# Une approche Multi-agents à Architecture P2P pour l'apprentissage collaboratif

Hanaa Mazyad

▶ **To cite this version:**

Hanaa Mazyad. Une approche Multi-agents à Architecture P2P pour l'apprentissage collaboratif. Autre [cs.OH]. Université du Littoral Côte d'Opale, 2013. Français. NNT : 2013DUNK0321 . tel-00845225

**HAL Id: tel-00845225**
**https://theses.hal.science/tel-00845225**

Submitted on 16 Jul 2013

**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.

**UNIVERSITE DU LITORAL CÔTE D'OPALE**

**LABORATOIRE D'INFORMATIQUE SIGNAL ET IMAGE DE LA CÔTE D'OPALE**

## THESE

Pour l'obtention du titre de

Docteur de l'Université du Littoral Côte d'Opale

Spécialité : Informatique

# Une Approche Multi-agents à Architecture P2P pour l'Apprentissage Collaboratif

Thèse soutenue le 31 janvier 2013

par

Hanaa MAZYAD

Devant le jury composé de :

| | | |
|---|---|---|
| M. Claude GODART | Professeur à l'Université de Lorraine | Rapporteur |
| M. Mourad OUSSALAH | Professeur à l'Université de Nantes | Rapporteur |
| Mme Zahia Guessoum | Maître de Conférences HDR à l'Université de Reims | Examinateur |
| M.Christophe Renaud | Professeur à l'Université du Littoral Côte d'Opale | Examinateur |
| M. Henri BASSON | Professeur à l'Université du Littoral Côte d'Opale | Directeur |
| Mme. Insaf TNAZEFTI-KERKENI | Maître de Conférences à l'Université du Littoral Côte d'Opale | Co-directeur |

# Résumé

Les Systèmes multi-agents ou SMA proposent une approche originale de conception de systèmes intelligents et coopératifs. Ils se caractérisent par la distribution du contrôle global du système et par la présence d'agents autonomes évoluant dans un environnement partagé et dynamique. De plus, il existe plusieurs interdépendances entre les buts des agents, leurs capacités et les ressources qu'ils utilisent, donc afin d'éviter d'éventuels conflits, de favoriser la synergie des activités des agents et de partager les ressources de l'environnement commun, il est important que les agents coordonnent leurs actions.

Nous nous sommes intéressés, dans le cadre de cette thèse, au déploiement des systèmes multi-agents sur une architecture pair à pair (Peer-to-Peer ou P2P) et ceci dans le but d'établir la communication entre ces agents et les relier entre eux. Cependant, du fait de la nature dynamique des systèmes P2P où chaque pair peut apparaître et disparaître à tout moment, des nouveaux problèmes se posent pour la coordination d'agents nécessitant des mécanismes de coordination adaptés au contexte spécifique des P2P. Pour cela, nous avons proposé une méthode de formation de groupe comme solution à ces problèmes. Nous nous sommes intéressés ensuite à l'application des systèmes multi-agents à architecture P2P au domaine d'apprentissage collaboratif en ligne où des apprenants contribuent aux apprentissages du groupe, et en retour, le groupe contribue à ceux des apprenants et c'est la cohérence du collectif qui permet d'atteindre l'objectif. Cependant, l'apprentissage collaboratif à distance implique des nouveaux rôles pour l'enseignant ainsi que pour les apprenants. Il est donc essentiel de définir ces rôles pour identifier les besoins qui en découlent pour pouvoir intégrer à l'outil informatique des fonctionnalités afin de satisfaire ces besoins. En effet, il est essentiel de fournir aux enseignants et apprenants la possibilité d'avoir des informations sur la progression de leur apprentissage ainsi que sur les niveaux de collaboration et de sociabilité de chaque apprenant et du groupe. Enfin, nous avons proposé, comme application de nos travaux, un système appelé COLYPAN (COllaborative Learning sYstem for Project mANagment) conçu pour l'apprentissage à distance et de façon collaborative de la gestion des projets.

**Mots clés :** SMA, P2P, Coordination, Formation de groupes, Apprentissage collaboratif

# Abstract

Multi-Agents systems (MAS) propose an original approach to design intelligent and cooperative systems. They are characterized by the distribution of the overall system control and the presence of autonomous agents operating in a shared and dynamic environment. In addition, there are many interdependencies between: agents' goals, their abilities and used resources. So, in order to avoid possible conflicts, promote synergy of agents activities and share resources of the common environment, it is important that the agents coordinate their actions.

We are interested, in the context of this thesis, in the deployment of multi-agents systems on Peer-to-Peer (P2P) networks in order to establish communication between these agents. However, because of the dynamic nature of P2P systems where each peer may appear and disappear at any time, new problems arise concerning the coordination of agents. Thus, coordination mechanisms adapted to the specific context of P2P are required. For that, we have proposed a group formation method to solve these problems. Then, we were interested in the usage of multi-agents systems with P2P architecture in the field of collaborative e-learning. In such applications, each learner contributes in the learning process of the group, and in return, the group contributes in the learning process of its members. The consistency of the whole group allows to achieve the goal. However, collaborative e-learning implies new roles for teachers as well as for learners. It is therefore essential to define these roles in order to identify the users needs and integrate, in the platform, the functionalities that allows us to satisfy such needs. Indeed, it is essential to provide teachers and learners with the opportunity to obtain information about the progress of their learning processes as well as the level of collaboration and sociability of each learner in the group. Finally, we have proposed, as an application of our work, a system called COLYPAN (COllaborative Learning sYstem for Project management) designed for the collaborative e-Learning project management.

**Keywords**: MAS, P2P, Coordination, Group formation, Collaborative learning

A mon père,
ma mère,
mon mari et ma fille,
mes frères et sœurs,
ma famille et mes amis,
Je dédie cette thèse.

# Remerciements

Mes remerciements vont avant tout aux membres de jury qui ont accepté d'évaluer ce travail de thèse. Je remercie vivement et particulièrement : M. Claude Godart, professeur à l'Université de Lorraine et M. Mourad Oussalah, professeur à l'Université de Nantes, qui ont accepté de juger ce travail et d'en être les rapporteurs.

Je tiens à exprimer ma reconnaissance à Mme Zahia Guessoum, maître de conférences à l'université de Reims, d'avoir accepté d'être l'examinatrice de ce travail.

Je remercie également M. Christophe Renaud, directeur du Laboratoire d'Informatique, Signal et Image de la Côte d'opale (LISIC), pour avoir accepté d'examiner mon travail de thèse. Je le remercie aussi pour m'avoir accepté comme doctorante au sein du laboratoire ainsi que pour son soutien et son encouragement durant ces années de doctorat.

Je remercie M. Henri Basson, professeur à l'Université du Littoral Côte d'Opale et directeur de la thèse, pour son soutien de tous les jours qui m'a permis de mener à terme ces travaux. Je tiens tout particulièrement à lui exprimer ma plus profonde gratitude pour avoir dirigé ces travaux et m'avoir soutenu dans cette étude. La pertinence de ses remarques sur mes travaux de recherche et son expérience m'ont beaucoup appris. Je le remercie pour son extrême gentillesse et sa disponibilité de tous les instants.

Je remercie Mme Insaf Tnazefti-Kerkeni, pour avoir co-dirigé mes travaux durant ces années de thèse et l'année de DEA ainsi que pour l'ensemble de conseils qu'elle m'a prodigué. Son expérience m'a été grandement profitable.

Je remercie mes collègues au LISIC pour les encouragements et l'aide qui m'ont permis de finir cette thèse et surtout ceux qui m'ont fait profiter de leur grande expérience et avec qui j'ai eu de nombreuses discussions enrichissantes. Un merci particulier à Mourad, Virginie, Denis, Grégory, Arnaud, Bénédicte et Magali.

Je souhaite aussi remercier les doctorants du B125, les anciens et les nouveaux, mes compagnons de labeur qui ont su contribuer à rendre ces années agréables en y maintenant une ambiance chaleureuse. Je remercie particulièrement Adeel, Oussama K. et Hala.

Merci également à tous les collègues et personnels de l'ULCO de Calais et de l'IUT de Calais-Boulogne pour leur extrême gentillesse.

Une seule ligne ne saurait exprimer ma gratitude envers ma mère, mes frères Khalil et Ahmad et mes sœurs Hiba, Rawan et Layla qui m'ont soutenu et encouragé pendant toutes ces années d'études. Merci pour votre présence à mes côtés à chaque instant.

Un grand merci également à ma famille, ma belle-famille et mes amis qui ont su être présents dans les moments difficiles et sans qui cette thèse n'aurait pas été possible. Mhamad, merci pour tes encouragements. Merci également à Racha et Ibrahim.

Je remercie mon mari Oussama qui a su toujours m'apporter du soutien quand j'en avais besoin. Il est évident que je n'aurais pas pu terminer cette thèse sans son appui indéfectible. Je le remercie, pour son amour sans faille, pour sa patience et sa générosité, du fond de mon cœur. Merci à ma fille, Luna, qui m'apporte tant de bonheur dans ma vie.

Cette thèse a été ton rêve pour moi...Tu as été présent à chaque instant dans mon cœur et mes pensées, cher père malgré ton absence et j'espère être à la hauteur de ce que tu as espéré pour moi.

Qu'ils trouvent ici le témoignage de ma reconnaissance.

Calais, le 1er juin 2012

# Table des matières

**Résumé** ..... **3**

**Abstract** ..... **4**

**Chapitre I. Introduction générale** ..... **13**

1.1 Cadre de la thèse ..... 13

1.2 Contexte de recherche et éléments de problématique ..... 13

1.3 Objectifs de la thèse ..... 15

1.4 Organisation du document ..... 16

**Chapitre II. Agents et Systèmes multi-agents** ..... **19**

2.1 Avant-propos ..... 19

2.2 Le concept d'agent ..... 19

2.2.1 Agent versus objet ..... 19

2.2.2 Caractéristiques des agents ..... 20

2.2.3 Classification des agents ..... 21

2.3 Système multi-agents ..... 22

2.3.1 Définition d'un système multi-agents ..... 22

2.3.2 Caractéristiques d'un système multi-agents ..... 23

2.3.2.1 Organisation sociale ..... 23

2.3.2.2 Interaction ..... 24

2.3.2.3 La coopération ..... 24

2.3.2.4 La négociation ..... 25

2.3.2.5 La coordination ..... 25

2.3.2.6 La communication ..... 26

2.3.3 Intérêts des systèmes multi-agents ..... 26

2.4 Méthodologies de conception des systèmes multi-agents ..... 27

2.4.1 Les méthodologies utilisant UML et constituant une extension des méthodes orientées objet ..... 27

2.4.1.1 MaSE ..... 28

2.4.1.2 AAII ..... 31

2.4.2 Les méthodologies organisationnelles ..... 32

2.4.2.1 Aalaadin ..... 32
2.4.2.2 GAIA ..... 36
2.4.3 Les méthodologies formelles ..... 37
2.4.3.1 DESIRE ..... 38
2.4.3.2 PASSI ..... 38
2.5 Comparaison des méthodologies de conception des SMA ..... 39
2.6 Domaines d'application des systèmes multi-agents ..... 41
2.7 Conclusion ..... 42
**Chapitre III. Le P2P pour le déploiement des SMA ..... 45**
3.1 Avant-propos ..... 45
3.2 Introduction ..... 45
3.3 Présentation des systèmes P2P ..... 45
3.3.1 Caractéristiques des systèmes P2P ..... 46
3.3.2 Fonctionnement du P2P ..... 47
3.3.3 Client-serveur versus P2P ..... 48
3.4 La coordination des agents sur une architecture P2P ..... 50
3.4.1 Discussion ..... 51
3.5 Les plateformes de développement ..... 52
3.5.1 JXTA ..... 53
3.5.2 JADE ..... 55
3.5.3 Madkit ..... 57
3.5.4 Comparaison des plateformes ..... 59
3.6 Conclusion ..... 61
**Chapitre IV. L'apprentissage : de la théorie à la pratique ..... 62**
4.1 Avant-propos ..... 62
4.2 L'apprentissage à distance et l'apprentissage en ligne ..... 62
4.3 Les théories d'apprentissage ..... 63
4.3.1 Le béhaviorisme ..... 63
4.3.2 Le cognitivisme ..... 63
4.3.3 Le constructivisme ..... 64
4.4 L'apprentissage collaboratif ..... 64

4.4.1 Différence entre apprentissage collaboratif et coopératif ..... 65
4.4.2 Habilités de l'apprenant et rôles du tuteur ..... 65
4.4.3 Le groupe dans l'apprentissage collaboratif ..... 66
4.5 Plateformes d'e-Learning classiques ..... 66
4.5.1 Sakai ..... 67
4.5.2 Claroline ..... 68
4.5.3 Moodle ..... 69
4.5.4 Discussion ..... 70
4.6 Travaux similaires ..... 71
4.6.1 Travaux utilisant le paradigme objet ..... 71
4.6.2 Travaux utilisant le paradigme agent ..... 72
4.6.3 Discussion ..... 74
4.7 Conclusion ..... 76
**Chapitre V. COLYPAN : un système d'apprentissage collaboratif de la gestion de projet . ..... 80**
5.1 Avant-propos ..... 80
5.2 Introduction ..... 80
5.3 Rôles et besoins du tuteur ..... 81
5.4 Rôles et besoins de l'apprenant et du groupe ..... 84
5.5 La formation de groupe ..... 84
5.5.1 Méthode adoptée pour la formation d'un groupe ..... 85
5.5.2 La division récursive en sous groupes selon les compétences ..... 86
5.6 Pédagogie par projet ..... 87
5.6.1 MAETIC ..... 88
5.7 Le système d'apprentissage collaboratif COLYPAN ..... 90
5.8 Description du système ..... 90
5.9 Objectifs du système ..... 91
5.10 Modélisation du système ..... 91
5.10.1 Fonctionnalités du système ..... 92
5.10.2 La phase de conception ..... 94
5.10.2.1 La conception des bases ..... 94
5.10.2.2 La conception des groupes ..... 94

5.10.2.3 Fonctionnement des groupes ..... 97
5.11 Conclusion ..... 98
**Chapitre VI. Implémentation du système COLYPAN ..... 100**
6.1 Introduction ..... 100
6.2 Implémentation de COLYPAN ..... 100
6.2.1 Les agents du système ..... 101
6.2.2 L'interface de COLYPAN ..... 103
6.2.3 COLYPAN Pas-à-Pas ..... 105
6.3 Le scénario fictif ..... 106
6.3.1 Les principes de COLYPAN ..... 107
6.3.2 Déroulement de l'apprentissage ..... 108
6.4 Conclusion ..... 112
**Chapitre VII. Conclusion et perspectives ..... 114**
7.1 Conclusion ..... 114
7.2 Contributions de nos travaux de thèse ..... 114
7.3 Perspectives ..... 116
**Liste des figures ..... 118**
**Liste des tableaux ..... 119**
**Bibliographie ..... 121**

# Chapitre I. Introduction générale

## 1.1 Cadre de la thèse

Les travaux de cette thèse ont été effectués au sein de l'équipe ModEL (Modélisation et Évolution du Logiciel) du Laboratoire d'Informatique Signal et Image de la Côte d'opale (LISIC). Les travaux de l'équipe ModEL s'articulent autour du contrôle de l'évolution du logiciel. Ils ont pour principal objectif l'amélioration des processus de développement et d'évolution des logiciels ainsi que leur assurance qualité. Une partie de l'équipe Model s'intéresse plus particulièrement au développement de modèles et plateformes destinés aux activités de l'enseignement et à l'ingénierie pédagogique dont s'inscrivent les travaux de cette thèse.

## 1.2 Contexte de recherche et éléments de problématique

Les Technologies de l'Information et de la Communication pour l'Enseignement (TICE) ont permis aux développeurs et aux utilisateurs dispersés dans le temps ou l'espace de collaborer pour améliorer leur créativité et l'efficience de ce qu'ils ont constitué comme groupes de travail. Des espaces plus ou moins ouverts de collaboration ont été créés dans différents domaines d'activités où les facilités des échanges sont génératrices de synergie et de motivation mutuelle entre les acteurs de la collaboration. Ainsi dans un domaine tel que l'E-learning, l'efficacité d'un apprentissage collaboratif dépend de la motivation de ses membres à collaborer. Elle dépend également de leur compétences, de la qualité des outils de collaboration dont ils disposent, et du temps qu'ils peuvent consacrer aux échanges. En effet, une démarche collaborative d'apprentissage signifie que chaque apprenant coopère pour le développement et l'avancement des connaissances de l'ensemble de groupe. En retour, le groupe délivre un ensemble de connaissances structurées et enrichies à disposition de chaque apprenant. La cohérence du collectif devient ainsi primordiale pour atteindre l'objectif de l'avancement de l'ensemble dans son apprentissage et la responsabilité de cette cohérence concerne chaque apprenant.

Cependant, l'apprentissage collaboratif entre acteurs distants implique des nouveaux rôles, on peut à l'évidence y distinguer, les deux rôles majeurs incluant le rôle d'enseignant ainsi que celui d'apprenant. Au bon fonctionnement de chaque rôle correspondent des besoins spécifiques à définir avec précision. Le but étant de concevoir et de réaliser des fonctionnalités, dans la plateforme partagée par l'ensemble de collaborateurs, permettant de satisfaire adéquatement leurs besoins. En prenant le cas de l'apprenant à distance, il est autonome, il est souvent sans contact direct avec l'enseignant ou les autres apprenants. Il est donc supposé gérer efficacement les ressources offertes par la plateforme utilisée ainsi que ses interactions avec les autres apprenants. L'apprenant est également supposé bien gérer son temps, le rythme de sa progression, et son acquisition des savoirs et des savoir-

faire. L'autonomie de l'apprenant est un aspect principal à prendre en compte dans tout apprentissage collaboratif. Cependant, cette autonomie ne peut être considérée comme un acquis, puisqu' elle ne dépend pas uniquement des apprenants car il faut d'abord créer le cadre où peut s'exercer cette autonomie [chasseneuil, 2000] [Oubahssi, 2005]. En considérant la collaboration comme un cadre d'application P2P (Pair à Pair), par exemple, la volatilité du nombre des pairs complique davantage l'apprentissage. Sans l'accompagnement de l'apprenant dans la fluctuation de sa détermination et sa motivation, son apprentissage peut tendre vers un échec au vu du ralentissement de son rythme d'acquisition des connaissances. Il est alors essentiel de faciliter la tâche de l'enseignant en lui permettant à tout moment d'observer l'état et les activités de chaque apprenant. Ceci permet à l'enseignant, en fonction de la progression du travail observé du groupe, de les motiver et les guider pour prévenir l'échec ou l'abandon des apprenants.

Pour cette problématique d'apprentissage collaboratif à distance, les Systèmes Multi-Agents (SMA) proposent une approche intéressante de conception des systèmes intelligents et coopératifs. Ils se caractérisent par la distribution du contrôle global du système et par la présence d'agents autonomes évoluant dans un environnement partagé et dynamique. Pour les connaissances réparties dans les différents composants du système, les agents n'en ont qu'une vision partielle. Ceci provoque un accroissement de l'incertitude dans les arguments associés aux actions menées par chaque agent. Le manque dans les connaissances dont il dispose le mène souvent à prendre de mauvaises décisions. D'autre part, il existe plusieurs interdépendances entre les buts des agents, leurs capacités et les ressources qu'ils utilisent. Afin d'éviter d'éventuels conflits, des règles de coordination favorisant la synergie des activités des agents doivent être adoptées. Elles permettent, entre autres, de partager les ressources de l'environnement commun et une coordination de leurs différentes actions. La coordination s'avère ainsi comme essentielle pour tout groupe d'agents en lui évitant tout risque de dégradation de ses activités ou sa dérive vers un groupe collaboratif chaotique.

Dans ce contexte d'apprentissage collaboratif, les apprenants et les enseignants sont considérés comme étant à la fois producteur et fournisseur de l'information. Il est alors essentiel de fournir aux enseignants et apprenants la possibilité d'avoir des informations sur les niveaux de collaboration, de la sociabilité de chaque apprenant, et permettre à chaque apprenant de percevoir sa propre situation d'apprentissage.

Nous nous sommes intéressés dans ce contexte au déploiement du système multi-agents sur une architecture Pair à Pair (Peer-to-Peer ou P2P) dans le but de créer des facilités adaptées de communications entre les agents. Ce choix vient du fait que, dans une architecture P2P, chaque Pair possède une autonomie significative, et qu'un Pair peut se joindre à différents groupes de Pairs afin de réaliser différents buts. En plus, certains concepts du P2P, tels que le groupe de Pairs et l'autonomie, sont des concepts typiques des systèmes multi-agents. Enfin, le concept de groupe de Pairs s'avère de grande adéquation pour représenter l'aspect dynamique des organisations sociales au sein des systèmes multi-agents.

Nous nous sommes ensuite intéressés à l'application des systèmes multi-agents à architecture P2P au domaine d'apprentissage collaboratif en ligne. 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'entre-aide et de soutien mutuel et comme lieu privilégié d'interaction pour la construction collective des connaissances.

Ce choix est motivé par le fait que les systèmes multi-agents se prêtent bien à la conception d'un environnement d'apprentissage où chaque membre doit gérer et échanger ses connaissances et collaborer avec les autres afin de réaliser ses buts. En plus, dans des tels environnements ouverts, dynamiques et complexes, il y a un besoin de distribuer les données, le contrôle ainsi que l'expertise ce qui rend l'utilisation des systèmes P2P bénéfique.

Enfin, nous avons proposé, comme application de nos travaux, un système appelé COLYPAN (Collaborative Learning sYstem for Project mANagment) conçu pour l'apprentissage à distance et de façon collaborative de la gestion des projets. Ce système implémente la méthode MAETIC (Méthode pédAgogique InstrumentEe par les TIC) réalisés au sein de notre laboratoire et visant à mettre en place une pédagogie de groupe par projet. MAETIC décrit un ensemble de démarches formalisées et appliquées selon des principes définis. L'objectif de MAETIC est ainsi de permettre à un étudiant de développer les savoirs et savoir-faire sollicités par la mise en œuvre d'un processus de développement d'un « produit » et de l'entraîner aux techniques de gestion de projet. Pour l'enseignant, l'objectif de MAETIC est de favoriser la mise en place d'une démarche qui facilitera ses actions pédagogiques.

## 1.3 Objectifs de la thèse

Cette thèse est le résultat d'un intérêt pour les systèmes complexes, ouverts et dynamiques tels que les systèmes d'apprentissage à distance. Les caractéristiques de ces systèmes rendent les SMA très adaptés pour modéliser, concevoir, et implémenter ces systèmes. En effet, la distribution des agents, des données et du contrôle associés, ainsi que les interactions qui ont lieu entre les agents et qui facilitent le travail collectif rend très intéressante l'utilisation de SMA. L'adoption d'une architecture P2P à la place d'une architecture client-serveur pour déployer le système le rend plus ouvert mais renvoie certains problèmes particulièrement au niveau de la coordination d'agents.

L'objectif premier de cette thèse est de résoudre ces problèmes de coordination d'agents dans le contexte d'une architecture P2P. Plusieurs mécanismes de coordination existent dans la littérature. Cependant, nous avons choisi la coordination par formation de groupe comme solution.

Le deuxième objectif est d'utiliser cette association SMA et P2P pour concevoir un système d'apprentissage collaboratif à distance. Ce deuxième objectif renvoie le problème

d'identifier les nouveaux rôles qu'impose ce contexte d'apprentissage collaboratif à distance à l'enseignant, l'apprenant, le groupe ainsi que les besoins qui en découlent.

Le troisième objectif est celui d'analyser les besoins d'un système collaboratif pour en concevoir des fonctionnalités à intégrer au système dédié à satisfaire ces besoins. Ces fonctionnalités ont trois buts. Le premier est de faciliter la tâche du tuteur en satisfaisant ses besoins et en lui permettant d'apprécier les activités des apprenants et de suivre leur progression dans l'apprentissage ainsi que leur niveau de collaboration. Le deuxième est d'éviter l'isolement des apprenants et de les motiver et le dernier est d'encourager le travail collaboratif et l'utilisation des outils permettant le travail collaboratif tels que le forum, le chat, la messagerie instantanée.

## 1.4 Organisation du document

Le document de mémoire est organisé en deux parties dont la première comprend trois chapitres consacrés respectivement aux systèmes multi-agents, aux systèmes P2P et à la problématique de l'apprentissage collaboratif. La deuxième partie, composée des deux chapitres descriptifs des apports des travaux et de leur validation. Le document se termine par les conclusions et les perspectives des travaux.

**Le chapitre 2** est consacré aux systèmes multi-agents. Nous y définissons les notions d'agents, de systèmes multi-agents, et nous présentons leurs caractéristiques afin de montrer l'importance de l'organisation et des interactions et surtout de la coordination dans ces systèmes. Nous dressons ensuite une comparaison entre les méthodologies orientées-agent afin de choisir une qui soit la plus adaptée pour la conception d'un système ouvert, distribué et dynamique tel que celui d'e-Learning qui s'avère être un domaine intéressant des applications des SMA.

**Le chapitre 3** présente les systèmes P2P ainsi que leurs caractéristiques. La nature ouverte et dynamique de ces systèmes où les pairs apparaissent et disparaissent à tout moment complique davantage la coordination d'agents et rend les mécanismes de coordination inadaptés dans un contexte P2P. Nous nous intéressons en particulier à la coordination par formation de coalitions.

**Le chapitre 4** est consacré à l'apprentissage et plus particulièrement celui de l'apprentissage collaboratif en ligne. Nous présentons quelques plateformes d'e-Learning ainsi que d'autres travaux dans ce domaine. Ainsi dans le but de situer nos travaux par rapport à l'existant, nous essayons de voir ce qui a été fait concernant la prise en compte de l'apprentissage collaboratif. Celui-ci a requis l'examen des outils proposés pour le travail en groupe et le suivi des activités des apprenants.

**Le chapitre 5** est une élaboration des rôles de l'enseignant, de l'apprenant et du groupe imposés par l'apprentissage collaboratif à distance. Cette étude nous permettra d'identifier les difficultés liées à ces rôles et donc de définir les besoins pour surmonter ces difficultés. Ainsi à l'aide de la méthodologie Aalaadin, nous analysons les besoins afin de proposer des

fonctionnalités capables de les satisfaire à intégrer au système. Nous proposons aussi dans ce chapitre une méthode de formation de groupe permettant la coordination d'agents dans le contexte des systèmes P2P.

**Le chapitre 6** est consacré à l'implémentation du système d'apprentissage collaboratif COLYPAN à l'aide de la plateforme multi-agents MadKit. Nous proposons donc un scénario pour simuler l'utilisation du système et montrer ainsi toutes ses fonctionnalités.

Le document se termine par les conclusions générales sur les contributions, les apports ainsi que les perspectives des travaux.

# Chapitre II. Agents et Systèmes multi-agents

## 2.1 Avant-propos

De nos jours, les systèmes multi-agents ont pris une place de plus en plus importante en informatique, que ce soit dans le domaine de l'intelligence artificielle, dans celui des systèmes informatique distribués, de la robotique et du génie logiciel. C'est une discipline qui s'intéresse aux comportements collectifs produits par les interactions de plusieurs entités autonomes et flexibles appelés agents. Ce chapitre introduit d'abord, les notions d'agents et de systèmes multi-agents (SMA), il détaille par la suite les méthodologies de modélisation de SMA. Le chapitre se termine par un aperçu des différents domaines d'application des SMA.

## 2.2 Le concept d'agent

Le concept d'agent est d'une généralité qui le rend difficile à définir. Il existe en effet plusieurs définitions ou significations données à cette notion. Nous présentons ci-dessous quelques unes :

Jennings et Wooldridge [Jennings et Wooldridge, 1998] définissent l'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.

Deloach et al [Deloach et al, 2001] pour leur part définissent un agent comme un ensemble de processus qui communiquent entre eux pour atteindre un objectif donné.

Selon Chaib-Draa [Chaib-Draa et al, 2001], un agent est un système informatique, situé dans un environnement, qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu.

Nous retenons la définition de Ferber [Ferber, 1989] qui définit un agent comme étant une entité (physique ou virtuelle) capable d'agir sur elle-même et sur son environnement, disposant d'une représentation partielle de cet environnement, pouvant communiquer avec d'autres agents et dont le comportement est la conséquence de ses objectifs, de sa perception, de ses représentations, de ses compétences et des communications qu'elle peut avoir avec les autres agents.

### 2.2.1 Agent versus objet

Plusieurs chercheurs définissent la notion d'agent par rapport à celle d'objet, leur argumentation se base sur le fait que la maîtrise d'une nouvelle technologie ou d'un nouveau concept est facilitée par sa 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 et qui sont capables d'accomplir des actions (ou méthodes dans cet état) tout en communicant par échange de messages. Bien qu'il existe

plusieurs similitudes entre les agents et les objets, il n'en ait pas moins qu'il existe d'importantes différences [Wooldridge, 2002]:

La première concerne le degré d'autonomie : les agents comme les objets peuvent encapsuler leur état interne, et agir sur cet état par le biais de leurs méthodes. Pour les objets, on assimile parfois une invocation de méthode à un envoi de message mais une telle méthode doit être explicitement invoquée par un autre objet pour agir tandis qu'un agent recevant une information décidera par lui-même de l'action subséquente [Wooldridge, 2002 ; Jarras et Chaib-Draa, 2002]. La seconde différence entre les objets et les agents provient de la flexibilité du comportement d'un agent. Bien qu'il soit possible de construire des objets disposant de ces caractéristiques, on notera que le modèle standard d'un objet ne dit rien à propos de ces types de comportements [Wooldridge, 2002 ; Jarras et Chaib-Draa, 2002]. La troisième différence entre les objets et les agents vient du fait que ces derniers sont censés avoir chacun son fil d'exécution (thread) [Wooldridge, 2002 ; Jarras et Chaib-Draa, 2002]. 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 programmation orientée-objet (POO) avec des langages comme Java offrant 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 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 [Hikolo, 2003]. Odell et al. [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 et refuser de répondre à une sollicitation.

### 2.2.2 Caractéristiques des agents

Il existe de nombreuses caractéristiques relatives aux agents qui procurent la force au paradigme agent et qui les distinguent des autres paradigmes logiciels tels que les systèmes orientés objets, les systèmes distribués et les systèmes experts. Nous citons les plus intéressantes afin d'avoir une meilleure vision d'un agent, qui cependant, ne possède pas forcément toutes ces caractéristiques :

- l'autonomie : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne ;
- la capacité représentationnelle ou la perception: l'agent peut avoir une vision très locale de son environnement, mais aussi une représentation plus large de cet environnement et notamment des agents qui l'entourent ;
- la 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 ;
- la réactivité : l'agent reçoit des informations de son environnement, et il doit être capable de réagir en conséquence ;

- l'intentionnalité ou pro-activité : un agent ne réagit pas simplement aux changements de l'environnement mais il fait preuve d'un comportement dirigé vers ses buts et prend des initiatives lorsqu'elles lui semblent appropriées ;
- la communication avec les autres agents, plus ou moins étendue ;
- l'anticipation : l'agent possède plus ou moins les capacités d'anticiper les événements futurs ;
- la rationalité : les agents rationnels disposent de critères d'évaluation de leurs actions et de choix des meilleures pour atteindre le but ;
- l'adaptabilité ou la flexibilité: un agent adaptable est un agent capable de moduler ses aptitudes selon l'état de l'environnement. L'agent est capable d'apprentissage et sait résoudre un nouveau problème à partir de son expérience ;
- la sociabilité : un agent interagit avec les autres agents (logiciels et humains) quand la situation l'exige afin d'accomplir ses tâches et aide les autres à atteindre leurs buts.

### 2.2.3 Classification des agents

Des chercheurs issus de l'intelligence artificielle distribuée comme Moulin et Chaib-Draa [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. Il ne possède pas de représentations de lui-même, de son environnement ou des autres agents ;
- un agent intentionnel est une entité qui perçoit et agit sur son environnement, qui est capable de raisonner à propos de ses intentions et ses croyances, qui possède des buts explicites et qui 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.

Ferber [Ferber, 1995] classe les agents en fonction de leur caractère cognitif ou réactif. Les agents cognitifs, dont la plupart sont intentionnels, disposent chacun d'une base de connaissances comprenant l'ensemble d'informations et de savoir-faire leur permettant de réaliser leur tâche et de gérer les interactions avec les autres agents et avec leur environnement. Ils sont capables d'anticipation et peuvent planifier leur comportement. Par contre, les agents réactifs sont incapables d'anticiper ou de planifier leur comportement. Il les classe aussi en fonction de leurs comportements téléonomiques dirigés vers des buts explicites ou bien des comportements réflexes régis par les perceptions.

Nwana [Nwana et al, 1996] propose sept catégories d'agents :

- Les agents réactifs ;

- Les agents collaboratifs : sont des agents autonomes et qui coopèrent et négocient avec les autres agents afin d'atteindre des ententes lors de la résolution distribuée de problèmes;
- Les agents d'interface : ces agents fournissent une assistance à l'utilisateur. Un agent d'interface est capable de s'adapter aux habitudes de l'utilisateur et à ses préférences ;
- Les agents mobiles : ce sont des agents qui peuvent se déplacer d'un site à un autre en cours d'exécution pour se rapprocher des données ou des ressources [Bernard et al, 2002];
- Les agents d'information ou d'Internet : ces agents ont pour rôle d'administrer, manipuler ou collecter les informations à partir de plusieurs sources d'informations distribuées. ;
- Les agents hybrides : Un agent hybride consiste en la combinaison de plusieurs caractéristiques des autres agents au sein d'un même agent tels que la mobilité, la collaboration, l'autonomie, la capacité à apprendre, etc.
- Et les agents intelligents : Nwana pense que les agents logiciels intelligents n'existent réellement pas encore mais d'après Jennings et Wooldridge [Jennings et Wooldridge, 1998], un agent intelligent se caractérise par son autonomie, sa réactivité, sa capacité à agir ainsi que sa sociabilité.

Nous considérons dans nos travaux que les agents sont intelligents selon la définition de Jennings et Wooldridge.

Dans la section suivante, nous nous intéressons aux systèmes multi-agents. Nous présentons ses caractéristiques ainsi que l'intérêt de les utiliser.

## 2.3 Système multi-agents

Les systèmes multi-agents ont vu le jour avec l'avènement de l'intelligence artificielle distribuée (IAD). A la différence de l'intelligence artificielle classique (IA), qui modélise le comportement intelligent d'un seul agent, l'intelligence artificielle distribuée (IAD) s'intéresse à des comportements intelligents qui sont les produits de l'activité coopérative de plusieurs agents. Le passage du comportement individuel aux comportements collectifs est considéré, non seulement comme une extension, mais aussi un enrichissement de l'intelligence artificielle, d'où émergent de nouvelles propriétés et de nouveaux comportements.

