informations à r_SI

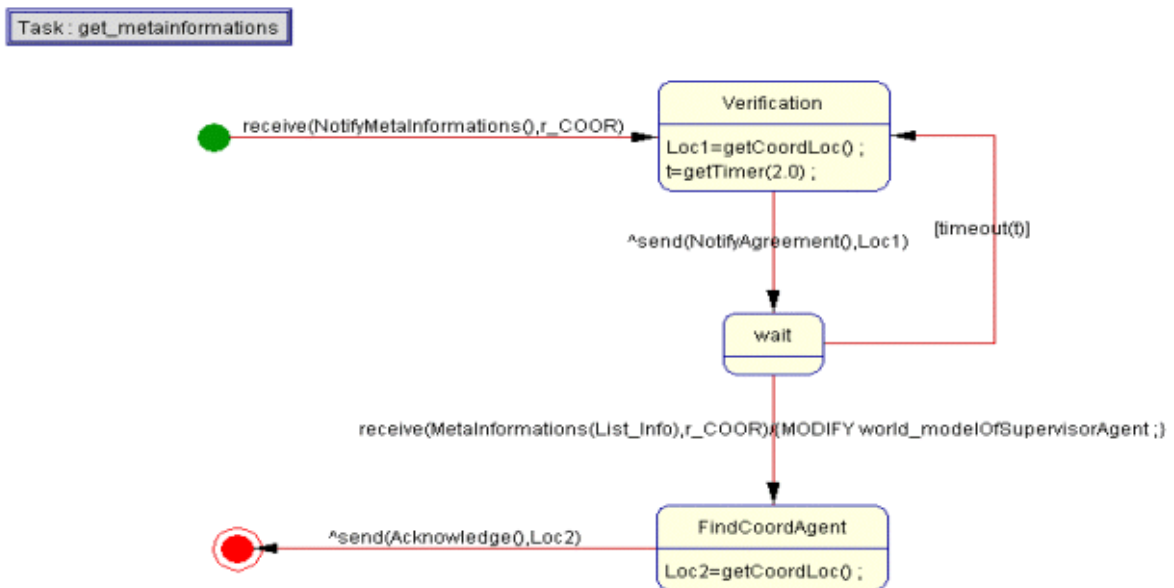


Figure B.8 : Etats et transitions de r_SI dans la réception des méta-informations

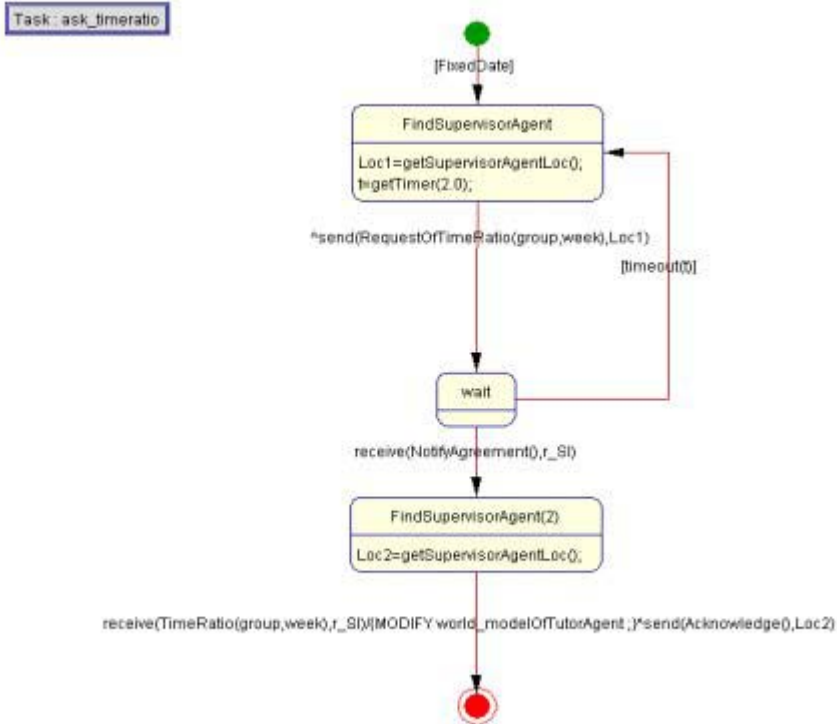


Figure B.9 : Etats et transitions de r_TUT pour obtenir les ratios de temps

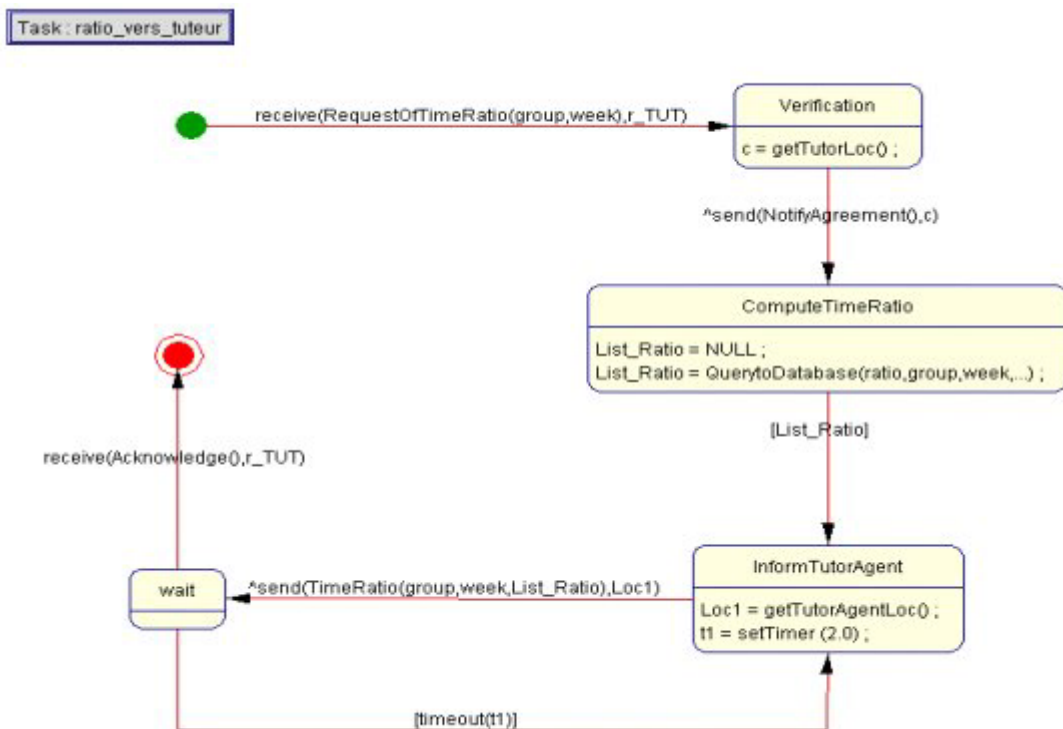


Figure B.10 : Etats et transitions de r_SI pour fournir les ratios de temps

Task : ask_etatgroupe

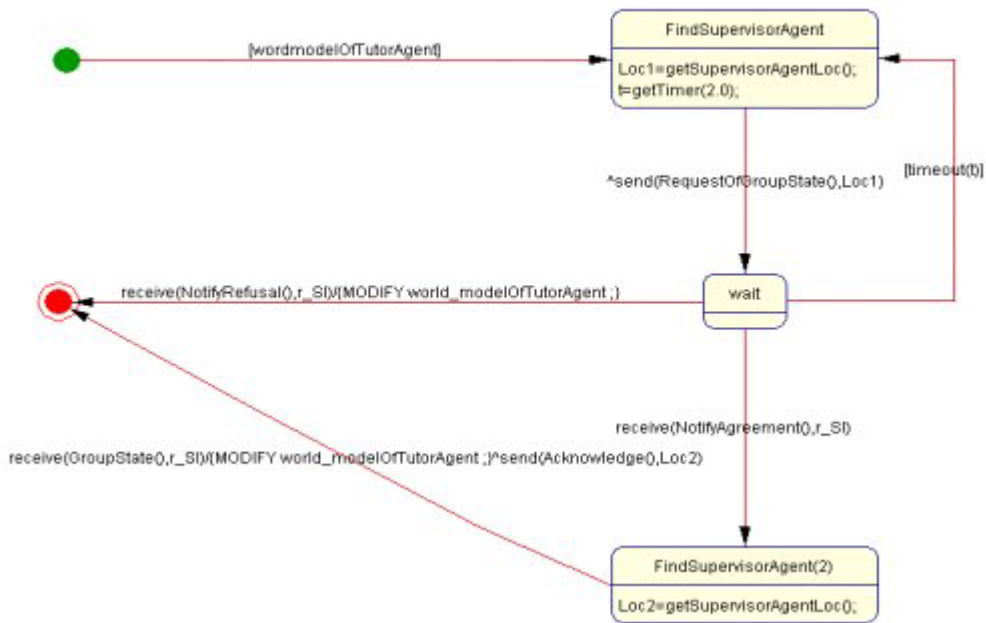


Figure B.11 : Etats et transitions de r_TUT pour recevoir l'état du groupe

Task : etatgroupe_vers_tuteur

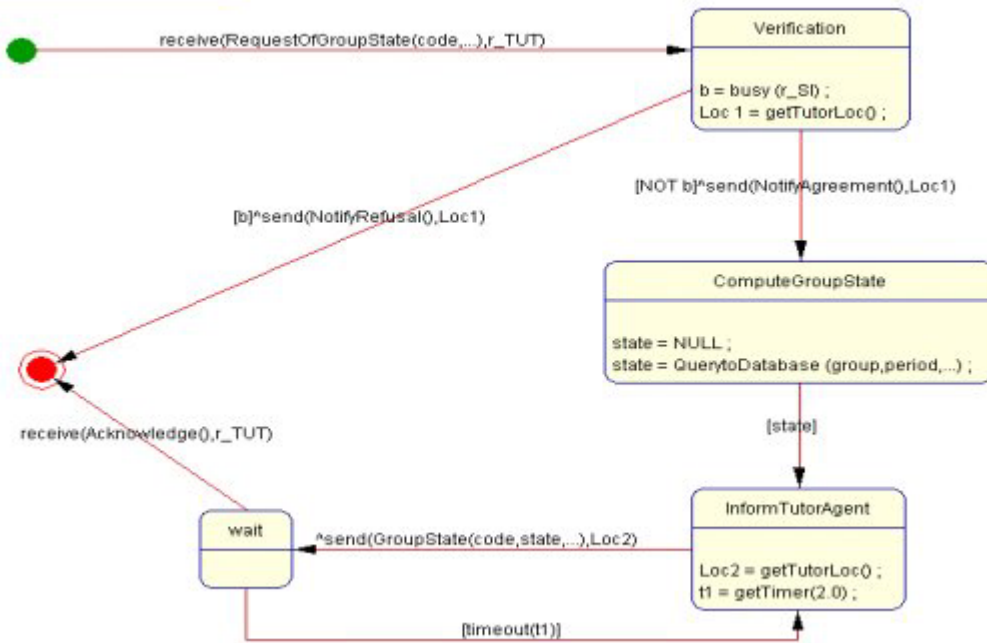


Figure B.12 : Etats et transitions de r_SI pour fournir l'état du groupe

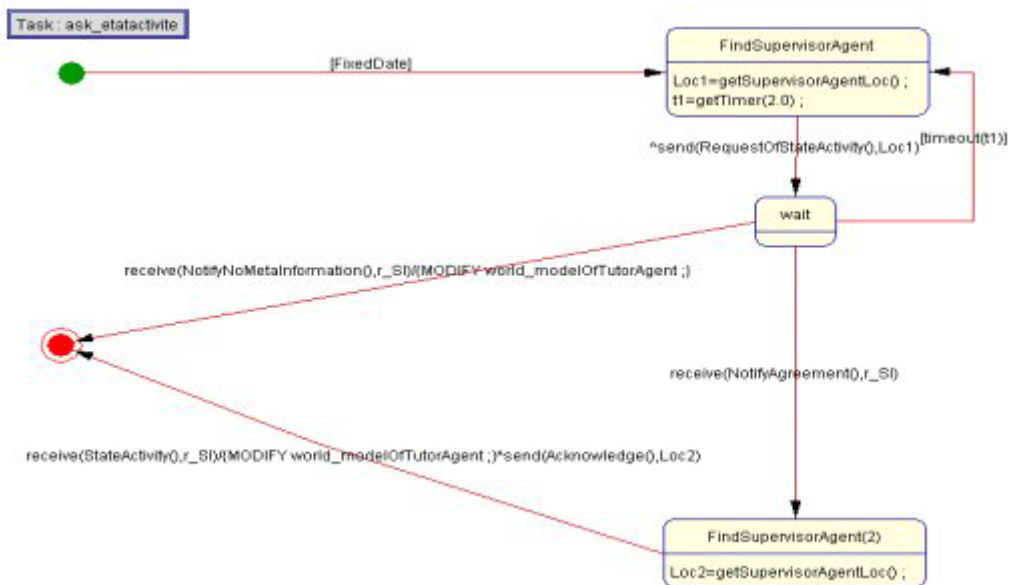


Figure B.13 : Etats et transitions de r_TUT dans l'indication de l'état de réalisation d'une activité

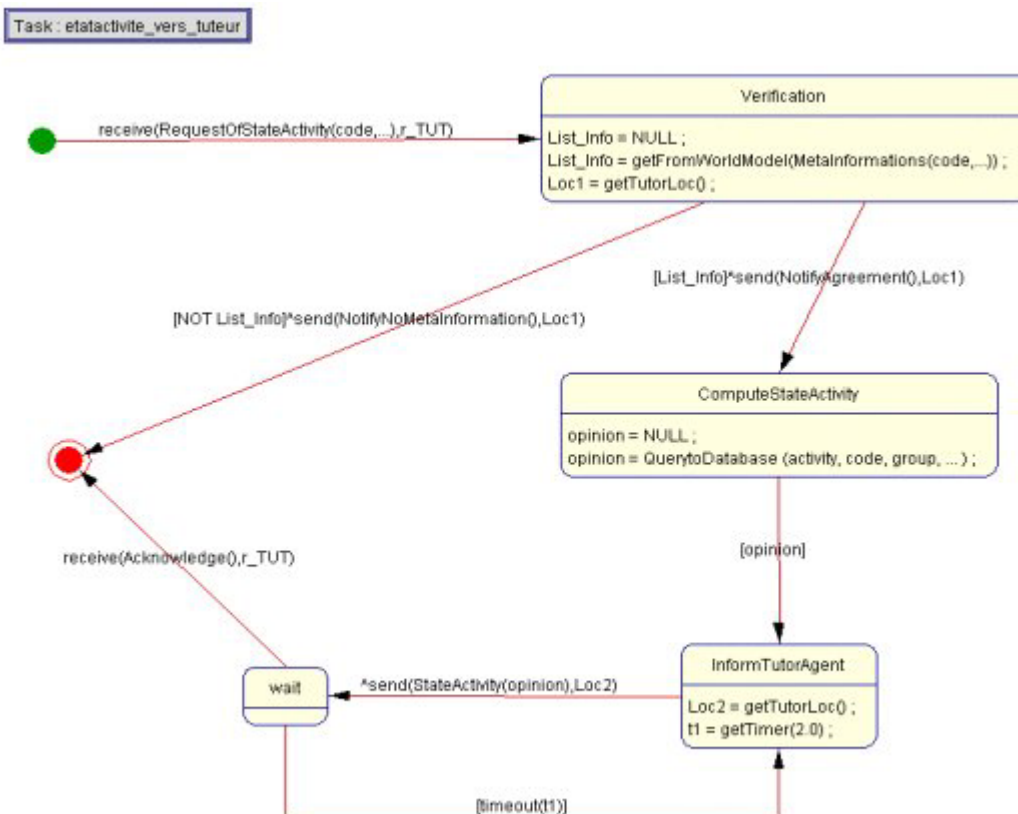


Figure B.14 : Etats et transitions de r_SI dans l'indication de l'état de réalisation d'une activité