Les SMA permettent d'introduire dans un système, un ensemble d'agents qui sont capables d'interagir et sont dotés de connaissances, d'intentions et de capacité d'évolution différente.

### 2.3.1 Définition d'un système multi-agents

Un système multi-agents est un ensemble d'entités (physiques ou virtuelles) appelées agents, partageant un environnement commun (physique ou virtuel), qu'elles sont capables

de percevoir et sur lequel elles peuvent agir. Les perceptions permettent aux agents d'acquérir des informations sur l'évolution de leur environnement, et leurs actions leur permettent entre autres de modifier l'environnement [Guessoum, 2003]. Les agents interagissent entre eux directement ou indirectement, et exhibent des comportements corrélés créant ainsi une synergie permettant à l'ensemble d'agents de former un collectif organisé.

Selon [Chaib-Draa & al, 2001], un système multi-agents est un système distribué composé d'un ensemble d'agents interagissant, le plus souvent, selon des modes de coopération, de concurrence ou de coexistence.

Tandis que [Ferber, 1995] le définit comme un système composé d'un environnement E, d'un ensemble d'objets O, d'un ensemble A d'agents, d'un ensemble de relation R qui unissent des objets (et donc des agents) entre eux, d'un ensemble d'opérations OP permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler les objets de O et 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.

Parmi les différentes définitions des systèmes multi-agents nous retiendrons celle d'Yves Demazeau [Demazeau, 1995], un SMA est un système composé d'agents qui évoluent dans un environnement et interagissent de manière organisée. Ainsi, un SMA est défini selon l'équation ci-dessous :

SMA = Agents + Environnement + Interactions + Organisations (AEIO)

### 2.3.2 Caractéristiques d'un système multi-agents

Les Systèmes multi-agents sont issus du domaine de l'intelligence artificielle distribuée. Ainsi, dans un SMA, les traitements et les données sont distribués mais aussi les compétences, les rôles et les buts des agents. Ces agents ont chacun un point de vue partiel et il n'y a aucun contrôle global du système. En plus, ils sont en activité permanente et prennent leurs propres décisions en fonction de leurs objectifs et leurs connaissances et sont capables de communiquer entre eux selon des langages plus ou moins élaborés.

Les agents sont situés dans un environnement où ils interagissent en vue de réaliser conjointement une tâche ou d'atteindre conjointement un but particulier. Ainsi, c'est la notion d'interaction qui distingue un SMA d'une collection d'agents indépendants [Chaib-draa, 199]. Par ailleurs, cette notion est une caractéristique principale d'un SMA en plus de l'organisation qui est produite par l'interaction entre les agents.

#### 2.3.2.1 Organisation sociale

Un système multi-agents est caractérisé par son organisation sociale [Gasser, 1990] c'est à dire la manière dont le groupe est constitué, à un instant donné, pour pouvoir fonctionner. Cette organisation décrit l'ensemble de composants fonctionnels du système, leurs natures, leurs responsabilités et leurs besoins en ressources ainsi que les liens de communication entre les agents. Selon [Ferber, 1999], toute organisation est le résultat d'une interaction

entre agents et le comportement des agents est contraint par l'ensemble de structures organisatrices.

Dans un système ouvert, une organisation n'a pas nécessairement un objectif global et les agents la constituant, ne partagent pas forcément les mêmes intérêts et peuvent être en concurrence. En effet, les agents ont une activité qui n'est pas nécessairement convergente. L'organisation nécessite donc un minimum de structuration afin que l'ensemble, formé par des membres à priori hétérogènes, soit cohérent [Ferber, 1999].

#### 2.3.2.2 Interaction

La caractéristique principale d'un système multi-agents réside dans la notion d'interaction pouvant être définie comme une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques [Ferber, 1999]. Les interactions s'expriment ainsi à partir d'une série d'actions dont les conséquences exercent en retour une influence sur le comportement futur des agents. Par ailleurs, la motivation des agents pour interagir dépend de leurs buts, leurs capacités à réaliser certaines tâches et les ressources dont ils disposent.

L'interaction est par nature distribuée et elle a lieu généralement entre une entité informatique autonome (un agent) et son environnement. Celui-ci pouvant être peuplé, soit d'autres agents, soit d'opérateurs humains, soit d'objets inanimés. Elle est importante dans la mesure où c'est à travers elle que les agents peuvent combiner les efforts et s'entraider ou bien entrer en compétition et négocier [Chaib-Draa et Demolombe, 2002]. En effet, dans un SMA, les agents peuvent coexister, être en compétition ou coopérer.

S'ils ne font que coexister dans le même environnement alors les agents ne communiquent qu'indirectement et ceci en ayant une perception les uns des autres ce qui permet à chaque agent de mieux éviter les autres agents. S'ils sont en compétition, alors chaque agent agit afin de satisfaire ses propres buts ce qui se fait généralement aux dépens des autres agents et donc engendre des conflits que les agents se voient obliger de résoudre habituellement en négociant. Cependant, si les agents sont en coopération, alors ils interagissent pour réaliser leur propre but mais aussi pour contribuer à la réussite du groupe et ceci en communiquant, en s'échangeant des informations sur l'environnement ou sur leurs intentions afin que chaque agent ait une idée de ce que les autres font.

Nous présentons dans ce qui suit les différentes formes d'interaction entres agents à savoir : la coopération, la négociation, la coordination et la communication.

#### 2.3.2.3 La coopération

La coopération est la forme générale d'interaction. Une activité de coopération [Bond, 1990] prend place si un ensemble particulier d'agents converge vers un but global par l'achèvement de leurs propres buts locaux. Dans ce cas, la coopération est une attitude intentionnelle adoptée par les agents [Ferber, 1999]. Cependant, certains auteurs [Durfee et al 1989] [Bouron, 1992] considèrent la coopération du point de vue d'un observateur extérieur qui interprète à posteriori les comportements des agents en les qualifiants de

coopératifs ou non à partir de critères sociaux, tels que l'interdépendance des actions ou le nombre de communications effectuées et ceci sans avoir accès à leurs états mentaux.

La coopération permet à un agent d'accomplir des tâches qu'il ne peut pas réaliser tout seul et d'améliorer l'utilisation des ressources. Ceci revient alors à augmenter la productivité de chaque agent et par conséquent celle du système et à améliorer les performances, autrement dit, augmenter le nombre de tâches réalisées et diminuer le temps de réalisation d'une tâche [Ferber, 1999].

#### 2.3.2.4 La négociation

La négociation [Durfee et Lesser, 1989] est le processus d'améliorer les accords en réduisant les inconsistances et l'incertitude sur des points de vue communs ou des plans d'action grâce à l'échange structuré d'informations pertinentes. C'est une composante de base de l'interaction surtout parce que les agents sont autonomes [Jennings et al, 2000].

Les activités des agents dans un système distribué sont souvent interdépendantes et entraînent des conflits. Pour les résoudre, il faut considérer les points de vue des agents, les négocier et utiliser des mécanismes de décision concernant les buts sur lesquels le système doit se focaliser [Chevrier, 1992]. En négociant, un agent peut soit trouver un compromis soit modifier les croyances d'autres agents pour faire prévaloir son point de vue.

Un des protocoles les plus étudiés pour la négociation est le protocole « Contract Net » ou réseau contractuel [Smith, 1988]. Ce protocole s'appuie sur une métaphore organisationnelle : les agents coordonnent leurs activités grâce à l'établissement de contrats afin d'atteindre des buts spécifiques.

#### 2.3.2.5 La coordination

La coordination est une question centrale pour les SMA et la résolution de systèmes distribués [Chaib-draa et al, 2001]. Elle peut être définie comme l'articulation des actions individuelles accomplies par chacun des agents de manière à ce que l'ensemble aboutisse à un tout cohérent et performant [Ferber, 1999]. Les actions des agents doivent être coordonnées [Jennings, 1996] pour plusieurs raisons : 1) Il y a des dépendances entre les actions des agents ainsi dans le cas contraire, les agents peuvent agir indépendamment les uns des autres; 2) Aucun agent n'a suffisamment de compétence, de ressources et d'information pour atteindre tout seul le but du système complet ; 3) Il faut éviter les redondances dans la résolution de problèmes.

La coordination améliore l'action du groupe en diminuant les conflits et en augmentant les performances. En effet, elle permet aux agents de bénéficier des capacités des autres agents et à la communauté d'éviter de dégénérer en une collection chaotique d'individus [Nwana et al, 1996]. Cependant, pour que la coordination soit efficace, il est important que les agents puissent éviter les interactions négatives survenant entre eux.

Nous nous intéressons particulièrement à la coordination des agents puisque c'est une caractéristique très importante pour les SMA. Nous revenons là-dessus dans le prochain chapitre.

#### 2.3.2.6 La communication

La communication dans les systèmes multi-agents est à la base des interactions et de l'organisation des agents mais aussi de la résolution coopérative des problèmes [Vernadet et Azena, 1992]. Elle permet de synchroniser les actions des agents et de résoudre les conflits de ressources et de buts par la négociation. En effet, sans communication, les agents sont incapables de coopérer, de négocier, de coordonner leurs actions ou de réaliser des tâches en commun. Deux modes de communication se distinguent dans la littérature : la communication indirecte qui se fait par transmission de signaux via l'environnement et la communication directe qui correspond aux échanges des messages entre les agents.

Des langages de communication ont vu le jour afin de faciliter l'échange et l'interprétation des messages et l'interopérabilité entre les agents. Le premier langage qui a été introduit est KQML (pour Knowledge Query and Manipulation Language) [Labrou et Finin, 1997]. Le deuxième est le FIPA-ACL (FIPA[1] Agent Communication Language) qui est une extension du langage KQML.

### 2.3.3 Intérêts des systèmes multi-agents

L'utilisation des Systèmes multi-agents (SMA) présente une série d'avantages [Brandolese et al, 2000]. Les systèmes multi-agents héritent des bénéfices envisageables du domaine de l'intelligence artificielle comme le traitement symbolique (au niveau des connaissances), la facilité de maintenance, la réutilisation et la portabilité ainsi que des avantages traditionnels de la résolution distribuée de problèmes : [Chaib-Draa et al, 2001] [Le Bars, 2003]

- L'ouverture : c'est la capacité d'ajouter ou de retirer dynamiquement dans le système des agents ou des fonctionnalités et des services [Klein, 2009]. Ainsi, il est impossible de savoir lors de la conception quels seront les composants du SMA ni comment ils vont interagir les uns avec les autres [Wooldridge, 2001].
- L'interaction : les SMA permettent de modéliser un ensemble d'agents qui interagissent. Ces agents ont une influence localement sur le comportement des autres agents, généralement sur un pied d'égalité. On rencontre de nombreuses interactions entre agents telles que la coordination (organiser la résolution d'un problème de telle sorte que les interactions nuisibles soient évitées, ou que les interactions bénéfiques soient exploitées), la négociation (parvenir à un accord acceptable pour toutes les parties concernées), la coopération (travailler ensemble à la résolution d'un but commun) [Chaib-Draa et al, 2001]. Cette approche est particulièrement bien adaptée à la simulation des systèmes complexes dont le fonctionnement global émerge des actions des individus. Les SMA permettent de faire vivre virtuellement des agents autonomes sur ordinateur et d'y effectuer des expériences difficiles, voire impossible à mener dans la réalité, d'où la qualification de laboratoires virtuels [Treuil et Mullon, 1997].

[1] *Foundation for Intelligent Physical Agents – http://www.fipa.org*

- La souplesse de l'outil informatique : elles permettent de rendre la programmation plus simple. Elles permettent aussi de modifier le comportement des agents, d'ajouter ou de supprimer des actions possibles [Le Bars, 2003].
- La décentralisation : la décentralisation des agents permettent au SMA de supporter l'échec individuel d'un des éléments ou l'ajout d'un nouveau sans dégrader le système dans son ensemble. Le fonctionnement du système ne dépend en effet d'aucun agent particulier. Il n'y a pas de contrôle global : le contrôle et les responsabilités sont partagés entre les différents agents, ce qui le rend fiable, robuste et flexible.
- La résolution distribuée de problèmes : un problème peut être décomposé en sous-parties, chacune pouvant être résolues de façon indépendante pour aboutir à une solution stable.
- La vitesse d'exécution: elle est principalement due au parallélisme, car plusieurs agents peuvent travailler en même temps pour la résolution d'un problème.

Nous consacrons la section suivante à l'étude des méthodologies de conception des SMA. En effet, cette étude nous permettra de choisir une méthodologie pour concevoir notre SMA.

## 2.4 Méthodologies de conception des systèmes multi-agents

Les systèmes multi-agents coopératifs dans lesquels les agents doivent interagir ensemble pour accomplir leurs buts est un domaine de recherche très actif et extrêmement riche. Cette richesse rend les SMA plus complexes et conduit à de très nombreux modèles d'agent, d'environnement, d'interaction et d'organisation qui sont d'ailleurs souvent combinés au sein d'un même système multi-agents. Cependant, la maîtrise de la complexité des SMA impose une certaine discipline dans les méthodologies de développement de ces derniers. Or, la facilité avec laquelle les applications pourront être développées rend la diffusion industrielle ou commerciale des SMA plus rapide.

L'évolution des méthodologies de modélisation de SMA est contrainte par la difficulté à séparer nettement l'agent des autres paradigmes logiciels (en l'occurrence les objets), la capacité à concevoir des systèmes ouverts et enfin la possibilité de prévoir la mise à l'échelle du système [Wooldridge, 2001]. Aussi le manque d'architectures et de langages d'agents standard gêne l'opérationnalisation des modèles. En effet, la conception d'agents doit être adaptée à chaque architecture d'agents. L'analyse quant à elle est indépendante des architectures d'agents, elle définit ce que fait le système et pas comment il le fait.

De nombreuses méthodologies ont été proposées pour le développement des systèmes multi-agents. Nous les classons en trois catégories.

### 2.4.1 Les méthodologies utilisant UML et constituant une extension des méthodes orientées objet

Depuis l'apparition d'UML en 1997, l'envie d'utiliser une notation normalisée et claire pour décrire les systèmes multi agents s'est faite plus urgente. UML est donc le premier candidat puisqu'il est désormais la norme pour la programmation Objet.

Plusieurs travaux ont vu le jour pour vérifier l'applicabilité d'UML à la description du SMA et ils n'ont pas hésité de proposer des augmentations ou des spécifications d'UML pour le rendre apte à décrire des SMA.

#### 2.4.1.1 MaSE

MaSE (*Multiagent Software Engineering*) est une approche complète pour le développement des systèmes multi-agents de l'analyse au déploiement [Deloach 1999] [Deloach 2001]. L'objectif de MaSE est d'aider le concepteur à analyser, concevoir et implémenter un SMA à partir d'un cahier de charges initial.

Elle utilise de nombreux modèles graphiques pour décrire le type d'agents et les interfaces inter-agents [Deloach 2002]. En effet, de nombreux diagrammes sont utilisés dans cette méthodologie. Ils sont très proches de ceux utilisés dans la conception orientée objet, avec quelques caractéristiques de plus et quelques modifications de la sémantique du paradigme objet pour pouvoir capter les concepts d'agent et les comportements coopératifs des agents (*figure II-1*) :

- *la hiérarchie de buts* est obtenue en décomposant le but principal du système en sous-buts à partir du cahier des charges. Elle est définie grâce à un arbre dont les nœuds sont les buts et les liens représentent les décompositions;

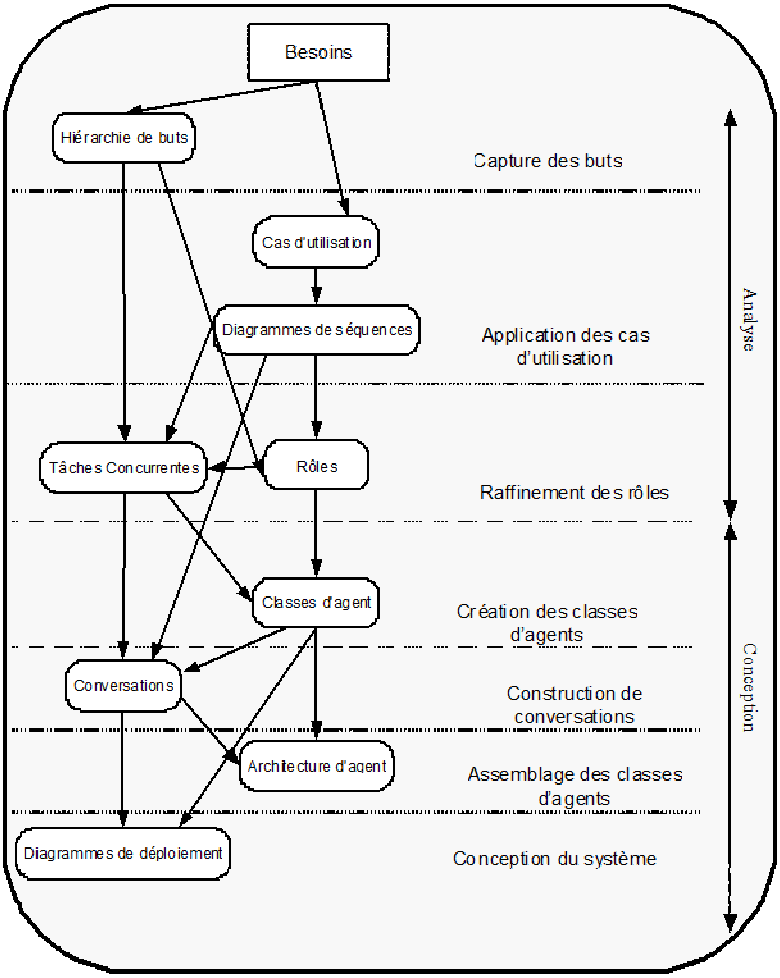


*Figure II-1 Les différentes étapes et les modèles de MaSE*

- *les cas d'utilisation* qui définissent des scénarios de base que le système peut exécuter. Ce modèle est souvent retrouvé dans les processus de conception objet afin de décrire à la fois la fonctionnalité du système et son environnement en termes d'utilisateurs et d'acteurs. Dans MaSE, il permet notamment d'identifier les *rôles* joués par le système (et donc ses agents). Les cas d'utilisation apparaissent au même niveau que les buts ;
- *les diagrammes de séquences* décrivent les cas d'utilisations comme un ensemble d'événements survenant entre les rôles constituant le système. Dans ce modèle, les interactions entre rôles sont décrites comme des envois de messages. Ces diagrammes de séquences sont quasi similaires aux diagrammes UML, (Unifieded Modelling Language) [Muller, 1997], avec la différence que ce sont des rôles et non des classes qui communiquent
- *les rôles* constituent la base de la définition des classes d'agents et sont vus comme le moyen d'atteindre les buts fixés au système. Généralement, un but est associé à un rôle unique, mais ceci n'est pas une règle systématique ;
- *les tâches concurrentes* décrivent les moyens pour les rôles d'atteindre les buts. Elles sont spécifiées au moyen des automates à états finis ;

- *les classes d'agents* sont créées à partir des rôles identifiés. Une classe d'agent est une extension d'une classe d'objet qui peut jouer plusieurs rôles, et qui possède des attributs et des méthodes. Les diagrammes de classes d'agents sont des diagrammes de classes UML ;
- *les conversations* définissent en détail les interactions entre agents. Une interaction consiste en deux automates à états finis, un pour l'initiateur et un pour le récepteur ;
- *l'architecture d'agent* consiste en un ensemble de composants qui peuvent être des classes ou des ensembles de classes. Elle détaille les différents types d'agents ou objets présents dans le système ;
- *les diagrammes de déploiement* permettent de spécifier le type et la localisation des agents sur les plateformes utilisées ainsi que les liens de discussion nécessaires entre les plates-formes.

MaSE fournit un outil, *agentTool*, permettant une génération partielle du code [Deloach 2000]. Elle utilise deux langages pour décrire les agents et les systèmes multi-agents : *Agent Modeling language* (AgML) et *Agent Definition Language* (AgDL).

MaSE se décompose en sept étapes réparties en deux phases (*figure II.1*). Chaque étape a pour résultat un ou plusieurs diagrammes. La phase d'analyse se décompose en trois étapes :

1. **Identification des rôles** à partir des besoins de l'utilisateur. Le résultat de cette étape est le diagramme hiérarchisé des buts ;
2. **Identification des cas d'utilisations et création des diagrammes de séquences** associés pour aider à l'identification d'un ensemble initial de rôles et de voies de communication ;
3. **Raffinement des rôles** dont le résultat est :
    a. le modèle des rôles de MaSE, permettant de saisir les rôles et les tâches associées
    b. un ensemble de diagrammes de tâches concurrentes associés à chaque tâche permettant de définir les comportements des rôles.

La phase de conception comporte quatre étapes :

1. **Création de classes d'agents** dont le résultat est le diagramme de classes d'agents qui montre l'organisation du système en classes d'agents et les interactions entre elles en examinant les modèles de tâches concurrentes et en se basant sur les rôles joués par chaque classe ;
2. **Construction des conversations** permettant de définir le détail des interactions entre agents ceci en extrayant les messages et les états définis dans les diagrammes de tâches concurrentes ;
3. **Assemblage des classes d'agents** à partir de l'architecture de chaque classe d'agent en utilisant des composants et des connecteurs. Il faut alors s'assurer que chaque action

apparaissant dans une conversation est bien implémentée comme une méthode au sein de l'architecture d'agent ;

4. **Conception du système** en utilisant des diagrammes de déploiement pour l'instanciation des classes d'agents préalablement définies.

***Discussion***

MaSE a l'avantage de couvrir tout le cycle de vie du système. La définition des étapes est très pertinente et les différents modèles et les livrables à produire sont clairement identifiables. Cependant, MaSE est difficilement applicable au-delà de dix classes d'agents. En effet, MaSE ne permet pas l'ouverture du système : il n'y a pas de création, de destruction ou de déplacement d'agent pendant l'exécution du système.

#### 2.4.1.2 AAII

La méthodologie AAII (*Australian Artificial Intelligence Institute)* a été développée en se basant sur l'expérience accumulée dans les travaux effectués pour la gestion du trafic aérien [Kinny 1996]. Elle constitue un mélange de concepts des méthodologies orientées objet avec certains concepts orientés agents. Cette méthodologie vise à construire un ensemble de modèles qui définissent des spécifications du système d'agent à base d'architecture BDI (Belief, Desires, Intentions – Croyance, Désirs, Intentions) [Rao et al, 1995].

AAII fournit deux modèles :

1. Le modèle *externe* (le point de vue externe) : il a pour but de définir les relations entre les classes d'agents, l'organisation, les rôles, les services et les responsabilités. Il est divisé en deux sous-modèles :
    a. Le modèle d'agents qui se divise aussi en deux modèles. Le premier est le modèle de classes d'agents qui décrit l'hiérarchie entre agents, le second est le modèle d'instances d'agents qui identifie les multiples instances possibles.
    b. Le modèle d'interaction qui décrit les responsabilités, les services et les interactions entre les agents.
2. Le modèle *interne* (le point de vue interne) : il représente les comportements des agents. Il est décrit suivant les trois modèles suivants :
    a. Le *modèle de croyance* (*belief*) qui décrit les informations concernant l'environnement, l'état interne des agents et leurs actions possibles ;
    b. Le *modèle de but* (goal) qui décrit les buts que l'agent peut adopter et les évènements auxquels il peut répondre ;
    c. Le *modèle de plans* qui décrit sous forme de plans, les moyens que l'agent peut utiliser pour atteindre ses buts.

Les différents modèles de cette méthodologie reposent principalement sur des notations de type UML. Les modèles d'agents et d'instances sont décrits moyennant des diagrammes de

classes ou de collaboration. La hiérarchie entre les agents et l'organisation est définie grâce aux relations de composition, d'agrégation ou d'héritage.

Le processus de développement d'un SMA selon AAII est composé de quatre étapes :

1. **Identification des rôles de l'application** : définition de l'organisation structurelle et fonctionnelle ainsi qu'une hiérarchie des classes d'agent ;
2. **Identification des responsabilités et des services** qui sont liés à chaque rôle d'agent *;*
3. **Identification des interactions** : définir les actes de langage nécessaire à la communication et c'est à ce niveau que la définition interne de chaque agent peut commencer ;
4. **Raffinement de la hiérarchie d'agents** : introduire de nouvelles classes ou sous-classes d'agents, en utilisant la composition des classes. Enfin, établir les instances d'agents nécessaires.

***Discussion***

AAII ne prend pas en compte les systèmes ouverts (en termes de nombre variable d'agents), et impose une hiérarchie au sens objet du terme. De plus, elle exige l'utilisation de l'architecture BDI qui est contrainte par l'utilisation de simples graphes ET/OU pour modéliser les buts. Elle ne présente aucun support pour la conception et le développement.

L'utilisation de cette méthodologie n'est pas très difficile si l'on connaît un peu la conception orientée objet. Cependant, AAI est une méthode assez ancienne, et peu de ressources sont disponibles pour des concepteurs désirant l'utiliser.

### 2.4.2 Les méthodologies organisationnelles

Elles s'intéressent à la notion d'agent et des sociétés d'agents. Elles partent du principe qu'un système multi agent est un ensemble d'agent autonomes, possédant chacun un ou plusieurs buts, et, surtout, socialement situé.

#### 2.4.2.1 Aalaadin

Aalaadin (*figure II-2*) est une approche organisationnelle pour le développement des SMA fondée sur le concept AGR (Agent, Groupes et Rôles [Ferber 2003]. Il s'agit, d'une part, d'un cadre de développement des systèmes multi-agents, fournissant des indications méthodologiques et d'autre part, d'un environnement de prototypage et d'exécution pour des agents reposant sur les notions du modèle AGR [Ferber 1998].

Un agent est une simple entité autonome qui joue des rôles dans des groupes. Il peut avoir plusieurs rôles et être membre de plusieurs groupes. L'agent dans Aalaadin n'a aucune contrainte ou pré-requis ni sur l'architecture interne de l'agent ni sur son modèle de comportement.

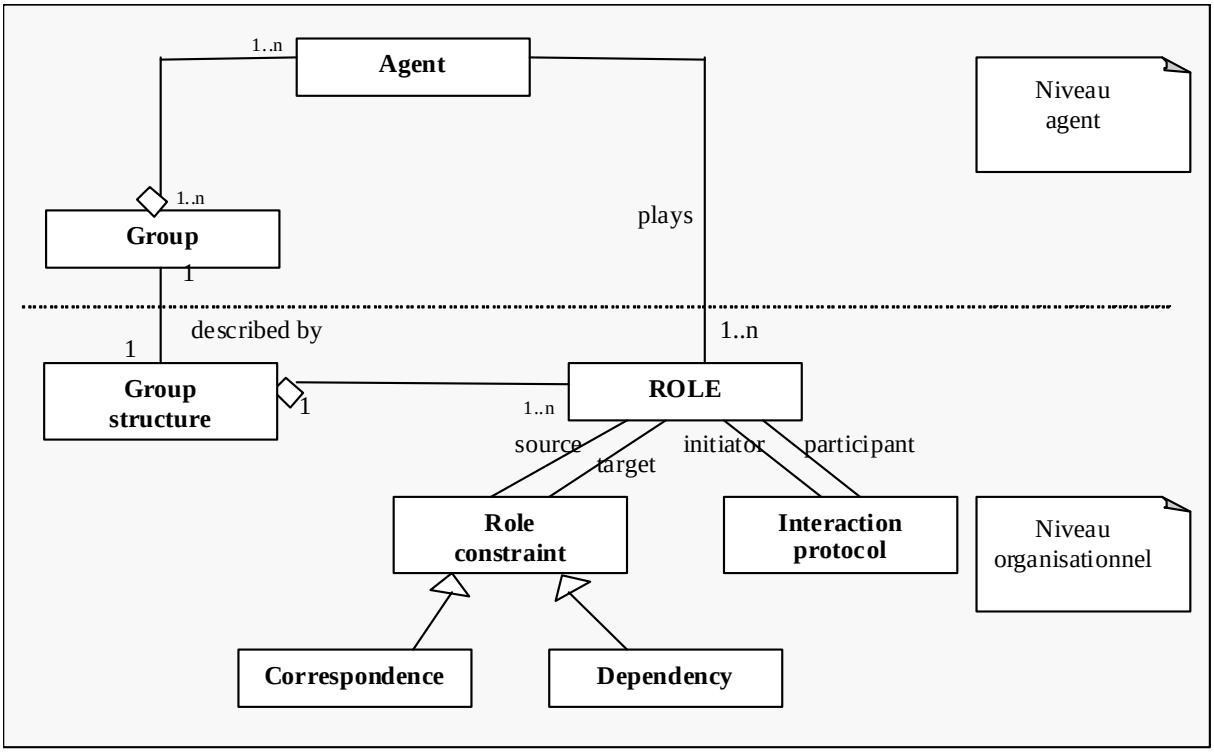


*Figure II-2 Le modèle organisationnel*

Un agent est identifié dans son groupe par un rôle qui est une représentation abstraite d'une fonction ou d'un service qui définit un ensemble de contraintes que l'agent jouant ce rôle doit respecter. Un rôle peut être joué par plusieurs agents.

Un groupe est un regroupement d'ensembles d'agents partageant une caractéristique. La communication entre deux agents appartenant aux groupes différents n'est pas possible.

Aalaadin propose une vision centrée organisation plutôt que centrée agent pour des raisons de sécurité, de normes et de responsabilités [Ferber 2003]. Une organisation, au sens d'Aalaadin, est un ensemble de groupes liés entre eux par des agents qui jouent un rôle dans un des groupes et un autre dans un autre groupe. Une organisation est découpée en deux niveaux (*figure II-3*) :

- *La structure organisationnelle* représente les relations qui font d'une agrégation d'individus un tout identifiable, c'est-à-dire, un invariant organisationnel ;
- *L'organisation concrète* correspond à une organisation décomposée en groupes. Elle est une instance de structure organisationnelle avec des agents des rôles et des groupes particuliers.

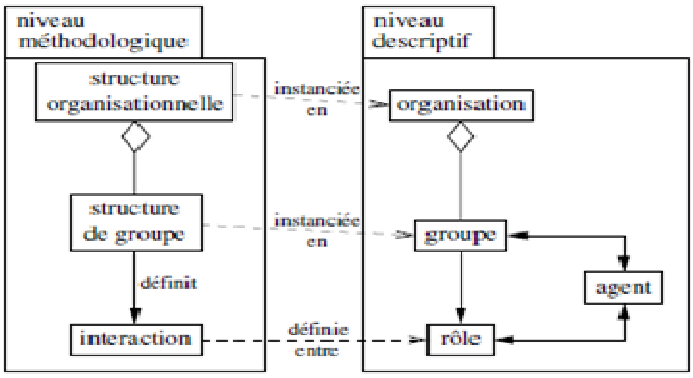


*Figure II-3 Les concepts du modèle Aalaadin*

Trois types de diagrammes sont utilisés dans cette méthodologie (*figure II-4*) :

1. **Les diagrammes de structures organisationnelles** dans lesquelles sont représentés les structures de groupes (rectangles), les rôles (hexagones), les contraintes (flèches) et les interactions (rectangles) entre rôles.
2. **Les diagrammes d'organisations concrètes** permettent de représenter l'appartenance d'agents (quilles) qui jouent des rôles dans des groupes (ovales). L'appartenance d'un Agent à plusieurs groupes se fait en étirant la quille représentant l'agent sur tous les ovales des groupes d'appartenance.
3. **Les diagrammes organisationnels d'interaction** spécifient les interactions entre groupes et/ou rôles. Ils correspondent aux diagrammes de séquences en UML, dans lesquels les lignes de vies sont attribuées à des rôles et non à des instances de classes. Ces diagrammes permettent notamment de représenter les créations ou destructions de groupes.

Le processus de développement d'Aalaadin est basé essentiellement sur la phase de conception et peut se résumer en trois phases [Gutknecht, 1999] :

1. **L'analyse** sert à identifier les fonctions du système et les dépendances au sein de communautés identifiées. Il convient de définir quels sont les contextes pour la conception des groupes et les modes de coordination et d'interactions entre les entités d'analyse.
2. **La conception** assure l'identification des groupes et des rôles dans les diagrammes de structures organisationnelles. Elle permet aussi la description des schémas d'interactions (protocoles et messages) et la définition d'autres modèles décrivant les différentes actions et tâches ainsi que les objectifs à atteindre par chaque agent.

3. **La réalisation** : à partir des résultats des phases d'analyse et de conception, on choisit l'architecture d'agent et l'implémentation des protocoles d'interactions. Puis la gestion des entités du domaine permet d'implanter le système à partir d'organisations concrètes. La plateforme MadKit[2] qui supporte cette architecture organisationnelle serait bien adapté comme plateforme de prototypage et de simulation.

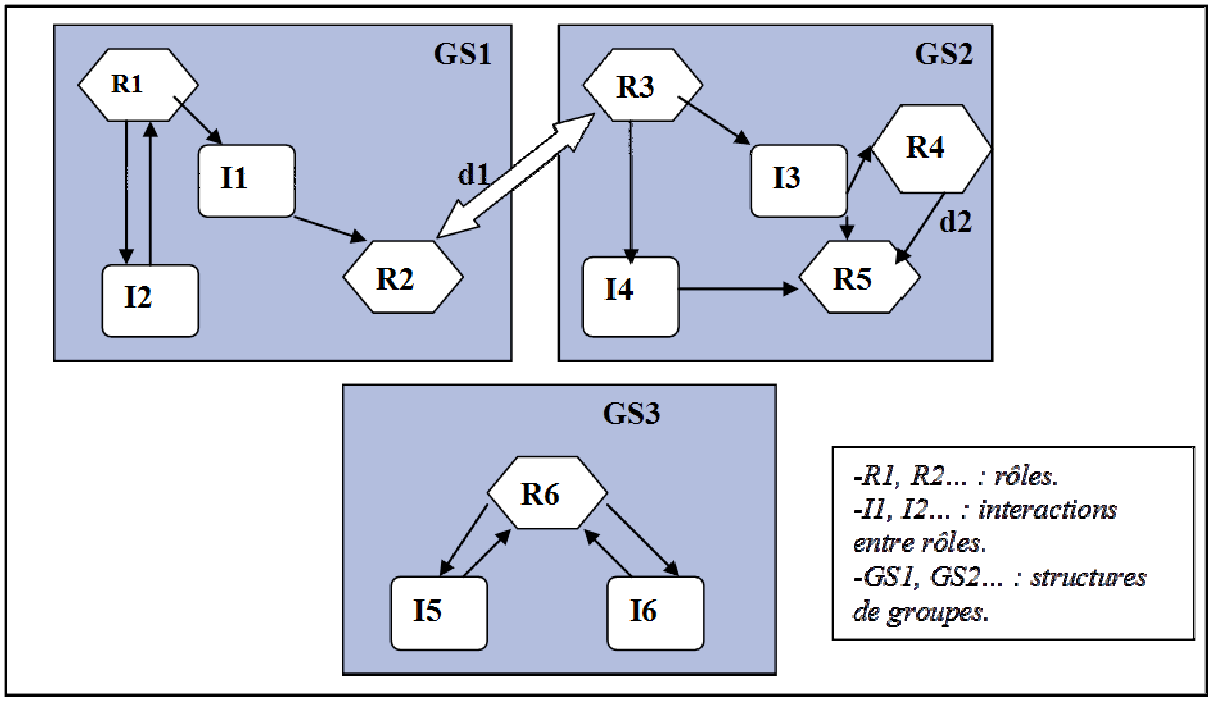


*Figure II-4 Les diagrammes de structure organisationnelle dans Aalaadin*

***Discussion***

Le modèle AGR d'Aalaadin a l'avantage d'être générique : il ne dépend ni du contexte de l'application ni du langage, ce qui permet aux développeurs :

a. d'implémenter eux-mêmes les comportements et les obligations attendus par le rôle,
b. d'exprimer les contraintes de cardinalité et de comptabilité offrant des moyens pour bâtir une dynamique au niveau de l'organisation,
c. de décrire explicitement au niveau des agents, les rôles qu'ils sont susceptibles d'utiliser.

Cependant, cette généricité est insuffisante à elle seule de pouvoir représenter tous les aspects multi-agents (agent, interaction, environnement et organisation). Toutefois, la plateforme MadKit reposant sur ce modèle doit alléger le travail du développeur.

[2] http://www.madkit.org

#### 2.4.2.2 GAIA

La méthodologie GAIA utilise une approche centrée sur l'organisation pour analyser et concevoir un SMA. Elle reprend et développe les concepts d'Aalaadin [Wooldridge, 2000]. Cette méthodologie s'inspire de la méthodologie orientée objet FUSION [Coleman et al, 1994]. Les concepts principaux de GAIA sont : Rôle, Permission, Responsabilité, Protocole, Propriété d'animation et Propriété de sécurité.

Un *rôle* est défini par quatre attributs : responsabilités, permissions, activités et protocoles. Une *responsabilité* détermine la fonctionnalité principale liée à un rôle. Elle est décrite par des *propriétés d'animation*, décrivant le cycle de vie du rôle et des *propriétés de sécurité*, qui sont des conditions que doivent vérifier certaines variables afin de garantir la survie du rôle.

Les *permissions* sont les droits de l'agent sur les ressources utilisées par un rôle. Ces droits peuvent être de type lecture, modification ou génération. Les permissions énoncent les limites de ressources dans lesquelles l'exécuteur de rôle doit opérer.

Une *activité* représente une opération interne de l'agent, sans qu'il n'ait nécessairement recours à une interaction avec d'autres agents.

Les *protocoles* définissent la manière dont les rôles peuvent agir avec d'autres rôles.

Comme le montre la figure II-5, le processus de développement de GAIA comprend une phase d'analyse et une phase de conception.

La phase d'analyse a pour objectif de comprendre le système et sa structure sans se préoccuper des détails d'implémentation. Elle regroupe le modèle d'environnement et le modèle d'organisation.

1. **Le modèle de rôle** : il énumère les différents rôles pouvant être joués par les agents du système et donne leurs caractéristiques.
2. **Le modèle d'interactions** : il décrit les dépendances et les relations entre les différents rôles du système. Il définit les protocoles de communication entre les agents.

La phase de conception transforme les modèles d'analyse en modèles plus spécifiques en procédant à des raffinements de ces modèles par l'analyse des structures organisationnelles. Elle produit un d'agents, un modèle de services et un modèle d'organisation.

1. **Le modèle d'agent** qui attribue les rôles aux différents agents du système. Il définit les différents types d'agents et leurs instances. Il est construit à partir du modèle de rôles défini à la phase d'analyse.
2. **Le modèle de services** a pour objectif d'identifier les services associés à chaque rôle d'agent. Un service est vu comme une fonction d'un agent et possède les caractéristiques suivantes : des entrées, des sorties, des pré-conditions d'exécution et des post-conditions.

3. **Le modèle d'organisation** (ou d'accointances) représente les liens de communication entre les différents types d'agents.

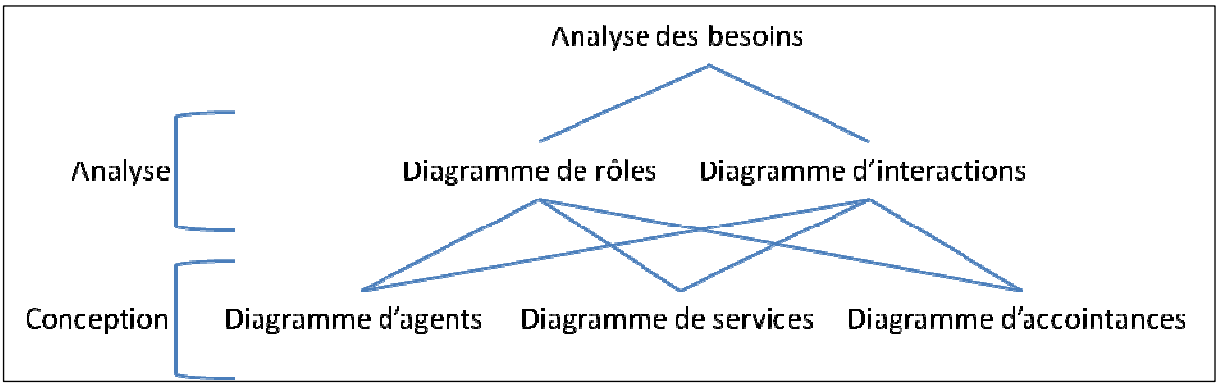


*Figure II-5 Le processus de développement de Gaia et les modèles manipulés*

***Discussion***

GAIA se focalise uniquement sur les premières phases de conception. Elle ne couvre ni la phase de vérification ni celle d'implémentation. Les SMA évolutifs ou réactifs, ou les agents compétitifs ne peuvent pas être modélisés avec cette méthodologie.

Gaia ne fournit pas de notations graphiques. Ses différents concepts sont principalement décrits moyennant des tables.

GAIA est une méthodologie complète et générique : elle n'est pas dédiée à un domaine spécifique d'application. Cependant, GAIA ne peut pas être ouvert ou utilisable à grande échelle [Mansour, 2007] pour diverses raisons :

- La méthodologie fixe à l'avance les types d'agents que peut contenir le système et qui ne changent pas par la suite.
- Les relations entre agents sont statiques et guidées par les rôles.
- Le concept d'organisation n'a pas de trace réelle, et même les relations entre rôles ne sont pas structurées.
- Les permissions et droits définis pour les rôles concernent des ressources qui sont supposées être connues à l'avance ce qui est inconcevable dans un environnement ouvert.

### 2.4.3 Les méthodologies formelles

Ces méthodes utilisent des formalismes mathématiques poussées. Elles sont spécifiées par la vérification et la validation des applications multi-agents en utilisant différents langages et notations. Nous détaillons dans cette section les méthodologies DESIRE [BRAZIER et AL., 1997, 1998] et PASSI [Cossentino et al, 2005 ; Cossentino et Potts, 2002].

### 2.4.3.1 DESIRE

DESIRE (Design and Specification of Interacting Reasoning components) est directement issue de l'ingénierie des connaissances. Elle se base sur un modèle qui traite la connaissance, l'interaction, la coordination des tâches et les possibilités de raisonnement dans les SMA.

DESIRE est basée sur trois concepts clés :

1. *la représentation de connaissances* sous forme de simples graphes nœuds/lien,
2. *le modèle générique d'agents* : un agent est considéré comme un composant décomposé en tâches,
3. *le modèle de composition de tâches* : permet de définir les tâches et leurs relations. Les tâches sont décomposées en sous tâches et sont présentées sous forme de composants avec des informations d'entrée et de sortie.

Le processus de développement de DESIRE se décompose comme suit :

1. **La description du problème** qui inclut les besoins imposés à la conception ;
2. **Les principes de conception** ou « design rational » pour spécifier les choix faits lors de la conception ;
3. **La conception conceptuelle** ou « conceptual design » qui définit un modèle conceptuel pour chaque agent, le monde extérieur et les interactions entre agents ;
4. **La conception détaillée** qui va spécifier les aspects de connaissance et de comportement ;
5. **La conception opérationnelle** qui va spécifier les paramètres nécessaires à l'implémentation.

***Discussion***

DESIRE est très orientée tâche et elle est parmi les méthodologies les plus utilisées dans l'industrie, certes avec des créateurs aux commandes, pour développer par exemple un logiciel d'aide au diagnostic de maladies cardiaques ou bien de la conception de tâches en entreprise.

Elle ne prend pas en compte un grand nombre de caractéristiques d'un SMA, tels que la socialité et la proactivité des agents. Son processus de développement est assez limité, et ne fait référence qu'à la phase de conception sans informations pour le passage d'une phase à une autre.

### 2.4.3.2 PASSI

PASSI (Process for Agent Societies Specification and Implementation) est une méthodologie se basant sur les modèles de conception et concepts provenant de l'ingénierie orientée objet et de l'intelligence artificielle en utilisant UML. Elle a la particularité de prendre en

considération la modélisation des agents mobiles. Cette modélisation respecte les standards définis par FIPA.

Dans PASSI, chaque agent est censé satisfaire ses besoins fonctionnels exprimés sous la forme de cas d'utilisation. Ces besoins sont attribués à l'agent dans la phase d'identification d'agent. Il peut jouer plusieurs rôles impliqués dans des scénarios et pouvant fournir des services à la société d'agents.

Le rôle que joue l'agent remplit au moins une tâche. Ces rôles sont aussi des mediums d'information par message. Chaque message contient la connaissance échangée et le langage utilisé pour décrire son contenu.

Le processus de PASSI est réitérable. Il est composé de cinq phases :

1. **La phase de définition des besoins du système** qui permet d'exprimer les besoins et de décrire le contexte du système.
2. **La phase de définition de la société d'agents** qui permet d'identifier les interactions sociales et les dépendances entre les agents.
3. **La phase d'implémentation** qui permet de définir la structure et le comportement du SMA et ceux de chaque agent.
4. **La phase de codage** qui permet d'identifier les modules réutilisables en cas de développement préexistants. Elle permet de rendre exécutable le code des agents.
5. **La phase de déploiement** qui permet de spécifier la configuration du système.

***Discussion***

PASSI est une méthodologie qui couvre toutes les phases du développement logiciel en partant de la spécification des besoins au déploiement des agents. Son utilisation est facile puisqu'elle est basée sur une extension du langage UML. Les modèles à fournir tout au long du processus sont clairement définis. PASSI est une méthode pouvant s'appliquer à n'importe quel domaine, mais elle est plutôt orientée *agent mobile*.

D'autre part, PASSI est riche en ressources et possède notamment un site[3] expliquant le processus pas à pas ainsi que tous les modèles. Son application ne nécessite aucune expertise particulière et est complètement dissociée des plateformes d'implémentation.

Cependant, PASSI souffre du manque de formalisme rendant ainsi difficile l'expression des propriétés, telles que la vivacité et la sûreté.

## 2.5 Comparaison des méthodologies de conception des SMA

Il faut savoir que les méthodes de développement des SMA sont nombreuses et ne cessent d'évoluer rapidement. Ainsi, il s'avère intéressant de faire une comparaison de l'existant afin de pouvoir bien cerner, d'une part les apports de ses méthodologies et d'autre part leurs lacunes. Le tableau suivant fait une synthèse des principales caractéristiques que nous avons

[3] *http://mozart.csai.unipa.it/passi/*

dégagées des méthodes présentées précédemment. Ces caractéristiques sont classées en fonction de leur pertinence.

| | AAII | Aaladin | DESIRE | GAIA | MaSe | PASSI |
|---|---|---|---|---|---|---|
| **Autonomie** | *oui* | *oui* | *oui* | *oui* | *oui* | *Oui* |
| **Groupe** | *non* | *oui* | *non* | *possible* | *non* | *Oui* |
| **Interaction** | *oui* | *oui* | *oui* | *oui* | *oui* | *Oui* |
| **Rôle** | *non* | *oui* | *oui* | *oui* | *oui* | *Oui* |
| **Adaptation** | *non* | *oui* | *non* | *non* | *non* | *Possible* |
| **Analyse** | *oui* | *oui* | *oui* | *oui* | *oui* | *oui* |
| **Conception** | *oui* | *oui* | *oui* | *oui* | *oui* | *oui* |
| **Implémentation** | *non* | *oui* | *possible* | *non* | *oui* | *oui* |
| **Buts** | *oui* | *possible* | *oui* | *possible* | *oui* | *Possible* |
| **Croyances** | *oui* | *possible* | *oui* | *possible* | *oui* | *Oui* |
| **Désirs** | *oui* | *possible* | *oui* | *possible* | *oui* | *Oui* |
| **Environnement** | *non* | *possible* | *non* | *possible* | *non* | *Possible* |
| **Pro activité** | *oui* | *oui* | *oui* | *oui* | *oui* | *Oui* |
| **Réactivité** | *oui* | *oui* | *oui* | *oui* | *non* | *Oui* |
| **Tâche** | *non* | *possible* | *oui* | *oui* | *oui* | *Oui* |
| **Accessibilité** | *oui* | *oui* | *possible* | *possible* | *possible* | *Oui* |
| **Analysabilité** | *oui* | *oui* | *oui* | *non* | *oui* | *oui* |
| **Complexité** | *oui* | *oui* | *oui* | *non* | *possible* | *Oui* |
| **Consistance** | *oui* | *possible* | *oui* | *oui* | *possible* | *Oui* |
| **Exécution** | *non* | *oui* | *oui* | *non* | *oui* | *Oui* |
| **Modularité** | *oui* | *oui* | *oui* | *oui* | *possible* | *Oui* |
| **Raffinage** | *oui* | *oui* | *oui* | *oui* | *oui* | *Oui* |
| **Traçabilité** | *oui* | *oui* | *oui* | *oui* | *non* | *Oui* |
| **Besoins** | *non* | *non* | *non* | *non* | *non* | *Oui* |
| **Test** | *non* | *non* | *oui* | *non* | *possible* | *possible* |
| **Délivrables** | *oui* | *non* | *non* | *oui* | *oui* | *oui* |
| **Gestion de qualité** | *non* | *non* | *non* | *non* | *non* | *non* |
| **Ressources** | *non* | *oui* | *non* | *non* | *oui* | *oui* |
| **Expertise requise** | *oui* | *non* | *oui* | *oui* | *non* | *non* |
| **Langages spécifiques** | *non* | *oui* | *oui* | *non* | *non* | *non* |

**Tableau II-1 Comparatif des différentes méthodes**

***Notation*** : -***oui :*** *la méthodologie a pris en compte cette valeur ;*
-***non :*** *la méthodologie n'a pas pris en compte cette valeur ;*
-***possible :*** *la méthodologie pourrait prendre en compte la valeur implicitement.*

***Discussion***

L'analyse des différentes méthodologies fait ressortir quelques caractéristiques communes :

- Elles ont presque le même découpage en phases de la tâche de modélisation : la spécification des besoins, l'analyse et la conception.
- L'architecture interne de l'agent ainsi que ses fonctionnalités sont définies dans les modèles d'agent et dans les services.
- L'environnement dans lequel évoluent les agents est défini à travers un modèle d'interface d'environnement.
- Les interactions qu'entretiennent les agents entre eux sont représentées à travers des modèles d'interaction et de rôles.

Cependant, très peu de méthodologies offrent une dynamique aux systèmes construits. Or, il est inconcevable de modéliser un SMA sans dynamique, étant entendu que c'est l'un des traits caractéristiques du système. Le modèle AGR d'Aalaadin a l'avantage d'assurer la généricité ainsi c'est aux développeurs de fournir les moyens pour bâtir une dynamique au système qu'ils construisent. Un autre point fort d'AGR est qu'une plateforme appelée MadKit reposant sur ce modèle a été conçue. Nous présentons cette plateforme dans le chapitre suivant consacré aux systèmes Pair à Pair.

Nous présentons, dans la section suivante, les principales catégories d'applications des SMA afin de souligner les multitudes des possibilités d'utilisation de ces systèmes et de confirmer donc l'intérêt que peuvent apporter ces systèmes aux domaines cités.

## 2.6 Domaines d'application des systèmes multi-agents

Les domaines d'application des SMA sont particulièrement riches : Support et aide à la décision, commerce électronique, systèmes manufacturiers, télécommunications, supervision de réseaux, robotique, simulation de systèmes sociaux et naturels, applications embarquées et sans fil, applications Internet, filtrage et recherche d'information, gestion des transports et du trafic, ingénierie médicale, jeux, e-Learning. En effet, les SMA ont attiré l'attention à cause de leur grande souplesse d'utilisation puisqu'il n'y a aucune limite fixée sur les comportements des agents ni sur leur manière de s'exprimer mais aussi à cause de leur pouvoir de représentation d'un univers intuitif où chaque composante d'un système peut être modélisée en un agent [Glisse, 2005]. Il existe cependant quelques grandes catégories d'applications des SMA dont [Ferber, 1999], [Chaib-draa et al, 2001] :

- La résolution de problèmes qui concerne toutes les situations dans lesquelles des agents logiciels, purement informatiques, accomplissent des tâches utiles aux êtres humains.

- La robotique distribuée porte sur la réalisation d'un ensemble de robots qui coopèrent pour accomplir une mission et utilise des agents concrets qui se déplacent dans un environnement réel.
- La simulation multi-agents : la simulation est la démarche scientifique qui consiste à réaliser une reproduction artificielle, appelée modèle, d'un phénomène réel que l'on désire étudier, à observer le comportement de cette reproduction lorsqu'on en fait varier certains paramètres, et à en induire ce qui se passerait dans la réalité sous l'influence de variations analogues. La simulation multi-agents offre un outil puissant pour expliquer les changements au sein d'un système naturel ou artificiel en fonction des changements des individus qui le composent [Guessoum, 2003]. Les phénomènes émergents peuvent être détectés et expliqués par un observateur externe qui est l'utilisateur de la simulation. La détection de ces phénomènes émergeants repose souvent sur l'interprétation par l'observateur externe de résultats de simulation [Muller, 2000]. Elle est utilisée pour l'étude de phénomènes complexes réels ou virtuels dans les domaines tels que la physique, la chimie, la biologie, l'écologie, la géographie et les sciences sociales. Elle permet de représenter et de simuler des systèmes faisant intervenir un grand nombre d'individus (écosystèmes, sociétés d'insectes, dynamiques de populations), de modéliser des environnements virtuels (jeux vidéo, cinéma). Enfin, elle permet d'expliquer l'impact de comportements individuels à un niveau global de description.

La réussite des applications fondées sur les SMA développés dans plusieurs domaines d'application comme la simulation, la gestion d'informations, le commerce électronique démontre que les SMA paraissent être l'approche la plus adaptée à un nombre important de domaines [Jennings, 1998]. Ceci est valable aussi pour le domaine d'e-Learning où l'utilisation du paradigme agent a montré son utilité [Milgrom et al, 2001] [Wilson et al, 2004]. Nous nous sommes intéressés à l'utilisation des systèmes multi-agents dans le domaine d'e-Learning que nous traitons dans le chapitre 4.

## 2.7 Conclusion

Les systèmes multi-agents sont des sociétés constituées d'un certain nombre d'agents autonomes travaillant ensemble pour résoudre un problème donné. Cependant, cette autonomie implique que chaque agent peut poursuivre ses propres buts et ne communiquer avec les autres que pour collecter des informations ou bien coopérer avec les autres pour résoudre un but global. Cependant, aucun agent n'a suffisamment de connaissance ou de compétence pour résoudre seul un problème. Ainsi, dans de tels systèmes, il est important d'interagir, de coopérer, de communiquer mais aussi de coordonner ses actions avec les autres. En effet, la coordination est une notion clé pour les SMA. Elle permet aux agents de bénéficier des capacités des autres agents et à la communauté d'éviter de dégénérer en une collection chaotique d'individus.

Nous avons présenté dans ce chapitre les agents et les systèmes multi-agents ainsi que leurs caractéristiques. De plus, nous avons étudié les méthodologies de conception des SMA afin de choisir une qui satisfait nos besoins c'est-à-dire qui est appropriée pour concevoir un

système ouvert, distribué et à large échelle. En effet, les caractéristiques des SMA rendent leur utilisation dans des systèmes complexes, ouverts, distribués et dynamiques très appropriée. Nous retrouvons ces caractéristiques dans les systèmes d'e-Learning. Pour cela, notre choix s'est orienté vers ce domaine là : l'e-Learning.

Cependant, dans ce chapitre nous n'avons pas évoqué les plateformes de développement des systèmes multi-agents qui seront néanmoins présentées dans le chapitre suivant. En effet dans le prochain chapitre, nous nous intéressons au déploiement des systèmes multi-agents sur des architectures P2P. Pour cela, nous voulons étudier les plateformes qui supportent cette combinaison.

Nous proposons donc, dans le chapitre suivant, d'étudier les systèmes P2P et de présenter les avantages qu'apportent la combinaison entre les technologies agents et Pair à Pair dans la construction des applications ouvertes et distribuées.

# Chapitre III. Le P2P pour le déploiement des SMA

## 3.1 Avant-propos

Ce chapitre aborde l'utilisation des systèmes P2P en association avec les systèmes multi-agents afin de résoudre les problèmes dus à l'hétérogénéité, l'ouverture et la dynamique des systèmes [Sycara, 1998]. Le Peer to Peer (P2P) aussi appelé "pair à pair", "poste à poste" ou encore "d'égal à égal" en français est un modèle d'architecture de réseau où les postes interconnectés communiquent directement et partagent leurs ressources. Nous présentons dans ce chapitre, l'intérêt d'utiliser les P2P dans un contexte multi-agents ainsi que les problèmes qui en découlent.

## 3.2 Introduction

Les systèmes P2P sont des systèmes décentralisés, autonomes et tolérants aux pannes. Ces caractéristiques sont très utiles pour les systèmes multi-agents. Nous proposons dans ce chapitre d'associer les systèmes multi-agents aux systèmes P2P et ceci en déployant un SMA sur une architecture P2P afin de profiter des caractéristiques qu'offrent les systèmes P2P.

Les systèmes multi-agents sont souvent considérés comme un ensemble de pairs coordonnant leurs actions et partageant leurs connaissances afin d'accomplir un but commun. Ainsi, la coordination des actions des agents est primordiale pour les SMA. Cependant, dans un système P2P, les pairs ne connaissent pas l'ensemble global des participants du système et ne peuvent communiquer directement qu'avec leurs voisins. Par ailleurs, dû au dynamisme du réseau, ce voisinage change aussi dynamiquement. Une des difficultés majeures consiste alors à adapter les mécanismes de coordination tels qu'ils sont connus dans les systèmes multi-agents pour le contexte spécifique des systèmes P2P.

Dans ce chapitre, nous allons tout d'abord présenter le paradigme P2P. Nous présenterons ensuite les travaux concernant la coordination d'actions des agents et nous terminons par la présentation des plateformes de développement capable de supporter l'association des systèmes multi-agents avec les systèmes P2P.

## 3.3 Présentation des systèmes P2P

Les systèmes P2P [Milojicic et al, 2002] sont des systèmes répartis à l'échelle du réseau Internet. Ils sont composés des pairs capables de s'auto-organiser au sein d'un réseau logique dans le but d'échanger ou partager de services et de ressources telles que des données, des programmes, des capacités de stockage ou de calcul.

Il existe trois types d'architecture de systèmes P2P, en fonction du mode de mise en relation entre demandeur et fournisseur, c'est-à-dire en fonction du mécanisme de localisation des ressources.

1) L'architecture centralisée dont le représentant le plus connu est ***Napster***. Les systèmes centralisés sont à la limite du P2P et du client-serveur et se basent sur un serveur central auquel se connectent les utilisateurs. Ce serveur est chargé de les mettre en relation directe. L'intérêt de cette technique réside dans l'indexation centralisée de tous les répertoires et intitulés de fichiers partagés par les abonnés sur le réseau. Cependant, cette architecture pose des problèmes de sécurité, de robustesse et de limitation de la bande passante qui sont directement issus de l'utilisation de serveurs dont le seul but est de posséder l'annuaire des clients.

2) L'architecture décentralisée ou « P2P Pur » dont le représentant le plus connu est ***Gnutella***. Le fait de vouloir une architecture sans serveurs centraux, il faudra alors trouver le moyen de constituer un annuaire incluant chaque client, puis de les faire communiquer. C'est sur ce mécanisme que se basent les réseaux Peer to Peer décentralisés. Avec l'absence des serveurs centraux, ce sont tous les éléments du réseau qui vont jouer ce rôle. Cependant, localiser une ressource devient une opération critique qui demande a priori le parcours d'une partie plus ou moins importante du réseau. Ce parcours ralentit souvent les échanges de données entre les machines.

3) Les architectures hybrides sur lesquels sont basés les logiciels Skype, eDonkey et Kazaa. Il s'agit ici d'un modèle que l'on pourrait qualifier d'hybride ou semi-décentralisés entre le modèle centralisé et pur. En effet, ces architectures conservent le principe d'un serveur de localisation mais s'appuient sur un ensemble de serveurs répartis afin de minimiser les conséquences des pannes et de réduire les risques de contention [Arcangeli et Leriche, 2006].

Les systèmes P2P ont un domaine applicatif très vaste. Les applications les plus célèbres restent celles permettant l'échange des fichiers comme Gnutella [Ripeanu et al, 2002], eMule[4] et bitTorrent [Cohen, 2003]. D'autres applications fonctionnent sur les réseaux P2P, telles que Skype[5], qui autorisent leurs utilisateurs à communiquer en direct mais aussi les applications scientifiques de calcul distribué et celles de travail collaboratif.

Nous nous intéressons dans ce qui suit à identifier les points forts ainsi que les inconvénients des systèmes P2P. Nous comparons également l'architecture de ces systèmes à celle du modèle client-serveur.

### 3.3.1 Caractéristiques des systèmes P2P

Les systèmes P2P sont des systèmes ouverts et distribués à grande échelle. Ils se caractérisent par la capacité de chaque membre (pair) du système à : (1) gérer un ensemble de technologies, des applications et des ressources (autonomie), (2) à réaliser ses buts indépendamment des autres pairs (décentralisation), (3) fournir des ressources ou des services aux autres membres afin de pouvoir joindre et utiliser le système (coopération).

[4] http://www.emule-project.net/

[5] http://www.skype.com/

Dans ce contexte, les pairs et les ressources peuvent apparaître et disparaître à n'importe quel moment (dynamisme).

Cependant, ces caractéristiques des systèmes P2P posent des nombreux problèmes. En effet, leur échelle, correspondant au nombre de pairs et aux dimensions du réseau, étant une caractéristique principale des systèmes P2P, rend les interactions entre les pairs beaucoup plus nombreuses ce qui influe fortement sur le fonctionnement et la performance du système. Cette caractéristique implique aussi l'ouverture des systèmes. Les pairs peuvent également intégrer ou quitter le système dynamiquement sans contrôle global et sans qu'il ne soit possible de prévoir l'ensemble des évolutions. Ce dynamisme conduit à une volatilité de l'ensemble des ressources et des services requis au bon fonctionnement du système. De plus, les pairs peuvent ne disposer que d'une connaissance partielle ou non actualisée de sa structure. Il n'est alors pas possible de connaître précisément à chaque instant la structure du réseau et l'ensemble des informations et des services disponibles.

Les points négatifs des systèmes P2P peuvent être donc considérés comme étant les effets secondaires de leurs points forts. Pour cela, nous avons choisi d'utiliser ces systèmes. En effet, plusieurs de leurs caractéristiques (autonomie, décentralisation, tolérance aux pannes,…) sont très utiles aux systèmes multi-agents. D'ailleurs, nous avons fait une comparaison avec le modèle client-serveur afin de nous situer par rapport à ces caractéristiques.

### 3.3.2 Fonctionnement du P2P

Localiser une ressource est une fonctionnalité essentielle pour le P2P. En effet, du fait de l'échelle, la communauté ne peut pas être informée des changements qui ont lieu sur le réseau tels que les connexions/déconnexions et les pannes. Ainsi, un pair voulant accéder à une ressouce, n'a a priori qu'une connaissance partielle de la disponibilité des serveurs et de la qualité des services

Le mécanisme de localisation des ressources diffèrent selon l'architecture du système P2P (centralisée, décentralisée ou hybride). Cependant, les échanges de données et plus généralement l'exploitation des ressources se font directement entre pairs sans intermédiaire. Nous détaillerons par la suite le mécanisme de localisation des ressources selon les 3 architectures du P2P.

Dans le cas d'une architecture centralisée, le serveur central fonctionne comme un annuaire. Il dispose principalement de deux types d'informations: celles sur le fichier (nom, taille, ...), et celles sur l'utilisateur (nom utilisé, IP, nombre de fichiers, type de connexion, ...). Ainsi, lorsqu'un poste souhaite partager un fichier, il le déclare au serveur central. Celui-ci répertorie alors son adresse IP ainsi qu'un numéro de port donné par le client où il pourra être contacté pour un téléchargement. Si un utilisateur désire un fichier, il interroge l'index central du serveur qui lui indique alors la liste des postes (adresses IP) sur lesquels il est susceptible de le trouver. Interroger l'index se fait de la même manière qu'une recherche avec un moteur de recherche classique. L'utilisateur choisit alors parmi les réponses celle qui lui convient le mieux (pertinence de la réponse, bande passante, taille du fichier) et contacte

directement le ou les postes choisis. Il faut noter que le fichier en lui-même ne transite pas par le serveur central.

Dans une architecture décentralisée où chaque machine peut jouer le rôle de serveur, un client qui souhaite se connecter doit envoyer un message en diffusion (ou broadcast) afin de savoir quelles autres personnes du réseau sont actives. Seules ces personnes répondront au message broadcast et le client est alors connecté au réseau. Cependant afin de garder des informations cohérentes, il n'est pas connecté directement à plus de 3 ou 4 poste et n'a qu'une vision limitée du réseau. Pour obtenir une ressource, un poste client transmet une requête à ses voisins qui font de même et ainsi de suite, on parle d'inondation. Pour limiter la génération d'un nombre exponentiel de messages par ce procédé, on limite la propagation des messages jusqu'à un horizon, défini par un paramètre de durée de vie de la requête (TTL Time To Live), décrémenté à chaque poste. Une fois un poste possédant la ressource est retrouvé, une connexion directe s'établit entre les postes. Ce mécanisme de recherche présente néanmoins une limite, une requête peut être stoppée par une expiration de TTL sans avoir parcouru l'intégralité du réseau et retourner une réponse négative.

Dans une architecture hybride, .lorsqu'un poste se connecte, il est affecté à l'un des groupes des super-pairs. Si le poste requiert une ressource, il effectue une requête classique client/serveur à son super-pair. Si aucun poste du groupe de ce super-air ne possède le fichier demandé, alors deux cas se présentent :

- soit il transmet la requête aux autres super-pairs,
- soit il transmet l'adresse d'autres super-pairs au client qui doit ainsi renouveler la requête.

Si la ressource est retrouvée, une connexion s'établit alors entre les deux pairs quel que soit leur niveau pair ou super-pair. Entre deux super-pairs, le système continue à fonctionner comme les réseaux décentralisés. Cependant, la propagation des données est plus rapide, puisqu'elle n'utilise plus que les connexions haut débit.

### 3.3.3 Client-serveur versus P2P

Le modèle P2P est une alternative au modèle client-serveur classique. En effet, dans le modèle client-serveur, les services sont centralisés. Les clients ne disposent pas d'informations mais sont capables d'aller en chercher sur les serveurs qui sont situés sur des machines puissantes afin de pouvoir servir plusieurs clients en même temps. Ce caractère centralisé est sujette à des goulots d'étranglement des ressources. Par conséquent, ce modèle peut être facilement attaqué, puis il est difficile et coûteux à modifier en raison de son positionnement stratégique au sein de l'infrastructure du réseau. Par ailleurs, toute panne du serveur entraîne l'arrêt total du fonctionnement du système dans sa globalité. En contraste, dans un modèle P2P, les pairs peuvent jouer en même temps le rôle de serveur et le rôle du client et sont à égalité de devoirs et de droits : chacun peut rendre accessible des ressources dont il dispose et exploiter des ressources fournies par d'autres et ceci sans aucun contrôle central ce qui rend le système flexible et tolérant aux pannes. Ainsi, en cas de

panne d'un pair, le dysfonctionnement reste localisé et le système continue tout de même à fonctionner. Cette propriété suscite parfois une controverse par ceux qui la désapprouvent comme n'étant pas valable pour les architectures centralisées. Néanmoins, le serveur dans les architectures centralisées est un serveur de localisation des pairs et ne détient pas les ressources recherchées par ces derniers. Par contre, cette architecture, contrairement aux deux autres est particulièrement vulnérable puisqu'il suffit d'un blocage du serveur pour déconnecter tous les utilisateurs et stopper le fonctionnement de l'ensemble du réseau.

Une autre différence contrastant les deux modèles vient du fait que les performances du modèle client-serveur se dégradent au fur et à mesure que le nombre d'utilisateurs augmente alors qu'une des caractéristiques des réseaux P2P est que la qualité et la quantité des données disponibles augmentent à mesure que le nombre d'utilisateurs augmente. La valeur du réseau augmente donc avec sa popularité. Cependant, dans le modèle client-serveur, des solutions pour les problèmes relatifs au passage à large échelle sont bien connues, utiliser des machines plus puissantes par exemple mais ceci a un coût considérable. Par contre, les systèmes P2P décentralisés ont souvent besoin des solutions algorithmiques pour résoudre les problèmes de passage à large échelle puisqu'il n'y a rien de centralisé pour ajouter plus des ressources informatiques. Ces algorithmes distribués ont tendance à être parmi les plus difficiles à développer car ils nécessitent des décisions à faire sur chaque pair local, généralement avec peu de connaissance globales.