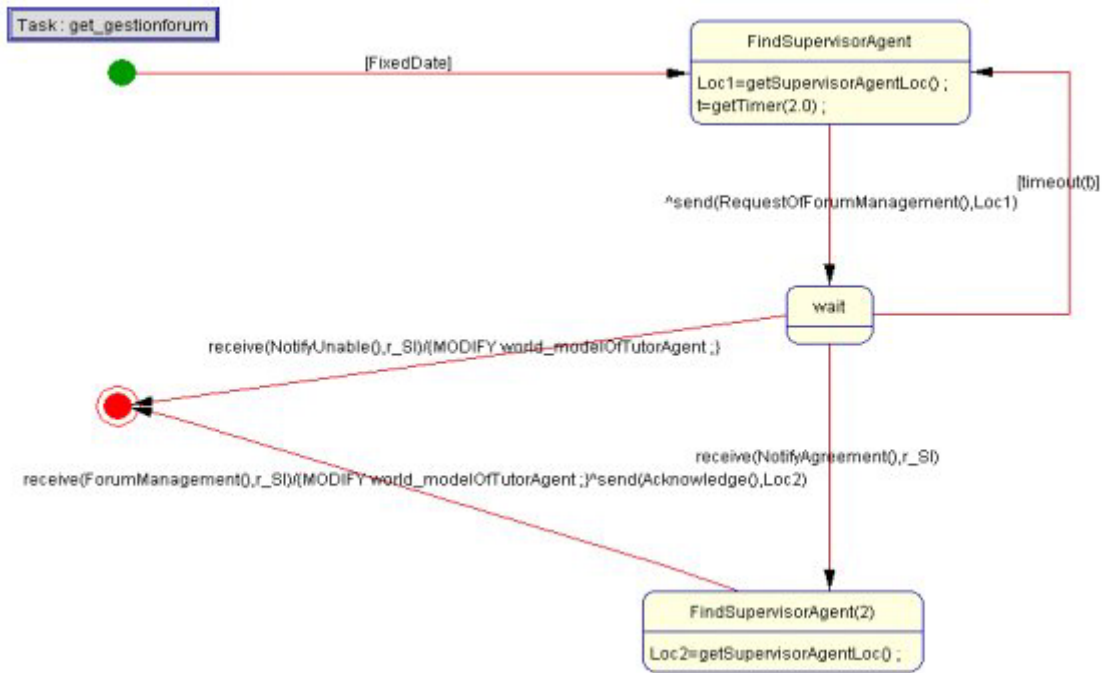


Figure B.15 : Etats et transitions de r_TUT pour la demande et la réception de l'avis de gestion des forums

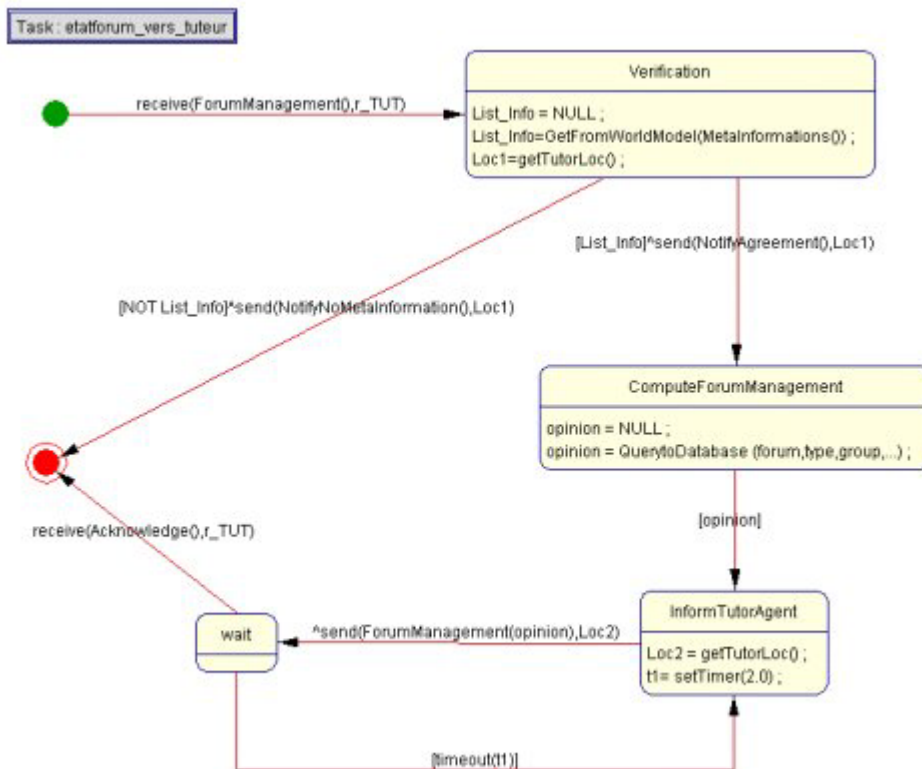


Figure B.16 : Etats et transitions de r_SI lors de la fourniture de l'avis de gestion des forums

C – Application de SIGFAD aux données de SimuLigne

Nous présentons ici deux tableaux montrant les résultats fournis par SIGFAD pour les deux groupes de base Gallia et Narbonensis. Nous avons présenté sur les tableaux 5.1 et 5.2, les mêmes résultats pour les groupes Aquitania et Lugdunensis.

Période	Gallia			
	Présents	Absents	Dormants	Etat du groupe
02 avr. – 22 avr. PREMJ1 PREMJ2	Tim Ben Mike Alan Arthur Erica Babeth Eliane John	<i>Pas d'absents sur cette période</i>	<i>Pas de dormants sur cette période</i>	<i>Très Dynamique</i> <u>9 apprenants actifs</u> Tim Ben Mike Alan Arthur Erica Babeth Eliane John
29 avr. – 06 mai E0A1J1 E0A1J2 E0A2J1	Ben Mike Alan Arthur Erica Babeth Eliane John	Tim	Mike	<i>Dynamique</i> <u>9 apprenants actifs</u> Tim Ben Mike Alan Arthur Erica Babeth Eliane John
30 avril-13 mai E0A3 E0A4J1 E0A4J2 E0A5 INTER0	Tim Ben Mike Alan Arthur Erica Babeth Eliane John	<i>Pas d'absent sur cette période</i>	<i>Pas de dormants sur cette période</i>	<i>Dynamique</i> <u>6 apprenants actifs</u> Tim Ben Alan Arthur Eliane John
14 mai - 20 mai I1P1	Tim Ben Mike Alan Arthur Erica Babeth John	Eliane	<i>Pas de dormants sur cette période</i>	<i>Peu dynamique</i> <u>3 apprenants actifs</u> Tim Mike Alan
22 mai - 03 juin I1P3 E1A3J2	Tim Ben Mike Alan Arthur Erica John	Babeth Eliane	<i>Pas de dormants sur cette période</i>	<i>Dynamique</i> <u>6 apprenants actifs</u> Tim Ben Mike Alan Arthur Erica

Tableau C.1 : Quelques résultats fournies par l'agent du tuteur de Gallia

	Narbonensis			
Période	Présents	Absents	Dormants	Etat du groupe
02 avr. – 22 avr. PREMJ1 PREMJ2	Lucia Tony Bill Jacques Enrico Bertha Julia Alisson Nelly	Joe	<i>Pas de dormants sur cette période</i>	Très Dynamique. <u>8 apprenants actifs</u> Lucia Bill Jacques Enrico Bertha Julia Alisson Nelly
29 avr. – 06 mai E0A1J1 E0A1J2 E0A2J1	Lucia Tony Bill Jacques Enrico Julia Alisson Nelly	Bertha Joe	<i>Pas de dormants sur cette période</i>	Assez dynamique <u>8 apprenants actifs</u> Lucia Bill Jacques Enrico Alisson
30 avril-13 mai E0A3 E0A4J1 E0A4J2 E0A5 INTER0	Lucia Tony Bill Jacques Enrico Bertha Julia Alisson Nelly Joe	<i>Pas d'absent sur cette période</i>	<i>Pas de dormants sur cette période</i>	Assez dynamique <u>4 apprenants actifs</u> Lucia Bill Enrico Kelly
14 mai - 20 mai I1P1	Lucia Tony Bill Enrico Alisson Nelly Joe	Jacques Bertha Julia	Nelly	Peu dynamique <u>2 apprenants actifs</u> Bill Kelly
22 mai - 03 juin I1P3 E1A3J2	Lucia Bill Enrico Alisson Joe	Tony Jacques Bertha Julia Nelly	<i>Pas de dormants sur cette période</i>	Assez dynamique <u>3 apprenants actifs</u> Lucia Bill Alisson

Tableau C.2 : Quelques résultats fournis par l'agent du tuteur Narbonensis

D – Code source de SIGFAD

Nous présentons le code source de quelques-uns des différents programmes constituant SIGFAD. Il est impossible de présenter le code de tous les programmes dans le cadre de ce mémoire. En particulier, nous présentons le code de l'agent superviseur des interactions, le code d'un exemple d'interface agent-utilisateur : l'interface de l'agent du tuteur Aquitania. Nous présentons ensuite trois exemples de primitives écrites et intégrées à l'environnement JAM pour permettre le bon fonctionnement de SIGFAD. La première primitive est celle par laquelle l'agent superviseur des interactions reçoit une requête de l'agent du tuteur. La deuxième primitive est écrite pour extraire la liste des absents de la base de données des interactions. Enfin, la dernière primitive indique comment le résultat de la requête est

transmis de l'agent superviseur des interactions à l'agent du tuteur. Signalons simplement que nous avons écrit une cinquantaine de primitives pour assurer les fonctions de SIGFAD.

D – 1 Code source de l'agent superviseur des interactions

```
// -*- Mode: Java -*-
GOALS:
  PERFORM requete_tutors;
FACTS:
  // Membres du Groupe Aquitania
  FACT IN "Aquitania" "boillon" "natif";
  FACT IN "Aquitania" "broa" "apprenant";
  FACT IN "Aquitania" "concepteur" "tuteur";
  FACT IN "Aquitania" "dornier" "natif";
  FACT IN "Aquitania" "elliott" "apprenant";
  FACT IN "Aquitania" "farnell" "apprenant";
  FACT IN "Aquitania" "gardner" "apprenant";
  FACT IN "Aquitania" "giejgo" "apprenant";
  FACT IN "Aquitania" "grinan" "natif";
  FACT IN "Aquitania" "mannion" "apprenant";
  FACT IN "Aquitania" "maxhuni" "apprenant";
  FACT IN "Aquitania" "millar" "apprenant";
  FACT IN "Aquitania" "parsons" "apprenant";
  FACT IN "Aquitania" "stacy" "apprenant";
  FACT IN "Aquitania" "vetter" "tuteur";
  // Membres du Groupe Gallia
  FACT IN "Gallia" "eardley" "tuteur";
  FACT IN "Gallia" "garces" "apprenant";
  FACT IN "Gallia" "gatehouse" "apprenant";
  FACT IN "Gallia" "genre" "natif";
  FACT IN "Gallia" "green" "apprenant";
  FACT IN "Gallia" "james" "apprenant";
  FACT IN "Gallia" "jamfrey" "apprenant";
  FACT IN "Gallia" "leraitre" "natif";
  FACT IN "Gallia" "malecki" "apprenant";
  FACT IN "Gallia" "morran" "apprenant";
  FACT IN "Gallia" "spencer" "apprenant";
  FACT IN "Gallia" "vergere" "apprenant";
  // Membres du Groupe Lugdunensis
  FACT IN "Lugdunensis" "atkinson" "apprenant";
  FACT IN "Lugdunensis" "bialkowicz" "natif";
  FACT IN "Lugdunensis" "evans" "apprenant";
  FACT IN "Lugdunensis" "concepteur" "tuteur";
  FACT IN "Lugdunensis" "ferraris" "natif";
  FACT IN "Lugdunensis" "jeffs" "apprenant";
  FACT IN "Lugdunensis" "kemp" "apprenant";
  FACT IN "Lugdunensis" "macnaughtan" "apprenant";
  FACT IN "Lugdunensis" "morgans" "apprenant";
  FACT IN "Lugdunensis" "piva" "natif";
  FACT IN "Lugdunensis" "probert" "tuteur";
  FACT IN "Lugdunensis" "seymour" "apprenant";
  FACT IN "Lugdunensis" "snow" "apprenant";
  FACT IN "Lugdunensis" "wilson" "apprenant";
  // Membres du Groupe Narbonensis
  FACT IN "Narbonensis" "bermont" "natif";
  FACT IN "Narbonensis" "booker" "apprenant";
  FACT IN "Narbonensis" "duxburry" "apprenant";
  FACT IN "Narbonensis" "graham" "apprenant";
  FACT IN "Narbonensis" "grattidge" "apprenant";
  FACT IN "Narbonensis" "jones" "apprenant";
  FACT IN "Narbonensis" "martin" "apprenant";
  FACT IN "Narbonensis" "pickering" "apprenant";
  FACT IN "Narbonensis" "pourchot" "natif";
  FACT IN "Narbonensis" "powell" "apprenant";
  FACT IN "Narbonensis" "price" "apprenant";
  FACT IN "Narbonensis" "smith" "apprenant";
  FACT IN "Narbonensis" "willis" "tuteur";
  //Echéancier des activités et Jalons
```