Sur un autre, le modèle client-serveur offre plus de sécurité que le modèle P2P. En effet, le modèle client-serveur permet l'identification des clients et a un contrôle sur les transactions qui se produisent sur le serveur ce qui n'est pas le cas du modèle P2P qui offre à chaque pair une autonomie significative sans possibilité de contrôle et lui permet de se connecter de manière intermittente avec des adresses IP variables. En conséquence, l'identification de pair est difficile pour reconnaître les membres qui auraient eu des comportements éventuellement malveillants.

Finalement, il est important de préciser qu'il est essentiel qu'un nombre minimal des pairs soit disponible pour que le système P2P continue à fonctionner ce qui n'est pas le cas pour le modèle client-serveur qui continue à fonctionner aussi longtemps que le serveur conserve le service en cours d'exécution.

Nous avons montré les différences qui existent entre le modèle client-serveur et les systèmes P2P. Chaque modèle a ses points forts et ses points faibles mais nous choisissons d'utiliser les systèmes P2P plutôt que le modèle client-serveur parce que le premier possède les caractéristiques que nous recherchons.

L'autonomie des agents nécessite la coordination des actions d'agents afin d'accomplir les tâches menant à la réalisation du but et/ou la résolution du problème. Ainsi, mettre en place des mécanismes de coordination est primordial pour les SMA. Nous discutons de ce sujet dans la section suivante.

## 3.4 La coordination des agents sur une architecture P2P

Les interdépendances, existantes dans la majorité des systèmes multi-agents, entre les buts des agents, leurs capacités et les ressources qu'ils utilisent rendent leur coordination très importante. En effet, cette coordination permet de garder les bénéfices d'un groupe d'agents ce qui empêche la communauté de devenir une collection chaotique d'individus. Cependant, l'atteinte d'une bonne coordination n'est pas une chose aisée. En effet, les agents sont autonomes, ils ont leurs propres buts et actions. Par ailleurs, le contrôle et les données sont distribués entre eux. Par conséquent, les connaissances sont réparties dans tout le système et chaque agent n'a de l'ensemble du système qu'une vision locale sur laquelle il doit se baser pour agir. Ceci peut l'amener à prendre des mauvaises décisions. Ainsi, pour résoudre efficacement des problèmes de coordination, un agent doit savoir quand, où, avec quels agents, et de quelle manière il doit coordonner ses actions pour réaliser ses buts et ceux de la collectivité et ceci en bénéficiant efficacement des capacités des autres. Dans la littérature, plusieurs modèles sont proposés pour résoudre le problème de coordination dans les systèmes multi-agents comme l'anticipation, la planification distribuée, la formation de coalition, etc.

**L'anticipation :** L'anticipation peut être définie comme étant la prédiction de l'état futur de l'environnement ainsi que de l'interprétation et de l'utilisation de ces prédictions afin de pouvoir changer cet état. C'est un concept relativement large qui à l'origine a été étudié par les biologistes et les psychologues qui cherchaient à expliquer les comportements adaptatifs et complexes des animaux. Plusieurs chercheurs ont proposé des modèles basés sur l'anticipation pour la coordination multi-agents. Par exemple, [Doniec et al, 2006] a proposé un modèle de sélection d'actions basée sur l'anticipation permettant la détection a priori des situations inter-bloquantes qui peuvent avoir lieu dans un contexte de coordination multi-agents qu'il a appliqué sur une simulation de trafic routier en carrefour.

**La planification distribuée:** L'approche de coordination par planification distribuée s'intéresse aux interactions qui existent entre les plans des agents [Von Martial, 1990]. En effet dans la planification distribuée, chaque agent planifie individuellement les actions qu'il compte accomplir en fonction de ses propres buts. Ainsi, chaque agent dispose des buts et des plans. Le problème consiste pour ces agents à échanger des informations portant sur leurs plans et leurs buts, afin que chacun puisse satisfaire ses objectifs [Ferber, 1997]. La planification contribue à la coordination, dans la mesure où lorsque les agents adoptent un plan « bien fait », ils agissent généralement de manière coordonnée [Chaib-draa et al, 2001]. Plusieurs chercheurs se sont intéressés à ce domaine notamment [Von Martial, 1990] [El-Fallah et al, 1996] [Durfee, 2001].

**La formation de coalitions**: Le domaine de formation de coalition dans les SMA est évocateur pour les organisations dynamiques. En effet, les coalitions sont des regroupements ponctuels d'agents compétitifs pour la réalisation d'un projet commun [Caillou et al, 2001]. Une coalition peut être définie comme une organisation à court terme basée sur des engagements spécifiques et contextuels, ce qui permet aux agents de bénéficier de leurs compétences respectives [Charif, 2007]. Le principe de la formation de

coalition a été proposé pour l'allocation de tâches pour la résolution de problèmes [Shehory et al, 1998], [Vauvert, 2002]. Dans [Belleili, 2005], une méthode de formation de coalition a été proposée pour maximiser l'utilité de la réponse des requêtes des agents.

Cependant, l'utilisation des systèmes P2P complique davantage la coordination des actions d'agents. En effet, les systèmes P2P se caractérisent par leur environnement fortement dynamique où chaque pair peut apparaître ou disparaître dans le réseau ou même devenir inaccessible (déconnexion, panne). En plus, les pairs n'ont qu'une vision limitée du système et ne peuvent communiquer directement qu'avec leurs voisins qui changent aussi dynamiquement. Des travaux de recherche ont essayé d'adapter les mécanismes de coordination tels qu'ils sont connus dans les SMA au contexte spécifique des systèmes P2P notamment dans le domaine de formation de coalition.

Dans [Belmonte, 2006], les auteurs proposent une méthode de formation de coalition basée sur les mécanismes d'incitation pour des systèmes P2P hybrides d'échange des fichiers qui empêche le comportement de cavalier seul (qui prend sans rien donner en échange). Un index de réputation est introduit afin de mesurer la contribution de chaque agent ainsi plus l'agent échange des fichiers avec les autres plus son index est élevé ce qui le rend prioritaire dans la queue pour l'échange d'un fichier. [Aknine, 2007] propose une méthode de formation de coalition évolutive adaptée pour les systèmes distribués à large échelle. Cette méthode est formée de deux phases principales : une phase de négociation durant laquelle les agents échangent leurs buts et discutent des propositions de structure de coalition et ceci jusqu'à l'atteinte d'un accord ou l'expiration du temps. En cas d'accord, la deuxième phase commence, c'est la phase d'engagement atomique non bloqué durant laquelle la structure décidée de coalition qui satisfait le mieux les buts des agents est formée. Plus récemment dans [Panah et Khorsandi, 2011], une méthode de formation de coalition dynamique a été proposée pour la construction du réseau logique recouvrant[6] (overlay en anglais) et ceci afin d'améliorer la performance de recherche dans les réseaux P2P. Cette méthode repose sur deux stratégies : la stratégie myope où le joueur cherche à maximiser son gain pour la prochaine période et la stratégie clairvoyante où le joueur choisit la coalition qui lui apporte plus de gain à l'avenir mais pas nécessairement dans la prochaine période.

### 3.4.1 Discussion

Récemment, une étude expérimentale de l'effet de l'identité de groupe sur la formation des coalitions et la distribution des richesses qui en résulte a été faite [Tremewan, 2010]. L'étude consiste en deux étapes : dans la première étape, une identité de groupe basé sur des goûts communs pour des peintures a été induite chez les sujets. Dans une deuxième étape, les sujets jouent par trois un jeu symétrique avec vote à la majorité. Le résultat principal est que lorsque parmi les trois sujets qui doivent prendre une décision, deux sujets appartiennent au

[6] Dans un système P2P, chaque pair n'a qu'une vision limitée du système. Afin de pouvoir participer au système, il a une liste de pairs voisins (voisinage logique). L'ensemble de ces listes définit un graphe : le réseau logique recouvrant ou overlay en anglais.

même groupe, ils gagnent significativement plus que le joueur de l'autre groupe. L'auteur explique ceci soit par un changement de préférences sociales, soit par le fait que le partage avantageant les deux joueurs du même groupe est rendu plus marquant par le fait que les deux joueurs appartiennent au même groupe, ce qui rend plus facile la coordination.

Cette étude vient réconforter notre idée qui est de proposer une méthode de formation de GROUPE comme solution pour la coordination d'actions d'agents dans une application P2P. En effet dans cette solution, nous essayons de reproduire ce qui se passe dans la réalité, en rassemblant des agents qui vont bien ensemble afin de mettre en commun leurs compétences et ainsi réaliser un but global. Ces agents auront ainsi le temps de se connaître et d'avoir une appréciation des autres agents avant que le groupe, atteignant une certaine taille, ne se divise pour former des nouveaux groupes et ceci selon les préférences des agents qui ont une idée sur les compétences mais aussi sur la sociabilité (niveau de collaboration et d'interaction) des autres agents.

Nous avons proposé dans ce chapitre de déployer les systèmes multi-agents sur une architecture P2P. Cependant, nous avons besoin d'une plateforme capable de supporter un tel système. Pour cela nous étudions, dans la section suivante, les plateformes de développement permettant de construire un système multi-agents à architecture P2P afin d'en choisir une.

## 3.5 Les plateformes de développement

Nous avons étudié des plateformes de développement des SMA ainsi que des plateformes permettant de construire des applications P2P.

En effet, plusieurs environnements de développement permettant de construire des applications P2P existent, nous citons pour exemple, XtremWeb[7] , Anthill[8], JXTA[9] (de l'anglais Juxtapose). Cette dernière, que nous présentons ci-après, est de loin la plus connue et permet le développement d'un système multi-agents.

Cependant, nous trouvons plus judicieux d'utiliser des plateformes multi-agents pour construire des systèmes multi-agents. En effet, les plateformes multi-agents permettent aux développeurs de concevoir et réaliser leurs applications sans perdre de temps à réaliser des fonctions de base pour la création et l'interaction entre agents et éliminent, dans la plupart des cas, la nécessité d'être familier avec les différents concepts théoriques des systèmes multi-agents. Ainsi, plusieurs plateformes permettant la conception et l'implémentation d'un SMA existent, nous citons pour exemple : SYNERGIC [Carpua et Regis, 1995], DIMA [Guessoum, 1996], GEAMAS [Guyoumard et al, 1999], etc. Cependant, il faut que la plateforme multi-agents choisie permette le développement d'un système qui respecte le modèle P2P. Nous présentons, donc, dans cette section deux plateformes qui utilisent le

[7] http://www.xtremweb.net/

[8] http://www.cs.unibo.it/projects/anthill/

[9] http://jxta.kenai.com/

paradigme agent et qui permettent en plus le développement des applications P2P : Jade[10] et Madkit[11].

### 3.5.1 JXTA

JXTA est un projet open source lancé par Sun Microsystems qui cherche à l'imposer comme un standard du P2P. Le but de JXTA est de proposer un ensemble de protocoles ouverts pour interconnecter des dispositifs allant du téléphone cellulaire jusqu'aux serveurs, en faisant abstraction de la topologie physique. JXTA offre des mécanismes de recherche et de publication efficaces ainsi qu'une solution transparente de communication indispensable dans un environnement comme Internet [Mansour, 2007]. Le fonctionnement repose sur le découpage de parties du réseau réel en réseaux virtuels, dans lesquels chaque pair peut accéder aux ressources des autres sans se préoccuper de leur localisation ou de leur environnement d'exécution [Leriche, 2006].

JXTA appartient à la famille de P2P hybride. En effet, certains pairs sont spécialisés afin d'offrir un service de découverte à la communauté à laquelle ils appartiennent, ce sont les pairs de rendez-vous. D'autres jouent un rôle de routage, ce sont les pairs relais. Ainsi, les pairs sont des éléments de base de JXTA. En plus de pairs rendez-vous et relais, il y a un autre type de pair qui est le plus basique c'est le pair simple qui permet de consommer et de proposer des services à la communauté. Notant que les pairs rendez-vous et relais sont aussi de pairs simples proposant d'autres services en plus. Chaque pair sur JXTA peut agir comme l'un de ces types (simple, relais, rendez-vous) et même endosser les trois en même temps. Ces pairs utilisent des canaux de communication virtuels pour échanger des messages appelés pipes. Un « pipe » offre un mode de transfert de messages asynchrone et non discriminant par rapport au contenu. En effet, ce message échangé entre deux pairs est formé d'une séquence ordonnée d'éléments, l'élément est un couple (nom, type) plus un contenu qui peut être une chaîne de caractères, un objet, une séquence binaire. En plus, JXTA prend en compte la notion de communauté et la formalise sous forme de PeerGroup ou groupe de pairs. Un PeerGroup est une collection de pairs qui se sont mis d'accord pour le partage d'un certain nombre de services. Sachant qu'un pair peut appartenir à plusieurs groupes en même temps, et tous les pairs dès leurs créations appartiennent automatiquement à un groupe par défaut dit « World Peer Group ».

JXTA est composé de six protocoles indépendants mais qui travaillent ensemble pour faciliter la découverte, l'organisation, le monitoring et la communication entre pairs. Les protocoles JXTA sont conçus sans imposer des contraintes sur l'environnement d'exécution des pairs. Sachant qu'un pair n'est pas obligé de tous les implémenter :

- Le protocole Peer Discovery (PDP) qui est le protocole de découverte des pairs
- Le protocole Peer Information (PIP) permet d'obtenir au moyen d'échange des messages des informations sur les capacités et les statuts des autres pairs.

[10] http://jade.tilab.com/

[11] http://www.madkit.org/

- Le protocole Peer Resolver (PRP) permet d'envoyer une requête à un ou plusieurs pairs et de recevoir une ou plusieurs réponses à une requête.
- Le protocole Rendez-vous (RVP) permet la propagation des messages au sein d'un groupe des pairs.
- Le protocole Pipe Binding (PBP) permet à un pair d'établir une communication via un pipe avec un ou plusieurs autres pairs.
- Le protocole Endpoint Routing (ERP) permet à un pair de trouver les routes disponibles pour l'envoi d'un message à un autre pair.

Ces protocoles sont implémentés par l'architecture en trois couches de JXTA (*figure III-1*) et plus précisément au niveau de la couche noyau. Les deux autres couches de l'architecture de JXTA sont la couche service qui comme son nom l'indique, est la couche des services disponibles pour les utilisateurs afin de les aider dans la création d'applications P2P et leur faciliter cette tâche ; et la couche application qui est la couche permettant aux développeurs de créer leur application P2P.

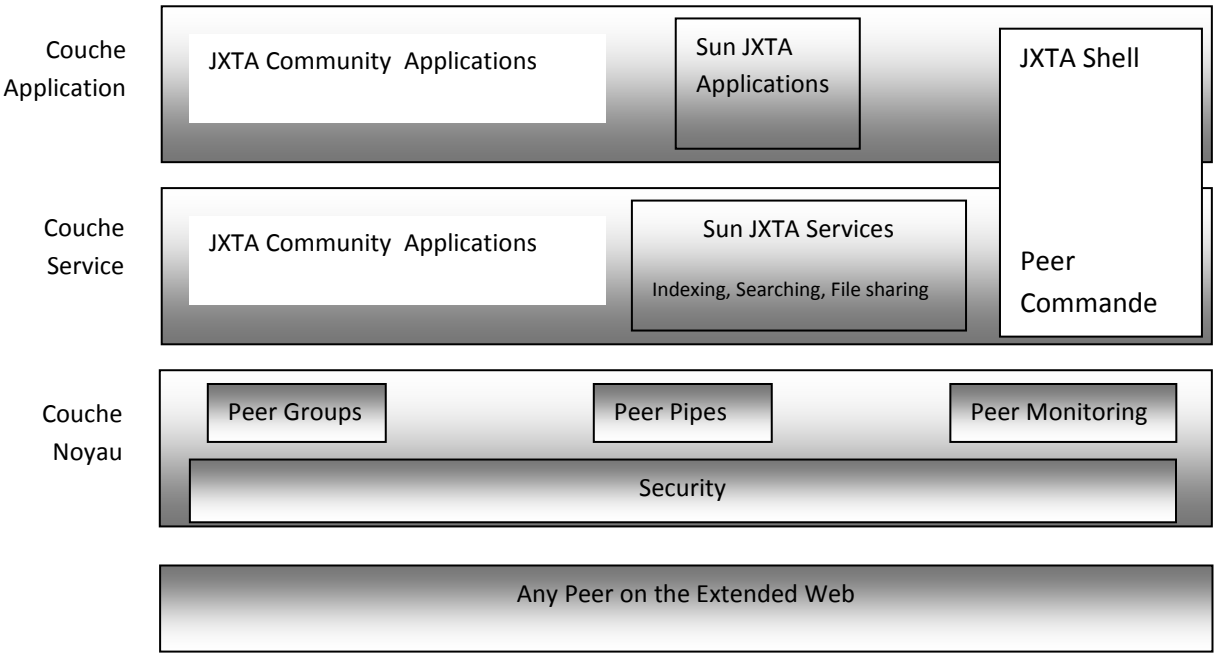


*Figure III-1 Architecture de la plateforme JXTA*

### 3.5.2 JADE

JADE (Java Agent DEvelopment Framework) [Bellefemine, 2004] est une plateforme implémentée en Java qui permet l'implémentation des systèmes multi-agents et des applications conformes aux standards FIPA, notamment pour permettre l'interopérabilité entre agents. Elle permet de réaliser des systèmes P2P dans lesquels chaque pair peut fonctionner de manière proactive, de communiquer sans se soucier de la localisation et de se coordonner pour résoudre un problème complexe grâce à l'utilisation d'agents.

La plateforme Jade se compose principalement de deux composants (*figure III-2*) : Une plateforme agents compatible FIPA qui est fournie avec une License open source LGPL et des packages offrants aux développeurs un ensemble de fonctionnalités pré-développées et prenant en charge la gestion des threads agents, le transport des messages, l'architecture distribuée, etc. JADE utilise le langage de communication FIPA-ACL (Agent Communication Language) et a implémenté le protocole Contract-Net qui a été normalisé par le FIPA.

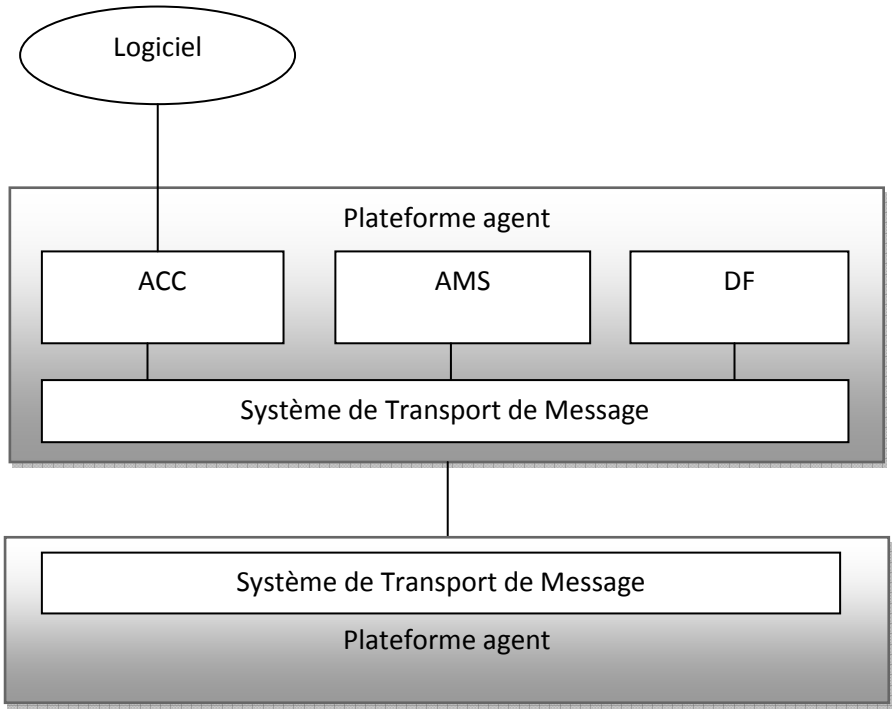


*Figure III-2 Architecture de la plateforme JADE*

Chaque agent JADE correspond à un thread JAVA. D'un point de vue architecture logicielle, un agent peut être vu comme une liste de comportements (classe Behaviour). Chaque comportement est une routine JAVA exécutable. L'ensemble de comportements forme une file d'attente et ceux-ci sont exécutés successivement par un ordonnanceur. Un comportement est composé de deux méthodes, la méthode 'action()', qui est exécutée en premier et qui correspond au code du comportement défini pour l'agent, et la méthode 'done()' qui est exécutée après la méthode action(), et qui vérifie si le comportement doit

être remis dans la file d'attente de l'ordonnanceur ou non. Un comportement peut donc être exécuté plusieurs fois. Un agent n'exécute qu'un seul comportement à la fois [Genin, 2010].

A chaque démarrage de la plateforme, trois modules sont activés :

- le DF (pour Director Facilitor) qui fournit à la plateforme un service de pages jaunes ;
- l'ACC (pour Agent Communication Channel » qui gère la communication entre les agents et
- l'AMS (pour Agent Management System) qui sert de superviseur pour la plateforme.

Tout agent créé doit s'enregistrer auprès de l'AMS pour obtenir son identifiant (AID), indispensable à la communication. En effet, chaque agent peut obtenir la liste de tous les AID des agents de la plateforme et ainsi les agents peuvent communiquer en s'envoyant des messages à partir des AID. Cependant, ceci est un point de centralisation qui constitue un frein à l'adoption de Jade dans des systèmes répartis à grande échelle, et restreint son usage à des clusters ou des grilles de petite taille [Leriche, 2006].

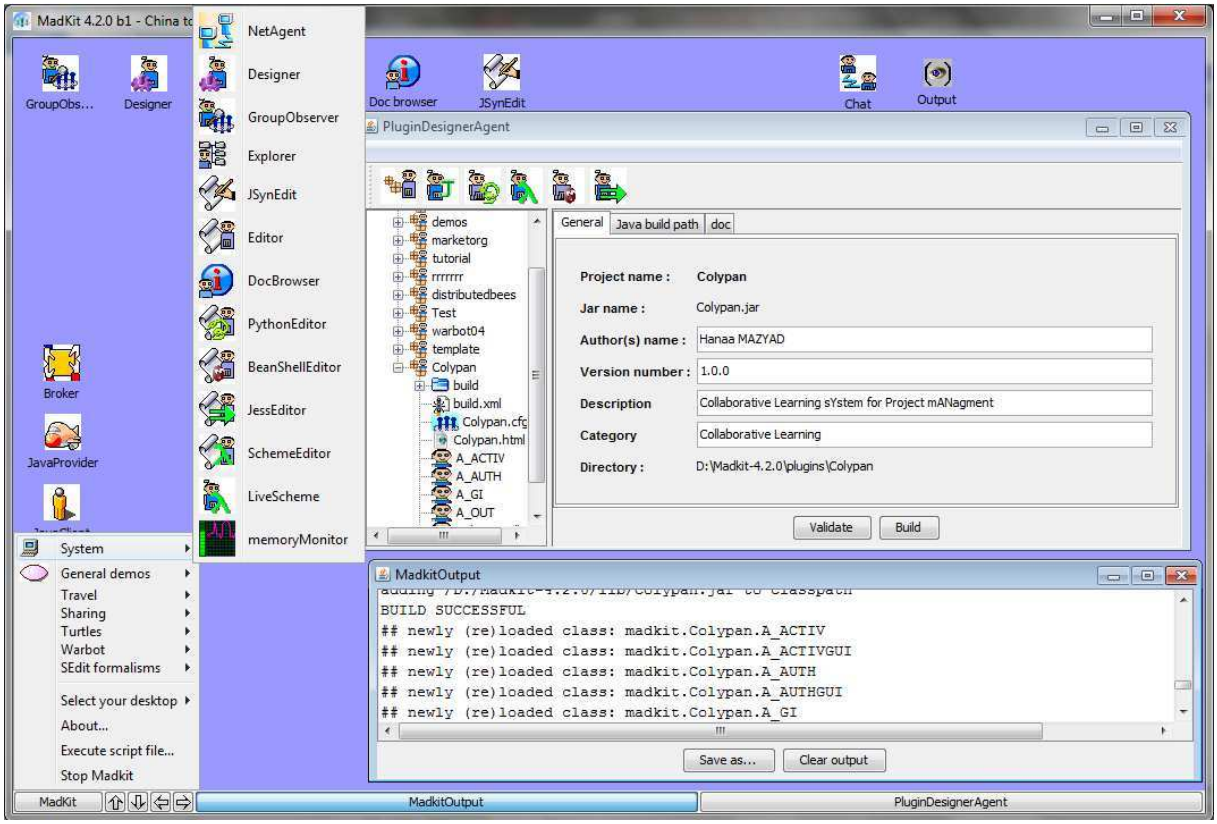


*Figure III-3 Interface de Madkit*

### 3.5.3 Madkit

**Madkit** (Multi-Agent Developpement Kit) est une plateforme multi-agents modulaire et scalable écrite en java *(figures III-3 et III-4*). Elle permet la création des SMA en se basant sur le modèle organisationnel AGR. Ce modèle sert à la réalisation des applications implémentées sur Madkit, mais aussi pour le fonctionnement interne de la plateforme. Madkit se base sur trois principes architecturaux : un micro noyau, une agentification des services et un modèle graphique componentiel [Michel et al, 2005].

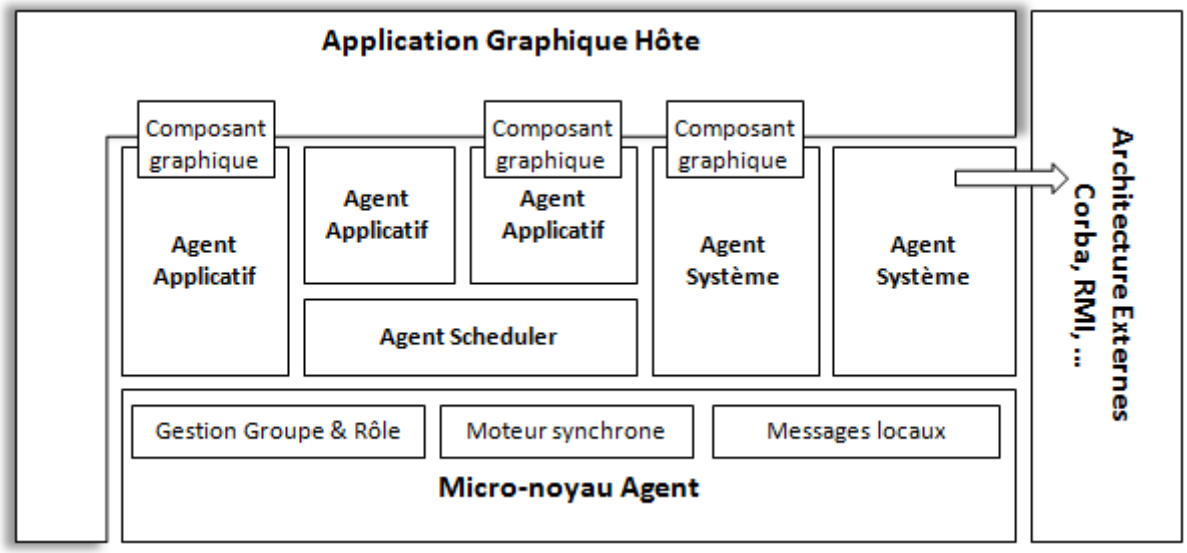


*Figure III-4 Architecture de la plateforme Madkit*

*a) Le micro-noyau*

Le terme "micro-noyau" est utilisé ici de façon intentionnelle. En effet, le système est inspiré du modèle des systèmes d'exploitation à micro-noyau. Le Micro-noyau de MadKit est très petit (moins de 100Ko de code). En effet, il contient un nombre réduit d'outils qui facilitent le déploiement des services du système et se charge de la gestion des mécanismes permettant l'utilisation des notions de groupes et de rôles. En effet, il a la responsabilité de maintenir une information correcte sur les membres des groupes et les rôles tenus. Il définit un ensemble de méthodes qui permettent à un agent de rejoindre un groupe, puis d'y demander un rôle particulier (par exemple : les méthodes createGroup pour créer un groupe ou requestRole pour demander un rôle). Il gère aussi le cycle de vie de chaque agent qui est constitué par : la création, l'activation, l'exécution et la destruction.

Le micro-noyau se charge aussi de la communication à l'intérieur de la plateforme. Différents types de messages sont disponibles et faciles à réaliser et à transmettre. Le message peut être local échangé entre agents sur la même plateforme et se fait par un simple échange de référence ou bien distribué c'est à dire le destinataire se situe dans une plateforme externe sur le réseau et dans ce cas, ce n'est pas le micro-noyau qui se chargera de l'envoi du message mais un agent responsable de la communication distante. Cependant, c'est le noyau qui intercepte l'envoi de message et vérifie s'il est local ou distribué. Il faut noter aussi que Madkit possède une classe AgentAdress qui représente l'adresse d'un agent

où il faut envoyer le message car étant donné qu'un agent peut jouer différents rôles au sein d'un ou plusieurs groupes et communautés, il peut ainsi posséder une adresse pour chaque rôle.

Le noyau lui-même est en fait encapsulé dans un agent particulier, le KernelAgent qui est créé automatiquement à l'amorçage de la plate-forme. Il agit comme représentant du noyau dans le monde agent : toutes les demandes de surveillance ou de contrôle sont alors écrites comme une interaction inter-agent et préservent l'unicité du modèle [Gutknecht et Ferber, 2000].

*b) L'agentification des services*

Le Micro-noyau de Madkit maintient un ensemble minimal de fonctionnalités permettant le déploiement d'autres systèmes. Tous les autres services ne font pas partie du micro noyau et en particulier la communication en distribué. Ces services sont assurés par des agents encapsulés par un seul agent initial : c'est l'agentification des services. Par exemple, l'agent *SiteAgent* assure la communication inter-noyaux en utilisant d'autres agents qui offrent différents modes de communication.

L'agentification des services présente plusieurs avantages. En effet, les interactions entre ces agents systèmes, le noyau, et les agents des applications sont décrites dans le modèle AGR ce qui implique qu'en respectant les groupes, rôles et interactions, un agent offrant une fonctionnalité donnée peut être remplacé par un autre de façon transparente (et éventuellement durant le fonctionnement d'une application). Ceci a également l'effet de faciliter une montée en puissance de l'application puisqu'un agent commence au départ avec un certain nombre de rôles, ensuite, au fur et à mesure que les besoins de l'application grandissent, il délègue dynamiquement à des nouveaux agents certains de ses rôles pour réduire sa charge. Avoir découplé les services du noyau permet aussi de ne pas alourdir les applications: deux plateformes exécutant un SMA localement ou en distribué ne diffèrent que par le lancement ou non des agents de communication.

*c) Le modèle graphique componentiel*

L'interface graphique de Madkit est componentielle et découplée du noyau et des agents ce qui permet de supporter différentes modes d'utilisation et d'exploitation de la plateforme. En effet, Madkit est dotée d'un modèle graphique basé sur des composants graphiques indépendants, utilisant la spécification JAVA Beans dans la version standard. Chaque agent est responsable de ses propres interfaces graphiques. Un noyau graphique se charge de lancer les interfaces et les gérer dans une interface globale. Etant donné que ce noyau graphique est un module logiciel indépendant, il peut être réalisé comme un agent pour une meilleure flexibilité, permettant le contrôle des interfaces des autres agents par un agent MadKit qui peut participer à n'importe quel scénario d'interaction [Gutknecht et Ferber, 2000].

*d) Structure et fonctions d'un agent*

Tout agent qui tourne sous la plateforme Madkit est issu d'une classe d'agent abstraite nommée *AbstractAgent*. Cette classe d'agent offre un certain nombre des fonctionnalités de

base, nécessaires pour le bon fonctionnement d'un agent quel que soit son modèle conceptuel. En effet, l'agent dispose de quatre états (création, activation, exécution et destruction) et a la possibilité de démarrer d'autres agents sur le noyau local. En plus, tout agent peut invoquer de primitives permettant d'observer son organisation locale (connaître les groupes et rôles courants) et d'y agir (prise de rôle, entrée et retraits de groupes) [Gutknecht et Ferber, 2000]. D'autres fonctionnalités existent aussi : les outils de communication et la manipulation d'une interface graphique qui lui est associé.

Un agent crée, ne débutera réellement son cycle de vie qu'une fois il est activé. En fait à l'activation de l'agent, le noyau lui affecte une adresse unique, il devient ainsi identifiable par les autres membres de la société et pourra aussi faire partie de l'organisation et jouer des rôles dans différents groupes [Mansour, 2007]. MadKit n'impose aux développeurs aucune architecture particulière ni aucun concept d'exécution pour l'agent. Cela permet ainsi aux développeurs d'implémenter l'architecture de leur choix et de programmer le comportement de l'agent [Ferber, 2009].

### 3.5.4 Comparaison des plateformes

Nous avons besoin, pour construire notre système, d'une plateforme de développement de systèmes multi-agents capable de supporter les architectures P2P. Pour cela, nous avons comparé les deux plateformes multi-agents présentées ci-dessus à savoir : Jade et Madkit.

Nous nous sommes basés sur les critères[12] définis par [Guessoum et al, 2002] pour choisir ceux qui correspondent à nos besoins.

| | Jade | Madkit |
|---|---|---|
| Langages utilisés | Développée en Java | Développée en Java |
| Systèmes d'exploitation compatibles | Tous | Tous |
| Logiciel libre | Oui | Oui |
| Organisation | | |
| Type de structure organisationnelle | ***Basé sur un modèle P2P centralisé*** | ***Basé sur le modèle P2P sans nécessiter de serveur dédié*** |
| Utilisation du concept du groupe | Oui | Oui |

[12] http://www-poleia.lip6.fr/~guessoum/asa/criteres.html

| Agent | | |
|---|---|---|
| Caractéristiques | Un thread qui a une liste des comportements (objets de type Behaviour) | Aucune architecture n'est imposée<br>Chaque agent est issu de la Classe « AbstractAgent » et dispose de 4 états (création, activation, exécution et destruction) |
| Interaction / communication | | |
| Langage de communication | FIPA ACL | Pas de langage prédéfini de communication |
| Communication distribuée | Oui | Oui |
| Environnement de développement | | |
| Méthodologie d'analyse | Non (construite selon les normes FIPA) | Basée sur le modèle AGR |
| Nécessite la connaissance du langage de développement | Oui | Possibilité d'interprétation du code avec d'autres langages (Scheme, Python, BeanShell et Jess) |
| Standard | Conforme aux spécifications FIPA | Non |
| Bibliothèque | Disponible | Disponible |
| Documentation | Riche | Riche |
| Tutoriel | Disponible | Disponible |
| Environnement d'exécution | | |
| Aide en ligne | Oui | Oui |
| Manuels d'installation | Disponible | Disponible |
| Manuels d'utilisation | Disponible | Disponible |

**Tableau III-1comparatif des plateformes**

Nous avons choisi Madkit pour l'implémentation de notre système. Ce choix est motivé par le fait que cette plateforme est particulièrement destinée à des SMA fondés sur des critères organisationnels et qu'elle fonctionne en mode distribuée de manière transparente à partir d'une architecture P2P sans nécessiter de serveur dédié ce qui n'est pas le cas de Jade qui est basé sur un modèle P2P centralisé.

En plus, on reconnait généralement à MadKit les qualités suivantes [Ferber, 2009]:

- Simplicité de mise en œuvre et de prise en main,
- Intégration très facile de la distribution au sein d'un réseau,
- L'aspect pratique et efficace des concepts organisationnels pour créer différents types d'applications,
- Hétérogénéité des applications et des types d'agents utilisables: on peut faire tourner sur MadKit aussi bien des applications utilisant des agents réactifs simples que des applications disposant d'agents cognitifs sophistiqués.

## 3.6 Conclusion

Dans ce chapitre, nous avons proposé d'associer les systèmes multi-agents avec les systèmes P2P. Nous avons montré que cette association a ses avantages mais pose aussi des défis notamment à cause du caractère dynamique des systèmes P2P. Nous nous sommes intéressés particulièrement à la coordination d'agents dans un contexte P2P. Nous avons présenté des travaux permettant d'adapter les mécanismes de coordination des systèmes multi-agents pour prendre en compte les caractéristiques des systèmes P2P et principalement la formation de coalition. Cependant, nous pensons que le fait de faire partie d'un groupe a des effets positifs sur la coalition. Ceci a été confirmé par une étude faite par [Tremewan, 2010]. Ainsi, afin de résoudre les problèmes de coordination d'agents dans un contexte P2P, nous proposons dans le chapitre 5 de former un groupe d'agents qui partagent des buts communs, qui ont les différentes ressources nécessaires pour accomplir ces buts et qui vont collaborer afin de les réaliser.

Dans la suite de cette thèse, nous nous intéressons à l'application des systèmes multi-agents à architecture P2P au domaine d'apprentissage collaboratif en ligne. En effet, les systèmes multi-agents s'adaptent bien à la conception d'un environnement d'apprentissage où chaque membre doit gérer et échanger ses connaissances et collaborer avec les autres afin de réaliser ses buts. En plus, dans des tels environnements ouverts, dynamiques et complexes, il y a un besoin de distribuer les données, le contrôle ainsi que l'expertise ce qui rend l'utilisation des systèmes P2P bénéfique.

Nous commençons d'abord dans le chapitre suivant par la présentation de l'apprentissage collaboratif en ligne. Dans ce type d'apprentissage, le groupe prend tout son sens et l'apprenant et l'enseignant ont des nouveaux rôles à jouer. Nous étudierons quelques plateformes d'e-Learning classiques ainsi que quelques-unes orienté-agent afin de situer nos travaux par rapport à l'existant.

# Chapitre IV. L'apprentissage : de la théorie à la pratique

## 4.1 Avant-propos

L'apprenant acquiert ses connaissances et compétences selon deux formes d'apprentissage. Le premier est l'apprentissage structuré, contrôlé par un tuteur et dispensé au sein de l'école ou des structures similaires et qui utilise des techniques traditionnelles qui aboutissent à une reconnaissance de type diplômes, certificat, attestation ou autre. Le deuxième est l'apprentissage par la pratique sociale où chaque apprenant accumule des compétences et des connaissances à travers les interactions avec les apprenants ou autres personnes contribuant à cet apprentissage. C'est un processus de longue durée qui n'est pas structuré ou contrôlé par les tuteurs. Nous abordons dans ce chapitre, l'apprentissage à distance et plus précisément l'apprentissage collaboratif à distance qui prend en compte ces deux formes d'apprentissage.

## 4.2 L'apprentissage à distance et l'apprentissage en ligne

L'apprentissage à distance est une méthode de formation conçue initialement pour les adultes. Son objectif est de rendre la formation accessible au plus grand nombre et de permettre à l'apprenant de contrôler sa démarche. En effet, c'est l'apprenant qui choisit le moment, le lieu et le rythme de son apprentissage.

Plusieurs modèles d'apprentissage à distance ont vu le jour. Ces modèles utilisent des technologies et des médias différents mais ils ont tous les mêmes intentions pédagogiques et le même objectif d'accessibilité. On cite pour exemple : l'enseignement par correspondance, télé-enseignement, etc.

Les Technologies de l'Information et de la Communication en Education (TICE) ont fait apparaître un autre modèle d'apprentissage à distance, il s'agit de l'apprentissage en ligne (ou e-Learning). Ces technologies sont, en peu de temps, devenues l'un des piliers de la société moderne au point de considérer la compréhension de ces technologies et la maîtrise de leurs principaux concepts et savoir-faire comme partie intégrante de l'éducation de base, au même titre que la lecture, l'écriture et le calcul [Unesco, 2004]. Ainsi, le fait de combiner les technologies issues de l'informatique avec ceux de la communication et les mettre au service d'e-Learning est censé améliorer la qualité des apprentissages mais ceci n'est possible qu'en proposant, d'une part, un modèle pédagogique adapté et en intégrant, d'autre part, les apprenants et leurs tuteurs dans des communautés d'apprentissage. En effet, ceci permettra l'échange, la discussion et la négociation entre les constituants de cette communauté qui constituent des facteurs essentiels dans le processus d'appropriation des connaissances tout en évitant l'isolement des apprenants et en favorisant leur suivi.

Nous commençons dans ce chapitre par présenter les différentes théories d'apprentissage existantes afin d'introduire le travail collaboratif dans un contexte d'e-Learning. Nous

étudions, ensuite, quelques plateformes d'e-Learning ainsi que quelques travaux sur l'utilisation des systèmes multi-agents dans le domaine d'e-Learning.

## 4.3 Les théories d'apprentissage

Le béhaviorisme dominait la recherche en éducation avant que d'autres théories émergent à la fin des années 50, notamment le cognitivisme et le constructivisme. Nous adoptons le point de vue du constructivisme qui, contrairement au behaviorisme qui limitait trop l'apprentissage à l'association stimulus-réponse, s'est intéressé à l'activité de l'apprenant qui construit ses connaissances de façon continue grâce aux interactions incessantes avec les autres apprenants ainsi qu'avec son environnement.

### 4.3.1 Le béhaviorisme

Le béhaviorisme appelé aussi la théorie comportementale a émergé au début du 20ème siècle par Watson [Watson, 1930] qui a souligné le travail empirique basé sur des observations des comportements. En effet, selon les béhavioristes, l'apprentissage se définit par des comportements observables et mesurables sans prendre en compte ce qui se passe dans la tête des individus au cours du processus d'apprentissage. Ce déroulement est perçu comme une boîte noire où se contente de limiter l'observation à l'interface. Dans ce modèle, l'enseignant est au centre d'intérêt où il stimule l'apprenant en posant des questions pour l'amener à émettre une réponse, l'encourage en récompensant son travail et augmente sa rétention des connaissances en se basant sur une répétition de l'apprentissage des leçons visées.

Pour les béhavioristes, les médias sont essentiellement des véhicules de transmission de l'information et de contrôle du comportement de l'apprenant. Leurs outils privilégiés ont été d'abord l'imprimé, les médias audiovisuels et l'enseignement assisté par ordinateur [Basque, 1999].

### 4.3.2 Le cognitivisme

La révolution cognitive a eu lieu [Gardner, 1993] à partir des années 1950. Ce courant prône une nouvelle approche pour expliquer le comportement humain : l'approche du traitement de l'information. Les cognitivistes s'intéressent essentiellement à ce qui se passe dans la tête des individus lorsque ces derniers apprennent, résolvent des problèmes ou effectuent diverses tâches [Basque, 1999].

La vie psychique est pour le cognitivisme constituée d'un certain nombre d'opérations logiques de contrôle, de régulation, de calcul et de mémoire, tout comme un ordinateur. L'esprit est constructeur de représentations symboliques qu'il organise, gère et manipule comme peut le faire un ordinateur: consultation de données, exécution d'opérations.

Les cognitivistes visent des apprentissages durables, flexibles, transférables et autorégulés, ce qui signifie que les connaissances doivent pouvoir être utilisées sur demande, sans aide externe et de manière appropriée à la situation [Di Vesta et Rieber, 1987].

Les cognitivistes tendent à favoriser les médias interactifs, qui exigent un engagement et un effort mental de la part des étudiants; l'humain et l'ordinateur sont deux « médias » particulièrement prisés par les cognitivistes à cause précisément de leur grande capacité interactive [Bonner, 1988]. Les cognitivistes favorisent également la création d'environnements d'apprentissage adaptatifs ou réactifs, c'est-à-dire qui peuvent s'adapter aux différences individuelles [Bonner, 1988; Fleury, 1994; Merrill, Jones et Li, 1992; Winn et Snyder, 1996], qu'il s'agisse de systèmes tutoriels intelligents [Bonner, 1988; Jonassen, 1994 ; Johnson et Thomas, 1994; Winn et Snyder, 1996] ou d'environnements multimédias ou hypermédias [Fleury, 1994; Winn et Snyder, 1996].

### 4.3.3 Le constructivisme

En s'opposant au réalisme, le constructivisme adopte une vision subjectiviste du monde qu'on associe aux positions épistémologiques d'auteurs comme Piaget, Von Foerster, Bateson et Varela [Henri et Lundgren-Cayrol, 1998]. L'essence de la philosophie constructiviste s'exprime dans les conceptions qu'elle propose de la réalité, de la connaissance, de l'interaction humaine et de la science [Wilson, 1992].

Le constructivisme se focalise sur le raisonnement de l'apprenant ainsi que sur sa façon de penser. Selon les constructivistes, l'apprenant se base d'une façon ou d'une autre sur ce qu'il a précédemment acquis pour comprendre toute nouvelle connaissance. Ainsi, l'apprentissage est un processus actif de construction des connaissances. Les constructivistes considèrent que le monde n'existe pas indépendamment de l'activité mentale ; chacun construit ses propres interprétations [Jonassen, 1991]. Les connaissances ne sont pas des vérités absolues ; ce sont simplement des interprétations viables à un moment donné de notre contexte [Resnick, 1987] [Henri et Lundgren-Cayrol, 1997].

Les constructivistes considèrent les médias avant tout comme des outils d'apprentissage et non des outils d'enseignement [Duffy et Cunningham, 1996]. Les technologies informatiques ont une place de choix dans les environnements d'apprentissage constructivistes, qu'il s'agisse des technologies hypertextes, hypermédias et multimédias [Duffy et Cunningham, 1996] [Honebein, Duffy et Fishman, 1993] [Jonassen, 1990] [Spiro et al, 1991], de l'internet [Duffy et Cunningham, 1996] ou d'environnements de réalité virtuelle [Winn et Snyder, 1996].

Nous nous intéressons particulièrement à l'apprentissage collaboratif qui repose sur sept valeurs qui puisent leurs sources dans la philosophie constructiviste : collaboration, réflexivité, générativité, engagement actif, pertinence personnelle et pluralisme [Lebow, 1993]. Nous tenterons de définir cet apprentissage dans la section suivante.

## 4.4 L'apprentissage collaboratif

L'apprentissage collaboratif [Henri et Lundgren-Cayrol, 2001] est défini comme 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'entre-aide et

de soutien mutuel et comme lieu privilégié d'interaction pour la construction collective des connaissances.

Il faut préciser que l'apprentissage collaboratif n'est pas synonyme d'apprentissage coopératif. Ainsi, nous montrons, dans ce qui suit, la différence entre ces deux termes. Nous mettons l'accent ensuite sur le rôle du groupe et du tuteur ainsi que sur les habilités que doivent posséder les apprenants pour réussir un apprentissage collaboratif.

### 4.4.1 Différence entre apprentissage collaboratif et coopératif

On confond souvent les termes apprentissage collaboratif et apprentissage coopératif. Il est important de montrer que coopération n'est pas synonyme de collaboration et les deux démarches collaborative et coopérative sont différentes. Plusieurs auteurs [Adams et Hamm, 1990; Cavalier, Klein et Cavalier, 1995] conçoivent la démarche coopérative comme une méthode d'initiation ou de préparation à la collaboration véritable.

Ce qui distingue essentiellement les deux démarches, c'est le degré d'autonomie des apprenants et le contrôle qu'ils exercent sur leur apprentissage. En effet, la démarche coopérative convient au profil des jeunes apprenants. Ils sont moins autonomes, n'ont pas acquis beaucoup de maturité cognitive et ne maîtrisent pas encore tous les stratégies d'apprentissage. Dans la démarche coopérative, c'est le tuteur qui exerce le contrôle sur l'apprentissage pour développer graduellement chez l'apprenant des habilités de collaboration et l'amener à devenir autonome. Cependant, la démarche collaborative convient aux adultes. Elle propose aux apprenants plus de liberté.

Dans les deux démarches, les groupes d'apprenants travaillent afin d'atteindre un but commun. Cependant, dans la démarche coopérative, chaque membre est responsable d'une activité. La tâche est divisée en sous-tâches et chaque apprenant est responsable, individuellement, d'une sous-tâche. La tâche est complète lorsque tous les membres du groupe mettent en commun leurs réalisations. Ainsi, l'ensemble permet au groupe d'atteindre son but ce qui rend les apprenants dépendants l'un de l'autre.

La démarche collaborative de son côté fait en sorte que chaque membre, individuellement, atteigne le but mais trouve auprès du groupe inspiration, soutien et appui. En effet, tous les membres d'un groupe collaboratif s'engagent de manière volontaire et spontanée, participent à des travaux de groupe, mettent en commun leurs idées et font part de leurs réalisations afin d'atteindre le but. Ainsi, les contributions sont à la mesure de ce que chacun est prêt à donner.

### 4.4.2 Habilités de l'apprenant et rôles du tuteur

L'idée principale de l'apprentissage collaboratif est de mettre l'étudiant au centre du système [Dillenbourg, 1999]. Les apprenants sont censés participer activement dans le processus d'apprentissage en discutant, partageant leurs connaissances, posant et répondant aux questions tout au long de ce processus. Cependant, les apprenants doivent être prêts avant de s'engager dans la collaboration afin qu'ils puissent assumer la

responsabilité de leur apprentissage. En effet, la collaboration nécessite la maîtrise de trois grandes habiletés : l'analyse, la conception et l'évaluation. Chaque apprenant doit être capable de rechercher et d'analyser l'information ; d'argumenter ; d'utiliser ses connaissances ; de négocier ; de critiquer de façon constructive ; de faire de concessions ; etc.

L'apprenant acquiert petit à petit ces habiletés en jouant divers rôle à chaque phase de la collaboration ainsi il devient plus autonome et prend de plus en plus de responsabilité dans son apprentissage. Le tuteur a alors comme rôle de faciliter l'acquisition des connaissances [Feenberg, 1989] et d'accompagner les apprenants dans leur démarche d'apprentissage. Il les conseille, les guide, les aide à comprendre le contenu d'apprentissage et à collaborer pour aboutir à leur objectif commun et les encourage à devenir autonome. Cependant, le rôle de tuteur évolue au rythme de l'évolution des apprenants eux-mêmes qui graduellement deviennent de plus en plus autonome et de plus en plus habiles à collaborer [Henri et Lundgren-Cayrol, 1998] alors que le tuteur s'efface graduellement [Lave et Wenger, 1991]. L'évolution du rôle du tuteur permet à l'apprenant d'exercer un contrôle plus grand, de développer sa responsabilité et son autonomie face à son apprentissage. Toutefois, le tuteur demeure toujours évaluateur afin de pouvoir intervenir adéquatement auprès des apprenants.

### 4.4.3 Le groupe dans l'apprentissage collaboratif

La notion de groupe est très essentielle dans l'apprentissage collaboratif. En effet, les apprenants ne sont plus vus d'une façon individuelle, même si chacun travaille de façon autonome, mais plutôt comme une communauté qui partage un savoir commun et essaye d'atteindre un objectif commun de façon collective. Chaque apprenant doit avoir le désir de réussir, la volonté de participer et de collaborer, s'engager dans la réalisation d'activités collectives et accepter le mode de fonctionnement du groupe avec ce qu'il apporte comme contraintes mais aussi comme gratification et satisfaction. Ce sont des conditions essentielles pour que le groupe survive et réussisse à atteindre ses objectifs.

Dans la section suivante, nous présentons une étude de l'existant afin de situer nos travaux dans le domaine d'apprentissage et plus particulièrement celui de l'apprentissage collaboratif. Pour cela, nous ferons un tour d'horizon sur les plateformes d'apprentissage collaboratif et nous présenterons ensuite quelques travaux qui se sont intéressés à l'utilisation du paradigme agent dans l'implémentation de leur outil dédié à l'apprentissage ainsi que des travaux sur le suivi des apprenants.

## 4.5 Plateformes d'e-Learning classiques

Une plate-forme d'e-Learning est un logiciel de création et de gestion de contenus pédagogiques et de support aux acteurs destiné à trois types d'utilisateurs : L'enseignant, l'apprenant et l'administrateur. Elle regroupe les outils nécessaires aux trois types d'intervenants permettant d'incorporer des ressources pédagogiques multimédias, de

participer à des activités et d'effectuer le suivi pédagogique et administratif des apprenants [Oubahssi, 2005].

Dans une plate-forme d'e-Learning, l'apprenant suit depuis son poste de travail une formation dont le contenu est organisé selon un parcours pédagogique défini par le formateur en fonction des compétences possédées par l'apprenant avant la formation, des objectifs pédagogiques visés par la formation et des résultats obtenus lors des évaluations intermédiaires. Il organise et a une vue de l'évolution de son travail, s'auto-évalue, et transmet des travaux à corriger. Quant au tuteur, son rôle est fondamental : il ne s'agit plus de transmettre des connaissances, mais d'accompagner l'apprenant dans ses acquisitions, de lui faire acquérir un maximum d'autonomie. Cependant, il a aussi pour rôle de créer les parcours de formation types, d'incorporer des ressources pédagogiques multimédias et d'effectuer un suivi des activités des apprenants.

Le groupe est là pour combattre la sensation d'isolement que pourrait ressentir l'apprenant dans une classe virtuelle et pour maintenir sa motivation. L'administrateur assure la maintenance du système, gère les comptes et les droits des utilisateurs, crée des liens avec les systèmes d'information externes (dossiers administratifs, catalogues, ressources pédagogiques, etc.).

Il existe des multitudes de plateformes d'e-Learning et leur nombre ne cesse de croître, il nous est impossible de toutes les recenser. Le site de Thot[13] cursus recense plusieurs centaines de logiciels de ce type.

Nous présentons dans ce qui suit trois plateformes libres et gratuits et qui supportent le travail collaboratif, à savoir: Claroline[14], Sakai[15], Moodle[16].

### 4.5.1 Sakai

Sakai est une plateforme d'enseignement développée au départ par quatre universités américaines. Elle se distingue des plus employées en France, par l'accent mis beaucoup plus sur l'aspect travail collaboratif que sur l'aspect suivi pédagogique semaine par semaine. Cet environnement de travail collaboratif peut d'ailleurs être employé dans d'autres contextes et, à la création d'un cours, on doit choisir entre deux configurations : cours ou projet. Les outils proposés, les rôles sont alors différents. Certaines universités emploient Sakai uniquement dans cette seconde configuration dans un contexte beaucoup plus large que l'enseignement[17].

Sakai est développé en Java et utilise plusieurs technologies issues de J2EE. Son installation et les mises à jour sont complexes mais très documentées. La plateforme est facile à prendre en main tant par les enseignants que par les apprenants.

[13] http://cursus.edu/institutions-formations-ressources/formation/13486/plates-formes-learning-formation-2010/
[14] http://www.claroline.net
[15] [http://www.sakaiproject.org]
[16] http://moodle.org
[17] https://www.projet-plume.org/fiche/sakai

Sakai dispose d'une communauté de plus de 80 universités dans plusieurs pays dans le monde. Sa documentation[18] en langue française, à l'origine canadienne, a été entièrement revue à l'UPMC en coopération avec l'université du Littoral.

Sakai dispose des outils permettant le travail collaboratif tels que le wiki, les forums, chat. Les inscrits à un cours peuvent être organisés en groupes et de nombreux outils permettent de s'adresser à des groupes sélectionnés.

Sakai propose un outil pour le suivi des apprenants qui n'est disponible que pour les enseignants. Il permet de suivre l'usage d'un cours par les étudiants : qui se connecte, quand, quelles ressources sont employées. Cependant, les statistiques sont journalières et ne peuvent être remises à zéro. En effet, cet outil prend en considération tous les événements depuis la date de création d'un cours. Il est possible d'exporter les résultats dans des tableaux compatibles Excel pour effectuer des traitements plus complexes sur les données.

### 4.5.2 Claroline

Claroline est une plate-forme de formation à distance et de travail collaboratif développée en 2002 par l'université de Louvain en Belgique. Elle permet aux formateurs de créer des espaces de cours en ligne et de gérer des activités de formation sur Internet.

Claroline est développée en PHP et est compatible avec tous les systèmes d'exploitation et tous les navigateurs. La plateforme est facile à prendre en main tant du côté des étudiants que du côté des enseignants.

Claroline bénéficie de l'appui d'une communauté mondiale d'utilisateurs et de développeurs. Sa documentation[19] est complète et organisée en plusieurs langues de l'installation à l'utilisation.

Claroline dispose des outils permettant le travail collaboratif tels que le wiki, forum, chat mais aussi un outil Groupes qui permet au responsable de cours de créer des groupes de travail avec les étudiants inscrits dans son cours, en spécifiant des critères d'inscription. Le responsable de cours décide de laisser les apprenants s'inscrire eux-mêmes dans le/les groupes, de les inscrire lui-même de façon manuelle ou bien automatiquement en les dispersant dans les différents groupes. Un groupe rassemble des utilisateurs qui partagent certains outils : une zone de documents, un forum, un espace de discussion et un Wiki. L'outil Groupes permet aux apprenants de collaborer efficacement lors d'un travail de groupe ou d'une séance de travail à distance.

Claroline propose en plus un outil destiné aux responsables du cours leur permettant d'avoir une vue globale de l'activité dans leurs cours. Ils peuvent ainsi connaître les étudiants qui ne sont jamais connectés ou ceux qui ne sont pas connectés récemment ; le nombre de connexion à chaque cours ; les outils utilisés, les documents téléchargés, les exercices

[18] http://wiki.cpm.jussieu.fr/doku.php?id=libre:sakai:bienvenue

[19] http://www.claroline.net/documentation/

effectués, et les statistiques de fréquentation du forum. Le responsable peut aussi avoir ces statistiques par rapport à un utilisateur donné : Utilisation du forum, travaux envoyés, documents téléchargés…

### 4.5.3 Moodle

Moodle (acronyme de Modular Object-Oriented Dynamic Learning Environment) est un logiciel libre créée à la base par Martin Dougiamas à l'université de Curtin en Australie en 2002. C'est une plateforme d'apprentissage en ligne servant à créer des communautés d'apprenants autour de contenus et d'activités pédagogiques. À un système de gestion de contenu (SGC), Moodle ajoute des fonctions pédagogiques ou communicatives pour créer un environnement d'apprentissage en ligne : c'est une application permettant de créer, par l'intermédiaire du réseau, des interactions entre des pédagogues, des apprenants et des ressources pédagogiques.

Moodle est principalement développé sur Linux avec Apache, Mysql et PHP et est compatible avec tous les systèmes d'exploitation ainsi que les navigateurs Web. En plus, aucune configuration spéciale n'est requise pour l'utilisation de Moodle côté client.

L'interface de Moodle se présente sous forme de différents blocs : le bloc central présente les documents et les activités du cours. Les blocs latéraux affichés sur les pages web donnent accès aux différents outils et liens du cours. Cependant, Moodle est très riche en fonctionnalités ce qui nécessite un temps d'adaptation aux apprenants pour qu'ils puissent la prendre en main. Par ailleurs, la diversité et la spécificité de tous les paramétrages des outils peuvent être complexes pour les enseignants.

Moodle est utilisé dans des nombreuses universités et organismes de formation, bénéficie d'un développement très actif à l'échelle mondiale et dispose d'une communauté active sur internet de plus de 200 000 utilisateurs enregistrés, parlant plus de 75 langues dans plus de 175 pays. Sa documentation[20] en ligne est très détaillée et bien organisée.

La fonctionnalité Groupes permet à un enseignant de constituer des groupes d'étudiants à l'intérieur d'un cours. Les participants d'un cours peuvent appartenir à plusieurs groupes. Il est possible de créer également des groupements (groupes de groupes) et des cohortes qui sont des groupes d'étudiants totalement indépendants mais qui partagent les mêmes ressources dans le même espace. Les groupes peuvent être créés par l'enseignant ou automatiquement.

Les groupes peuvent être séparés ce qui signifie que chaque groupe ne peut voir que son propre groupe et que les contributions aux activités des membres des autres groupes sont invisibles ou bien visibles ce qui signifie que chaque utilisateur ne contribue qu'au sein de son propre groupe, mais peut accéder uniquement en lecture aux contributions des membres des autres groupes.

[20] http://docs.moodle.org/overview/

Pour le suivi des apprenants, Moodle propose une page Rapports qui permet aux enseignants de consulter les rapports d'activité et historiques (logs) pour un cours ou pour le site entier pour ceux qui possèdent les droits d'administrateur. Les rapports de cours montrent l'activité à l'intérieur d'un cours sous forme des logs. Ils permettent aux tuteurs de voir quelles ressources sont consultées, et quand, ou encore de vérifier qu'un étudiant précis a bien vu la ressource qu'il prétend avoir lu. Un rapport d'activité permet de connaitre l'activité effectuée globalement pour un cours par les différents utilisateurs et ceci en affichant le nombre des documents vus (views) et le nombre des actions effectués (Posts), ou d'avoir le détail des activités d'un utilisateur. Un rapport de participation fournit un moyen facile de surveiller la participation des étudiants. Ce rapport liste les actions à faire et détermine si oui ou non l'étudiant l'a effectuée. Les rapports d'activités peuvent être générés sous forme graphique ou dans des tables.

L'enseignant dispose de 6 options de filtrage sous la forme de listes déroulantes pour avoir son rapport : Nom du cours ; Utilisateurs concernés : tous les participants ou un seul ; Période de temps concernée (jours) ; Activités concernées : toutes ou une seule ; Actions concernées : toutes, une seule, ou toutes les modifications ; Format des résultats : affichage web, ou export Texte/ODS/Excel.

### 4.5.4 Discussion

Nous avons présenté ci-dessus trois plateformes d'e-Learning libres et gratuits et qui supportent le travail collaboratif. Nous avons voulu mettre la lumière sur les trois aspects qui nous intéressent le plus c'est-à-dire les outils permettant le travail collaboratif, le groupe ainsi que le suivi des apprenants.

On a vu que les groupes se forment par l'enseignant ou automatiquement de façon aléatoire. Claroline permet une troisième option l'inscription de l'apprenant lui-même dans un ou plusieurs groupes.

Quant aux suivis de l'apprenant, chaque plateforme traite le sujet différemment. Sakai propose un suivi basique en récoltant des informations générales sur l'utilisation de la plateforme (qui se connecte, quand, ressources utilisées…). Claroline va plus loin en récoltant les informations sur les utilisateurs ainsi que sur leurs activités. Moodle, de son côté, récolte toutes les activités produites sur le site et dans un cours. Cependant, les informations récoltées sont affichées la plupart du temps, sous forme des logs. Par ailleurs, les logs deviennent rapidement difficiles à traiter quand le nombre d'activités tracées devient grand.

En conclusion, Sakai permet un suivi minimal de l'apprenant. Moodle permet le suivi des apprenants en récoltant les traces des activités de chaque apprenant mais engendre un tas d'information difficile à exploiter ce qui rend la tâche plus difficile pour l'enseignant qui se trouve ainsi rapidement débordé. De son côté, Claroline proposent un suivi bien structuré et organisé. Cependant, nous trouvons que ces plateformes en particulier et les plateformes d'e-Learning en général doivent prendre plus en compte l'enseignant. En effet, le suivi des apprenants est une tâche parmi beaucoup d'autres que doit accomplir l'enseignant. Ainsi,

pour que le suivi des apprenants soit le plus efficace possible, il faut doter les plateformes des outils capables de traiter les informations récoltées et ainsi renseigner les enseignants à tout moment sur l'état des apprenants (actif, passif…) ainsi que sur la progression de leur travail.

## 4.6 Travaux similaires

Nous avons présentés ci-dessus quelques plateformes d'e-Learning. Ces plateformes offrent des nombreuses fonctionnalités. Cependant, d'autres plateformes moins connues et qui utilisent le paradigme objet ou agent existent.

### 4.6.1 Travaux utilisant le paradigme objet

**Reflet** [Després et Coffinet 2004] est un outil, développé en Java et destiné à être couplé aux plateformes informatiques de formation ouverte et à distance (FOAD). Il permet de visualiser l'état d'avancement d'un étudiant et d'une promotion d'étudiants en FOAD. Dans Reflet, chaque activité est décomposée en une ou plusieurs tâches. L'étudiant fournit l'information sur son état d'avancement en déclarant l'achèvement de chaque tâche à réaliser. Ainsi, l'étudiant peut visualiser son avancement par rapport aux objectifs de la formation et par rapport aux autres étudiants de sa promotion dont il peut aussi visualiser leur avancement mais anonymement. Quant au tuteur, il peut visualiser l'avancement d'un étudiant dont il a la charge ou comparer l'avancement de tous les étudiants d'un même groupe. Il peut également « contrevalider » des tâches déjà validées par les apprenants.

**Croisières** [Gueye, 2005] est un projet qui a pour but de définir des services accompagnant l'apprenant dans son trajet d'apprentissage [Teutsch et al, 2004] et de l'assister en situation d'autonomie. Sa structuration modulaire permet aux apprenants de sélectionner leurs activités d'apprentissage en fonction de leurs objectifs et de leurs stratégies d'apprentissage. Les recherches ont été centrées sur le suivi à distance et la régulation des activités des apprenants dans le cadre d'une formation ouverte et à distance en Français comme langue étrangère. La problématique traitée par ce projet est l'observation des activités des apprenants et la synthèse des données observées pour définir celles qui seront utiles au tuteur ainsi que la définition des outils permettant la présentation de ces données au tuteur afin de pouvoir exercer les rôles qui lui incombent auprès de l'apprenant.

**FORMID** (FORmation Interactive à Distance) [Guéraud et al, 2007] est une plateforme, basée sur une architecture client-serveur, et qui vise à faciliter la tâche de suivi des apprenants par le tuteur en synchrone et à distance en s'appuyant sur l'utilisation des scénarios pédagogiques. En effet, cette plateforme propose une solution générique pour le suivi de classes d'apprenants engagés dans des Situations Actives d'Apprentissage (SAA) exploitant des logiciels tels que des simulations. FORMID permet à l'enseignant de concevoir un exercice ainsi qu'un scénario indiquant les éléments significatifs à observer pour le suivi des apprenants sur cet exercice. Ensuite, l'outil FORMID-Elève met en œuvre l'exercice pour l'élève tout en exécutant le scénario proposé. Ainsi, il permet de générer les traces associées: étapes validées avec succès ou non, détection de situations à observer. Ces traces

laissées par les élèves sur l'ensemble de SAA de la séance suivie sont alors observées par le tuteur grâce à l'environnement FORMID-Suivi.

### 4.6.2 Travaux utilisant le paradigme agent

Beaucoup des chercheurs se sont intéressés à l'application des systèmes multi-agents au domaine d'e-Learning. En effet, les systèmes multi-agents proposent une approche originale de conception de systèmes intelligents et coopératifs et apportent une solution intéressante aussi bien aux problèmes de structuration et d'exploitation des connaissances qu'aux problèmes liés aux mécanismes de coordination et de communication. Les SMA sont donc appropriés pour la conception d'un environnement d'apprentissage où chaque membre doit gérer et échanger ses connaissances et collaborer avec les autres afin de réaliser ses buts. Ainsi, beaucoup des plateformes ont vu le jour :

**Baghera** [Pesty et al, 2003] est une plateforme développée au Laboratoire Leibniz. Elle est conçue comme un système multi-agents qui permet à des élèves, des enseignants et des agents artificiels de collaborer à distance sur Internet pour résoudre des problèmes et produire des preuves en Géométrie. L'architecture multi-agents de Baghera a été conçue à l'aide de la méthodologie d'analyse et de conception « AIEO » (Agents, Interaction, Environnement, Organisation) [Demazeau, 1995]. En plus, cette plateforme a été implémentée avec l'environnement de développement d'applications multi-agents JATLite [Heecheol et al, 2000]. Baghera propose aux utilisateurs un outil de travail collaboratif. Il s'agit d'une zone de dialogue entre les étudiants et les enseignants connectés. Cependant, la notion de groupe est absente dans la plateforme. Le suivi des apprenants proposé est minimal. En effet, la plateforme propose aux enseignants la possibilité de vérifier le temps de connexion, le contenu du cartable de chaque étudiant.

**SIGFAD** [Hikolo, 2003] est un système multi-agents destiné à être couplé aux plateformes informatiques de formation à distance (FAD). Ce système est modélisé à l'aide de la méthodologie MaSE couplée à agentTool [Deloach, 2001] qui est un environnement de développement permettant d'implémenter MaSE et est développé en Java. SIGFAD fournit des appréciations au niveau du groupe (l'état du groupe par rapport à la réalisation des activités, connexion des membres), au niveau d'un individu (sa productivité et sa sociabilité) ainsi qu'au niveau de l'activité (niveau de réalisation d'une activité par tous les participants) afin de permettre au tuteur de maintenir les groupes, les dynamiser et de bien conduire la séance de FAD.

**iPédagogique** [Fougères, 2010] est une plate-forme auteur pour l'enseignement en présentiel et à distance d'unités de valeurs scientifiques et techniques dont la pédagogie est orientée projet. Les auteurs définissent deux principaux objectifs à cette plateforme : structurer et instrumenter les activités coopératives de suivi, de gestion et d'évaluation de projets d'étudiants ; et offrir une assistance aux utilisateurs étudiants et enseignants engagés dans cette coopération. Ces utilisateurs ont quatre possibilités d'utilisation d'iPédagogique qui sont l'enseignement, l'apprentissage, la réalisation et l'interaction.

La plateforme iPédagogique dispose aussi d'un système de médiation qui a pour but de faciliter son utilisation et de le rendre moins complexe pour les utilisateurs. Il est composé d'une interface homme-machine, d'une base de connaissance et d'un système de multi-assistance qui est la composante centrale du système de médiation et est conçu selon une approche multi-agents.