FACT JALON "PREMJ1" "2001-04-02" "2001-04-22" "groupe" "1" "Forum(principal)";
 FACT JALON "PREMJ2" "2001-04-02" "2001-04-22" "apprenant" "1" "Questionnaire(OIP)";
 FACT JALON "E0A1J1" "2001-04-29" "2001-05-06" "groupe" "1" "Forum(Principal)";
 FACT JALON "E0A1J2" "2001-04-29" "2001-05-06" "groupe" "1" "Forum(E0A3_RDV_bavardage)";
 FACT JALON "E0A2J1" "2001-04-29" "2001-05-04" "apprenant" "1" "AR(biblio)";
 FACT JALON "E0A2J2" "2001-04-29" "2001-05-07" "tuteur" "1" "courriel(biographie)";
 FACT JALON "E0A3" "2001-05-07" "2001-05-12" "apprenant" "1" "Bavardage(e0a3)";
 FACT JALON "E0A4J1" "2001-05-07" "2001-05-11" "apprenant" "1" "Plublication(biographie)";
 FACT JALON "E0A4J2" "2001-05-07" "2001-05-11" "apprenant" "1" "Plublication(photo)";
 FACT JALON "E0A5" "2001-05-07" "2001-05-11" "apprenant" "1" "AR(oracle)";
 FACT JALON "E1A1J1" "2001-05-11" "2001-05-18" "apprenant" "1" "AR(e1a1_villes_francaises)";
 FACT JALON "E1A1J2" "2001-05-11" "2001-05-21" "tuteur" "0" "Courriel(participant)";
 FACT JALON "E1A1J3" "2001-05-18" "2001-05-21" "tuteur" "0" "Publication(e1a1_villes_francaises)";
 FACT JALON "E1A2J1" "2001-05-11" "2001-05-15" "groupe" "1" "Forum(e1a2_choisir_theme)";
 FACT JALON "E1A2J2" "2001-05-11" "2001-05-23" "apprenant" "1" "AR(e1a2_lieux)";
 FACT JALON "E1A2J3" "2001-05-23" "2001-05-25" "tuteur" "0" "Publication(e1_ville)";
 FACT JALON "E1A3J1" "2001-05-11" "2001-05-24" "groupe" "1" "Forum(e1a3_RDV_bavardage)";
 FACT JALON "E1A3J2" "2001-05-28" "2001-06-01" "groupe" "1" "Bavardage(e1a3)";
 FACT JALON "E1A3J3" "2001-06-02" "2001-06-04" "technicien" "0" "Publication(e1_ville)";
 FACT JALON "E1A4J1" "2001-05-11" "2001-05-25" "apprenant" "1" "AR(e1a4_fiche)";
 FACT JALON "E1A4J2" "2001-05-11" "2001-05-25" "tuteur" "0" "Publications(e1a4_fiche)";
 FACT JALON "E1A5J1" "2001-05-11" "2001-05-24" "apprenant" "1" "Forum(e1a5_choix_fonction)";
 FACT JALON "E1A5J2" "2001-05-11" "2001-05-30" "apprenant" "1" "Publications(e1a5_fonction)AR(e1a5_fonction)";
 FACT JALON "E1A5J3" "2001-05-11" "2001-06-01" "tuteur" "0" "Courriel(e1a5_fonction)";
 FACT JALON "E1A5J4" "2001-05-30" "2001-06-01" "tuteur" "0" "Publications(e1_identites)";
 FACT JALON "E2A1J1" "2001-06-01" "2001-06-06" "apprenant" "1" "AR(e2a1_description)";
 FACT JALON "E2A1J2" "2001-06-06" "2001-06-10" "apprenant" "1" "Questionnaire(e2a1_description)";
 FACT JALON "E2A1J3" "2001-06-07" "2001-06-07" "tuteur" "0" "Publication(e2a1_description)";
 FACT JALON "E2A2J1" "2001-06-01" "2001-06-10" "tuteur" "0" "Courriel(sous_groupe)";
 FACT JALON "E2A2J2" "2001-06-01" "2001-06-11" "apprenant" "1" "Courriel(sous_groupe)";
 FACT JALON "E2A2J3" "2001-06-11" "2001-06-20" "groupe" "1" "Forum(e2a2_hymne)AR(e2a2_recit)";
 FACT JALON "E2A2J4" "2001-06-11" "2001-06-18" "tuteur" "0" "Publications(e2_notre_ville)";
 FACT JALON "E2A2J5" "2001-06-18" "2001-06-22" "apprenant" "1" "Questionnaire(e2a2_recit)";
 FACT JALON "E2A3J1" "2001-06-02" "2001-06-08" "apprenant" "1" "Courriel(premier)";
 FACT JALON "E2A3J2" "2001-06-02" "2001-06-15" "apprenant" "1" "Courriel(deuxième)";
 FACT JALON "E2A4J1" "2001-06-11" "2001-06-15" "apprenant" "1" "Courriel(binome)";
 FACT JALON "E2A4J2" "2001-06-11" "2001-06-15" "apprenant" "1" "Forum(e2a4_RDV_bavardage)";
 FACT JALON "E2A4J3" "2001-06-18" "2001-06-21" "apprenant" "1"
 "Bavardage(e2a4_cult/e2a4_vente/e2a4_menu/e2a4_invit)";
 FACT JALON "E2A5J1" "2001-06-01" "2001-06-21" "apprenant" "1" "Forum(e2a5_incidents)";
 FACT JALON "E2A5J2" "2001-06-23" "2001-06-23" "tuteur" "0" "Publications(e2_interactions)";
 FACT JALON "E3A1J1" "2001-06-27" "2001-06-27" "technicien" "0" "Publication(e3_villes_candidates)";
 FACT JALON "E3A1J2" "2001-06-27" "2001-07-01" "apprenant" "1" "Questionnaire(e3_vote_final)";
 FACT JALON "E3A1J3" "2001-06-27" "2001-07-06" "apprenant" "1" "Forum(Principal/Principal_Monde)";
 FACT JALON "E3A2J1" "2001-07-02" "2001-07-06" "apprenant" "0" "Publication(e3_resultats)";
 FACT JALON "E3A2J2" "2001-07-02" "2001-07-06" "apprenant" "1" "Forum(Principal/Principal_Monde)";
 FACT JALON "E3A3J1" "2001-07-02" "2001-07-06" "apprenant" "1" "Questionnaire(e3_bilan)";
 FACT JALON "E3A3J2" "2001-07-02" "2001-07-06" "apprenant" "1" "Forum(Principal/e3_bilan_Monde)";
 FACT JALON "INTER0" "2001-04-30" "2001-05-13" "Monde" "1" "Forum(InterCulture_prep)";
 FACT JALON "I1P1" "2001-05-14" "2001-05-20" "Monde" "1" "Formulaire(mots)";
 FACT JALON "I1P3" "2001-05-22" "2001-06-03" "Monde" "1" "Forums(2,3)";
 FACT JALON "I2P1" "2001-06-11" "2001-06-17" "Monde" "1" "Formulaire(phrases_situations)";
 FACT JALON "I2P3" "2001-06-19" "2001-07-01" "Monde" "1" "Forums(2,3)";
 //Énumération des jalons concernant chaque activité
 FACT COMPREND "PREM" "PREMJ1" "PREMJ2";
 FACT COMPREND "E0A1" "E0A1J1" "E0A1J2";
 FACT COMPREND "E0A3" "E0A3";
 FACT COMPREND "E0A4" "E0A4J1" "E0A4J2";
 FACT COMPREND "E0A5" "E0A5";
 FACT COMPREND "E1A1" "E1A1J1" "E1A1J2" "E1A1J3";
 FACT COMPREND "E1A2" "E1A2J1" "E1A2J2" "E1A2J3";
 FACT COMPREND "E1A3" "E1A3J1" "E1A3J2" "E1A3J3";
 FACT COMPREND "E1A4" "E1A4J1" "E0A4J2";
 FACT COMPREND "E1A5" "E1A5J1" "E0A5J2" "E1A5J3" "E0A5J4";
 FACT COMPREND "E2A1" "E2A1J1" "E2A1J2" "E2A1J3";
 FACT COMPREND "E2A2" "E2A2J1" "E2A2J2" "E2A2J3" "E2A2J4" "E2A2J5";
 FACT COMPREND "E2A3" "E2A3J1" "E2A3J2";
 FACT COMPREND "E2A4" "E2A4J1" "E2A4J2" "E2A4J3";
 FACT COMPREND "E2A5" "E2A5J1" "E2A5J2";
 FACT COMPREND "E3A1" "E3A1J1" "E3A1J2" "E3A1J3";
 FACT COMPREND "E3A2" "E3A2J2";
 FACT COMPREND "E3A3" "E3A3J1" "E3A3J2";
 FACT COMPREND "Inter0" "Inter0";
 FACT COMPREND "I1P1" "I1P1";
 FACT COMPREND "I1P3" "I1P3";

```

FACT COMPREND "I2P1" "I2P1";
FACT COMPREND "I2P3" "I2P3";
FACT initialized "False";
Plan: {
NAME: "Top-level plan"
GOAL:
  PERFORM requete_tutors;
CONTEXT:
BODY:
  //Reservation des ports
  ASSIGN $port_server 8080; // c'est le port sur lequel est lancé l'agent Superviseur
  ASSIGN $port_aqui 8090; // c'est le port de l'agent Tuteur de Aquitania
  ASSIGN $port_gall 8010; // c'est le port de l'agent Tuteur de Gallia
  ASSIGN $port_Lugdu 8020; // c'est le port de l'agent Tuteur de Lugdunensis
  ASSIGN $port_narbo 8030; // c'est le port de l'agent Tuteur de Narbonensis
  ASSIGN $port_coord 8040; // c'est le port de l'agent Coordonnateur
  ASSIGN $type_present "Presents";
  ASSIGN $type_absent "Absents";
  ASSIGN $type_dormant "Dormants";
  ASSIGN $type_groupe "Etat du groupe";
  ASSIGN $type_indiv "Etat individu";
  ASSIGN $type_socia "Sociabilite";
  ASSIGN $type_activite "Niveau activite";
  DO
  {
  // Reception d'une requête venant d'un tuteur
  EXECUTE print "Waiting as Server for tutors' connections.\n";
  EXECUTE com.irs.jam.primitives.ReceiveQueryTutor.execute $port_server $port_groupe $host_client $Group $Date1
  $Date2 $type;

  // Traitement de la requête reçue
  EXECUTE print "Traitement de la requete en cours...\n";
  WHEN : TEST (== $type $type_present)
  {
  EXECUTE com.irs.jam.primitives.PresentsInGroup.execute $Group $Date1 $Date2 $resultat;
  };

  WHEN : TEST (== $type $type_absent)
  {
  EXECUTE com.irs.jam.primitives.AbsentsInGroup.execute $Group $Date1 $Date2 $resultat;
  };

  WHEN : TEST (== $type $type_dormant)
  {
  EXECUTE com.irs.jam.primitives.DormantsInGroup.execute $Group $Date1 $Date2 $resultat;
  };

  WHEN : TEST (== $type $type_socia)
  {
  //Recuperation du groupe de l'apprenant, $Group contient le nom de l'individu, $Groupe le nom du groupe
  EXECUTE com.irs.jam.primitives.SocialIndividu.execute $Group $Date1 $Date2 $resultat;
  };

  // Fourniture de l'état de productivite d'un individu
  WHEN : TEST (== $type $type_indiv)
  {
  // Récuperer les jalons et activités à la date de la requête
  RETRIEVEALL $ONFACTS JALON $jalon $debut $fin $concernes $alerte $outil;
  ASSIGN $i 0; // $i compte le nombre de jalons voir la relation JALON
  ASSIGN $jalon_revolus " ";
  ASSIGN $jalon_possibles " ";
  DO
  {
  ASSIGN $i (+ $i 1);
  NEXTFACT $ONFACTS JALON $jalon $debut $fin $concernes $alerte $outil;
  ASSIGN $nom_jalon $jalon;
  ASSIGN $date_debut $debut;
  ASSIGN $date_fin $fin;
  ASSIGN $pers_concerne $concernes;
  ASSIGN $indic_alerte $alerte;
  ASSIGN $nom_outil $outil;
  //La variable $Group contient le nom de l'individu et $Groupe le nom du groupe
  FACT IN $Groupe $Group $statut;
  ASSIGN $jalon0 $jalon_revolus;
  ASSIGN $jalon1 $jalon_possibles;

```

```

//Traitement pour les apprenants
WHEN : TEST (== $statut "apprenant")
{
  WHEN : TEST (|| (== $pers_concerne "apprenant") (== $pers_concerne "groupe") (== $pers_concerne "Monde"))
  {
    //Renvoi la liste des jalons concernées sur la période
    EXECUTE com.irs.jam.primitives.JalonsConcernees.execute $Group $Date1 $Date2 $date_debut $date_fin $nom_jalon
$jalon0 $jalon1
    $jalon_revolus $jalon_possibles;
  };
};
// Traitement pour le Tuteur
WHEN : TEST (== $statut "tuteur")
{
  WHEN : TEST (|| (== $pers_concerne "tuteur") (== $pers_concerne "groupe") (== $pers_concerne "Monde"))
  {
    EXECUTE com.irs.jam.primitives.JalonsConcernees.execute $Group $Date1 $Date2 $date_debut $date_fin $nom_jalon
$jalon0 $jalon1
    $jalon_revolus $jalon_possibles;
  };
};

} WHILE : TEST (< $i 52);
//$Groupe contient le nom de l'individu et $Groupe est le groupe ...
EXECUTE com.irs.jam.primitives.PointIndividu.execute $Group $Groupe $Date1 $Date2 $jalon_revolus $jalon_possibles
$resultat;
};
// Debut procédure de fourniture de l'état d'un groupe
WHEN : TEST (== $type $type_groupe)
{
  EXECUTE print "Debut Procedure de Etat du groupe.\n";
  RETRIEVEALL $ONFACTS JALON $jalon $debut $fin $concernees $alerte $outil;
  ASSIGN $i 0; // $i compte le nombre de jalons voir la relation JALON
  ASSIGN $JalonsRev " ";
  DO
  {
    ASSIGN $i (+ $i 1);
    NEXTFACT $ONFACTS JALON $jalon $debut $fin $concernees $alerte $outil;
    ASSIGN $nom_jalon $jalon;
    ASSIGN $date_debut $debut;
    ASSIGN $date_fin $fin;
    ASSIGN $pers_concerne $concernees;
    ASSIGN $indic_alerte $alerte;
    ASSIGN $nom_outil $outil;
    ASSIGN $jalon0 $JalonsRev;
    // Ne prendre que les jalons pour les apprenants
    WHEN : TEST (|| (== $pers_concerne "apprenant") (== $pers_concerne "groupe") (== $pers_concerne "Monde"))
    {
      EXECUTE com.irs.jam.primitives.JalonsGroupeConcernees.execute $Date1 $Date2 $date_debut $date_fin $nom_jalon
$jalon0 $JalonsRev;
    };
  } WHILE : TEST (< $i 52);
  EXECUTE com.irs.jam.primitives.EtatGroupe.execute $Group $JalonsRev $resultat;
};
// Fin procédure de fourniture de l'état d'un groupe
//Debut procedure fourniture niveau d'exécution d'une activité
WHEN : TEST (== $type $type_activite)
{
  ASSIGN $JalonsRevolus "";
  ASSIGN $JalonsPossibles "";
  ASSIGN $activite $Group;
  FACT COMPREND $activite $jal1 $jal2 $jal3 $jal4 $jal5;
  ASSIGN $jalon1 $jal1;
  ASSIGN $jalon2 $jal2;
  ASSIGN $jalon3 $jal3;
  ASSIGN $jalon4 $jal4;
  ASSIGN $jalon5 $jal5;
  RETRIEVEALL $ONFACTS JALON $jalon $debut $fin $concernees $alerte $outil;
  ASSIGN $j 0; // $j compte le nombre de jalons voir la relation JALON
  DO
  {
    ASSIGN $j (+ $j 1);
    NEXTFACT $ONFACTS JALON $jalon $debut $fin $concernees $alerte $outil;
    ASSIGN $nom_jalon $jalon;
    ASSIGN $DateDebut $debut;
  }
};

```

```

ASSIGN $DateFin $fin;
ASSIGN $pers_concernes $concernes;
ASSIGN $pers_alerte $alerte;
ASSIGN $nom_outil $outil;
WHEN : TEST (&& (== $jalon1 $nom_jalon) (!= $pers_concernes "tuteur") (!= $pers_concernes "technicien") )
{
  //comparer $Date1 avec $DateDebut et $DateFin
  EXECUTE com.irs.jam.primitives.PlusrecenteDate.execute $DateFin $Date1 $res;
  WHEN : TEST (== $res 1)
  {
    ASSIGN $JalonsRevolus (+ $JalonsRevolus $jalon1 " ");
  };
  WHEN : TEST (== $res 0)
  {
    EXECUTE com.irs.jam.primitives.PlusrecenteDate.execute $DateDebut $Date1 $res;
    WHEN : TEST (== $res 1)
    {
      ASSIGN $JalonsPossibles (+ $JalonsPossibles $jalon1 " ");
    };
  };
};
WHEN : TEST (&& (== $jalon2 $nom_jalon) (!= $pers_concernes "tuteur") (!= $pers_concernes "technicien") )
{
  //comparer $Date1 avec $DateDebut et $DateFin
  EXECUTE com.irs.jam.primitives.PlusrecenteDate.execute $DateFin $Date1 $res;
  WHEN : TEST (== $res 1)
  {
    ASSIGN $JalonsRevolus (+ $JalonsRevolus $jalon2 " ");
  };
  WHEN : TEST (== $res 0)
  {
    EXECUTE com.irs.jam.primitives.PlusrecenteDate.execute $DateDebut $Date1 $res;
    WHEN : TEST (== $res 1)
    {
      ASSIGN $JalonsPossibles (+ $JalonsPossibles $jalon2 " ");
    };
  };
};
WHEN : TEST (&& (== $jalon3 $nom_jalon) (!= $pers_concernes "tuteur") (!= $pers_concernes "technicien") )
{
  //comparer $Date1 avec $DateDebut et $DateFin
  EXECUTE com.irs.jam.primitives.PlusrecenteDate.execute $DateFin $Date1 $res;
  WHEN : TEST (== $res 1)
  {
    ASSIGN $JalonsRevolus (+ $JalonsRevolus $jalon3 " ");
  };
  WHEN : TEST (== $res 0)
  {
    EXECUTE com.irs.jam.primitives.PlusrecenteDate.execute $DateDebut $Date1 $res;
    WHEN : TEST (== $res 1)
    {
      ASSIGN $JalonsPossibles (+ $JalonsPossibles $jalon3 " ");
    };
  };
};
WHEN : TEST (&& (== $jalon4 $nom_jalon) (!= $pers_concernes "tuteur") (!= $pers_concernes "technicien") )
{
  //comparer $Date1 avec $DateDebut et $DateFin
  EXECUTE com.irs.jam.primitives.PlusrecenteDate.execute $DateFin $Date1 $res;
  WHEN : TEST (== $res 1)
  {
    ASSIGN $JalonsRevolus (+ $JalonsRevolus $jalon4 " ");
  };
  WHEN : TEST (== $res 0)
  {
    EXECUTE com.irs.jam.primitives.PlusrecenteDate.execute $DateDebut $Date1 $res;
    WHEN : TEST (== $res 1)
    {
      ASSIGN $JalonsPossibles (+ $JalonsPossibles $jalon4 " ");
    };
  };
};
WHEN : TEST (&& (== $jalon5 $nom_jalon) (!= $pers_concernes "tuteur") (!= $pers_concernes "technicien") )
{
  //comparer $Date1 avec $DateDebut et $DateFin

```

```

EXECUTE com.irs.jam.primitives.PlusrecenteDate.execute $DateFin $Date1 $res;
WHEN : TEST (== $res 1)
{
  ASSIGN $JalonsRevolus (+ $JalonsRevolus $jalon5 " ");
};
WHEN : TEST (== $res 0)
{
  EXECUTE com.irs.jam.primitives.PlusrecenteDate.execute $DateDebut $Date1 $res;
  WHEN : TEST (== $res 1)
  {
    ASSIGN $JalonsPossibles (+ $JalonsPossibles $jalon5 " ");
  };
};
};
} WHILE : TEST (< $j 52); //52 représente le nombre de jalons actuels
//Donner l'état de réalisation de l'activité
EXECUTE com.irs.jam.primitives.EtatActivite.execute $activite $port_groupe $JalonsRevolus $JalonsPossibles $resultat;
}; //Fin procédure de fourniture niveau d'exécution d'une activité
//Se connecte sur le tuteur comme client pour lui envoyer le résultat de sa requête
EXECUTE com.irs.jam.primitives.SendResultToTutor.execute $port_groupe $host_client $resultat $type;
} WHILE : TEST (== 1 1);
FAILURE:
EXECUTE println "Plan failed!";
}

```

D.2 - Exemple de code source d'un agent tuteur : L'agent du tuteur du groupe Aquitania

```

// -*- Mode: Java -*-
import java.sql.*;
import java.io.*;
import java.net.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.awt.*;
import java.util.*;
//import com.irs.jam.*;
public class agtAquitania extends JFrame {
  static agtAquitania maFenetre;
  JTextArea result_area = new JTextArea (3,30);
  JScrollPane scrollPane = new JScrollPane (result_area);
  JTextField portserver_tx = new JTextField ("8080");
  JTextField portclient_tx = new JTextField ("8090");
  JTextField client_tx = new JTextField ("localhost");
  JTextField hote_tx = new JTextField ("pcjn.univ-fcomte.fr");
  JTextField date1_tx = new JTextField ();
  JTextField date2_tx = new JTextField ();
  Choice nom_tx = new Choice();
  Choice groupe_tx = new Choice();
  Choice activite_tx = new Choice();
  Choice typeI_tx = new Choice ();
  Choice typeG_tx = new Choice ();
  JTextField typeA_tx = new JTextField ("Niveau activite");
  JLabel L5 = new JLabel("Date de début");
  JLabel L6 = new JLabel("Date de fin");
  JLabel L7 = new JLabel("Nom");
  JLabel L8 = new JLabel("Requête sur individu");
  JLabel L9 = new JLabel("Groupe");
  JLabel L10 = new JLabel("Requête sur Groupe");
  JLabel L11 = new JLabel("Activité");
  JLabel L12 = new JLabel("Requête sur Activité");
  JButton B1 = new JButton("Envoyer Individu");
  JButton B2 = new JButton("Envoyer Groupe");
  JButton B3 = new JButton("Envoyer Activité");
  JButton B4 = new JButton("Annuler");
  ButtonGroup Onglet = new ButtonGroup();

```

```

JRadioButton BoutonGroupe = new JRadioButton("Requete sur Groupe",false);
JRadioButton BoutonApprenant = new JRadioButton("Requete sur Apprenant",false);
JRadioButton BoutonActivite = new JRadioButton("Requete sur Activite",false);
agtAquitania ()
{
    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e)
        {
            System.exit(0);
        }
    }
);
getContentPane().setLayout(new BorderLayout());
JPanel panOnglet = new JPanel();
panOnglet.setBackground(Color.lightGray);
Onglet.add(BoutonGroupe);
Onglet.add(BoutonApprenant);
Onglet.add(BoutonActivite);
panOnglet.add(BoutonGroupe);
panOnglet.add(BoutonApprenant);
panOnglet.add(BoutonActivite);
Border etched = BorderFactory.createEtchedBorder();
Border titled = BorderFactory.createTitledBorder(etched, "Indiquer sur quoi porte votre requête SVP");
panOnglet.setBorder(titled);
getContentPane().add("North",panOnglet);
JPanel panRequete = new JPanel();
    GridBagLayout gridbag = new GridBagLayout();
    GridBagConstraints constraints = new GridBagConstraints();
panRequete.setLayout(gridbag);
    JLabel L1 = new JLabel("Port Serveur");
    buildConstraints (constraints,0,0,1,1,10,30);
    constraints.fill=GridBagConstraints.NONE;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (L1, constraints);
    panRequete.add(L1);
L1.setVisible(false);
    buildConstraints (constraints,1,0,1,1,0,0);
    constraints.fill=GridBagConstraints.HORIZONTAL;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (portserver_tx, constraints);
    panRequete.add(portserver_tx);
portserver_tx.setVisible(false);
    JLabel L3 = new JLabel("Nom Serveur");
    buildConstraints (constraints,2,0,1,1,10,30);
    constraints.fill=GridBagConstraints.NONE;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (L3, constraints);
    panRequete.add(L3);
L3.setVisible(false);
    buildConstraints (constraints,3,0,1,1,0,0);
    constraints.fill=GridBagConstraints.HORIZONTAL;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (hote_tx, constraints);
    panRequete.add(hote_tx);
hote_tx.setVisible(false);
    JLabel L2 = new JLabel("Port Client");
    buildConstraints (constraints,0,1,1,1,10,30);
    constraints.fill=GridBagConstraints.NONE;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (L2, constraints);
    panRequete.add(L2);
L2.setVisible(false);
    buildConstraints (constraints,1,1,1,1,0,0);
    constraints.fill=GridBagConstraints.HORIZONTAL;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (portclient_tx, constraints);
    panRequete.add(portclient_tx);
portclient_tx.setVisible(false);
    JLabel L4 = new JLabel("Nom Client");
    buildConstraints (constraints,2,1,1,1,10,30);
    constraints.fill=GridBagConstraints.NONE;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (L4, constraints);
    panRequete.add(L4);
L4.setVisible(false);

```

```

        buildConstraints (constraints,3,1,1,1,0,0);
        constraints.fill=GridBagConstraints.HORIZONTAL;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (client_tx, constraints);
        panRequete.add(client_tx);
client_tx.setVisible(false);
        buildConstraints (constraints,0,2,1,1,10,30);
        constraints.fill=GridBagConstraints.NONE;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (L5, constraints);
        panRequete.add(L5);
        buildConstraints (constraints,1,2,1,1,10,30);
        constraints.fill=GridBagConstraints.HORIZONTAL;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (date1_tx, constraints);
        panRequete.add(date1_tx);
        buildConstraints (constraints,2,2,1,1,10,30);
        constraints.fill=GridBagConstraints.NONE;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (L6, constraints);
        panRequete.add(L6);
        buildConstraints (constraints,3,2,1,1,10,30);
        constraints.fill=GridBagConstraints.HORIZONTAL;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (date2_tx, constraints);
        panRequete.add(date2_tx);
        buildConstraints (constraints,0,3,1,1,10,30);
        constraints.fill=GridBagConstraints.NONE;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (L7, constraints);
        panRequete.add(L7);
        buildConstraints (constraints,1,3,1,1,0,0);
        constraints.fill=GridBagConstraints.HORIZONTAL;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (nom_tx, constraints);
        panRequete.add(nom_tx);
        nom_tx.add("broa");
        nom_tx.add("elliott");
        nom_tx.add("farnell");
        nom_tx.add("giejgo");
        nom_tx.add("mannion");
        nom_tx.add("maxhuni");
        nom_tx.add("millar");
        nom_tx.add("parsons");
        nom_tx.add("stacy");
        buildConstraints (constraints,2,3,1,1,10,30);
        constraints.fill=GridBagConstraints.NONE;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (L8, constraints);
        panRequete.add(L8);
        buildConstraints (constraints,3,3,1,1,0,0);
        constraints.fill=GridBagConstraints.NONE;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (type1_tx, constraints);
        panRequete.add(type1_tx);
        type1_tx.add("Etat individu");
        type1_tx.add("Sociabilite");
        buildConstraints (constraints,0,4,1,1,10,30);
        constraints.fill=GridBagConstraints.NONE;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (L9, constraints);
        panRequete.add(L9);
        buildConstraints (constraints,1,4,1,1,0,0);
        constraints.fill=GridBagConstraints.HORIZONTAL;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (groupe_tx, constraints);
        panRequete.add(groupe_tx);
        groupe_tx.add("Aquitania");
        groupe_tx.add("Gallia");
        groupe_tx.add("Lugdunensis");
        groupe_tx.add("Narbonensis");
        groupe_tx.add("Monde");
        buildConstraints (constraints,2,4,1,1,10,30);
        constraints.fill=GridBagConstraints.NONE;
        constraints.anchor=GridBagConstraints.CENTER;

```

```

        gridbag.setConstraints (L10, constraints);
        panRequete.add(L10);
    buildConstraints (constraints,3,4,1,1,0,0);
    constraints.fill=GridBagConstraints.HORIZONTAL;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (typeG_tx, constraints);
    panRequete.add(typeG_tx);
    typeG_tx.add("Presents");
typeG_tx.add("Absents");
typeG_tx.add("Dormants");
typeG_tx.add("Etat du groupe");
        buildConstraints (constraints,0,5,1,1,10,30);
        constraints.fill=GridBagConstraints.NONE;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (L11, constraints);
        panRequete.add(L11);
    buildConstraints (constraints,1,5,1,1,0,0);
    constraints.fill=GridBagConstraints.HORIZONTAL;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (activite_tx, constraints);
    panRequete.add(activite_tx);
    activite_tx.add("PREM");
    activite_tx.add("E0A1");
    activite_tx.add("E0A2");
    activite_tx.add("E0A3");
    activite_tx.add("E0A4");
    activite_tx.add("E0A5");
    activite_tx.add("E1A1");
    activite_tx.add("E1A2");
    activite_tx.add("E1A3");
    activite_tx.add("E1A4");
    activite_tx.add("E1A5");
    activite_tx.add("E2A1");
    activite_tx.add("E2A2");
    activite_tx.add("E2A3");
    activite_tx.add("E2A4");
    activite_tx.add("E2A5");
    activite_tx.add("E3A1");
    activite_tx.add("E3A2");
    activite_tx.add("E3A3");
    activite_tx.add("Inter0");
    activite_tx.add("I1P1");
    activite_tx.add("I1P3");
    activite_tx.add("I2P1");
    activite_tx.add("I2P3");
        buildConstraints (constraints,2,5,1,1,10,30);
        constraints.fill=GridBagConstraints.NONE;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (L12, constraints);
        panRequete.add(L12);
    buildConstraints (constraints,3,5,1,1,0,0);
    constraints.fill=GridBagConstraints.HORIZONTAL;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (typeA_tx, constraints);
    panRequete.add(typeA_tx);
    buildConstraints (constraints,0,6,1,1,10,30);
    constraints.fill=GridBagConstraints.NONE;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (B1, constraints);
    panRequete.add(B1);
    B1.addActionListener(new AL1());
    buildConstraints (constraints,1,6,1,1,10,30);
    constraints.fill=GridBagConstraints.NONE;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (B2, constraints);
    panRequete.add(B2);
    B2.addActionListener(new AL2());
    buildConstraints (constraints,2,6,1,1,10,30);
    constraints.fill=GridBagConstraints.NONE;
    constraints.anchor=GridBagConstraints.CENTER;
    gridbag.setConstraints (B3, constraints);
    panRequete.add(B3);
    B3.addActionListener(new AL3());
    buildConstraints (constraints,3,6,1,1,10,30);
    constraints.fill=GridBagConstraints.NONE;

```



```

        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (B4, constraints);
        panRequete.add(B4);
//Masquer tout au depart
L5.setEnabled(false); //Date de début
date1_tx.setEnabled(false);
L6.setEnabled(false); // date de fin
date2_tx.setEnabled(false);
L7.setEnabled(false); // Nom apprenant
nom_tx.setEnabled(false);
L8.setEnabled(false); // requete sur apprenant
typeI_tx.setEnabled(false);
L9.setEnabled(false); //nom du groupe
groupe_tx.setEnabled(false);
L10.setEnabled(false); // requete sur groupe
typeG_tx.setEnabled(false);
L11.setEnabled(false); // nom activite
activite_tx.setEnabled(false);
L12.setEnabled(false); //requete sur activite
typeA_tx.setEnabled(false);
B1.setEnabled(false); //individu
B2.setEnabled(false); // groupe
B3.setEnabled(false); // activité
B4.addActionListener(new AL4());
        JLabel L13 = new JLabel("Résultat Requête");
        buildConstraints (constraints,0,7,1,1,10,30);
        constraints.fill=GridBagConstraints.NONE;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (L13, constraints);
        panRequete.add(L13);
        buildConstraints (constraints,0,8,4,3,0,0);
        constraints.fill=GridBagConstraints.HORIZONTAL;
        constraints.anchor=GridBagConstraints.CENTER;
        gridbag.setConstraints (result_area, constraints);
panRequete.add(scrollPane, constraints);
setBounds(calculBounds(L1));
        setResizable(true);
setTitle("Agent du Tuteur de Aquitania");
getContentPane().add("Center",panRequete);
setVisible(true);
BoutonGroupe.addActionListener(new OB1());
BoutonApprenant.addActionListener(new OB2());
BoutonActivite.addActionListener(new OB3());
}
class OB1 implements ActionListener
{
    public void actionPerformed (ActionEvent e) {
        L5.setEnabled(true); //Date de début
        date1_tx.setEnabled(true);
        L6.setEnabled(true); // date de fin
        date2_tx.setEnabled(true);
        L7.setEnabled(false); // Nom apprenant
        nom_tx.setEnabled(false);
        L8.setEnabled(false); // requete sur apprenant
        typeI_tx.setEnabled(false);
        L9.setEnabled(true); //nom du groupe
        groupe_tx.setEnabled(true);
        L10.setEnabled(true); // requete sur groupe
        typeG_tx.setEnabled(true);
        L11.setEnabled(false); // nom activite
        activite_tx.setEnabled(false);
        L12.setEnabled(false); //requete sur activite
        typeA_tx.setEnabled(false);
        B1.setEnabled(false); //individu
        B2.setEnabled(true); // groupe
        B3.setEnabled(false); // activité
    }
}
class OB2 implements ActionListener
{
    public void actionPerformed (ActionEvent e) {
        L5.setEnabled(true); //Date de début
        date1_tx.setEnabled(true);
        L6.setEnabled(true); // date de fin
        date2_tx.setEnabled(true);
    }
}

```

```

L7.setEnabled(true); // Nom apprenant
nom_tx.setEnabled(true);
L8.setEnabled(true); // requete sur apprenant
type1_tx.setEnabled(true);
L9.setEnabled(false); //nom du groupe
groupe_tx.setEnabled(false);
L10.setEnabled(false); // requete sur groupe
typeG_tx.setEnabled(false);
L11.setEnabled(false); // nom activite
activite_tx.setEnabled(false);
L12.setEnabled(false); //requete sur activite
typeA_tx.setEnabled(false);
B1.setEnabled(true); //individu
B2.setEnabled(false); // groupe
B3.setEnabled(false); // activité
}
}
class OB3 implements ActionListener
{
    public void actionPerformed (ActionEvent e) {
        L5.setEnabled(true); //Date de début
        date1_tx.setEnabled(true);
        L6.setEnabled(false); // date de fin
        date2_tx.setEnabled(false);
        L7.setEnabled(false); // Nom apprenant
        nom_tx.setEnabled(false);
        L8.setEnabled(false); // requete sur apprenant
        type1_tx.setEnabled(false);
        L9.setEnabled(false); //nom du groupe
        groupe_tx.setEnabled(false);
        L10.setEnabled(false); // requete sur groupe
        typeG_tx.setEnabled(false);
        L11.setEnabled(true); // nom activite
        activite_tx.setEnabled(true);
        L12.setEnabled(true); //requete sur activite
        typeA_tx.setEnabled(true);
        B1.setEnabled(false); //individu
        B2.setEnabled(false); // groupe
        B3.setEnabled(true); // activité
    }
}
void buildConstraints (GridBagConstraints gbc, int gx, int gy, int gw, int gh, int wx, int wy)
{
    gbc.gridx = gx;
    gbc.gridy = gy;
    gbc.gridwidth = gw;
    gbc.gridheight = gh;
    gbc.weightx = wx;
    gbc.weighty = wy;
}
Rectangle calculBounds(JLabel jl)
{
    Dimension tailleEcran = Toolkit.getDefaultToolkit().getScreenSize();
    int largeurEcran = tailleEcran.width;
    int hauteurEcran = tailleEcran.height;
    int largeur = (int)((jl.getPreferredSize()).width + 500);
    int hauteur = (int)((jl.getPreferredSize()).height * 20);
    int xPos = (largeurEcran - largeur) / 2;
    int yPos = (hauteurEcran - hauteur) / 2;
    return new Rectangle(xPos, yPos, largeur, hauteur);
}
class AL1 implements ActionListener
{
    public void actionPerformed (ActionEvent e) {
        String monPortServer = portserver_tx.getText().trim();
        String monPortClient = portclient_tx.getText().trim();
        String monClient = client_tx.getText().trim();
        String monHote = hote_tx.getText().trim();
        String maDate1 = date1_tx.getText().trim();
        String maDate2 = date2_tx.getText().trim();
        String monType1 = type1_tx.getSelectedItem();
        String monNom = nom_tx.getSelectedItem();
        try {
            Integer portS = new Integer (monPortServer);
            int port = portS.intValue();