**MASCE** [Mahdi et Attia, 2010] est un système multi-agents pour l'apprentissage collaboratif à distance. Ce système complète l'apprentissage traditionnel en face-à-face. En effet, MASCE permet à l'apprenant de revoir les documents de cours, de demander de l'aide et d'évaluer l'aide fourni afin de permettre au système d'avoir une liste de meilleurs aideurs, d'interagir avec leurs tuteurs ou les autres apprenants à l'aide des outils de travail collaboratif (chat, email). La problématique de ce projet est de pouvoir continuer l'apprentissage entamé en classe et de permettre au tuteur d'interagir avec les apprenants afin de les aider dans leur apprentissage, de les évaluer et de suivre leurs activités.

**I-MINDS** [Khandaker et al, 2011] est un système conçu pour l'apprentissage collaboratif à distance. Il fournit une infrastructure multi-agents pour l'apprentissage collaboratif par ordinateur (Computer-Supported Collaborative Learning CSCL). Dans I-MINDS, des agents intelligents interagissent afin de servir les tuteurs et les apprenants. Ainsi, le tuteur peut interagir avec les apprenants, les évaluer, former des groupes et contrôler les performances de chaque apprenant ainsi que celles du groupe. Quant à l'apprenant, ce système lui permet d'apprendre, d'interagir avec les autres apprenants ainsi qu'avec le tuteur et de former de groupe. I-MINDS offre aussi aux utilisateurs des outils de travail collaboratif (chat, tableau de bord) [Soh et al, 2010].

Ces plateformes ont été conçues selon le modèle client-serveur. Bien que le modèle P2P soit utilisé dans beaucoup des domaines en association avec les systèmes multi-agents [Cabani et al, 2010], [Pommier et Bourdon, 2009], [Mari et al, 2008], [Lopes et Botelho, 2008], etc. son utilisation dans le domaine d'apprentissage reste très peu répandue. En effet, le P2P est utilisé pour échanger des documents, des informations ou des connaissances, le projet LL2 (de l'anglais « Learners Learn 2gether ») [Breu et al, 2005] est un exemple. Son objectif est de permettre aux étudiants de plusieurs universités d'échanger leurs documents.

I-Help [Greer et al. 2001] [Vassileva et al, 2003] et Help&Learn [Guizzardi et al, 2003] sont deux systèmes multi-agents à architecture P2P qui permettent de mettre en relation des apprenants dans le but de s'entre-aider et d'échanger des informations mais n'offrent pas une plateforme d'apprentissage à proprement dit.

**I-Help** Le système I-Help est un système de mise en relation entre pairs pour aider des apprenants engagés dans une activité de résolution de problème. En effet, I-Help propose à l'utilisateur un ensemble de ressources d'aides distribuées qui peuvent être humaines (l'aide d'un pair ou les conseils d'expert) ou bien numériques (des parties de discussions d'un forum, un sujet dans une FAQ ou une ressource web). Le système permet aux étudiants de donner et de recevoir de l'aide de façon synchrone ou asynchrone et utilise des services de

mise en correspondance (matchmaking en anglais) basés sur les modèles des utilisateurs pour mettre en relation les pairs entre eux.

**Help&Learn** est un système orienté-agent dont le but est de supporter la gestion de connaissances (ou knowledge management en anglais) dans des communautés d'apprentissage collaboratif. Dans Help&Learn, un utilisateur peut demander de l'aide (joué le rôle de demandeur d'aide (Helpee)) ou bien fournir l'aide demandé (joué le rôle de fournisseur d'aide (Helper)). Les connaissances sont échangées sous forme des objets d'aide (HelpItems) qui peuvent être des références bibliographique ou références web, des documents électroniques, etc. Help&Learn est construit selon une architecture P2P. Ainsi, la relation entre les apprenants et les enseignants n'est pas hiérarchique. En plus, tous les utilisateurs sont des pairs qui collaborent pour créer, intégrer et diffuser les connaissances. Dans ce système, des assistants personnels sont utilisés pour maintenir le profil de l'utilisateur ainsi que sa base de connaissance tandis que d'autres agents sont utilisés pour trouver le meilleur pair capable de fournir de l'aide ainsi que pour gérer les ressources.

### 4.6.3 Discussion

Nous avons étudié dans cette section quelques travaux dans le domaine d'e-Learning afin de les situer par rapport à nos travaux. iPédagogique et Baghera sont toutes les deux des plateformes dont le but est l'apprentissage mais le suivi des activités des apprenants est minimal. Cependant, iPédagogique dont la pédagogie est orientée projet offre une assistance aux utilisateurs. I-Help et Help&Learn ne proposent pas un suivi des apprenants cependant I-Help permet à l'étudiant demandeur d'aide d'évaluer les agents qui lui ont fourni l'aide. Au contraire, Formid est une plateforme d'apprentissage de géométrie qui permet à l'enseignant de suivre les activités des étudiants en observant leurs traces récoltés suivant un scénario qu'il a lui-même conçu. Dans le même principe, Croisières est un projet qui permet l'apprentissage de la langue française ainsi que le suivi asynchrone d'activité individuelle. MASCE et I-MINDS sont deux systèmes dont le but est d'offrir aux utilisateurs un support pour l'apprentissage collaboratif. Cependant, I-MINDS offre en plus aux utilisateurs la possibilité de former un groupe. Ces deux systèmes permettent d'évaluer les apprenants. MASCE permet à un apprenant d'évaluer l'aide fournit par un autre apprenant et I-MINDS permet au tuteur de suivre les activités des apprenants et des groupes.

Reflet et SiGFAD sont tous les deux des outils destinés à être couplés à des plateformes d'apprentissage. Cependant contrairement à SIGFAD, dans Reflet, il n'y a pas de collecte automatique des traces des étudiants. Ce sont les étudiants qui déclarent l'achèvement de chaque tâche de leur activité et permettent ainsi au tuteur de suivre leur avancement dans la réalisation des activités. Tandis que dans SIGFAD, le système récolte les traces des apprenants et celles des groupes pour fournir à l'enseignant des appréciations sur le groupe, l'apprenant ainsi que l'activité.

Nous constatons qu'aucun des travaux présentés ne s'intéresse au suivi de l'apprentissage dans un environnement P2P. En effet, les travaux s'intéressant au suivi des activités des

apprenants n'offrent qu'une réponse partielle. Cependant, ces outils et plateformes tels qu'ils sont ne suffisent plus pour satisfaire les nouvelles exigences que pose l'utilisation du modèle P2P dans un contexte d'apprentissage collaboratif. En effet, l'utilisation du modèle P2P pose quelques problèmes nécessitant la prise de mesures permettant de limiter les effets indésirables engendrés. La volatilité des pairs est un problème majeur surtout dans un contexte d'apprentissage collaboratif où le risque de perte de détermination et de motivation pour les apprenants est bien réel. Un autre problème qui se pose dans le contexte du P2P est le risque d'avoir des pairs qui puissent avoir des comportements malveillants ainsi que l'impossibilité de reconnaître ces pairs. Cependant, ces inconvénients du P2P ne constituent pas un obstacle à leur utilisation mais nécessite un mécanisme de coordination adapté. D'ailleurs, l'apport des systèmes P2P au domaine d'apprentissage collaboratif fait pencher la balance vers l'utilisation de ces systèmes. En effet, le P2P se caractérise par l'autonomie des pairs ce qui leur permet de gérer un ensemble de technologies et de ressources tout en réalisant leurs buts de façon décentralisée et indépendamment des autres pairs. Cette décentralisation ne les empêche pas par ailleurs de coopérer en fournissant des ressources ou des services aux autres membres dans le but de pouvoir joindre et utiliser le système. En plus, les P2P sont des systèmes flexibles et tolérants aux pannes. Ces caractéristiques (autonomie, décentralisation, coopération et tolérance aux pannes) sont très utiles aux systèmes multi-agents ce qui rend leur utilisation dans notre travail bénéfique.

La figure IV-1 montre l'apport du P2P aux domaines d'apprentissage et des SMA ainsi que les inconvénients qui peuvent être engendrés.

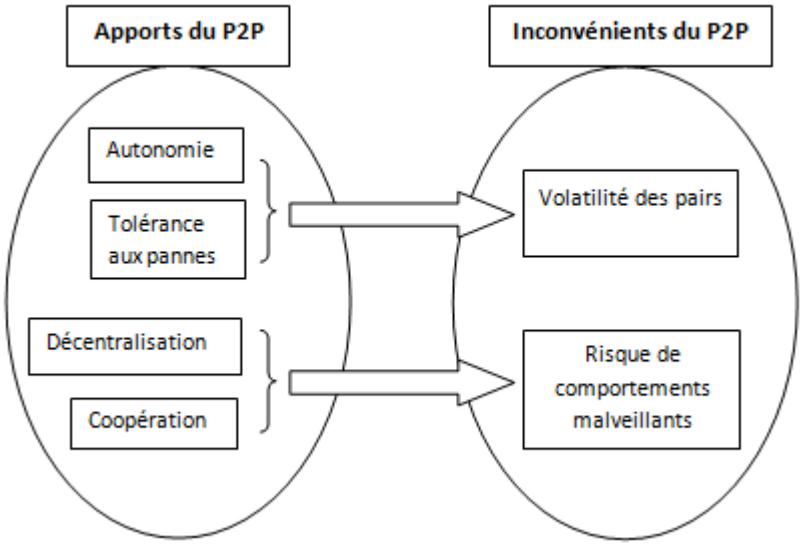


**Figure IV-1 Apports et inconvénients du P2P**

## 4.7 Conclusion

L'utilisation d'Internet à des fins d'enseignement sous des multiples formes s'est développée dans tous les secteurs de l'éducation et particulièrement dans le monde universitaire puisqu'elle offre des avantages dont celui notamment de faciliter l'enseignement à distance ainsi que l'e-Learning.

L'e-Learning implique un changement de rôle du formateur car la connaissance est disponible en dehors de lui : il devient un médiateur. L'e-Learning implique aussi l'autonomie de l'apprenant qui est une conséquence de son isolement et de l'absence du contact direct avec l'enseignant ou les autres apprenants. Il est donc supposé gérer efficacement les ressources offertes par l'outil informatique ainsi que ses interactions avec les autres apprenants mais surtout gérer son temps et le rythme de progression et d'acquisition des savoirs et des savoir-faire. Cependant, cette autonomie ne peut être considérée comme un pré-requis [chasseneuil, 2000] et ne dépend pas uniquement des apprenants car il faut d'abord créer le cadre où peut s'exercer cette autonomie [Oubahssi, 2005]. Par ailleurs, l'autonomie de l'apprenant est un point essentiel pour qu'il puisse s'engager dans un apprentissage collaboratif. En plus, dans le cadre d'une application P2P, la volatilité des pairs complique davantage l'apprentissage qui est voué à l'échec sans l'accompagnement de l'apprenant dans son apprentissage mais surtout sans sa détermination et sa motivation. En effet, il est essentiel de faciliter la tâche de l'enseignant en dotant la plateforme des outils capables de récolter et traiter les informations afin de renseigner les enseignants à tout moment sur l'état et les activités de chaque apprenant et du groupe ainsi que sur la progression de leur travail pour pouvoir les motiver et les guider et ainsi éviter l'échec ou l'abandon de ces apprenants. Cependant, dans un contexte d'apprentissage collaboratif où les apprenants et les enseignants sont considérés comme étant à la fois producteur et fournisseur de l'information, il est aussi essentiel d'offrir à l'apprenant les mêmes outils de perception de la situation d'apprentissage que le tuteur. Ceci afin que chaque apprenant impliqué dans le processus d'apprentissage prenne conscience de ses responsabilités envers lui-même et son groupe et contrôle ainsi ses actions et son apprentissage.

| | **Outils Collaboratifs et Notion de groupe** | **Méthode de formation de groupe** | **Paradigme objet ou agent** | **Architecture** | **Suivi de l'utilisateur** |
|---|---|---|---|---|---|
| **Sakai** | Oui | Non | Objet | Client/serveur | Suivi minimal |
| **Claroline** | Oui | Non | Objet | Client/serveur | Récolte les traces et les structure |
| **Moodle** | Oui | Formation de groupe de façon aléatoire | Objet | Client/serveur | Récolte les traces des apprenants |

| | | | | | |
|---|---|---|---|---|---|
| **Reflet** | Oui | Non | Objet | Client/serveur | Pas de collecte automatique de traces : les étudiants déclarent l'achèvement des activités et l'enseignant suit ainsi l'avancement de réalisation des tâches |
| **Croisières** | Oui | Non | Objet | Client/serveur | Oui |
| **Formid** | Oui | Non | Objet | Client/serveur | Environnement Formid-Suivi qui permet de suivre les activités des apprenants selon un scénario proposé par l'enseignant |
| **Baghera** | Oui | Non | Agent | Client/serveur | Suivi minimal |
| **SIGFAD** | Oui | Non | Agent | Client/serveur | Récolte les traces des apprenants et fournit à l'enseignant des appréciations sur le groupe, l'apprenant ainsi que les activités |
| **iPédagogique** | Oui | Non | Agent | Client/serveur | Suivi minimal + assistance aux utilisateurs |
| **MASCE** | Oui | Non | Agent | Client/serveur | Pas de suivi mais propose une liste des meilleurs aideurs |
| **I-MINDS** | Oui | Oui | Agent | Client/serveur | Oui |
| **I-Help** | Oui | Non | Agent | P2P | Non mais permet d'évaluer l'aide fourni par des apprenants |
| **Help&Learn** | Oui | Non | Agent | P2P | Non |

**Tableau IV-1Comparatif de l'existant**

Nous avons présenté dans le tableau IV-1, un comparatif des travaux existants. Ce tableau montre que l'architecture utilisée pour construire un système d'apprentissage collaboratif est le modèle client/serveur. Le modèle P2P est utilisé pour mettre en relation les apprenants et leur permettre d'échanger des informations.

Dans le chapitre suivant, nous proposons un système multi-agents à architecture P2P dédié à l'apprentissage collaboratif dans lequel plusieurs groupes d'apprenants interagissent et collaborent afin de réaliser un projet donné. Ce système offre aux enseignants des fonctionnalités pour le suivi des apprenants et aux apprenants, des fonctionnalités leur permettant d'apprécier leur travail et de suivre leur progression dans le processus

d’apprentissage. Nous définissons également une méthode de formation de groupe afin de permettre la coordination d’actions des agents.

# Chapitre V. COLYPAN : un système d'apprentissage collaboratif de la gestion de projet

## 5.1 Avant-propos

Les systèmes multi-agents sont mis à contribution pour la modélisation d'interactions dans des sociétés complexes d'individus humains ou artificiels. Le domaine applicatif qui illustre parfaitement cette problématique scientifique est très vaste. Nous distinguons, à titre d'exemple, les applications de travail collaboratif, sujet du présent chapitre.

## 5.2 Introduction

Nous proposons dans ce chapitre de concevoir un système multi-agents à architecture P2P pour l'apprentissage collaboratif à distance. En effet, les systèmes multi-agents s'adaptent bien à la conception d'un environnement d'apprentissage où chaque membre doit gérer et échanger ses connaissances et collaborer avec les autres afin de réaliser ses buts. En plus, dans des tels environnements ouverts, dynamiques et complexes, il y a un besoin de distribuer les données, le contrôle ainsi que l'expertise ce qui rend l'utilisation des systèmes P2P bénéfique.

Les Technologies de l'Information et de la Communication pour l'Enseignement (TICE) ont permis aux utilisateurs dispersés dans le temps et dans l'espace de collaborer et ainsi de maximiser la créativité et l'efficience d'un groupe. Cependant, l'efficacité et l'efficience d'un apprentissage collaboratif dépendent de la motivation de ses membres à collaborer, du nombre de ces membres, du temps qu'ils peuvent consacrer à ce travail et de leurs compétences. Ainsi, la responsabilité devient globale et collective. En effet, dans la démarche collaborative, les apprenants collaborent aux apprentissages du groupe, et en retour, le groupe collabore à ceux des apprenants et c'est la cohérence du collectif qui permet d'atteindre l'objectif. Ainsi, nous allons tout d'abord explorer le domaine d'apprentissage collaboratif afin de déceler les nouveaux rôles de l'enseignant, des apprenants et du groupe et les besoins qui en découlent pour les analyser et ainsi proposer des fonctionnalités capables de le satisfaire. Cependant, avant d'analyser ses besoins, nous nous intéressons aux problèmes de coordination d'agents sur une architecture P2P. En effet, nous présentons la méthode de formation de groupe que nous proposons pour résoudre ces problèmes de coordination. D'ailleurs, compte tenu de l'importance du groupe dans l'apprentissage collaboratif, il s'avère donc que notre proposition est aussi profitable pour l'application d'apprentissage choisie qui est la gestion des projets. En effet, le groupe crée des liens sociaux entre les apprenants les empêchant ainsi de se sentir isolés.

Nous proposons ensuite la modélisation, à l'aide de la méthodologie Aalaadin, du système COLYPAN (Collaborative Learning sYstem for Project mANagment) que nous avons conçu pour l'apprentissage à distance et de façon collaborative de la gestion de projet. Ce système

implémente la méthode MAETIC (Méthode pédAgogique InstrumentEe par les TIC) réalisés au sein de notre laboratoire et visant à mettre en place une pédagogie de groupe par projet. MAETIC décrit un ensemble de démarches formalisées et appliquées selon des principes définis. L'objectif de MAETIC est ainsi de permettre à un étudiant de développer les savoirs et savoir-faire sollicités par la mise en œuvre d'un processus de développement d'un « produit » et de l'entraîner aux techniques de gestion des projets. Pour l'enseignant, l'objectif de MAETIC est de favoriser la mise en place d'une démarche qui facilitera ses actions pédagogiques.

Ce système sera doté des fonctionnalités nécessaires pour satisfaire les besoins des utilisateurs et les accompagner dans leur apprentissage. En effet, d'après l'analyse de ces besoins, nous avons constaté qu'il est essentiel de faciliter la tâche de l'enseignant en lui renseignant à tout moment sur l'état et les activités de chaque apprenant et du groupe ainsi que sur la progression de leur travail pour pouvoir les motiver et les guider et ainsi éviter l'échec ou l'abandon de ces apprenants. D'ailleurs, dans un contexte d'apprentissage collaboratif où les apprenants et les enseignants sont considérés comme étant à la fois producteur et fournisseur de l'information, il est aussi essentiel d'offrir à l'apprenant la possibilité de percevoir sa situation d'apprentissage ainsi que celle des apprenants de son groupe afin qu'il prenne conscience de ses responsabilités envers lui-même et son groupe et contrôle ainsi ses actions et son apprentissage. De plus, il est aussi essentiel de fournir aux enseignants et apprenants la possibilité d'avoir des informations sur les niveaux de collaboration et de sociabilité de chaque apprenant et du groupe.

## 5.3 Rôles et besoins du tuteur

L'apprentissage collaboratif à distance (synchrone ou asynchrone) exige de l'enseignant de jouer des nouveaux rôles qu'il ne jouait pas dans l'apprentissage classique en classe. Il est alors essentiel de dégager tous ces rôles afin de pouvoir déceler les besoins qui en découlent et trouver les solutions que peuvent apporter la technologie informatique.

L'utilisation des Technologies de l'Information et de la Communication pour l'Apprentissage (TICE) impose à l'enseignant de revoir ses méthodes d'enseignement. En effet, l'enseignant a la charge de préparer tous les supports pédagogiques ainsi que les contenus nécessaires pour la formation. En plus, il doit concevoir des scénarios pour l'utilisation des contenus par les apprenants. Ce dernier aspect est un point essentiel pour la réussite de l'apprentissage d'une part parce qu'un contenu mal conçu rend l'apprentissage difficilement réalisable et d'autre part parce que dans le contexte d'apprentissage à distance qui contrairement aux situations d'apprentissage classique où l'enseignant gère le déroulement de l'apprentissage, exige de l'étudiant d'apprendre en suivant le scénario conçu par l'enseignant qui jouera dorénavant le rôle du tuteur qui l'accompagne dans son apprentissage. Nous présentons dans le *tableau V-1*, les rôles de l'enseignant dans l'apprentissage classique et l'équivalent dans l'apprentissage à distance.

Cependant, il faut souligner que ces nouveaux rôles ne sont pas obligatoirement joués par le même personnage. En effet, les rôles 1 et 2 peuvent être joués respectivement par le

responsable de la formation et l'administrateur de l'outil informatique ou la plateforme. Par contre, ceci n'empêche pas le fait que l'enseignant doit être familier avec les TIC et les logiciels mais surtout qu'il doit avoir les connaissances nécessaires pour exploiter au mieux les supports conçus ainsi que les capacités pour les adapter en fonction des besoins des apprenants.

| | l'enseignant dans l'apprentissage classique | L'enseignant dans l'apprentissage à distance |
|---|---|---|
| Rôle 1 | • Prépare cours<br>• Réalise les fiches de cours, TD et TP | • Prépare contenu pédagogique<br>• Réalise les scénarios pour l'apprentissage |
| Rôle 2 | • Photocopie les supports des cours<br>• les distribue aux étudiants | • Intègre le contenu à la plateforme |
| Rôle 3 | • Effectue des cours magistraux<br>• Dirige les TD et les TP | • Joue le rôle de tuteur |

**Tableau V-1 Rôles de l'enseignant : classique vs à distance**

Nous nous intéressons particulièrement au rôle 3 celui de tuteur puisque c'est lui qui sera en interaction avec les apprenants et les accompagnera dans leur apprentissage. Dans ce qui suit, nous essayons de définir les rôles que peut jouer le tuteur. Cependant, avant d'entrer dans les détails de ce rôle, il est important d'évoquer un fait dont on parle rarement c'est le besoin de l'enseignant d'apprendre à jouer ce rôle. En effet, on évoque souvent dans le contexte d'apprentissage à distance des problèmes d'isolement et d'abandon des apprenants. Cependant, on prête peu d'attention aux problèmes que peut rencontrer le tuteur dans ce type de formation. En effet, le tuteur ne peut pas appliquer dans l'apprentissage à distance, les méthodes d'apprentissage classiques. Ce type d'apprentissage nécessite des nouvelles méthodes et des nouvelles compétences que la majorité des tuteurs ne l'ont pas [Bennett et March, 2002]. Nous nous n'intéressons pas aux formations, si elles existent, que doit suivre le tuteur pour l'aider à acquérir ces compétences. Notre intérêt se porte sur l'apprentissage du tuteur de son métier de façon collaborative, tout comme les apprenants, en interagissant avec les autres tuteurs. Ainsi, cette interaction permet aux « tuteurs apprenants » d'échanger leurs expériences et leurs pratiques, de discuter des positions à adopter face à des situations précises. Les tuteurs peuvent aussi demander de l'aide aux autres tuteurs, c'est un point délicat parce que souvent les tuteurs n'osent pas évoquer leurs problèmes. Ainsi, encourager le tuteur à collaborer et à interagir avec les autres tuteurs permettent de lui donner plus de confiance, de le motiver. L'enseignant ne se sent pas alors isolé et lâché seul face à une tâche énorme qu'est le tutorat. D'ailleurs, un tuteur qui se sent isolé ou manque de motivation ne peut pas motiver ses apprenants. Comme le dit bien le proverbe : « on ne peut donner ce que l'on a pas ». En plus, l'apprentissage collaboratif est un mode d'apprentissage qui s'apprend au fur et à mesure des interactions. Il est alors important de s'engager dans un travail ou apprentissage

collaboratif avant de s'engager dans une responsabilité de tutorat d'apprentissage collaboratif.

On en déduit alors qu'il faut proposer au tuteur des outils qui lui permettent d'interagir avec d'autres tuteurs et de lui fournir les moyens pour qu'il puisse s'engager dans un travail / apprentissage collaboratif. Il faut aussi lui faciliter la tâche, en la simplifiant et en proposant des fonctionnalités répondant à ses besoins et capables d'alléger sa charge. Il faut donc définir tous les rôles possibles du tuteur (tableau V-2) dans le contexte d'un apprentissage collaboratif à distance.

Le premier rôle du tuteur est pédagogique. En effet, le tuteur doit guider les apprenants et les orienter vers les bonnes solutions. Ce premier rôle implique un deuxième qui est d'adapter le scénario d'apprentissage. En effet, le tuteur a besoin des informations sur la réalisation des activités afin de juger s'il y a besoin d'adapter le scénario proposé.

Son troisième rôle est d'encourager le travail collaboratif. Il doit inciter les étudiants à utiliser les outils collaboratifs pour interagir, discuter et échanger leurs points de vue. Pour cela, il a besoin des informations sur l'utilisation de ces outils.

Enfin, le tuteur a aussi pour rôle de motiver les apprenants en difficulté, de les encourager à participer à l'apprentissage. Cependant, pour pouvoir jouer ce rôle, il a besoin des informations sur l'état de chaque apprenant et du groupe (actif, passif, etc.). Ce rôle est essentiel pour la réussite de l'apprentissage. En effet, il est important de prédire les difficultés des apprenants pour éviter qu'ils se démotivent ou bien abandonnent la formation.

| | Rôles | Besoins |
|---|---|---|
| Tuteur | (R1)<br>• Guider les apprenants<br>• Adapter le scénario d'apprentissage | Infos sur les activités des apprenants de chaque groupe |
| | (R2)<br>• Encourager le travail collaboratif | Infos sur l'utilisation des outils collaboratifs |
| | (R3)<br>• Motiver les apprenants en difficulté | Infos sur l'état de chaque apprenant et chaque groupe |

**Tableau V-2 Rôles et besoins du tuteur**

## 5.4 Rôles et besoins de l'apprenant et du groupe

Les apprenants utilisent les TICE pour apprendre. S'agissant d'un apprentissage collaboratif et conformément à la théorie constructiviste, il se fait donc par les interactions des apprenants entre eux, avec le tuteur et l'environnement. Ainsi, l'outil informatique doit offrir aux apprenants (*figure V-1*), en plus du contenu pédagogique qui doit être clair, bien structuré et attractif, des outils permettant le travail collaboratif c'est-à-dire des outils de discussion et d'envoi de messages synchrone et asynchrone (Chat, Forum, ..). L'utilisation de ces outils permet à l'apprenant de ne pas se sentir seul et isolé et ceci en lui permettant d'interagir avec les tuteurs et les autres apprenants. En effet, les apprenants en interagissant se motivent les uns les autres. Ainsi, l'apprentissage devient plus convivial et animé et donc a plus de chance de réussir. La motivation de l'apprenant est donc primordiale pour réussir son apprentissage. Nous avons vu le rôle du tuteur dans la motivation et le suivi de l'apprenant. Cependant, un apprenant engagé dans un apprentissage collaboratif est suffisamment mature et responsable pour vouloir réussir cet apprentissage. Ainsi, un des moyens de le booster est de lui permettre de visualiser le suivi de ses activités ainsi que ceux des autres apprenants. En effet, le fait de pouvoir comparer ses activités avec ceux des autres est censé motiver l'apprenant à améliorer ses performances et à prendre conscience de ses points de faiblesse. L'apprenant a aussi besoin d'avoir une idée sur le comportement des autres apprenants au cas où il a besoin de solliciter quelqu'un pour l'aider ou bien pour former un nouveau groupe. Le groupe étant un élément essentiel dans l'apprentissage, il crée un lien social entre les apprenants ce qui s'avère très utile pour l'apprentissage à distance. En effet, les interactions qui ont lieu au sein d'un groupe permettent à chaque apprenant de ne pas se sentir isolé mais au contraire il se sent motivé pour apprendre. Ainsi, le taux d'abandon particulièrement élevé dans les formations à distance [Linard, 2000], est baissé. D'ailleurs, dans le cas d'un apprentissage collaboratif où chaque membre doit être actif pour réaliser les objectifs du groupe, le suivi de l'activité de chaque membre par les activités des autres apprenants du même groupe s'avère obligatoire afin de garder la synergie du groupe.

Nous décrivons dans la section suivante, la méthode de formation de groupe que nous proposons.

## 5.5 La formation de groupe

Nous avons décrit, dans le chapitre 3, les problèmes de coordination des systèmes multi-agents à architecture P2P et nous avons proposé comme solution pour résoudre ces problèmes de former un groupe. En effet, nous avons justifié ce choix en citant une étude faite par [Tremewan, 2010] qui confirme l'intérêt de faire partie d'un groupe. D'ailleurs, nous avons précédemment évoqué le rôle du groupe dans la réussite d'un apprentissage collaboratif et dans l'encouragement des apprenants. Cependant, cet effet positif du groupe ne peut avoir lieu que si les membres de ce groupe sont actifs dans leur apprentissage et partagent un but en commun. En effet, un groupe passif et désintéressé (ses membres sont passifs) ne peut encourager un membre à apprendre et à accomplir ses tâches. De plus, un

groupe dont ses membres ont chacun des compétences qui ne sont pas utiles aux autres ou des besoins différents ne peut résoudre efficacement ses objectifs. Pour cela, nous pensons qu'il est important de définir une méthode pour la formation de groupe qui prend en compte les besoins ainsi que les compétences de chaque apprenant. Nous détaillerons dans la suite la méthode proposée.

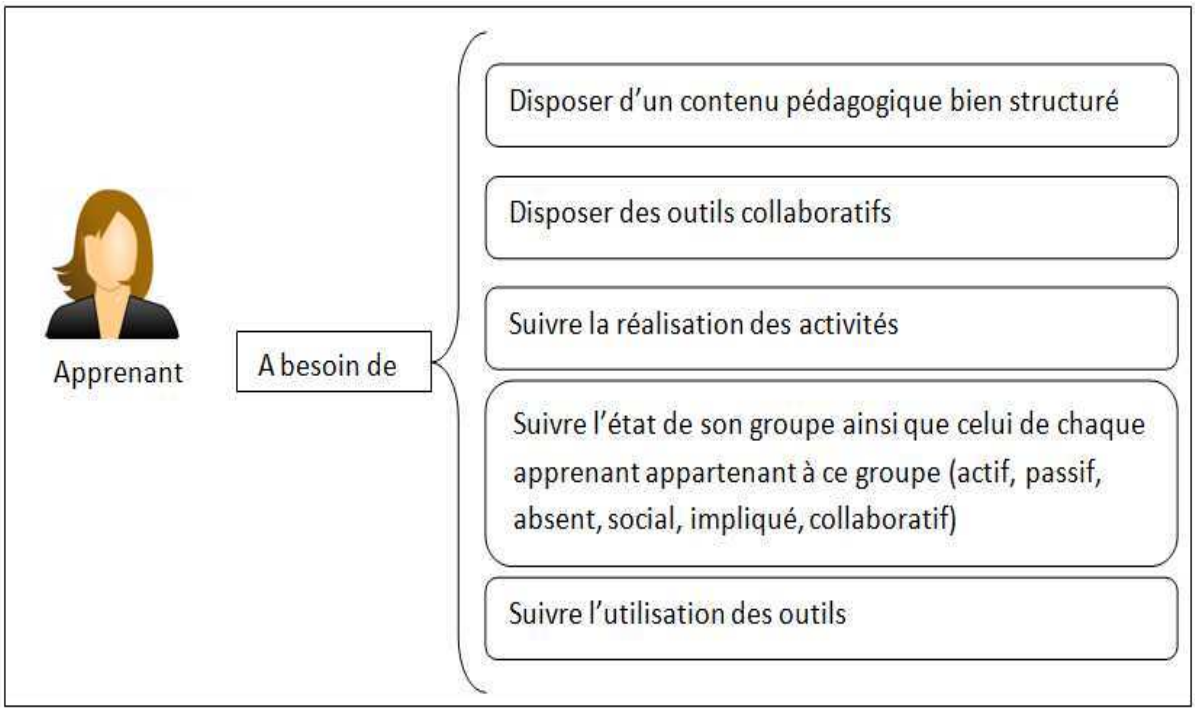


*Figure V-1 Besoins de l'apprenant*

### 5.5.1 Méthode adoptée pour la formation d'un groupe

Au début, une personne qui désire former un groupe, lance un appel pour attirer d'autres membres en précisant le thème du projet auquel les éventuels membres doivent travailler ainsi que les connaissances requises. Cette personne devient l'initiateur de groupe. Il précise le poids de chaque compétence en fonction de l'intérêt qu'apporte le fait de l'avoir au groupe afin de réaliser les buts de ce dernier.

Chaque pair voulant se joindre à un groupe donné commence par faire une demande d'admission à ce groupe. Pour cela, il doit préciser ses compétences.

Ensuite, l'agent chargé du groupe détient le droit d'accepter/refuser les candidats et ceci en comparant leurs compétences avec celles requises. En effet, chaque compétence a un poids, l'agent additionne le poids de compétences du candidat et le divise par le poids total de toutes les compétences. Si le candidat a la moyenne, sa demande est acceptée sinon, elle est refusée. De cette façon, un groupe ayant un même but général se forme. Cependant, la méthode de formation de groupe proposée étant ouverte, les membres du groupe ne tarderont pas à devenir trop nombreux. En plus, malgré que ces membres aient le même but en global, leurs propres buts peuvent diverger. Par exemple, tous les agents du groupe ont pour but l'échange des fichiers. Cependant, le type de fichiers à échanger diffère d'un agent

à un autre. Un agent X a besoin d'échanger des fichiers audio, d'autres de fichiers vidéo, etc. Pour cela, notre idée est que le groupe fonctionne de la façon décrite ci-dessus jusqu'à l'atteinte d'une taille minimale qui lui permette de fonctionner correctement. Une fois le groupe atteint la taille maximale, il se divise.

### 5.5.2 La division récursive en sous groupes selon les compétences

Chaque fois que les nombres de membres du groupe atteignent son maximum (le maximum est à définir par l'initiateur du groupe), les agents sont informés automatiquement que le groupe va se diviser pour former des nouveaux groupes et ceci de la façon suivante :

**Etape 1** : Chaque agent choisit sa liste de préférence (LP) en donnant des notes aux autres agents selon les ressources dont il a besoin, les compétences qu'ils peuvent offrir ainsi que selon leur niveau de collaboration et de sociabilité. En effet, chaque agent cherche à se mettre avec les meilleurs. En plus, chaque agent a une liste de blocage (LB) des agents avec lesquels il ne veut pas travailler.