```

```

        Socket socket = new Socket(monHote,port);
        BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        PrintWriter out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()),true);
        //Envoi des paramètres au Superviseur
        //Nom du port_client, Hote, nom, les dates de début et de fin de période, le type de requete
        out.println(monPortClient);
        String strr = in.readLine();
        out.println(monClient);
        String str0 = in.readLine();
        out.println(monNom);
        String str1 = in.readLine();
        out.println(monDate1);
        String str2 = in.readLine();
        out.println(monDate2);
        String str3 = in.readLine();
        out.println(monType1);
        String str = in.readLine();
        socket.close();
    }
    catch (Exception e1) {
        System.out.println("Erreur *" + e1 + "*" Envoi de la requete au a_SI !");
        e1.printStackTrace();
    }
    // Ici le tuteur devient serveur pour recevoir la liste
    try {
        Integer portC = new Integer (monPortClient);
        int portclient = portC.intValue();
        System.out.println("Le Tuteur devient serveur pour le resultat de sa requete");
        ServerSocket s = new ServerSocket(portclient);
        System.out.println("Started: " + s);
        //Le programme stoppe ici et attend une demande de connexion:
        Socket socket = s.accept();
        BufferedReader in1 = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        PrintWriter out2 = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()),true);
        String $resultat=in1.readLine();
        out2.println("2eme element");
        String $type=in1.readLine();
        out2.println("Le tuteur a bien eu la liste resultant de sa requete");
        socket.close();
        s.close();
        result_area.setText($resultat);
    }
    catch (Exception e2) {
        System.out.println("Erreur *" + e2 + "*" Reception du resultat de la requete!");
        e2.printStackTrace();
    }
}
}
} // Fin de AL1
class AL2 implements ActionListener
{
    public void actionPerformed (ActionEvent e) {
        String monPortServer = portserver_tx.getText().trim();
        String monPortClient = portclient_tx.getText().trim();
        String monClient = client_tx.getText().trim();
        String monHote = hote_tx.getText().trim();
        String maDate1 = date1_tx.getText().trim();
        String maDate2 = date2_tx.getText().trim();
        String monTypeG = typeG_tx.getSelectedItemId();
        String monGroupe = groupe_tx.getSelectedItemId();
        try {
            Integer portS = new Integer (monPortServer);
            int port = portS.intValue();
            Socket socket = new Socket(monHote,port);
            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()),true);
            out.println(monPortClient);
            String strr = in.readLine();
            out.println(monClient);
            String str0 = in.readLine();
            out.println(monGroupe);
            String str1 = in.readLine();
            out.println(monDate1);
            String str2 = in.readLine();
            out.println(monDate2);
            String str3 = in.readLine();

```

```

        out.println(monTypeG);
        String str = in.readLine();
        socket.close();
    }
    catch (Exception e1) {
        System.out.println("Erreur *** + e1 + *** Envoi de la requete au a_SI !");
        e1.printStackTrace();
    }
}
try {
    Integer portC = new Integer (monPortClient);
    int portclient = portC.intValue();
    System.out.println("Le Tuteur devient serveur pour le resultat de sa requete");
    ServerSocket s = new ServerSocket(portclient);
    System.out.println("Started: " + s);
    Socket socket = s.accept();
    BufferedReader in1 = new BufferedReader(new InputStreamReader(socket.getInputStream()));
    PrintWriter out2 = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()),true));
    String $resultat=in1.readLine();
    out2.println("2eme element");
    String $type=in1.readLine();
    out2.println("Le tuteur a bien eu la liste resultant de sa requete");
    socket.close();
    s.close();
    result_area.setText($resultat);
}
catch (Exception e2) {
    System.out.println("Erreur *** + e2 + *** Reception du resultat de la requete!");
    e2.printStackTrace();
}
}
}
} // Fin de AL2
class AL3 implements ActionListener
{
    public void actionPerformed (ActionEvent e) {
        String monPortServer = portserver_tx.getText().trim();
        String monPortClient = portclient_tx.getText().trim();
        String monClient = client_tx.getText().trim();
        String monHote = hote_tx.getText().trim();
        String maDate1 = date1_tx.getText().trim();
        String maDate2 = date2_tx.getText().trim();
        String monTypeA = typeA_tx.getText().trim();
        String monActivite = activite_tx.getSelectedItem();
        try {
            Integer portS = new Integer (monPortServer);
            int port = portS.intValue();
            Socket socket = new Socket(monHote,port);
            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()),true));
            out.println(monPortClient);
            String str = in.readLine();
            out.println(monClient);
            String str0 = in.readLine();
            out.println(monActivite);
            String str1 = in.readLine();
            out.println(maDate1);
            String str2 = in.readLine();
            out.println(maDate2);
            String str3 = in.readLine();
            out.println(monTypeA);
            String str = in.readLine();
            socket.close();
        }
        catch (Exception e1) {
            System.out.println("Erreur *** + e1 + *** Envoi de la requete au a_SI !");
            e1.printStackTrace();
        }
    }
    try {
        Integer portC = new Integer (monPortClient);
        int portclient = portC.intValue();
        System.out.println("Le Tuteur devient serveur pour le resultat de sa requete");
        ServerSocket s = new ServerSocket(portclient);
        System.out.println("Started: " + s);
        Socket socket = s.accept();
        BufferedReader in1 = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        PrintWriter out2 = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()),true));
    }
}

```

```

String $resultat=in1.readLine();
out2.println("2eme element");
String $type=in1.readLine();
out2.println("Le tuteur a bien eu la liste resultat de sa requete");
socket.close();
s.close();
result_area.setText($resultat);
}
catch (Exception e2) {
System.out.println("Erreur *" + e2 + "*" Reception du resultat de la requete!");
e2.printStackTrace();
}
}
}
} // Fin de AL3
class AL4 implements ActionListener
{
public void actionPerformed (ActionEvent e) {
dispose();
System.exit(0);
}
} // Fin de AL4
public static void main( String[] args ) {
maFenetre = new agtAquitania();
}
}

```

D.3 - Code de ReceiveQueryTutor.java

```

package com.irs.jam.primitives;
import java.io.*;
import java.net.*;
import com.irs.jam.*;
public class ReceiveQueryTutor implements PrimitiveAction {
public Value execute(String name, int arity, ExpList args, Binding binding, Goal currentGoal)
{
if (arity != 7)
{
System.out.println("Invalid number of arguments: " + arity + " to function \"" + name + "\"\n");
return Value.FALSE;
}
ExpListEnumerator ele = new ExpListEnumerator(args);
Expression exp = (Expression) ele.nextElement();
int port = (int) exp.eval(binding).getLong();
BufferedReader in;
PrintWriter out;
ServerSocket s;
try
{
System.out.println("Creation du Serversocket");
s = new ServerSocket(port);
String ia = s.toString();
System.out.println("Adresse du ServeSocket est : " + ia);
System.out.println("Started: " + s);
// Le programme stoppe ici et attend une demande de connexion:
Socket socket = s.accept();
in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socket.getOutputStream())),true);
// affecte à $port_groupe le contenu de in
exp = (Expression) ele.nextElement();
binding.setValue(exp, new Value (in.readLine()));
out.println("0eme element");
// affecte à $hote_client le contenu de in
exp = (Expression) ele.nextElement();
binding.setValue(exp, new Value (in.readLine()));
out.println("1er element");
// affecte à $Group le contenu de in
exp = (Expression) ele.nextElement();
binding.setValue(exp, new Value (in.readLine()));
out.println("2eme element");
// affecte à $Date1 le contenu de in
exp = (Expression) ele.nextElement();
}
}
}

```

```

        binding.setValue(exp, new Value (in.readLine()));
        out.println("3eme element");
        // affecte à $Date2 le contenu de in
        exp = (Expression) ele.nextElement();
        binding.setValue(exp, new Value (in.readLine()));
        out.println("4eme element");
        // affecte à $type le contenu de in
        exp = (Expression) ele.nextElement();
        binding.setValue(exp, new Value (in.readLine()));
        out.println("Bien reçu toutes les informations");
        socket.close();
        s.close();
        return Value.TRUE;
    }
    catch (IOException e)
    {
        System.out.println("JAM::ReceiveQueryTutor:IOException : " + e);
        return Value.FALSE;
    }
}
}
}

```

D.4 – Un exemple de la primitive : AbsentsInGroup

```

package com.irs.jam.primitives;
import java.io.*;
import java.net.*;
import java.sql.*;
import com.irs.jam.*;
public class AbsentsInGroup implements PrimitiveAction
{
    public Value execute(String name, int arity, ExpList args,
        Binding binding, Goal currentGoal)
    {
        if (arity != 4) {
            System.out.println("Invalid number of arguments: " + arity + " to fonction \"" + name + "\"\n");
            return Value.FALSE;
        }
        //Définition des tableaux contenant les membres des différents groupes
        String Aquil[]={"broa","elliott","farnell","gardner","giejgo","mannion","maxhuni","millar","parsons","stacy"};
        String Gall[]={"garces","gatehouse","green","james","jamfrey","malecki","morran","spencer","vergere"};
        String Lugdu[]={"atkinson","evans","jeffs","kemp","macnaughtan","morgans","seymour","snow","wilson"};
        String Narbo[]={"booker","dubury","graham","grattidge","jones","martin","pickering","powell","price","smith"};
        ExpListEnumerator ele = new ExpListEnumerator(args);
        Expression exp = (Expression) ele.nextElement();
        String monGroupe = (String) exp.eval(binding).getString();
        exp = (Expression) ele.nextElement();
        String maDate1 = exp.eval(binding).getString();
        exp = (Expression) ele.nextElement();
        String maDate2 = exp.eval(binding).getString();
        String $resultat="";
        String Group1 = "Aquitania";
        String Group2 = "Gallia";
        String Group3 = "Lugdunensis";
        String Group4 = "Narbonensis";
        try
        {
            Class.forName("org.gjt.mm.mysql.Driver");
            String url="jdbc:mysql://localhost/mysql/data";
            // Connexion
            Connection con = DriverManager.getConnection(url,"","");
            //Création d'une instruction
            Statement statement = con.createStatement();
            //Exécution d'une requête
            String query = "SELECT DISTINCT login, groupe FROM logs WHERE groupe= '"+ monGroupe + "' &&
action_WebCT='connexion' && date_action >='"+ maDate1+"' && date_action <= '"+ maDate2 +"' "+ "ORDER BY login ";
            ResultSet resultset = statement.executeQuery (query);
            //Traitement des résultats
            while (resultset.next())
            {
                String $present = resultset.getString(1);
                //System.out.println("Present : "+$present);
            }
        }
    }
}

```

```

    for (int j=0; j < Aqai.length;j++) if (Aqai[j].equals($present)) Aqai[j]="";
    for (int j=0; j < Gall.length;j++) if (Gall[j].equals($present)) Gall[j]="";
    for (int j=0; j < Lugdu.length;j++) if (Lugdu[j].equals($present)) Lugdu[j]="";
    for (int j=0; j < Narbo.length;j++) if (Narbo[j].equals($present)) Narbo[j]="";
}
if (monGroupe.equals(Group1))
{
    for (int k=0; k<Aqai.length;k++)
    {
        if (!(Aqai[k].equals(""))) $resultat = Aqai[k]+ " "+$resultat;
    }
}
// Groupe Gallia

if (monGroupe.equals(Group2))
{
    for (int g=0; g<Gall.length;g++)
    {
        if (!(Gall[g].equals(""))) $resultat = Gall[g] + " " +$resultat;
    }
}
// Groupe Lugdunensis

if (monGroupe.equals(Group3))
{
    for (int l=0; l<Lugdu.length;l++)
    {
        if (!(Lugdu[l].equals(""))) $resultat = Lugdu[l] + " " +$resultat;
    }
}
// Groupe Narbonensis
    if (monGroupe.equals(Group4))
    {
        for (int n=0; n<Narbo.length;n++)
        {
            if (!(Narbo[n].equals(""))) $resultat = Narbo[n] + " " +$resultat;
        }
    }
if ($resultat.equals("")) $resultat ="Pas d'absent sur cette periode";
//Affecte le résultat de la requête à la variable $result
exp = (Expression) ele.nextElement();
binding.setValue(exp, new Value ($resultat));
// Fermeture de la connexion
con.close();
}
catch (ClassNotFoundException e) {
    System.err.println ("Erreur lors du chargement du pilote : " + e);
}
catch (SQLException sqle) {
    System.err.print ("Erreur SQL : " + sqle);
}
}
return Value.TRUE;
}
}
}

```

D.5 - Code de la primitive SendResultToTutor.java

```
package com.irs.jam.primitives;
import java.io.*;
import java.net.*;
import com.irs.jam.*;
public class SendResultToTutor implements PrimitiveAction {
    public Value execute(String name, int arity, ExpList args, Binding binding, Goal currentGoal)
    {
        if (arity != 4)
        {
            System.out.println("Invalid number of arguments: " + arity + " to function \"" + name + "\"\n");
            return Value.FALSE;
        }

        ExpListEnumerator    ele = new ExpListEnumerator(args);
        Expression exp = (Expression) ele.nextElement();
        int port = (int) exp.eval(binding).getLong();
        exp = (Expression) ele.nextElement();
        String addr = exp.eval(binding).getString();
        exp = (Expression) ele.nextElement();
        String maListe = exp.eval(binding).getString();
        exp = (Expression) ele.nextElement();
        String monType = exp.eval(binding).getString();

        try
        {
            Socket socket = new Socket(addr,port);
            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socket.getOutputStream())),true);
            //Envoi des résultats de la requête au Tuteur
            out.println(maListe);
            String str2 = in.readLine();
            System.out.println(str2);
            out.println(monType);
            String str3 = in.readLine();
            socket.close();
            return Value.TRUE;
        }
        catch (IOException e)
        {
            System.out.println("JAM::SendResultToTutor:IOException : " + e);
            return Value.FALSE;
        }
    }
}
```


Liste des acronymes

- **ACAO** : Apprentissage Collaboratif Assisté par Ordinateur
- **ADL** : Advanced Distributed Learning. Initiative lancée par le ministère de la défense américain pour soutenir les efforts de normalisation dans le domaine de la e-formation
- **AFNOR** : Agence Française de Normalisation
- **AIED** : Artificial Intelligence in Education
- **AICC** : Aviation Industry CBT Committee.
- **AFIT** : Air Force Institute of Technology
- **AMR** : Agent Message Router
- **API** : Application Program Interface. Interface de programme d'application
- **ARIADNE** : Alliance of Remote Instructional Authoring & Distribution Networks for Europe
- **BDI** : Beliefs – Desires – Intentions. L'architecture BDI est utilisée pour décrire la structure interne des agents
- **CEN** : Comité Européen de Normalisation
- **CBT** : Computer-Based Training
- **CSCL** : Computer-Supported Collaborative Learning
- **DARE** : Distributed Activities in Reflexive Environment. C'est le projet développé par Bourguin (2000) dans sa thèse.
- **DPS** : Distributed Problem Solving
- **EIAH** : Environnement Informatique pour l'Apprentissage Humain
- **EIAO** : Environnement Interactif d'Apprentissage avec Ordinateur
- **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.
- **FLE** : Français Langue Etrangère
- **FOAD** : Formation Ouverte et A Distance
- **HH** : communication entre humains, donc entre acteurs au sens ICOGAD
- **HM** : communication Humain-Machine ou encore humain-système informatique
- **IAD** : Intelligence Artificielle Distribuée
- **ICOGAD** : Interactions et Changements cOgnitifs dans les Groupes d'Apprentissage à Distance.
- **IDL** : Interface Definition Language
- **IEEE** : Institute of Electrical and Electronics Engineers.
- **IHM** : le champ d'étude pluri-disciplinaire des interactions humain-machine.
- **ISO** : International Organisation for Standardisation

- **ITS** : Intelligent Tutoring Systems
- **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 développé au sein du FIPA afin de permettre aux agents de communiquer entre eux.
- **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.
- **LIFC** : Laboratoire d'Informatique de l'Université de Franche-Comté.
- **LMS** : Learning Management System
- **LOM** : Learning Object Metadata.
- **LTSC** : Learning Technology Standards Committee.
- **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.
- **MM** : Communication Machine-Machine, donc entre agents au sens informatique
- **OMA** : Object Manager Architecture
- **OMG** : Object Management Group. C'est une fondation qui travaille sur le développement de standards en programmation orientée objet.
- **ORB** : Object Request Broker
- **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.
- **PRS** : Procedural Reasoning System
- **RMI** : Remote Method Invocation
- **RPC** : Remote Procedure Call
- **SCO** : Sharable Content Object
- **SCORM** : Sharable Content Object Reference Model
- **SCS** : Structured Circuits Semantic
- **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 LIFC
- **SIGFAD** : Soutien des 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
- **SNA** : Social Network Analysis
- **SQL** : Structured Query Language
- **UIML** : User Markup Interface Language
- **UML** : Unified Modeling Language. Langage de modélisation utilisé en programmation orientée objet développé par l'Object Management Group et qui veut unifier toutes les méthodologies objet existantes.
- **TA** : Théorie de l'Activité
- **TCAO** : Travail Coopératif Assisté par Ordinateur
- **TIC / TICE** : Technologies de l'information et de la Communication / Technologies de l'information et de la Communication pour l'Education