**Etape 2** : Une comparaison est alors faite entre les listes de préférence des agents. En effet, il faut calculer la meilleure combinaison des groupes en tenant compte de liste des préférences de chaque agent ainsi que sa liste de blocage. Sachant que chaque agent peut appartenir à plusieurs groupes.

| Agents | Liste de préférences | Liste de blocage |
|---|---|---|
| a1 | {a10, a2, a3, a6, a7} | {a9} |
| a2 | {a3,  a6, a7} | {} |
| a3 | {a10, a4, a5, a8} | {} |
| a4 | {a1, a8, a9} | {a3} |
| a5 | {a5, a6, a1, a8, a3} | {} |
| a6 | {a1, a4, a7, a8, a10} | {a3, a9} |
| a7 | {a10, a8, a5, a6} | {a2, a3, a9, a1} |
| a8 | {a10, a4, a6, a3} | {a2} |
| a9 | {a1, a10, a8, a4} | {} |
| a10 | {a2, a5, a8} | {a9} |

**Tableau V-3 Exemple de LB et LP des agents**

Prenons pour exemple les agents {a1, a2, a3, a4, a5, a6, a7, a8, a9, a10} qui forment un groupe (*tableau V-3*). Ce groupe a atteint sa taille limite et il faut qu'il se divise. Dans ce cas, on regarde les listes (LP et LB) de chaque agent. On commence par éliminer les agents de la liste LB de chaque agent. Nous obtenons ainsi un premier groupe (G1) constitué de : {a4, a5, a6, a7, a8, a10}. Les éliminés sont donc : a1, a2, a3 et a9.

Nous plaçons ensuite ces agents éliminés avec les agents appartenant à leur LP ainsi que ceux qui l'ont mis dans leur liste LP. Nous obtenons alors :

(G2) : {a1, a10, a2, a3, a6, a7, a4, a5, a6, a9} cependant le a9 se trouve dans la liste LB de a1, il est alors viré. En plus, l'a7 a mis les a1, a2, a3 et a9 dans sa liste LB. Cependant, il s'agit de placer l'a1, c'est donc l'a7 qui sera éliminé. Par contre, l'a3 se trouve dans la liste LB d'a4 et a6, la décision sera donc prise selon la liste LP d'a1. Ainsi, dans la liste LB d'a1, il y a l'a3 et l'a6 et si on élimine l'a6, il faut alors éliminer l'a4 donc on élimine l'a3. Ainsi, nous obtenons le (G2) : {a1, a2, a4, a5, a6, a10}.

(G3) : {a2, a3, a6, a7, a1, a10}. De même, il faut éliminer l'a7 qui ne veut pas de l'a2. Autre incompatibilité entre l'a3 et l'a6. Tous les deux se trouvant dans la liste LP d'a2. Donc, il faut vérifier combien d'agents de G3 figurent dans les listes LP d'a3 et a6. Pour l'a3, il y a un seul a10 cependant pour l'a6, il y en a deux a1 et a10 donc c'est l'a3 qui sera éliminé. Ainsi, nous obtenons le (G3) : {a2, a1, a6, a10}.

Même chose pour l'a3, nous obtenons donc : (G4) : {a3, a10, a5, a8, a1, a5}.

Pour l'a9, nous obtenons (G5) : {a9, a4, a8}.

Ainsi, le groupe initial a été divisé en 5 groupes. Chaque groupe est composé des agents qui ont des intérêts en commun. Ces agents vont collaborer ensemble afin d'atteindre leur but.

Travailler dans un groupe s'avère donc très utile et fondamental dans l'apprentissage collaboratif. Dans cette vision de l'apprentissage où le groupe prend tout cet intérêt, une nouvelle technique est apparue appelée la pédagogie par projet. Des recherches ont indiqué que la pédagogie par projet, dans la formation à distance, permet aux apprenants de développer des habiletés cognitives de haut niveau comme l'analyse et la synthèse [Paquette et al, 1997], de même que l'initiative, l'inventivité et l'autonomie [Fung, 1996]. Dans ce qui suit, nous définissons la pédagogie par projet afin de présenter une méthode pédagogique appelée MAETIC [Talon et Leclet, 2008], développée par des chercheurs de notre laboratoire et basée sur la pédagogie par projet.

## 5.6 Pédagogie par projet

La pédagogie par projet trouve son origine au début du vingtième siècle dans le courant de l'éducation nouvelle [Dewey, 1922]. C'est une technique pédagogique fondée sur l'approche constructiviste selon lequel le processus d'apprentissage conduit à « une modification de la capacité à réaliser une tâche sous l'effet d'une interaction avec l'environnement » [Houdé et al, 1998]. La notion d'environnement désigne aussi bien les ressources matérielles que les ressources humaines.

La pédagogie par projet permet à l'apprenant de s'engager activement dans ses activités, d'investiguer des nouveaux problèmes, d'interagir avec son environnement. Elle demande aux apprenants de résoudre par eux-mêmes des problèmes mal structurés rencontrés dans le monde réel [Finkle et Torp, 1994]. Le but est de développer simultanément chez les apprenants des stratégies de résolution des problèmes ainsi que des connaissances du domaine et des habiletés. En effet, l'apprenant accomplit le projet, créé à partir des

situations réelles, de façon autonome et le formateur joue donc le rôle d'un facilitateur ou d'un consultant [Paquette et al, 1997]. C'est dans ce but que la méthode pédagogique MAETIC, présentée ci-dessous, a été conçue.

### 5.6.1 MAETIC

MAETIC (Méthode pédAgogique InstrumentEe par les TIC) est une méthode pédagogique qui, dans le cadre d'une pédagogie de groupe par projet, décrit un ensemble de démarches formalisées et appliquées selon des principes définis. L'objectif de MAETIC est ainsi de permettre à un étudiant de développer les savoirs et savoir-faire sollicités par la mise en œuvre d'un processus de développement d'un « produit » et de l'entraîner aux techniques de gestion de projet. Pour l'enseignant, l'objectif de MAETIC est de favoriser la mise en place d'une démarche qui facilitera ses actions pédagogiques

MAETIC préconise ainsi, pour les étudiants, la réalisation de cinq étapes communément adoptées dans les démarches de gestion de projet [Marchat, 2001] : l'*initialisation*, la *préparation*, la *planification*, le *suivi de projet* et la *recette* (*figure V-2*). Chaque étape met ainsi en place des activités, exige la production d'un ou des plusieurs livrables et se déroule sur une ou plusieurs séances. Le travail étant collectif, MAETIC préconise la mise en place d'une organisation dans le groupe projet (désignation de rôles) qui favorise ainsi l'acquisition ou le renforcement des compétences transversales nécessaires au travail en équipe. Le fait de faire produire au groupe des livrables développe également les qualités liées à la production écrite.

Ainsi, chaque groupe projet doit mettre en place son journal de bord. Ce journal a pour vocation de décrire la vie du projet. En dehors des informations générales sur le projet (sujet, membres), celui-ci est chargé de stocker toutes les notes sur la vie du projet (notamment des bilans de séance) et se charge également de collecter les livrables développés. La vocation du journal de bord « Étudiant » est double :

- Il a pour objectif de tenir l'enseignant informé de la progression du projet et de détecter ainsi des groupes ou étudiants « en sommeil » ou « peu actifs », et de les relancer le cas échéant. Il permet à l'enseignant de laisser des commentaires sur la qualité du travail réalisé. Il lui permet également de laisser des encouragements qui sont un facteur de motivation pour les étudiants. Il faut cependant que l'enseignant veille à consulter et à commenter systématiquement les journaux de bord des étudiants.
- Le journal est un outil qui a pour objectif d'obliger les étudiants à maintenir une activité régulière : les notes de bilan de séance imposent un point sur le travail effectué depuis la séance précédente et permettent de mesurer le retard pris par rapport à l'échéancier fourni par l'enseignant. L'obligation de déposer ses productions dans le journal les sensibilise ainsi à l'action.

MAETIC impose, également, la tenue d'un journal de bord « Enseignant ». Ce journal permet l'accès aux ressources (supports de cours, modèles de documents, etc.) mises à disposition des étudiants. Il a pour objectif de renseigner les étudiants sur la vie de l'unité d'enseignement (UE). L'enseignant doit ainsi y inscrire, selon son échéancier, des notes qui apportent des commentaires sur le déroulement de l'UE et sur les étapes du projet que les étudiants doivent réaliser, selon l'avancement (rappel de l'échéancier). L'enseignant est chargé de récupérer les fiches personnelles au début du projet. Il doit également consulter le bilan de séance sur les journaux de bord des étudiants, entre chaque séance, et rédiger des commentaires sur le travail réalisé. Le pilotage soutenu de l'enseignant vise à favoriser le respect de la démarche suggérée.

La méthode MAETIC a été conçu dans le cadre d'un apprentissage principalement en présentiel. Dans la section suivante, nous décrivons un système d'apprentissage collaboratif qui implémente cette méthode mais dans un cadre d'apprentissage à distance et à grande échelle.

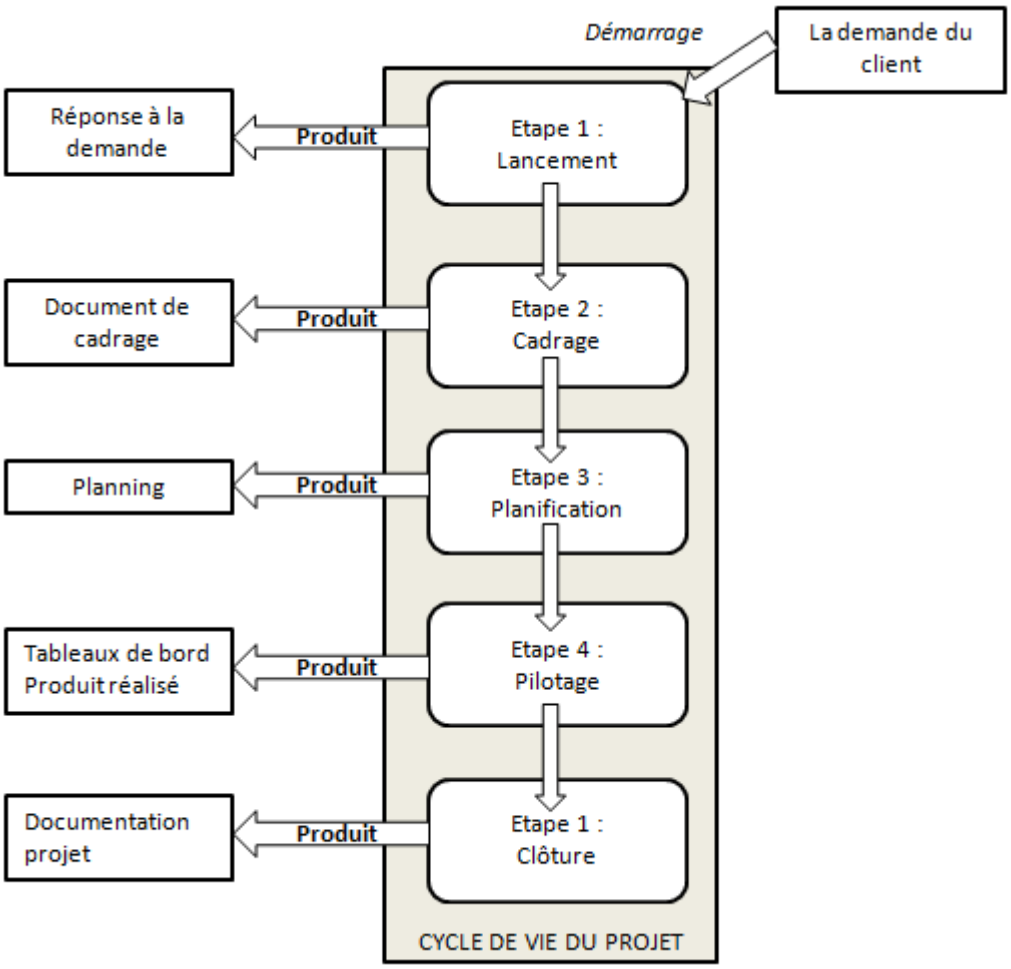


*Figure V-2 Cycle pédagogique de MAETIC*

## 5.7 Le système d'apprentissage collaboratif COLYPAN

COLYPAN (Collaborative Learning sYstem for Project mANagment) [Mazyad et al, 2010b] est un système conçu pour l'apprentissage à distance et de façon collaborative de la gestion de projet. Ce système implémente la méthode MAETIC et met à la disposition des utilisateurs, apprenants et tuteurs, les outils nécessaires pour l'apprentissage collaboratif : un forum, un chat, un tableau de bord, une messagerie instantanée ainsi que des outils de traitement de texte. D'ailleurs, il permet aussi aux utilisateurs de profiter de nombreuses sources pour acquérir des connaissances, échanger de ressources (livres, articles, etc.), rechercher sur l'Internet où apprenants et tuteurs peuvent consulter/télécharger des documents, trouver des références et des liens Web, échanger de connaissance en communiquant avec les différents utilisateurs.

## 5.8 Description du système

COLYPAN est un système destiné à être utilisé principalement par des tuteurs et des apprenants travaillant en groupe et collaborant pour réaliser leur projet. Ainsi, En l'utilisant, chaque apprenant peut apprendre en collaborant et en interagissant avec son groupe ainsi qu'avec le tuteur mais aussi en suivant la méthode MAETIC dans la réalisation de son projet. Il peut en plus utiliser le système pour envoyer/ recevoir des messages, demander de l'aide au tuteur ou aux membres de son groupe, échanger des informations/ documents avec les membres de son groupe ou le tuteur, etc.

Le système fournit aussi à l'apprenant des outils nécessaires pour le traitement de texte, ainsi que des outils de communication tels que le chat ou le forum qu'il peut utiliser pour discuter avec le tuteur ou les autres apprenants de son groupe. L'apprenant a aussi la possibilité d'avoir des informations sur les activités de chaque apprenant de son groupe ainsi que sur ses performances et celles de son groupe.

Le tuteur, de son côté, utilise le système pour interagir avec les apprenants et les groupes dont il a la charge, suivre leur progression dans la réalisation des tâches, les guider et les motiver lorsqu'il le juge nécessaire, leur venir en aide s'ils le sollicitent. Il a aussi la possibilité de connaître leur utilisation des outils mais surtout leur niveau de collaboration et de sociabilité. En plus, il peut utiliser les outils de communication (Forum, chat, messagerie instantanée) ou les outils de traitement de texte.

L'apprentissage dans COLYPAN, se fait de manière collaborative et par groupe d'apprenants. Chaque utilisateur peut être membre de plusieurs groupes mais ne peut communiquer qu'avec les groupes dont il est membre. La formation de groupe se fait en suivant la méthode proposée dans la section 5.5. Cette méthode permet aux utilisateurs de profiter des avantages du travail en groupe ainsi que de coordonner leurs actions afin d'achever leur but final et ceci en favorisant les bonnes actions et en évitant les comportements qui nuisent au bon fonctionnement du groupe ainsi que le comportement du cavalier seul.

## 5.9 Objectifs du système

COLYPAN a plusieurs buts. Le premier est de mettre à la disposition des apprenants et des tuteurs un environnement pour l'apprentissage par projet dotés des outils permettant l'apprentissage collaboratif des utilisateurs et leurs interactions ;

Tandis que le deuxième est de les accompagner dans leur apprentissage et dans la réalisation de leur projet en leur offrant des fonctionnalités pour le suivi de leurs activités ainsi que celles du groupe. En effet, le système permet d'analyser les éléments d'information récoltés sur l'apprenant ou sur un groupe d'apprenants afin de fournir aux tuteurs une information précise et adéquate pour leurs besoins propres sur l'évolution individuelle et collective des apprentissages. Ainsi, COLYPAN leur permette de Connaître le niveau de productivité de chaque apprenant en termes de réalisation des activités pédagogiques, ainsi que son niveau de communication avec les autres membres du groupe ; Connaître la progression de chaque groupe (dont il est membre) par rapport à la réalisation des activités mais aussi, les membres présents, absents, inactifs ; Connaître les niveaux de réalisation des activités par l'ensemble de groupes (dont il est membre) et ainsi ajuster le calendrier si besoin est. Les apprenants ont aussi accès à ces informations censées les responsabiliser, leur montrer leur performance et leur permettre de comparer cette performance avec celles des autres apprenants du même groupe. Sachant que le système permet en plus d'avertir les groupes en cas de retard. Et son dernier but est d'implémenter la méthode de formation de groupe que nous avons proposée.

Dans la section suivante, nous présentons la modélisation de COLYPAN avec la méthodologie Aalaadin que nous avons présentée dans le chapitre 2.

## 5.10 Modélisation du système

COLYPAN est donc un système d'apprentissage de gestion de projet de façon collaborative. Ce système doit être doté des fonctionnalités permettant la réussite de l'apprentissage et évitant l'abandon des utilisateurs, apprenants ou tuteurs. Nous avons choisi d'utiliser la technologie agent pour concevoir ce système. En plus, nous avons choisi une architecture P2P hybride pour son déploiement [Mazyad et al, 2008]. Nous avons montré dans le chapitre 3, les avantages de ce couplage. Cependant, cette association agent-P2P a aussi certains inconvénients qui proviennent essentiellement de la dynamique des systèmes P2P où les pairs peuvent apparaître ou disparaître à n'importe quel moment ce qui pose problème à la coordination des agents qui est pourtant essentielle pour un système multi-agents dont les agents sont constamment en interaction. Nous avons donc proposé une méthode de formation des groupes afin de résoudre ces problèmes de coordination d'agents.

Nous commençons la modélisation par la phase d'analyse, comme le préconise Aalaadin, cette phase sert à identifier les fonctionnalités du système et les dépendances entre les groupes [Mazyad et al, 2010c]. Nous avons étudié au début de ce chapitre les rôles du tuteur dans le contexte d'apprentissage collaboratif à distance et nous avons identifié ses besoins de façon générale ainsi que les besoins de l'apprenant et du groupe. Il s'agit maintenant

d'analyser ses besoins pour identifier les fonctionnalités dont il faut doter le système afin de les satisfaire.

### 5.10.1 Fonctionnalités du système

Les apprenants, acteurs principaux du système, ont besoin des informations sur la réalisation de leurs tâches, sur leur progression dans l'apprentissage, sur les performances du groupe entier ainsi que de chaque membre. Ces besoins imposent au système des fonctionnalités permettant de:

**Déterminer les apprenants absents appartenant au même groupe sur une période.** Le but étant d'éliminer les membres qui sont absents pendant une longue période et qui n'apportent rien au groupe. En effet, l'apprentissage étant collaboratif, le comportement de chaque membre du groupe influence le groupe en entier.

**Repérer les apprenants passifs appartenant à son groupe.** Ce sont les apprenants qui se connectent mais ne communiquent pas et ne réalisent pas les activités attendues. Le but est de les motiver, de les inciter à collaborer en les sollicitant constamment. Ceci permettra d'augmenter l'estime de soi chez les apprenants qui ne se sentent pas à l'aise dans ce type d'apprentissage et les encouragera à collaborer.

**Déterminer les apprenants actifs.** Ce sont les apprenants qui communiquent et réalisent les activités, qui ont un rendement strictement inférieur à 0,5 (0< R1< 0,5) ainsi que leur temps total de connexion ($T_i$) est proche de la moyenne (moy) de temps de connexion de tous les apprenants du groupe (Eq 3). Soit n le nombre des apprenants du groupe, $T_i$ est le temps total de connexion de l'apprenant i et moy est la moyenne qui est calculée selon la formule (Eq 2). Le rendement d'un apprenant i ($Ap_i$) est obtenu par un simple calcul comme le montre la formule (Eq 1).

**(Eq 1)** : Rendement $_{(Api)}$ = temps effectif de travail $_{(Api)}$ / temps total de connexion $_{(Api)}$

**(Eq 2)** : $moy = \sum_{i=1}^{i=n} T_i / n$

**(Eq 3)** : $T \geq 0{,}25 * moy$

**Déterminer les apprenants très actifs.** Tout comme les apprenants actifs, les apprenants très actifs communiquent et réalisent les activités mais ils ont en plus un rendement supérieur à 0,5 ($0{,}5 \leq R2 \leq 1$) ainsi que son temps total de connexion ($T_i$) est supérieur à la moyenne (moy) de temps de connexion de tous les apprenants du groupe (Eq 4). Soit n le nombre des apprenants du groupe, $T_i$ est le temps total de connexion de l'apprenant i et moy est la moyenne qui est calculée selon la formule (Eq 2).

**(Eq 4)** : $T \geq moy$

**Déterminer les apprenants sociables.** Ce sont les apprenants qui demandent l'aide des autres apprenants et répondent aux apprenants qui les sollicitent pour une aide.

**Déterminer les apprenants impliqués.** Ce sont les apprenants impliqués dans l'apprentissage collaboratif qui essayent de motiver les apprenants passifs en leur envoyant

des messages et en les sollicitant afin qu'ils participent à l'apprentissage et deviennent active. Nous considérons qu'un apprenant est impliqué s'il contacte plusieurs fois un apprenant passif ou une ou plusieurs fois plusieurs apprenants passifs.

**Déterminer les apprenants collaboratifs.** Ce sont les apprenants à la fois très actifs, sociables et impliqués. L'apprentissage étant en groupe, il est intéressant pour chaque apprenant de connaître les apprenants collaboratifs. En effet, cela peut être un facteur pour choisir les apprenants lors de la formation d'un nouveau groupe.

**Fournir à l'apprenant son rendement sur une période ainsi que pour chaque apprenant de son groupe.** Ceci permettra à l'apprenant d'être conscient de ses actes, plus son rendement (Eq 1) est bas (< 0,5) plus il doit s'impliquer dans l'apprentissage.

**Déterminer si le groupe réalise ou non une activité donnée.** Le but est de prendre conscience de performance du groupe auquel il appartient. Il doit aussi savoir s'il y a eu de retard dans la réalisation de cette activité et ceci dans le but que chacun organise son temps pour éviter de se mettre une nouvelle fois en retard.

**Déterminer l'état du groupe.** Si plus de la moitié des apprenants est respectivement passif, actif, très actif alors le groupe est dit passif, actif, très actif.

**Déterminer le niveau collaboratif du groupe.** Si plus de la moitié du groupe est sociable, impliqué ou collaboratif alors le groupe est dit sociable, impliqué ou collaboratif.

**Déterminer les outils utilisés par chaque apprenant et par le groupe ainsi que le nombre de fois où ils ont été utilisés.** Ces informations permettent à l'apprenant d'avoir une idée sur les outils les plus utilisés afin de les utiliser pour améliorer sa collaboration avec les autres. Soit nb le nombre total de fois où un outil a été utilisé par un groupe, $nb_i$ est celui où l'outil a été utilisé par un apprenant et n le nombre des membres d'un groupe. Ainsi, nb est la somme de tous les $nb_i$ où $1 \leq i \leq n$ (Eq 5).

$$\textbf{(Eq 5)}\ nb = \sum_{i=1}^{i=n} nb_i$$

Le tuteur a besoin d'estimer la progression des apprenants dans leur apprentissage, la régulation de l'activité d'apprentissage, la qualité du scénario pédagogique, le décalage entre le calendrier prévisionnel qu'il a défini et le calendrier réel et les outils utilisés pour soutenir l'apprentissage collaboratif ainsi que la fréquence d'utilisation de ces outils [Mazyad et al, 2010a].

Ces besoins imposent au système des fonctionnalités permettant de:

**Déterminer les apprenants présents/absents sur une période, dans un groupe donné**. Le tuteur peut ainsi essayer de contacter les apprenants absents sans raisons valables pour les encourager à revenir et éviter ainsi l'abandon de l'apprentissage ou le rejet du groupe.

**Déterminer l'état des apprenants (actifs, très actifs, passifs) ainsi que leur niveau collaboratif (sociables, impliqués et collaboratifs**) **et fournir pour chaque apprenant le rendement de ses activités prévu sur une période**. Ces informations sont obtenues de la même façon que pour les apprenants. Elles permettent au tuteur d'avoir une idée précise

sur les comportements des apprenants. Il peut ainsi motiver et guider les apprenants passifs ou bien encourager les apprenants à devenir plus collaboratifs.

**Déterminer l'état des groupes ainsi que leur niveau collaboratif.** Ces informations permettent au tuteur d'avoir une idée générale sur le fonctionnement de chaque groupe.

**Déterminer si un groupe réalise ou non une activité donnée**. Ces informations permettent au tuteur de mesurer la productivité des étudiants et leur progression dans le travail collaboratif.

**Déterminer les outils utilisés par chaque apprenant et par le groupe.** Ces informations permettent au tuteur de savoir si les apprenants profitent au mieux des outils disponibles et dans le cas contraire, de les inciter à utiliser ces outils pour améliorer les conditions de leur collaboration.

**Déterminer le nombre de fois où un outil a été utilisé par chaque apprenant et par le groupe.** Ces informations permettent au tuteur de connaître les outils les plus appréciés des apprenants. (Eq 5).

Toutes ces fonctionnalités doivent être stockées dans une base de données pour pouvoir les utiliser tout en respectant les principes du modèle P2P.

## 5.10.2 La phase de conception

### 5.10.2.1 La conception des bases

Les systèmes P2P imposent que chaque pair soit client et serveur à la fois. Ainsi, il n'y a pas de serveur centralisé qui contient toutes les données et les fichiers de chaque utilisateur. Ainsi, dans notre système, chaque utilisateur aura sa base de données locale (BD_L). Cette base va contenir toutes ses données et l'utilisateur doit permettre aux autres utilisateurs de son groupe d'y accéder pour récupérer des fichiers ou des données. Cependant, pour permettre à l'utilisateur de connaître la liste des utilisateurs appartenant à son groupe, nous optons pour le choix d'une entité centralisée. En effet, nous n'adoptons pas une architecture P2P pure mais une architecture hybride comme celle adoptée par Napster où les machines des utilisateurs sont reliées à une machine appelée super-pair qui joue le rôle d'un serveur contenant un annuaire. Cet annuaire contient les informations relatives aux utilisateurs (identifiant, adresse IP, etc.). Cependant, cette base centrale ne contiendra pas les fichiers des utilisateurs bien au contraire l'échange de fichier et de données se fera selon le modèle P2P où les machines des utilisateurs sont à la fois client et serveur.

### 5.10.2.2 La conception des groupes

Dans COLYPAN, nous définissons une structure de groupe (*figure V-3*) comme étant composée d'apprenants, de tuteurs et des agents artificiels qui joue chacun un ou plusieurs rôles. Les tuteurs possèdent les rôles identifiés dans la section 5.3 :

- Guider les apprenants et adapter le scénario d'apprentissage (R1) ;
- Encourager le travail collaboratif (R2) ;

- Motiver les apprenants en difficulté (R3).

Pour les apprenants, nous identifions les rôles suivants :

- L'aidant : qui aide les apprenants ;
- L'aidé : qui demande des questions et d'information aux autres apprenants ;
- L'auteur : qui rédige des documents ;
- L'étudiant : qui apprend.

Chaque tuteur et apprenant se voit attribuer un agent qui jouera un des rôles correspondants énumérés ci-dessus. Il s'agit de l'assistant personnel (**AP**). L'AP prend la place de l'humain dans le système. Il exécute toutes les requêtes de l'utilisateur qui doivent passer par lui d'abord et qu'il retransmet ensuite aux autres agents et retourne la réponse à l'utilisateur à la fin.

En plus, nous identifions les rôles suivants pour la collecte et la gestion des traces :

- Surveillant d'activité ou S_ACTIV
- Surveillant des outils ou S_OUT
- Gérant de la base de données locale (BD_L) ou gérant BD_L
- Gérant de la base de données centrale (BD_C) ou gérant BD_C.

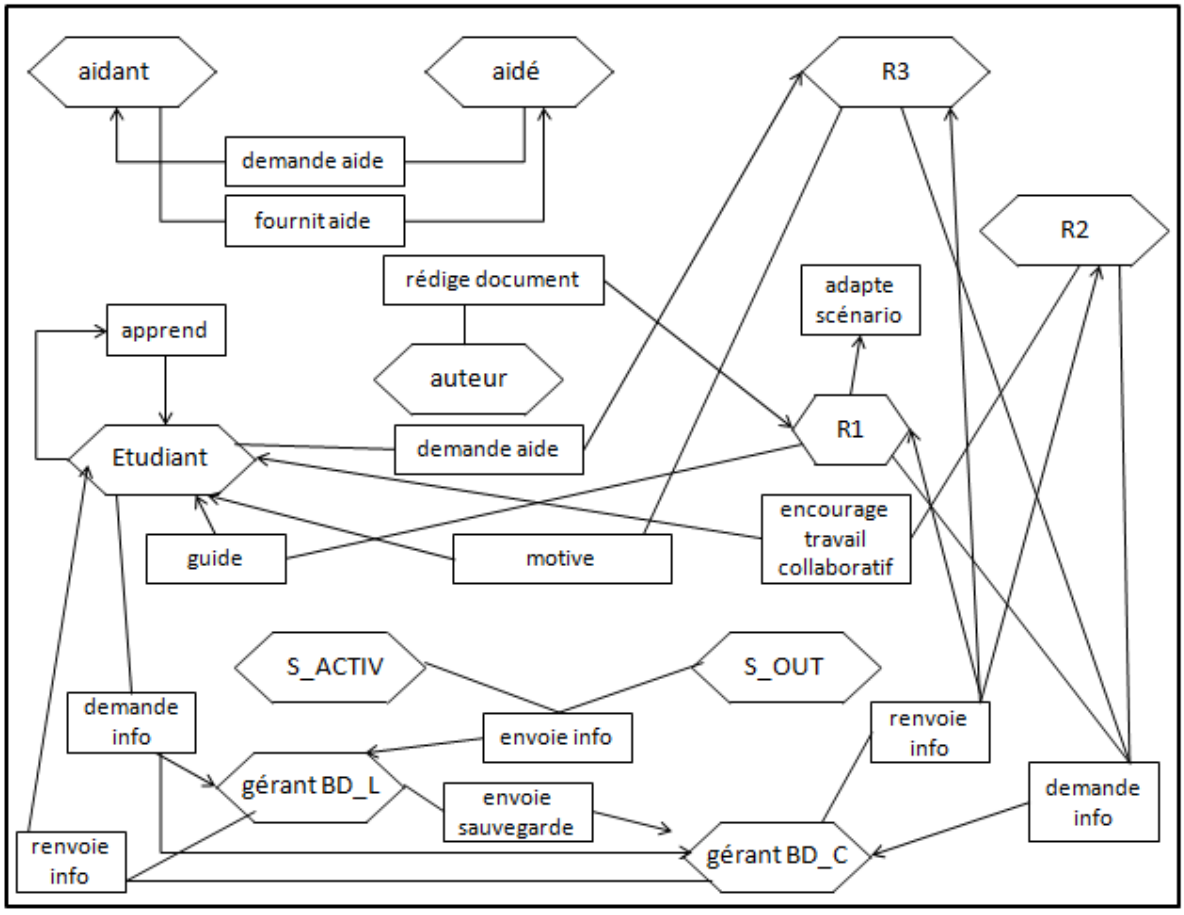


*Figure V-3 Structure organisationnelle de COLYPAN*

Chacun de ces rôles sera joué par un agent différent:

- **a_ACTIV** : Surveille les activités de l'apprenant pendant une session. Une activité est définie comme étant une étape du projet à réaliser (l'initialisation, la préparation, la planification, le suivi du projet ou la recette). Ainsi, cet agent collecte les traces de chaque utilisateur à chaque connexion et les stocke dans la BD_L de l'utilisateur en question. Il s'agit de toutes les actions effectuées par l'utilisateur (rédiger un document, envoyer un message, consulter une page, etc.).
- **a_OUT** : Surveille l'utilisation des outils. Il fournit des statistiques concernant l'utilisation des outils collaboratifs (Chat, Forum, messagerie instantanée,…). En effet, l'a_OUT a pour rôle d'enregistrer dans la base de données locale de chaque apprenant les outils qu'il a utilisés.
- **a_GI** : gère les interactions entre les agents du système et la BD_L ainsi qu'entre la BD_L et la BD_C. En effet, les agents doivent passer par l'a_GI pour accéder à la BD_L de l'utilisateur.
- **a_AUTH** : gère la base de données centrale.

Comme nous avons évoqué précédemment, COLYPAN adopte une architecture P2P hybride qui implique l'existence d'une entité centralisée, l'agent a_AUTH. Le but de ce choix est de permettre plus de flexibilité pour les utilisateurs du système en facilitant la localisation et donc l'accès aux données des autres pairs. Ainsi, les agents de COLYPAN se trouvent sur la machine cliente sauf l'a_AUTH, qui sera sur deux machines seulement (*figure V-4*).

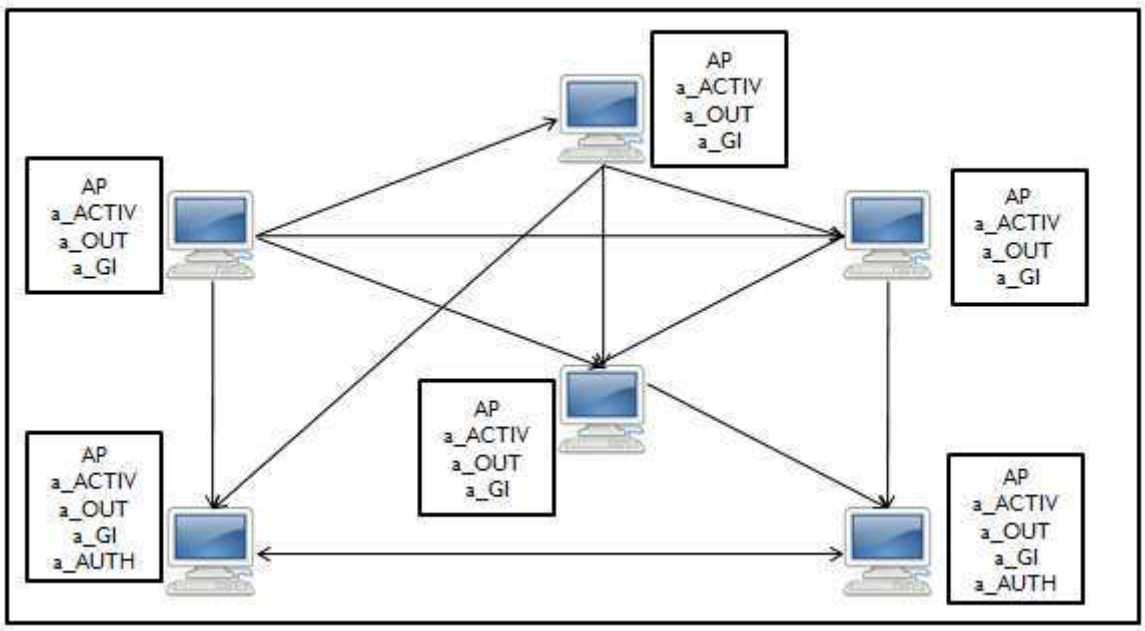


*Figure V-4 Déploiement des agents*

En dupliquant cette entité sur d'autres nœuds du réseau, nous pouvons surmonter les limitations concernant la sécurité et les performances du système.