Liste des Figures

Figure 1.1 : Vue d'ensemble d'un système de téléformation.....	37
Figure 1.2 : L'environnement d'exécution de SCORM.....	51
Figure 2.1 : Le modèle de Salmon pour le rôle des tuteurs en ligne, tiré de Salmon (2000).....	67
Figure 2.2 : Les niveaux hiérarchiques d'une activité.....	70
Figure 2.4 : La structure basique d'une activité tirée de (Bourguin, 2000).....	76
Figure 2.5 : La réutilisation dans l'ingénierie logicielle.....	82
Figure 2.6 : Organisation de SimuLigne.....	93
Figure 3.1 : le contexte de départ de SimuLigne.....	102
Figure 3.2 : Calendrier des activités de SimuLigne + InterCulture.....	110
Figure 3.3 : Le calendrier des activités de SimuLigne + InterCulture – Jalons et outils associés (une couleur par outil).....	113
Figure 3.4 : Bilan quantitatif de SimuLigne.....	116
Figure 3.5: Structure d'une activité dans SimuLigne.....	119
Figure 3.6 : Le modèle conceptuel d'activité de SimuLigne.....	120
Figure 3.7 : Le schéma général de PIFAD.....	124
Figure 3.8 : Informations fournies par PIFAD au tuteur sur son groupe.....	137
Figure 3.9 : Le tuteur peut disposer de plus d'informations sur un apprenant donné.....	138
Figure 4.1 : Un agent dans son environnement (Wooldridge, 1999).....	145
Figure 4.2 : Taxonomie des agents de Franklin et Graesser (1996).....	149
Figure 4.3 : Les étapes et les phases de la méthodologie MaSE.....	176
Figure 4.4 : L'architecture JAM tirée de (Huber, 2001).....	181
Figure 4.5 : Le Routeur de messages de JATLite.....	186
Figure 4.6 : L'architecture de JATLite est basée sur un ensemble de couches spécialisées et hiérarchiques.....	187
Figure 5.1 : Une vue d'ensemble de SIGFAD.....	193
Figure 5.2 : Le diagramme hiérarchisé des objectifs de SIGFAD.....	197
Figure 5.3 : Le diagramme de séquence fournissant la liste des apprenants absents.....	198
Figure 5.4 : Algorithme donnant l'état du groupe.....	200
Figure 5.5 : Algorithme de détermination de l'état de réalisation d'une activité.....	202
Figure 5.6 : Le diagramme de rôles de SIGFAD.....	206
Figure 5.7 : Etats et transitions du rôle émetteur (agent tuteur, coordonnateur, apprenant) d'une demande.....	207
Figure 5.8 : Etats et transitions de r_SI (agent superviseur des interactions).....	207
Figure 5.9 : Etats et transitions de r_SI dans la fourniture de la liste des apprenants à r_TUT.....	208
Figure 5.10 : Etats et transitions de r_TUT dans la demande de la liste d'apprenants à r_SI.....	209
Figure 5.11 : Le diagramme de classes d'agent de SIGFAD.....	211
Figure 5.12 : Le diagramme de déploiement de SIGFAD.....	212
Figure 5.13 : L'agent du tuteur du groupe Aquitania : tutor_gpe.jam.....	214
Figure 5.14 : Capture d'écran montrant l'exécution de l'agent tutor_gpe.jam.....	215
Figure 5.16 : L'écran d'entrée de l'interface agent – tuteur.....	219
Figure 5.17 : Ecran disponible si le tuteur veut des informations sur le groupe.....	219
Figure 5.18 : La vue d'entrée offerte au Coordonnateur par son agent.....	221
Figure 5.19 : La vue d'entrée de l'interface apprenant-SIGFAD.....	221
Figure 5.20 : L'algorithme exécuté par supervisor_v13nov.jam.....	223
Figure 5.21 : L'algorithme exécuté par un agent du tuteur.....	223
Figure 5.22 : Evolution de l'état de productivité des groupes entre le 02 avril et le 03 juin.....	230
Figure A.1 : Plan de la ville proposé par le groupe Aquitania.....	249
Figure A.2 : Plan du campus proposé par le groupe Gallia.....	249
Figure A.3 : Hymne de la ville proposé par le groupe Aquitania.....	250
Figure A.4 : Une proposition de menus par le Chef Cuisinier du groupe Narbonensis.....	250
Figure B.1 : Le diagramme de séquence pour fournir les ratios de temps par semaine.....	251
Figure B.2 : Le diagramme de séquence pour fournir l'état du groupe.....	251
Figure B.3 : Le diagramme de séquence pour recevoir la liste des méta-informations sur les activités.....	252
Figure B.4 : Le diagramme de séquence pour déterminer l'état de réalisation d'une activité.....	252
Figure B.5 : Le diagramme de séquence pour recevoir les méta-informations sur les forums.....	252
Figure B.6 : Le diagramme de séquence pour déterminer si un forum est bien suivi.....	253
Figure B.7 : Etats et transitions de r_COOR dans la fourniture des méta-informations à r_SI.....	254
Figure B.8 : Etats et transitions de r_SI dans la réception des méta-informations.....	254
Figure B.9 : Etats et transitions de r_TUT pour obtenir les ratios de temps.....	255

Figure B.10 : Etats et transitions de r_SI pour fournir les ratios de temps.....	255
Figure B.11 : Etats et transitions de r_TUT pour recevoir l'état du groupe.....	256
Figure B.12 : Etats et transitions de r_SI pour fournir l'état du groupe.....	256
Figure B.13 : Etats et transitions de r_TUT dans l'indication de l'état de réalisation d'une activité.....	257
Figure B.14 : Etats et transitions de r_SI dans l'indication de l'état de réalisation d'une activité.....	257
Figure B.15 : Etats et transitions de r_TUT pour la demande et la réception de l'avis de gestion des forums ...	258
Figure B.16 : Etats et transitions de r_SI lors de la fourniture de l'avis de gestion des forums.....	258

Liste des tableaux

Tableau 1.1 : les acteurs en téléformation et certains de leurs rôles, tiré de (Paquette, 2002).....	42
Tableau 2.1 : Résumé des caractéristiques générales des méthodes Catalysis, RUP, Select.....	84
Tableau 3.1 : Pourcentage cumulé du nombre de connexions à PIFAD.....	126
Tableau 3.2 : Pourcentage cumulé des temps de connexion à PIFAD.....	126
Tableau 3.3 : Pourcentage cumulé du nombre de courriels lus à l'intérieur du groupe.....	126
Tableau 3.4 : Pourcentage cumulé du nombre de courriels postés à l'intérieur du groupe.....	127
Tableau 3.5 : Pourcentage cumulé du nombre de messages lus dans les forums.....	127
Tableau 3.6 : Pourcentage cumulé du nombre de messages postés dans les forums.....	128
Tableau 3.7 : Durées de connexion aux pages de consignes.....	129
Tableau 3.8 : Pourcentages cumulés du nombre de messages lus par les apprenants du groupe Lugdunensis...	130
Tableau 3.9 : Nombre d'entrées dans le groupe Lugdunensis entre le 11 mai et le 25 mai.....	132
Tableau 3.10 : Nombre de messages lus par les apprenants dans les forums du Groupe entre le 11 et le 15 mai 2001.....	133
Tableau 3.11 : Nombre de messages postés par les apprenants dans les forums du Groupe entre le 11 et le 15 mai 2001.....	133
Tableau 3.12 : Nombre de fois où les apprenants du groupe ont consulté les pages de consignes entre le 11 mai et le 23 mai.....	133
Tableau 4.1 : Classification des agents proposée par Ferber (1995).....	150
Tableau 4.2 : Exemple d'un robot voulant franchir une porte (Ferber, 1995).....	150
Tableau 5.1 : Quelques résultats fournis par l'agent tuteur du groupe Aquitania.....	226
Tableau 5.2 : Quelques résultats fournis par l'agent du tuteur de Lugdunensis.....	227
Tableau 5.3 : Résultats de requêtes fournies par SIGFAD pour trois apprenants.....	228
Tableau 5.4 : Résultats de requêtes portant sur l'exécution des activités dans le groupe Aquitania.....	231
Tableau C.1 : Quelques résultats fournis par l'agent du tuteur de Gallia.....	259
Tableau C.2 : Quelques résultats fournis par l'agent du tuteur Narbonensis.....	260

Index des mots clés et d'auteurs cités

A

AAII 162
agent 95, 145, 148
Agent UML 164
AgentTool 191
apprentissage collaboratif 59, 63
Apprentissage collaboratif 55
Apprentissage machine 167
approche composants 80

B

BDI 191
Bourguin 75, 117

C

Cassiopea 164
Cerri 85
Chanier 10, 88, 96, 244
collaboration 60
composant 80
coordonnateur 66
CORBA 183
CSCL 57

D

DeLoach 95, 148
Depover 10
Després 28, 241
Dillenbourg 55, 57, 58, 101

E

Ecoutin 67
e-formation 24
EIAH 21

F

FAD 21, 22
Ferber 10, 146, 156, 160
Fjök 12
Fjuk et Ludvigsen 122
Fjök et Ludvigsen 77
FOAD 9, 21, 22

G

GAIA 163
George 28, 136, 241
groupe 65

H

Haythornthwaite 63, 129
Henri et Ludgren-Cayrol 64, 140
Henri et Lundgren-Cayrol 58
Huber 180

I

ICOGAD 96, 101
interactions 65

J

JAM 179, 181, 213
JATLite 185, 186, 187
JDBC 216, 217
Jennings 79, 145
Jermann et al 77

K

Keegan 22

L

Labat 66, 80
langage Z 164
LMS 36

M

MadKit 166
MaSE 172, 176
Mbala 135, 239, 244
modèle conceptuel d'activité 120
modèle de composants 81

N

Nwana 148, 150

O

Odell..... 146, 147, 160

P

Paquette..... 32, 42
PIFAD..... 86
Prometheus..... 165

R

Reffay..... 15, 135, 244
RMI..... 183

S

SALMON..... 68

SCORM 47, 50
SICAH 101
SIGFAD 86, 138, 193, 197, 206, 211, 212
SimuLigne..... 86, 93
SMA..... 154, 155, 156

T

téléformation 29
TICE 21
tuteur 65, 68

W

Wooldridge 79, 145, 147

Références bibliographiques

Bibliographie

- Allen, P., Frost, S. (1998). *Component-Based Development for Enterprise Systems : Applying the select perspective*, Cambridge University Press, Cambridge, UK.
- Alonso, E., Kudenko, D. (1999). "Machine learning techniques for adaptive logic-based multi-agent systems". *Proceedings of the Second Workshop of the UK Special Interest Group on Multi-Agent Systems (UKMAS-99)*, Hewlett-Packard Laboratories, Bristol.
- Andersen (2000). *L'irrésistible ascension du e-learning*. Consulté en juin 2001 sur <http://www.arthurandersen.fr>
- Apple Computer (1993). *AppleScript language Guide*. Readings, Mass : Addison-Wesley
- Ball, G., Ling, D., Kurlander, D., Miller, J., Pugh, D., Skelly, T., Stankosky, A., Thiel, D., Dantzich, M. V., Wax, T. (1996). Lifelike Computer Characters : The Persona Project at Microsoft Research. In *Software Agents*, ed. J.M. Bradshaw. Menlo Park, Californie, AAAI Press.
- Barbier, F., Cauvet, C., Oussalah, M., Rieu, D., Bennisri, S., Souveyet, C. (2002). "Composants dans l'ingénierie des systèmes d'information : concepts clés et techniques de réutilisation". Actes des deuxièmes assises nationales du GDR-I3. Décembre 2002. Consulté en avril 2003 sur <http://sis.univ-tln.fr/gdri3/assises.htm>
- Bates, J., (1994). "The role of emotion in believable agents". *Communications of the ACM*. 37 (7). pp. 122-125.
- Blaye, A. (1998). *Confrontation socio-cognitive et résolution de problèmes*. Thèse de doctorat de l'Université de Provence.
- Boissier, O., (1990). *La coopération entre systèmes à base de connaissances*. Rapport de recherche, Institut National Polytechnique de Grenoble.
- Bourguin, G., (2000). *Un support informatique à l'activité coopérative fondé sur la Théorie de l'Activité : le Projet DARE*. Thèse de Doctorat en Informatique. Université des Sciences et Technologies de Lille.
- Boy, G., (1991). *Intelligent Assistant Systems*. San Diego, Calif. : Academic Press.
- Brooks, R., A. (1999). *Cambrian Intelligence*. The MIT Press: Cambridge, MA.
- Cerri, S., Maraschi, D. "Relations entre les technologies d'apprentissage humain et les agents". A paraître dans *Journal Cognitique*. Consulté en mai 2003 sur <http://www.lirmm.fr/~maraschi/papers/>
- Chanier, T. (2001a). "Créer des communautés d'apprentissage à distance. Apprentissage collaboratif avec les plates-formes de téléformation". *Les dossiers de l'Ingénierie éducative, n° 36 sur Les communautés en ligne*, Octobre. Centre National de Documentation Pédagogique, CNDP. Montrouge. pp. 56-59
- Chanier, T. (2001b). *Individu et groupe d'apprentissage en téléformation*. Transparents séminaire "Hypermédia, éducation et formation", Paris, mars 2001. Besançon : Laboratoire LIFC, Université de Franche-Comté. 64 p.
- Chanier, T. (2000) "Hypermédia, interaction et apprentissage dans des systèmes d'information et de communication : résultats et agenda de recherche". In *Apprendre une langue dans un environnement multimédia*. Outremont, Ca : Les Editions Logiques. pp. 53-92.
- Charlier, B., Daele, A., Docq, F., Lebrun, M., Lusalusa, S., Peeters, R., Deschryver, N., (1999). "Tuteurs en ligne : quels rôles, quelle formation ?". CNED – Décembre 1999
- Charlier, B., Daele, A., Deschryver, N., (1998). "Apprendre en collaborant à distance : ouvrons la boîte noire – ARDEMI". Consulté en Mai 2001 : <http://tecfa.unige.ch/proj/learnett/public/public.html>
- Clancey W. J., (1993). Situated action : a neuropsychological interpretation : Response to Vera and Simon. *Cognitive Science*, 17, pp. 87-116
- Collectif de Chasseneuil (2000). Conférence de consensus – « Formations Ouvertes et à Distance. L'accompagnement pédagogique et organisationnel ». 27, 28 & 29 mars 2000. Consulté en Juin 2001 : <http://www.algora.org/kiosque/actu/consensus/ccfod.pdf>
- Collinot, A., Drogoul, A., Benhamou, P., (1996). "Agent oriented design of a soccer robot team". In *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96)*. pp. 41-47. Kyoto, Japan.
- Coutaz, J. (1990). *Interfaces Homme Ordinateur : Conception et Réalisation*. Editions Bordas. Paris (France).

- DeLoach, S. A., Wood, M. F., Sparkman, C. H. (2001). "Multiagent Systems Engineering". *International Journal on Software Engineering and Knowledge Engineering*, Vol. 11, N° 3, 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
- Depover, C., De Lièvre, B., Deschryver N. (1999). "Pour une refonte des modèles d'usage d'Internet pour l'enseignement : quelques exemples de dispositifs adaptés à la formation universitaire". CIPTE
- Depover, C., Marchand, L. (2002). E-learning et formation des adultes en contexte professionnel. Bruxelles : De Boeck Université.
- Després, C. (2001). *Modélisation et conception d'un environnement de suivi pédagogique synchrone d'activités d'apprentissage à distance*. Thèse de Doctorat en informatique, Université du Maine.
- 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.
- Doise, W., Mugny, G. (1984). *The social development of the intellect*. Oxford : Pergamon Press.
- D'Souza, D., Cameron Wills, A. (1999). *Objects, Component and Frameworks with UML, The Catalysis Approach*, Addison-Wesley.
- Ecoutin, E. (2001). "Mise en œuvre des plates-formes pour la formation ouverte et à distance – Fiche pratique 4 : les fonctions d'accompagnement". Consulté en avril 2003 sur <http://ressources.algora.org/ressources/envIRONNEMENTS/tel/accompagnement.pdf>
- Ecoutin, E. (2000). "Etude comparative technique et pédagogique des plates-formes pour la formation ouverte et à distance – Mise à jour novembre 2000". Consulté en Juin 2001 : <http://www.algora.org/kiosque/publicat/doc/pdf/pf2000.pdf>
- Engeström, Y., (1987). *Learning by expanding : an activity-theoretical approach to developmental research*, Orienta-Konsult Oy, Helsinki.
- Estlin, T., (1998). *Using Multi-Strategy Learning to Improve Planning Efficiency and Quality*. Ph.D. thesis, The University of Texas at Austin, Department of Computer Sciences. May 1998
- Ferber, J., (1995). *Les systèmes multi-agents : vers une intelligence collective*. Interditions.
- Ferber, J., Gutknecht, O. (1998). "A meta-model for the analysis and design of organisations in multi-agents systems". In *Third International Conference on Multi-Agent Systems (ICMAS'98) Proceedings*, pp. 128-135. IEEE Computer Society, 1998.
- Finin, T., Labrou, Y., Mayfield J., (1997). "KQML as an agent communication language". In *Software Agents*, ed. J.M. Bradshaw. Menlo Park, Californie, AAAI Press.
- Fjuk A., Ludvigsen, S. (2001). "The Complexity of distributive collaborative learning : Unit of Analysis". *Proceedings of Euro-CSCL 2001*, Maastricht, 22 - 24 March 2001.
- Fjuk, A., Nurminen, M. I., Smordal, O. (1997). "Taking Articulation Work Seriously – an Activity Theoretical Approach". *TUCS Technical Report N° 120*. 16 p.
- Franklin, S., Graesser, A., (1996). "Is it an agent or just a program ? A taxonomy for autonomous agents". In *Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages*. New-York : Springer-Verlag.
- Furness, T., Winn, W., Yu R. (1998). "The Impact of Three Dimensional Immersive Virtual Environments on Modern Pedagogy – Global Change, VR and Learning". Consulté en Juin 1999 : <http://www.hitl.washington.edu/publications/r-97-32/>
- Futtersack, M., Labat, J. M., (2000). "Quelle planification pédagogique dans les EIAH ?", *Revue STE*, vol 7, n° spécial "Education et Informatique, hommage à Martial Vivet " Ed. Hermes, pp. 165-177.
- Gemme (2002). "L'impact de la normalisation sur les dispositifs d'enseignement". Consulté en Avril 2003 sur http://www.gis-gemme.org/rapports/norm_standard2.pdf
- Genereseth, M. R. (1997). "An Agent-based Framework for Interoperability". In *Software Agents*, ed. J.M. Bradshaw. Menlo Park, Californie, AAAI Press.

- George, S. (2001). *Apprentissage collectif à distance. SPLACH : un environnement informatique support d'une pédagogie de projet*. Thèse de Doctorat en informatique, Université du Maine (2001).
- Gilly, M. (1989). The psychological mechanisms of cognitive constructions, experimental research and teaching perspectives. *International Journal of Educational Research*, N° 13, Vol. 6. pp. 607-621
- Guidon, J. (2002). "Note sur les normes et standards pour la formation". Consulté en avril 2003 sur <http://www.educnet.education.fr/superieur/normes.htm>
- Gutknecht, O., Ferber, J. (1998). "Un modèle d'analyse, de construction et d'exécution de systèmes multi-agents". In Jean-Pierre Barthes (ed) : *Actes des Journées Francophones en Intelligence Artificielle Distribuée et Systèmes Multi-Agents 1998*. Hermes, Novembre 1998.
- Hayes-Roth, B., Brownston, L., Gent, R. V. (1995). "Multiagent collaboration in Directed Improvisation". In *Proceedings of the First International Conference on Multi-agent Systems. (ICMAS-95)*. Ed V. Lesser, pp. 148-154. Menlo Park, Calif. : AAAI Press.
- Haythornthwaite, C. (1999). "Networks of Information Sharing among Computer-Supported Distance Learners". *Proceedings of the Third International Conference on Computer-Supported Collaborative Learning* pp. 218-222.
- Henri, F., Lundgren-Cayrol, K. (2001). *Apprentissage collaboratif à distance*. Presses de l'Université du Québec.
- Hewitt, C. (1971). "Procedural embedding of knowledge in planner". In D.C. Cooper (Ed) *Proceedings of the 2nd International Joint Conference on Artificial Intelligence*. William Kaufmann, London, UK. pp. 167-184.
- Huber, M. J. (2001). "JAM: A BDI-theoretic Mobile Agent Architecture". Consulté en septembre 2002 sur http://marcush.net/IRS/irs_papers.html.
- Hutchins, E. (1991). "The social Organization of Distributed Cognition". In L. Resnick, J. Levine, S. Teasley (Eds). *Perspectives on Socially Shared Cognition*. Hyattsville, MD: American Psychological Association. pp. 383-407.
- Jacobson, I., Booch, G., Rumbaugh, J. (1999). *The Unified Software Development Process*. Addison-Wesley Object Technology Series.
- Jennings, N. R., Wooldridge M., (1998). "Applications of Intelligent Agents" in *Agent Technology: Foundations, Applications, and Markets* (eds. N. R. Jennings and M. Wooldridge) Springer-Verlag : Berlin, Germany. pp. 3-28.
- Jennings, N. R., Sycara, K., Wooldridge, M., (1998). "A Roadmap of Agent Research and Development". *Autonomous Agents and Multi-Agent Systems, 1*, pp. 275-306. Kluwer Academic Publishers, Boston.
- 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*
- Keegan, D., (1996). *Foundations of Distance Education*. Third Edition. Routledge Studies in Distance Education. 224 pages.
- Kinny, D., Georgeff, M., Rao, A., (1996). "A methodology and modelling technique for systems of BDI agents". In W. Van de Velde and J. W. Perram (eds). *Agents breaking away : Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in Multi-Agent World*. pp. 56-71. Springer-Verlag, Berlin, Germany.
- Knobock, C. A., Ambite J. L. (1997). "Agents for Information Gathering". In *Software Agents*. Ed. J.M. Bradshaw. Menlo Park, Calif. : AAAI Press.
- 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.
- Labat, J. M., (2002) "EIAH : Quel retour d'informations pour le tuteur ? ", *Actes de la Conférence TICE 2002*, Novembre 2002. pp. 81-87.
- Labidi, S., Lejouad, W., (1993) *De l'intelligence artificielle aux systèmes multi-agents*. Rapport de recherche Unité de recherche INRIA Sophia-Antoplois 2004.

- Lebrun, M. (1999). *Des technologies pour enseigner et apprendre*. De Boeck, Bruxelles
- Le Preau (2000). "Choisir une solution de téléformation. Etude 2000, Plates-formes et portails de téléformation.", consulté en mars 2003 sur <http://www.preau.ccip.fr/teleformation/default.htm#table>
- Leroux, P., (2002). *Machines partenaires des enseignants et des apprenants – Etude dans le cadre des environnements supports de projets pédagogiques*. Habilitation à Diriger les Recherches de l'Université du Maine.
- Leroux P., (1995). Conception et réalisation d'un système coopératif d'apprentissage - Étude d'une double coopération : maître/ordinateur et ordinateur/groupe d'apprenants, Thèse de Doctorat en informatique, Université Paris 6.
- Luck, M., d'Iverno, M., (1995). "A formal framework for agency and autonomy". In *Proceedings of the First International Conference on Multi-agent Systems. (ICMAS-95)*. pp. 254-260. San Francisco, CA.
- Maes, P. (1997). "Agents that Reduce Work and Information Overload". In *Software Agents*, ed. J.M. Bradshaw. Menlo Park, Californie, AAAI Press.
- Mbala, A., Reffay, C., Chanier, T. (2003). "SIGFAD : Un système multi-agents pour soutenir les utilisateurs en formation à distance". Actes de la conférence EIAH 2003, Strasbourg (France). Eds. Desmoulin C., Marquet P., Bouhineau D. Institut National de Recherche Pédagogique. pp. 319-330.
- Mbala, A., Reffay, C., Chanier, T. (2002). "Integration of Automatic Tools for Displaying Interaction Data in Computer Environments for Distance Learning ". *6th International Conference of Intelligent Tutoring Systems (ITS'2002)*. S. A. Cerri, G. Gouardères, F. Paraguaçu (Eds). LNCS 2363. Springer. (2002). pp. 841-850.
- Mbala, A., Reffay, C., Tamo, T. T. (2002). "Le paradigme d'agent dans le développement des plates-formes de formation à distance". *Actes du 1^{er} Séminaire National sur la formation à distance*. 18-19 avril 2002. Presses Universitaires de Yaoundé. pp. 67-74.
- Mbala, A. (2002) "Specifying a Multiagent System to Support Users' Activities in Computer-Supported Distance Learning". *Proceedings of the Starting Artificial Intelligence Researchers Symposium (STAIRS 2002)*. Thierry Vidal, Paolo Liberatore (Eds). IOS Press. (2002). pp. 33-34.
- Minsky, M. (1987). "The society of mind". London : William Heinemann Ltd.
- Minsky, M., Riecken, D. (1994). "A Conversation with Marvin Minsky about Agents". *Communications of the ACM 37 (7)*. pp. 23-29.
- Montgomery, S. (1998). "Adressing Diverse Learning Styles Through the Use of Multimedia". Consulté en Juin 1999 : <http://fre.www.ecn.purdue.edu/FrE/asee/fie95/3a2/3a22/3a22.htm>
- Moulin, B., Chaib-draa, B. (1996). An Overview of Distributed Artificial Intelligence in Foundations of Distributed Artificial Intelligence. Eds O'Hare and Jennings. New-York : Wiley. pp. 3-55.
- Mundhe, M., Sen, S. (2000). "Evolving agent societies that avoid social dilemmas," in *Proceedings of GECCO-2000* held between July 8--12, 2000 in Las Vegas, Nevada. pp. 809-816
- Negroponte, N. (1997). "Agents : From Direct Manipulation to Delegation", In *Software Agents*. Ed. J.M. Bradshaw. Menlo Park, Californie. AAAI Press.
- Negroponte, N. (1995). *Being Digital*. New York : Alfred Knopf.
- Nwana, H. S. (1996). "Software Agents : An Overview". *Knowledge Engineering Review*, Vol. 11, N° 3, pp. 1-40.
- Nwana H. S., Ndumu D. T., (1999). "A perspective on software agents research". Consulté en Juin 2000 : <http://agents.umbc.edu>
- Odell, J., Van Dyke P., Bauer, B (2000). "Extending UML for Agents". *Proceedings of the Agent-Oriented Information Systems Workshop* at the 17th National conference on Artificial Intelligence, Gerd Wagner, Yves Lesperance, and Eric Yu eds., Austin, TX, pp. 3-17.
- OFEM (2002) *Les entreprises et la e.formation en France, enquête 2002*. Consulté en mars 2003 sur <http://www.pwcgloballearning.com/agenda/eformation.pdf>
- OFEM (2001) *Les entreprises et la e.formation en France : coûts et retour sur investissements*. Consulté en mars 2003 sur http://www3.ccip.fr/ofem/nte/Synthese_nte.pdf

- Padgham, L., Winikoff, M., (2002). "Prometheus : A Methodology for Developing Intelligents Agents". *Proceedings of the Agent Oriented System Engineering 2002 (AOSE'2002)*. 12 p.
- Paquette, G. (2002). *L'ingénierie pédagogique. Pour construire l'apprentissage en réseau*. 2002, Québec, Presses de l'Université du Québec.
- Piaget, J. (1965). *Les études sociologiques*. Genève : Droz.
- Piaget, J. (1928). *The language and thought of the child*. New-York: Harcourt.
- Pietrie, C. (2003). <http://cdr.stanford.edu/ProcessLink/kqml-proposed.html> consulté en avril 2003
- Reffay, C. (2001). *SimuDonnées : la base de données de Simuligne*. Document interne Rap4/0108 du projet ICOGAD, août 2001 et rapport LIFC R.01.01. Besançon : Laboratoire LIFC, Université de Franche-Comté. 62 p. + annexes de 54 p.
- Reffay, C., Chanier, T. (2003a). Mesurer la cohésion d'un groupe d'apprentissage en formation à distance. *Actes de la conférence EIAH 2003, Strasbourg (France)*. Eds. Desmoulins C., Marquet P., Bouhineau D. Institut National de Recherche Pédagogique. pp. 367-378.
- Reffay, C., Chanier, T. (2003b). « How social network analysis can help to measure cohesion in Collaborative Distance Learning » *Proceeding of Computer Supported Collaborative Learning conference (CSCL'2003), Kluwer Academic Publishers, June, Bergen, Norway*, pp. 343-352.
- Reffay C., Chanier T. (2002). "Social Network Analysis Used for Modelling Collaboration ". *6th International Conference of Intelligent Tutoring Systems (ITS'2002)*. S. A. Cerri, G. Gouardères, F. Paraguaçu (Eds). LNCS 2363. Springer. (2002). pp. 31-40.
- Reffay C., Chanier T. (2001). "CUMULI : construction d'une mémoire du groupe dans l'interaction en FAD", *Actes de la Conférence EIAO'2001, Sciences et Techniques Éducatives*, Vol. 8, 1-2, pp. 155-158.
- Renié, D., Chanier, T. (1996). "ELEONORE : un environnement collaboratif en français langue seconde". *Sciences et Techniques Educatives*, 3(3), pp. 353-380
- RH Info (2001). "Evolution du e-learning et prospective pour la formation". <http://www.rhinfo.com/jsp.clear/home/etudes/elearn.jsp>
- Rosselle, M. (2003). "Etude des objectifs d'une plate-forme de coopération pour les EIAH". *Actes de la conférence EIAH 2003, Strasbourg (France)*. Eds : Desmoulins C., Marquet P., Bouhineau D. Institut National de Recherche Pédagogique. pp. 379-390.
- Salmon, G. (2000). *E-moderating : The Key to Teaching and Learning Online*. Kogan Page. London.
- Shoham, Y. (1997). "An overview of Agent-oriented Programming Languages". In *Software Agents*, ed J.M. Bradshaw. Menlo Park, Californie. AAAI Press.
- Shoham, Y. (1993). "Agent-oriented programming". *Artificial Intelligence* 60(1): pp. 51-92.
- Schwartz, D., L. (1995). The emergence of abstract dyad representations in dyad problem solving. *The Journal of the Learning Sciences*, Vol. 4, N° 3, pp. 321-354.
- Simudonnées (2001). *Cédérom contenant toutes les données Simuligne stockées sur la plate-forme PIFAD (matériaux pédagogiques et traces de la formation) et les informations de* (Reffay , 2001), version 1/09/01, septembre 2001. Document interne CD2/0109 du projet ICOGAD. Besançon : Laboratoire LIFC, Université de Franche-Comté. 480 Mo, 30 000 fichiers.
- Simuligne (2001a). *Formation Simuligne : descriptif général*. Document interne Rap1/0102 du projet ICOGAD, février 2001. Besançon : Laboratoire LIFC, Université de Franche-Comté. 29 p.
- Simuligne (2001b). *Formation Simuligne : consignes tuteurs et apprenants pour le descriptif général, les étapes E0 et E1, PrepInterculture*. Document interne Rap2/0105 du projet ICOGAD, mai 2001. Besançon : Laboratoire LIFC, Université de Franche-Comté. 104 p.
- Simuligne (2001c). *Formation Simuligne : consignes tuteurs et apprenants pour le descriptif général, les étapes E2 et E3, Interculture*. Document interne Rap3/0105 du projet ICOGAD, mai 2001. Besançon : Laboratoire LIFC, Université de Franche-Comté. 101 p.
- Smith, R. G. (1980). The Contrat net protocol : High-level communication in a distributed problem solver. *IEEE Transaction on Computers*. N° 12. pp. 1104-1113.
- Spivey, M., (1992). *The Z Notation* (second edition). Prentice Hall International : Hemel Hempstead, England.

- Stone, P., Veloso, M., (2000). "Multiagent systems : a survey from machine learning perspective". *Autonomous Robots*, Vol. 8, n° 3, July 2000. 57p.
- Sugawara, T., Lesser, V. (1993). « Learning Control Rules for Coordination. » In *Proceedings of Multi-Agent and Cooperative Computation '93*, pp. 121-136.
- Torrent, S. (2001). *Le gestionnaire 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.
- Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA: Harvard University Press
- Vygotsky, L. S. (1962). *Thought and Language*. Cambridge, MA: MIT Press
- Webb, N., M., (1991) Task Related Verbal Interaction and Mathematical Learning in Small Groups. *Research in Mathematics Education*. N° 22, Vol. 5. pp. 366-389
- Webb, N., M., (1989) Peer interaction and learning in small groups. *International Journal of Educational research*. N° 13, Vol. 1. pp. 21-39
- WebCT (2000). *Plate-forme de téléformation, version française 3.1*. Consulté en mai 2001 : <http://www.webct.com/webct/>
- Weiss, G. (1999). "Learning in Multiagent Systems". In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 559-298. The MIT Press, Cambridge, MA.
- Weiss, G., Dillenbourg, P. (1999). What is 'multi' in multi-agent learning? In P. Dillenbourg (Ed). *Collaborative-learning Cognitive*.
- White, J. (1997). Mobile Agents. In *Software Agents*, ed. J.M. Bradshaw. Menlo Park, Californie, AAAI Press.
- Wiederhold, G. (1992). "Mediators in the Architecture of Future Information Systems". *IEEE Computer*, March, pp. 38-49.
- Wiederhold, G. (1989). *The Architecture of Future Information Systems*, Technical Report, Computer Science Department, Stanford University.
- Winn, W. (1997). "The Impact of Three-dimensional Immersive Virtual Environments on Modern Pedagogy". Consulté en Juin 1999 : <http://www.hitl.washington.edu/publications/r-97-15/>
- Winn, W. (1993). *A conceptual basis for Educational Applications of Virtual Reality*. HITL Technical Report R-93-9. Consulté en Juin 1999 : <http://www.hitl.washington.edu/publications/r-93-9/>
- Wooldridge, M. (1999). "Intelligent Agents" In G. Weiss, editor: *Multiagent Systems*, The MIT Press, April 1999.
- Wooldridge, M. J., Jennings, N. R.(1995). "Agents Theories, Architectures and Languages : A Survey. In Intelligent Agents": *ECAI-94 Workshop on ATAL*, eds Wooldridge and Jennings. Springer-Verlag. Berlin. pp. 1-39.
- Wooldridge, M., Jennings, N. R., Kinny, D., (1999). "A methodology for agent-oriented analysis and design". In *Proceedings of the Third International Conference on Autonomous Agents*. pp. 69-76, Seattle, USA.

Sites internet

- (ADLnet, http) <http://www.adlnet.org/>
- (Icogad, http) <http://lifc.univ-fcomte.fr/RECHERCHE/P7/Icogad/icogad.htm>
- (JAM, http) <http://marcush.net/IRS>
- (JT1SC36, http) <http://collab-tech.jtc1sc36.org/doc/>
- (MaSE, http) <http://www.cis.ksu.edu/~sdeloach/ai/mase.htm> ,
<http://en.afit.af.mil/ai/papers.htm>
- (MadKit, http) <http://www.madkit.org/>
- (RoboCup, http) <http://www.RoboCup.org/>

Table des matières

REMERCIEMENTS	1
SOMMAIRE.....	3
INTRODUCTION.....	7
CONTEXTE DE NOS TRAVAUX DE RECHERCHE	11
PROBLEMATIQUE	12
OBJECTIFS DE RECHERCHE	13
APPROCHE METHODOLOGIQUE	15
PLAN DU MEMOIRE.....	16
CHAPITRE 1 : FORMATION A DISTANCE, TELEFORMATION ET E-FORMATION.....	19
1.1 - FORMATION A DISTANCE ET E-FORMATION	22
1.1.1 - Définitions.....	22
1.1.2 – FOAD, FAD et e-formation.....	23
1.1.3 - Les enjeux de la FAD.....	25
1.1.3.1 - L'autonomie de l'apprenant.....	25
1.1.3.2 - L'accompagnement de l'apprenant.....	26
1.1.3.3 - Le développement des dispositifs de support pour la FAD.....	27
1.2 - LA TELEFORMATION	29
1.2.1 - Présentation générale.....	30
1.2.2 - Typologie des contextes d'utilisation de la formation en ligne en France.....	32
1.2.2.1 - Les sociétés spécialisées	32
1.2.2.2 - La téléformation en entreprise	33
1.2.2.3 - L'université et la téléformation.....	34
1.2.3 - Les plates-formes	36
1.2.3.1 - Présentation générale	36
1.2.3.2 - Acteurs et fonctionnalités d'un dispositif de téléformation.	40
1.3 – NORMES ET STANDARDS DE LA E-FORMATION.....	43
1.3.1 - Les travaux de normalisation et de standardisation	43
1.3.1.1 Au niveau international	43
1.3.1.2 En France	46
1.3.2 - Le modèle SCORM.....	47
1.3.2.1 - Description et Organisation de SCORM.....	47
1.3.2.2 - Pourquoi un modèle de référence commun.....	49
1.3.2.3 – L'environnement d'exécution de SCORM.....	50
CHAPITRE 2 : FONDEMENTS THEORIQUES	53
2.1 - APPRENTISSAGE COLLABORATIF.....	55
2.1.1 - La variété des échelles.....	56
2.1.3 - La variété des significations données au mot «apprentissage»	57
2.1.4 - La variété des significations données au mot : collaboration.....	60
2.2 - LE GROUPE ET LES INTERACTIONS.....	62
2.3 - LE ROLE DU COORDONNATEUR ET DU TUTEUR	65
2.4 - LA THEORIE DE L'ACTIVITE.....	68
2.4.1 - Les niveaux d'une activité.....	69
2.4.2 - Les propriétés de l'activité.....	72
2.4.3 - La structure basique d'une activité.....	73
2.5 - L'ECHELLE D'ANALYSE DE LA TACHE DANS NOTRE RECHERCHE.....	76
2.6 - LES RAISONS DU CHOIX DU PARADIGME D'AGENT.....	77
2.7 - L'APPROCHE COMPOSANTS DANS LES EIAH.....	80
2.7.1 Définitions.....	80
2.7.2 Quelques types de composants pour les systèmes d'information	81
2.7.3 Démarches d'ingénierie de systèmes à base de composants.....	84
2.8 - TERMINOLOGIE	86
2.8.1 - PIFAD, SIGFAD, SimuLigne.....	86

2.8.2 - Groupe / Cours:.....	87
2.8.3 - Espaces / e-monde / e-univers.....	87
2.8.4 - e-personnalité.....	88
2.8.5 - Utilisateurs et statuts PIFAD.....	88
2.8.6 - Acteur SimuLigne.....	90
2.8.7 - Implémentation des fonctions et statuts SimuLigne dans PIFAD :.....	92
2.8.8 - Module de formation / Etape / Activité / Tâche.....	93
2.8.9 - Outil PIFAD.....	94
2.8.10 - Agent / Système Multi-Agents / rôle MaSE / Tâche informatique concurrente / Objectif MaSE	95
2.8.11 - Interaction / Interactivité / Communication.....	96
CHAPITRE 3 - L'EXPERIMENTATION SIMULIGNE : LE MODELE CONCEPTUEL D'ACTIVITE ET L'ANALYSE DES DONNEES D'INTERACTIONS DANS LES GROUPES	99
3.1 - SIMULIGNE.....	101
3.1.1 - Contexte de départ.....	101
3.1.2 - Principes et objectifs pédagogiques de SimuLigne.....	103
3.1.2.1 - La simulation globale.....	103
3.1.2.2 - Le scénario prétexte de SimuLigne.....	105
3.1.2.3 - Modalités de formation.....	106
3.1.2.4 - Modalité d'évaluation à l'issue de la formation, bilan.....	115
3.2 - LES INSUFFISANCES CONCEPTUELLES DES PLATES-FORMES DE TELE-FORMATION	116
3.2.1 - Le modèle conceptuel d'activité dans SimuLigne.....	117
3.2.1.1 - Introduction de concepts complémentaires.....	117
3.2.1.2 - Structure d'une activité dans SimuLigne.....	119
3.2.1.3 - Le modèle conceptuel d'activité.....	119
3.3 - L'ANALYSE DES DONNEES D'INTERACTION OBTENUES	124
3.3.1 - Les données d'interaction relatives aux groupes.....	125
3.3.2 - Les données d'interaction relatives aux utilisateurs du groupe Lugdunensis.....	130
3.4 - LIMITES DE PIFAD ET PROPOSITION D'UN SMA : SIGFAD.....	135
3.4.1 - Les limites des plates-formes existantes.....	135
3.4.2 - Les fonctionnalités de SIGFAD.....	138
3.4.2.1 - Surveiller l'activité du groupe et des apprenants.....	139
3.4.2.2 - Indiquer l'état de réalisation d'une activité.....	139
3.4.2.3 - Gestion des forums par les tuteurs.....	140
CHAPITRE 4 : AGENTS, SYSTEMES MULTI-AGENTS ET METHODOLOGIES.....	141
4.1 - AGENTS.....	144
4.1.1 - Définitions.....	144
4.1.2 - Typologie des agents.....	148
4.1.2.1 - Les agents collaboratifs.....	151
4.1.2.2 - Les agents d'interface.....	151
4.1.2.3 - Les agents mobiles.....	152
4.1.2.4 - Les agents d'information.....	153
4.1.2.5 - Les agents logiciels réactifs.....	153
4.1.2.6 - Les agents hybrides.....	154
4.2 - LES SYSTEMES MULTI-AGENTS.....	154
4.2.1 - Définitions.....	156
4.2.2 - Raisonnement et communication dans les SMA.....	156
4.2.2.1 - Distribution des tâches et des compétences.....	157
4.2.2.2 - Protocoles de communication.....	158
Le modèle acteur.....	158
Abcl.....	159
Contract net protocol.....	159
4.2.2.3 - Modes de coopération.....	160
4.2.3 - Méthodologies et plates-formes de développement des systèmes à base d'agents.....	161
4.2.3.1 - La méthodologie AAIL.....	162
4.2.3.2 - La méthodologie GAIA.....	163
4.2.3.3 - Agent UML.....	164
4.2.3.4 - Cassiopea.....	164
4.2.3.5 - Utilisation du langage Z.....	164

4.2.3.6 - La méthodologie Prometheus.....	165
4.2.3.7 - La méthodologie MadKit.....	166
4.3 - APPRENTISSAGE MACHINE ET SYSTEMES MULTI-AGENTS.....	167
4.3.1 <i>Généralités</i>	167
4.3.2 <i>Définition et principes</i>	168
4.3.3 <i>L'apprentissage par les agents</i>	170
4.3.3.1 - Apprentissage mono-agent et apprentissage multi-agents.....	170
4.3.3.2 - Les méthodes d'apprentissage hors connexion et en ligne.....	170
4.3.4 <i>Les techniques d'apprentissage</i>	171
4.3.4.1 - Apprentissage logique et apprentissage réactif.....	171
4.3.4.2 - L'apprentissage social.....	172
4.4 - LES OUTILS DE DEVELOPPEMENT DE SIGFAD.....	172
4.4.1 <i>La méthodologie Multiagent System Engineering (MaSE)</i>	172
4.4.1.1 - La phase d'analyse.....	174
4.4.1.2 - Phase de conception.....	178
4.4.2 <i>L'architecture JAM</i>	179
4.4.2.1 - L'interpréteur.....	181
4.4.2.2 - Les buts.....	181
4.4.2.3 - Les plans.....	182
4.4.2.4 - Le modèle du monde.....	183
4.4.2.5 - L'observateur.....	183
4.4.3 <i>La communication entre les agents</i>	183
4.4.3.1 - Sockets, RMI et CORBA.....	184
4.4.3.2 - JATLite.....	185
CHAPITRE 5 : ANALYSE, CONCEPTION, SPECIFICATION ET IMPLEMENTATION DE SIGFAD	189
5.1 - SIGFAD.....	192
5.1.1 <i>Analyse, conception et spécification de SIGFAD</i>	192
5.1.1.1 - Le diagramme hiérarchisé des objectifs de SIGFAD.....	193
5.1.1.2 - L'application des cas d'utilisation.....	197
5.1.1.3 - Le diagramme de rôles de MaSE et les tâches concurrentes.....	205
5.1.1.4 - Le diagramme des classes d'agents de SIGFAD.....	210
5.1.1.5 - La construction des communications inter-agents de SIGFAD.....	211
5.1.1.6 - La conception du système.....	211
5.1.2 <i>Implémentation de SIGFAD</i>	213
5.1.2.1 - Le développement des agents suivant le modèle JAM.....	213
5.1.2.2 - L'interface agent – BDD.....	216
5.1.2.3 – Les interfaces SIGFAD – utilisateur.....	217
L'interface tuteur – SIGFAD.....	218
L'interface coordonnateur – SIGFAD.....	220
L'interface apprenant – SIGFAD.....	220
5.1.2.4 – La communication entre les agents.....	222
5.1.2.5 – Les difficultés rencontrées dans l'implémentation de SIGFAD.....	223
5.2 – L'APPLICATION DE SIGFAD AUX DONNEES DE SIMULIGNE.....	224
5.2.1 – <i>Les informations fournies sur les apprenants</i>	228
5.2.2 – <i>Les informations portant sur le groupe tout entier</i>	229
5.2.3 – <i>Les informations portant sur les activités</i>	231
5.3 – VERS UNE PRISE EN COMPTE DES DONNEES D'INTERACTION DANS LA NORMALISATION ET LA STANDARDISATION DE LA E-FORMATION.....	232
BILAN, PERSPECTIVES ET CONCLUSION	235
BILAN DES TRAVAUX.....	238
RESULTATS OBTENUS.....	240
APPORTS DE NOS TRAVAUX DE THESE.....	241
PERSPECTIVES.....	243
CONCLUSION.....	244
ANNEXES	247
A - LES PRODUCTIONS DE SIMULIGNE.....	249

B – APPLICATION DE LA METHODOLOGIE MASE A L'ANALYSE, LA CONCEPTION ET LA SPECIFICATION DE SIGFAD	251
<i>B.1 Les diagrammes de séquence décrivant les cas d'utilisation</i>	251
<i>B.2 Les automates à états finis décrivant les tâches concurrentes</i>	253
C – APPLICATION DE SIGFAD AUX DONNEES DE SIMULIGNE	259
D – CODE SOURCE DE SIGFAD	260
<i>D – 1 Code source de l'agent superviseur des interactions</i>	261
<i>D.2 - Exemple de code source d'un agent tuteur : L'agent du tuteur du groupe Aquitania</i>	266
<i>D.3 - Code de ReceiveQueryTutor.java</i>	274
<i>D.4 – Un exemple de la primitive : AbsentsInGroup</i>	275
<i>D.5 - Code de la primitive SendResultToTutor.java</i>	277
LISTE DES ACRONYMES	279
LISTE DES FIGURES	283
LISTE DES TABLEAUX	284
INDEX DES MOTS CLES ET D'AUTEURS CITES	285
REFERENCES BIBLIOGRAPHIQUES	287
BIBLIOGRAPHIE	289
SITES INTERNET	294
TABLE DES MATIERES	295

Résumé

Nos travaux de thèse s'inscrivent dans les EIAH (Environnements Informatiques d'Apprentissage Humain) et plus particulièrement dans le contexte de la formation à distance (FAD) médiatisée par les ordinateurs et les réseaux informatiques. Nous nous intéressons au soutien à apporter aux utilisateurs afin qu'ils puissent mener à bien leurs activités pédagogiques et contribuer à la réussite de la session de formation. Comme nous nous situons plus particulièrement dans une pédagogie basée sur une production collaborative dans des groupes de taille réduite (8 à 15 personnes) ; l'existence et la vitalité du groupe, l'apparition d'une quantité minimale d'interactions entre les membres du groupe sont pour nous des enjeux majeurs. C'est pourquoi nous mettons un accent particulier sur la compréhension de la tâche du tuteur et proposons des outils informatiques capables de l'aider à apprécier l'état de son groupe et le bon déroulement des activités pédagogiques.

Notre approche méthodologique basée sur une démarche expérimentale (nous avons participé à un projet pluridisciplinaire dont le point focal était la conduite d'une session de FAD baptisée SimuLigne) nous a permis de comprendre que l'usage des outils informatiques à mettre à la disposition des utilisateurs doit épargner autant que possible les ressources humaines, déjà si cruciales en FAD. Nous avons eu recours au paradigme d'agent pour proposer un système multi-agents qui regrouperait un ensemble de fonctionnalités qui, de notre point de vue, manquent actuellement aux plates-formes de téléformation et que l'on viendrait y greffer. C'est ainsi que nous avons identifié et mis en œuvre un ensemble d'outils pour développer SIGFAD, notre système multi-agents. SIGFAD a été analysé, conçu et spécifié suivant une méthodologie de développement de système multi-agents (MaSE et l'environnement associé AgentTool), les agents ont été construits suivant une l'architecture JAM basée sur le modèle BDI (*beliefs – desires – intentions*). Les agents ont été programmés en Java et ils communiquent à travers des *sockets* écrites dans le même langage. La volonté d'appliquer toutes les étapes du cycle logiciel (de l'analyse à l'implémentation) à SIGFAD nous a donné l'occasion de constater la carence d'outils de développement adaptés à l'implémentation de systèmes à base d'agents.

SIGFAD permet d'obtenir la liste des utilisateurs présents, absents et dormants (utilisateurs qui se connectent à la plate-forme mais ne laissent aucune trace de leur passage pour les autres membres de leur groupe) sur une période donnée. Le prototype donne l'appréciation de l'état du groupe (en fonction du nombre de personnes dites actives dans le groupe), l'état d'un apprenant (sa productivité en termes d'activités validées et sa sociabilité en fonction de sa propension à lire ou écrire des messages), et le déroulement d'une activité donnée (activité non débutée, en cours, non validée, validée). Nous appliquons SIGFAD au corpus de données recueillies durant SimuLigne et montrons ainsi l'usage qui peut être fait de notre prototype au cours d'une session de formation à distance.

Le souci de rendre SIGFAD suffisamment générique pour être couplé à différentes plates-formes nous a conduit à nous intéresser aux questions de normalisation et de standardisation de la e-formation. Nous proposons ainsi les données que l'on devrait retrouver dans les plates-formes de télé-formation afin d'améliorer le suivi des activités en apprentissage collaboratif distribué.

Mots clés : *EIAH, formation à distance, apprentissage collaboratif, interactions, tutorat, paradigme agent, systèmes multi-agents.*