### 5.10.2.3 Fonctionnement des groupes

Chaque groupe a un but et travaille sur un projet. Tous les membres du groupe doivent collaborer afin de l'atteindre. Le travail collaboratif exige que chaque membre contribue à part égal au travail. De ce fait, chaque membre doit être actif pour que le groupe atteigne son but. Chaque membre inactif affecte le groupe et retarde la réalisation des objectifs.

Donc afin d'assurer l'activité de tous les groupes, chaque groupe passif (plus de la moitié de ses membres sont passifs) reçoit, dans un premier lieu, un avertissement sous forme de notification envoyée automatiquement à tous les membres *(figure V-5)*. Si au bout d'une durée bien déterminée, le groupe reste inactif, le groupe doit bloquer ses membres passifs et ceci en reformant un groupe avec les membres actifs. S'ils jugent le groupe petit, ils peuvent faire appel à candidature.

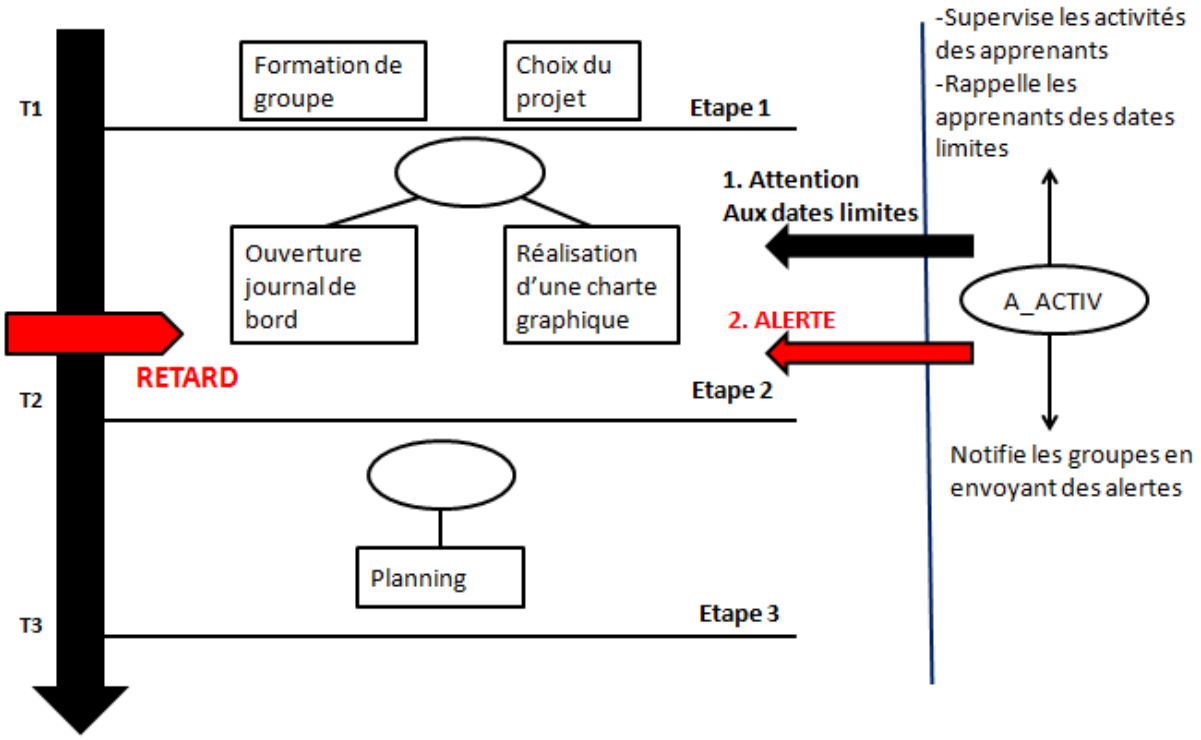


*Figure V-5 Notification des retardataires*

Dans le système, il n'y a pas une vision globale de tous les groupes. Chaque groupe connaît les membres de son groupe et ne peut communiquer qu'avec eux. Ainsi, il n'y a aucun lien direct entre les groupes à moins d'avoir un membre en commun.

Le fait qu'un membre peut appartenir à plusieurs groupes en même temps permet aux groupes de communiquer *(figure V-6)*. En effet, chaque personne ne connaît que les membres de son groupe et donc ne peut communiquer qu'avec eux. Ainsi, un membre en commun s'avère très utile pour l'échange entre deux groupes.

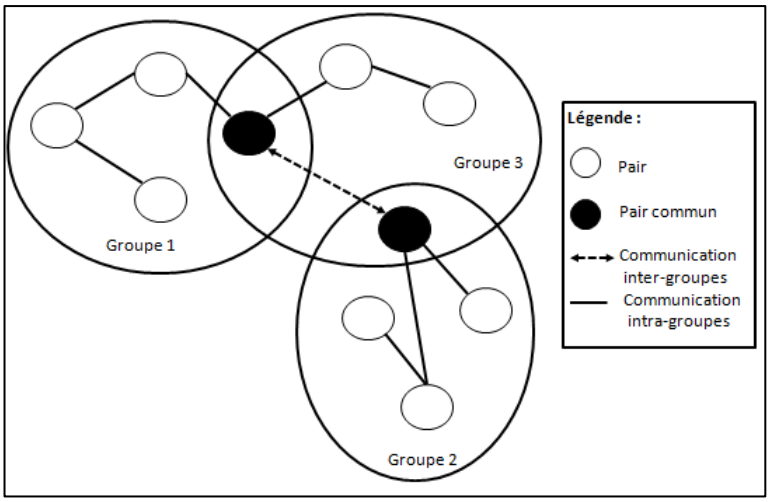


*Figure V-6 Communication inter et intra groupes*

## 5.11 Conclusion

L'identification des rôles des tuteurs, des apprenants et du groupe dans l'apprentissage collaboratif nous a permis d'identifier les besoins des tuteurs et des apprenants pour mener à bien leur tâche. Nous avons analysé ces besoins afin de proposer des fonctionnalités capables d'y répondre. Ces fonctionnalités ont donc trois buts. Le premier est de faciliter la tâche du tuteur en satisfaisant ses besoins et en lui proposant d'apprécier les activités des apprenants et de suivre leur progression dans l'apprentissage ainsi que leur niveau de collaboration. Le deuxième est d'éviter l'isolement des apprenants et de les motiver et le dernier est d'encourager le travail collaboratif et l'utilisation des outils le permettant tels que le forum, le chat, la messagerie instantanée. Nous sommes persuadés que c'est la combinaison de tous ces buts qui améliore la situation d'apprentissage collaboratif à distance et permet sa réussite.

Nous avons ensuite organisé ces fonctionnalités en objectifs à prendre en compte lors de la conception du système d'apprentissage collaboratif COLYPAN que nous avons modélisé avec la méthodologie Aalaadin. Le choix de cette méthodologie vient du fait que ce modèle a l'avantage d'assurer la généricité, ce qui nous permet d'implémenter nous-mêmes les comportements, services droits et obligations attendus par le rôle et de décrire explicitement au niveau des agents, les rôles qu'ils sont susceptibles d'utiliser. Cependant, cette généricité a rendu notre tâche complexe surtout pour bâtir une dynamique au niveau

de l'organisation. Toutefois, l'utilisation de la plateforme Madkit qui repose sur le modèle AGR d'Aalaadin a allégé ensuite le travail.

COLYPAN est un SMA à architecture P2P où des agents autonomes interagissent et communiquent afin de réaliser un but commun c'est-à-dire satisfaire les exigences des utilisateurs (apprenant ou tuteur). Il a pour objectif de permettre l'apprentissage collaboratif de gestion de projet. En effet, COLYPAN implémente la méthode d'apprentissage par projet MAETIC ainsi que la méthode de formation de groupe que nous avons proposé. Il permet ainsi de coordonner les actions d'agents en tenant compte de la nature dynamique des systèmes P2P. Dans COLYPAN, le tuteur a pour rôle de guider les apprenants et de faciliter leur apprentissage, le groupe a pour rôle de motiver les apprenants, de créer un lien social qui empêche le sentiment d'isolement. Quant aux apprenants, ils apprennent en interagissant avec les autres.

La méthode de formation de groupe que nous avons présenté a pour premier objectif la coordination d'agents dans un contexte P2P. En plus, elle s'avère bien utile dans le contexte d'apprentissage collaboratif puisqu'elle permette d'assembler des agents qui ont des intérêts en commun et qui souhaite collaborer et interagir. En effet, notre méthode évite de mettre dans le même groupe, les agents qui ne veulent pas travailler ensemble. De plus, elle permet d'éviter le comportement de cavalier seul, puisque chaque agent a intérêt à collaborer et à interagir pour ne pas se voir rejeter et placer dans la liste des blocages. Cette méthode peut être utilisée dans n'importe quel système multi-agents pour assurer la coordination des actions des agents.

Dans le chapitre suivant, nous illustrons, à l'aide d'un scénario, le fonctionnement du système COLYPAN. Nous présentons aussi les détails de son implémentation sur la plateforme Madkit.

# Chapitre VI. Implémentation du système COLYPAN

## 6.1 Introduction

Dans ce chapitre, nous allons vérifier la réalisabilité de notre proposition organisationnelle en implémentant notre système COLYPAN sur une plateforme multi-agents. Pour cela, nous avons choisi la plateforme Madkit présentée dans le chapitre 3. Ce choix vient du fait que Madkit est particulièrement destinée à des SMA fondés sur des critères organisationnels et qu'elle fonctionne en mode distribuée de manière transparente à partir d'une architecture P2P sans nécessiter de serveur dédié.

Nous décrivons donc l'implémentation de COLYPAN ainsi que les choix techniques que nous avons faits lors de cette implémentation. Nous présentons ensuite l'interface graphique de COLYPAN ainsi que les outils de travail collaboratif (chat, forum, messagerie instantanée, etc.) ainsi que les fonctionnalités qu'il propose. En effet, COLYPAN propose aux utilisateurs des fonctionnalités leur permettant de vérifier la réalisation des activités, de mesurer leur productivité et leur progression dans le travail collaboratif mais aussi de permettre au tuteur d'estimer la régulation de l'activité d'apprentissage ainsi que la qualité du scénario pédagogique.

Nous illustrons, à la fin de ce chapitre, le fonctionnement de COLYPAN par un scénario qui nous permettra d'expliquer son utilisation ainsi que le déroulement de l'apprentissage.

## 6.2 Implémentation de COLYPAN

Notre système est un SMA à architecture P2P hybride. Il est implementé en Java et est déployé sur la plateforme Madkit dont nous avons installé la version 5.0. L'utilisation de Java a nécessité l'installation de kit de développement Java (JDK de l'anglais « Java Development Kit »). JDK est l'environnement dans lequel le code Java est compilé pour être transformé en bytecode afin que la machine virtuelle Java (JVM de l'anglais « Java Virtual Machine ») puisse l'interpréter. Il contient une sélection d'outils de programmation, incluant : le compilateur javac ; l'archiveur jar ; le générateur de documentation javadoc ; et le débogueur jdb. L'environnement d'exécution Java (JRE de l'anglais « Java Runtime Environment »), un ensemble d'outils permettant l'exécution de programmes Java sur toutes les plateformes supportées, fait également partie du JDK.

Concernant la gestion de base de données, Java possède l'API JDBC (de l'anglais « Java DataBase Connectivity) qui permet la connexion à des bases de données. En effet, l'API JDBC permet à un programme de se connecter à n'importe quelle base de données en utilisant la même syntaxe et donc elle est indépendante du système de gestion des bases de données (SGBD). Nous avons choisi MySQL comme SGBD. Ce choix vient du fait que MySQL est un serveur très populaire et son utilisation est gratuite. Il fonctionne sur la plupart des systèmes d'exploitation et il est souvent utilisé conjointement avec Java.

S'agissant d'un modèle P2P hybride, notre système est constitué des pairs reliés à un super-pair. Ainsi, les machines avec une faible bande passante sont appelées des pairs. Les machines avec une bonne bande passante sont les super-pairs. Ces derniers jouent le rôle de serveur de localisation. Ce modèle a pour but d'utiliser les avantages des 2 types de réseaux P2P : le P2P centralisé et P2P décentralisé. En effet, sa structure permet, d'une part, de diminuer les nombres de connexions sur chaque serveur, et ainsi d'éviter les problèmes de bandes passantes. D'autre part, il utilise un mécanisme issu des réseaux décentralisés pour tenir à jour un annuaire client et un index des fichiers à partir des informations provenant des autres serveurs. Un serveur peut donc proposer à n'importe quel client toutes les informations contenues sur le réseau. L'utilisation d'un modèle hybride est essentielle pour le contexte d'apprentissage. En effet, cela nous permet de connaître les utilisateurs du système tout en gardant les avantages des systèmes P2P.

Dans COLYPAN, les super-pairs abritent la base de données centrale du système. Cependant, à aucun moment les fichiers des utilisateurs ne seront stockés sur ce serveur, en effet, le serveur permet de les localiser et les utilisateurs peuvent aller les chercher sur la machine qui les stocke dans un répertoire partagé et accessible à distance. En plus de cette base centrale, chaque utilisateur possède sur sa machine une base locale qui contient les informations qui le concernent et les données qui lui appartiennent. Les utilisateurs peuvent y accéder à condition que son propriétaire ait permis d'y accéder.

### 6.2.1 Les agents du système

Madkit propose une classe d'agent abstraite (AbstractAgent) qui offre des fonctionnalités de base pour assurer le bon fonctionnement d'un agent, tels que des outils de communication et la manipulation d'une interface graphique.

Nous avons créé nos agents en utilisant l'agent Designer de Madkit. C'est un outil qui a pour but de faciliter la création et le lancement des agents. Cependant, cet agent n'impose aucun concept d'exécution. En effet, il propose uniquement les méthodes pour démarrer le cycle de vie d'un agent (Activate, live, end) et c'est à l'utilisateur de programmer son comportement. D'ailleurs, il est important de préciser qu'un agent créé, ne débutera réellement son cycle de vie qu'une fois il est activé. En fait, à l'activation de l'agent le noyau lui affecte une adresse unique, il devient ainsi identifiable par les autres membres de la société et pourra aussi faire partie de l'organisation et jouer des rôles dans différents groupes.

Ainsi, notre système dispose de cinq agents artificiels (l'AP, a_OUT, a_ACTIV, a_GI, a_AUTH) (*figure VI-1*). Ces agents commencent leur cycle de vie au lancement du programme par l'utilisateur. Chacun de ces agents a un rôle bien défini à jouer.

L'agent AP est le remplaçant artificiel de l'utilisateur dans le système. Ainsi, toute requête de l'utilisateur doit passer par son AP. Par exemple, si l'utilisateur X veut envoyer un message à l'utilisateur Y, dans le système, c'est l'AP qui l'envoie à l'AP de Y.

Les autres agents sont des agents qui gèrent les fonctionnalités du système. Ainsi, l'agent a_AUTH, est la seule entité centralisée. Cependant, il existe sur plusieurs machines afin d'éviter les goulots d'étranglement ou le blocage du système s'il tombe en panne. Cet agent possède la liste des agents humains du système, leur adresse IP et d'autres informations les concernant. Cependant, il ne contient aucun fichier appartenant à ces agents humains mais seulement une indexation à ces fichiers. Ainsi, les agents humains font appel à lui à chaque fois qu'ils ont besoin de connaître la liste de leur groupe, d'intégrer un nouveau groupe ou d'en former un. Cet agent contient aussi un récapitulatif des activités des utilisateurs du système. En effet, à chaque connexion d'un utilisateur, son agent a_GI récupère les traces et les stocke dans la base de données locale. Il dresse ensuite un récapitulatif qu'il envoie à l'a_AUTH. Tant que l'agent humain est connecté, ce récapitulatif sera mis à jour toutes les un quart d'heure. L'agent a_GI est le seul agent capable d'accéder à la BD_L. En effet, chaque agent voulant y accéder est obligé de passer par l'a_GI.

Le récapitulatif envoyé à l'a_AUTH contient la liste des agents absents/ présents ainsi que leur temps de connexion, leur état (passif, actif, très actif) ainsi que leur niveau de collaboration (sociable, impliqué, collaboratif). Il contient aussi les outils utilisés par chaque utilisateur.

L'agent a_OUT est l'agent responsable de collecter les informations sur l'utilisation des outils collaboratifs. En effet, ces outils sont la clé de l'apprentissage collaboratif en ligne et sans eux, l'apprentissage ne peut pas avoir lieu. Pour cela, il est important que les utilisateurs puissent savoir s'ils sont suffisamment utilisés. L'agent a_OUT leur apporte la réponse. A chaque fois qu'un utilisateur utilise un outil, l'agent a_OUT envoie cette information à l'agent a_GI pour l'inclure dans la BD_L. Il récupère aussi le temps passé à utiliser cet outil, ceci dans le but de voir quel outil a le plus de succès et ainsi, encourager les apprenants passifs à l'utiliser pour participer à l'apprentissage. En effet, un apprenant passif a besoin d'être motivé, pour cela, il a besoin de se sentir entouré. Le fait de l'orienter vers les outils les plus utilisés augmentent la chance qu'il puisse avoir un retour ou bien qu'il puisse discuter et échanger avec les membres de son groupe.

Finalement, l'agent a_ACTIV récupère toutes les traces de l'utilisateur dès sa connexion jusqu'à sa déconnexion et les envoie à l'agent a_GI pour qu'il les enregistre dans la BD_L. Il récupère ainsi, le temps de connexion de l'utilisateur, ses activités dans le système (consulter/rédiger un document, discussion sur le chat, ouvrir un forum, commenter, demander ou répondre à une demande d'aide, le temps total de connexion, le temps effectif de travail, etc.).

Ainsi, COLYPAN définit deux agents pour collecter les traces des activités de chaque utilisateur. Ces agents récoltent les traces des activités de l'utilisateur dès sa connexion au système jusqu'à sa déconnexion. Les traces récoltées sont envoyées à l'agent qui gère la base de données pour les sauvegarder dans la base de données locale de chaque utilisateur qui permet, en structurant les données enregistrées d'avoir un corpus réellement exploitable pour produire des nouvelles connaissances sur les usages et en globalisant les observations d'affiner la portée ou l'interprétation des analyses.

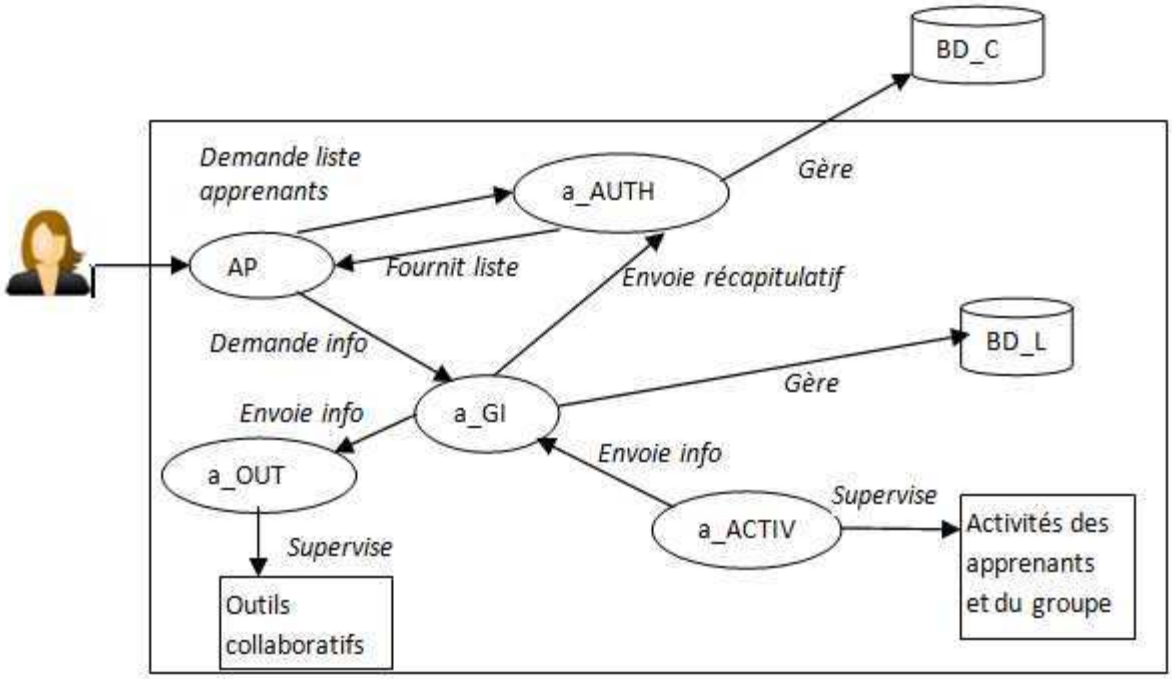


*Figure VI-1 Les agents dans COLYPAN*

### 6.2.2 L'interface de COLYPAN

COLYPAN est un système à architecture P2P. Dans un tel agent, il n'y a pas d'hiérarchie entre les apprenants et leurs tuteurs. Le système propose donc les mêmes fonctionnalités à l'apprenant et au tuteur. L'interface graphique de COLYPAN est simple à utiliser (*figure VI-2*). En effet, en se connectant au système, l'utilisateur accède à toutes les fonctionnalités et les outils du système. Il suffit alors de choisir ce qu'il veut faire. Il peut ainsi rédiger un document en utilisant les outils de traitement de texte, discuter en utilisant le chat, envoyer des messages privés aux membres de son groupe, proposer un sujet de discussion sur le forum, demander de l'aide ou répondre à une demande d'aide, poster des commentaires sur les documents des autres apprenants du groupe, lancer un appel pour former un groupe, demander de rejoindre un groupe, proposer un projet…

Nous avons choisi de proposer autant d'outils de communication pour pouvoir comprendre les traces récoltées. En effet, l'utilisateur peut envoyer un message pour demander de l'aide. Cependant, nous ne pouvons pas savoir qu'il s'agit d'une demande de l'aide qu'en le lisant ce qui demande une analyse syntaxique du contenu du message ce qui est long et lourd à

faire. Nous avons préféré d'offrir ces outils de communication suite à l'analyse des besoins des utilisateurs dans un contexte d'apprentissage collaboratif (cf. chapitre 5).

A partir de cette interface, un utilisateur peut lancer un appel à créer un groupe (par envoi de messages à la liste d'agents fournit par l'agent a_AUTH). Pour cela, il doit préciser le thème général et le projet sur lequel doit travailler le groupe ainsi que les compétences requises. Il peut aussi demander de rejoindre un groupe en envoyant par message ses compétences ainsi que ses besoins au membre de ces groupes. Il faut noter qu'un groupe travaille sur un seul projet mais plusieurs groupes peuvent travailler indépendamment sur le même projet sachant que les groupes ne communiquent que s'ils ont un membre en commun.

*Figure VI-2 Interface de COLYPAN*

Les utilisateurs, apprenants et tuteurs, peuvent aussi proposer un projet. Il faut alors préciser le sujet ainsi que le calendrier contenant les dates butoirs de chaque phase. Il faut noter que si le projet est proposé par le tuteur, ce dernier doit envoyer les informations concernant le projet proposé par message à la liste des apprenants fournie par l'agent a_AUTH. Cependant si c'est par l'apprenant, il doit d'abord en informer un tuteur en lui envoyant un message. Ensuite, il pourra envoyer le message à son groupe s'il fait déjà partie d'un groupe ou sinon, il lance un appel à former un groupe pour travailler sur ce projet.

Il est important de préciser que les informations concernant les projets ne sont pas stockées dans la base de données centrale BD_C gérée par l'agent a_AUTH. En effet, ces informations (le sujet et le calendrier) sont tout d'abord envoyées par messages aux apprenants afin qu'ils puissent décider de prendre ce projet ou de ne pas le faire. Ces informations seront alors stockées dans la base de données locale BD_L de chaque apprenant. Ainsi, chacun peut consulter ses projets en cours et les dates à respecter (*cf. figure VI-2*).

Enfin, ils peuvent utiliser les fonctionnalités de suivi qui leur fournissent trois grandes catégories d'appréciations :

- au niveau d'un individu : en appréciant sa productivité en termes de réalisation des activités pédagogiques (actif, passif, très actif), son niveau de collaboration (collaboratif, social, impliqué) avec les autres membres du groupe ;
- au niveau du groupe entier : en appréciant l'évolution de l'activité du groupe dans une période de temps et de l'implication de ses membres dans l'apprentissage collaboratif et ceci en indiquant les personnes présentes, absentes, l'état du groupe par rapport à la réalisation des activités pédagogiques ; le niveau de collaboration du groupe ;
- au niveau des outils : en appréciant le niveau d'utilisation des outils ;
- au niveau de l'activité : en appréciant le niveau de réalisation (à moitié finie, finie, en retard) d'une activité par l'ensemble de membres.

### 6.2.3 COLYPAN Pas-à-Pas

L'utilisation de COLYPAN nécessite d'abord le téléchargement du paquetage contenant le fichier exécutable nécessaire pour son installation ainsi que les bibliothèques indispensables pour son bon fonctionnement (madkit.jar). Il nécessite aussi l'installation d'un jdk comme nous avons expliqué dans la section 6.2.

Une fois installé, COLYPAN est prêt à être utilisé. Cependant, chaque utilisateur a une base de données locale qui contient toutes les informations le concernant lui et ses projets. Pour permettre aux apprenants de son groupe de récupérer des données de sa base, chaque apprenant doit les autoriser à y accéder à distance mais aussi aux tuteurs responsables de son groupe. Pour cela, chaque utilisateur doit configurer le serveur de base de données local afin qu'il autorise les connexions externes. Cette ouverture d'accès se fait en deux étapes :

1. Configuration de MySQL (*my.cnf*) : Il suffit de commenter les deux lignes suivantes, et procéder ensuite à un redémarrage du serveur MySQL.

```
# skip-networking
# bind-address
```

```
GRANT ALL PRIVILEGES ON colypan.* TO 'user_colypan'@'%' IDENTIFIED BY 'Password_Colypan';
```

2. Configuration des privilèges MySQL : il faut créer un nouvel utilisateur « user_colypan », et le configurer afin que les connexions à la base de données « colypan »puissent se faire de l'extérieur.

Ensuite, l'utilisateur doit créer un compte. La création de compte implique l'enregistrement de l'utilisateur auprès de l'a_AUTH. Ainsi, il pourra faire ses premiers pas dans le système et découvrir ses fonctionnalités. Cependant, il ne peut pas commencer son apprentissage puisque l'apprentissage ne se fait que par groupe. Il a donc deux choix, lancer un appel pour former un groupe (*figure VI-3*) ou bien rejoindre un groupe déjà existant. Le premier choix se

fera selon la méthode de formation de groupe que nous avons proposé et implémenté dans COLYPAN. Quant au deuxième choix, le choix du groupe se fera en fonction de ses compétences et des besoins du groupe. Concrètement, son AP envoie une requête à l'a_AUTH pour lui demander la liste des groupes existants et le thème de leur projet. L'a_AUTH lui renvoie alors la liste des groupes qui correspond au profil de l'utilisateur qui choisira donc de faire partie d'un ou plusieurs de ces groupes.

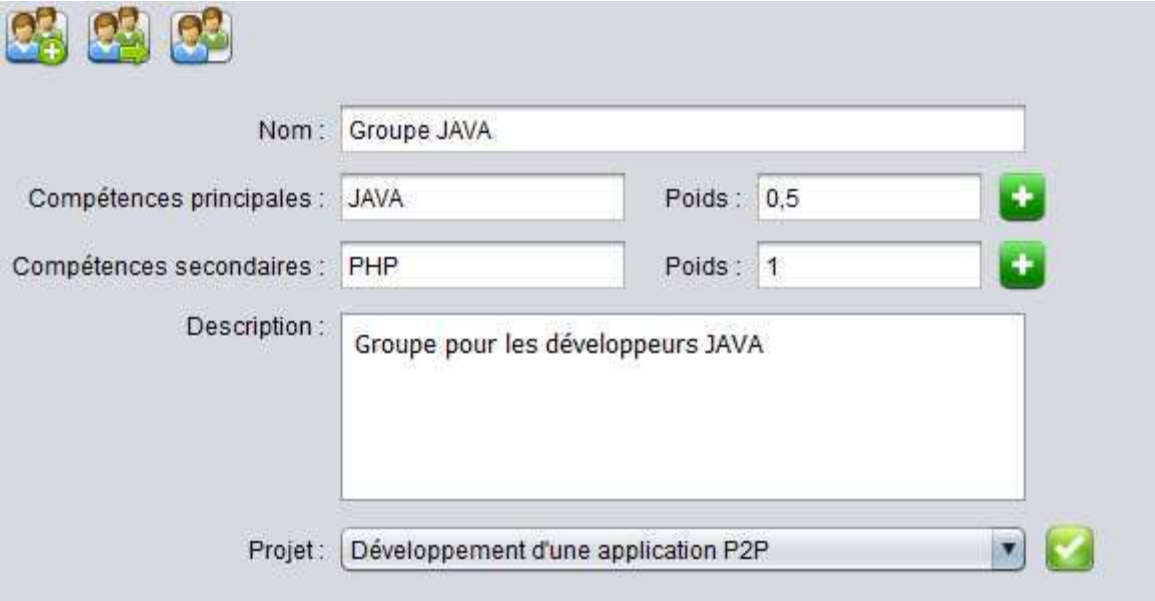


*Figure VI-3 Appel pour former un groupe*

Si l'apprenant est interesé par l'appel, il doit préciser ses compétences parmi celles requises pour rejoindre le groupe (*figure VI-4*) et il sera admis ou non selon le poids total de ses compétences.

*Figure VI-4 Rejoindre un groupe*

Nous présentons dans la section suivante un scénario fictif afin de décrire le fonctionnement du système.

## 6.3 Le scénario fictif

Nous utilisons un scénario fictif afin de démontrer la réalisabilité de la modélisation de notre système COLYPAN et de présenter ses fonctionnalités :

### 6.3.1 Les principes de COLYPAN

COLYPAN est un système pour l'apprentissage collaboratif de la gestion de projet qui suit la méthode pédagogique MAETIC. Ainsi, la formation est divisée en cinq étapes. Le tuteur fournit aux apprenants des documents expliquant cette méthode ainsi qu'un planning prévisionnel à respecter. Ce planning sera ajusté au fur et à mesure de l'avancement de la formation si le tuteur trouve qu'il y a besoin. A la fin de chaque étape, un document résumant le travail effectué qu'on appelle le délivrable, doit être envoyé au tuteur. En cas de retard, le groupe reçoit un avertissement.

Le travail se fait étape par étape, les apprenants ne peuvent pas passer à l'étape 2, s'ils n'ont pas fini l'étape 1. De plus, trois règles sont à respecter :

- Le travail s'effectue par groupe.
- Chaque groupe travaille sur un seul projet.
- Chaque apprenant ne peut communiquer qu'avec les membres de son groupe.

Cependant, chaque projet peut être traité par plusieurs groupes en même temps et les groupes peuvent communiquer uniquement s'ils ont un membre en commun. Toutefois, nous supposons qu'un apprenant ne peut pas faire membre de deux groupes travaillant sur le même projet. Il faut noter en plus que le projet peut être proposé par le tuteur ou par un des apprenants. En effet, une fois le projet choisi, l'apprenant peut faire appel pour former un groupe. Suite à cet appel, des apprenants intéressés par ce projet peuvent faire une demande d'admission dans ce groupe. Sachant qu'ils peuvent aussi lancer un appel à former un autre groupe pour le même projet (par exemple dans le cas où leur candidature a été rejetée par le premier groupe). La formation de groupe se fera selon la méthode décrite dans le chapitre précédent. Nous supposons qu'un groupe devient fonctionnel quand il compte au moins 3 membres. Cependant, étant un groupe ouvert, il continuera à s'agrandir même si l'apprentissage a déjà commencé. En effet, ces nouveaux membres peuvent compter sur les anciens pour les aider à rattraper leur retard. C'est l'un des nombreux avantages du travail en groupe.

Une fois le groupe est fonctionnel, l'apprentissage collaboratif peut commencer. Il se traduit par l'échange des connaissances et des expériences. En plus, chaque apprenant est responsable de son apprentissage et de celle de son groupe. Il est de son devoir de collaborer et d'être actif mais aussi de motiver les membres passifs et de les inciter à participer. En effet, les apprenants discutent en utilisant le chat ou le forum, ils échangent leur point de vue, fournissent des explications, demandent de l'aide ou proposent leur aide aux apprenants qui sont en difficulté, posent des questions. De plus, ils échangent des documents et des références. L'apprentissage se construit donc grâce à ces interactions. Ainsi, ces apprenants vont collaborer pour construire le délivrable à rendre au tuteur. Cependant, ceci ne constitue pas un devoir qui sera noté par le tuteur. Il s'agit juste d'évaluer les compétences acquises. C'est le moyen pour que le tuteur puisse prendre connaissance du rendu de chaque groupe afin de les guider et de les aider s'il juge qu'ils sont

en difficulté. Ce retour permet aussi au tuteur de juger de la pertinence du scénario pédagogique conçu ainsi que du calendrier et de les adapter si cela s'avère nécessaire.

Nous présentons dans la partie suivante le scénario fictif que nous proposons. Nous écrivons en italique, les parties du scénario, en gras l'explication sur les interactions des agents.

### 6.3.2 Déroulement de l'apprentissage

*Roger est un étudiant intéressé par une formation de gestion de projet de façon collaborative. Il désire s'y inscrire.*

*La formation est à distance et pour la suivre, il faut télécharger un système appelé COLYPAN. Ainsi, Roger installe COLYPAN sur son ordinateur en suivant le manuel et commence ses premiers pas. Il veut se familiariser avec l'environnement avant de commencer la formation. Roger est satisfait, ce système paraît simple à utiliser. En effet, il avait peur de rencontrer des difficultés en utilisant ce système et de devoir les gérer en plus de son apprentissage. Le voilà bien rassuré.*

**[En installant COLYPAN sur sa machine et en naviguant sur l'interface, Roger a activé les agents artificiels. En effet, il est maintenant enregistré dans la liste de l'agent a_AUTH, son AP est prêt à rendre service et les agents a_OUT et a_ACTIV ont déjà commencé la collecte de ses actions et ont envoyé les informations recueillies à l'agent a_GI.]**

*Roger ne tarde pas à recevoir un message privé du tuteur. Dans ce message, le tuteur M. Tresprof, lui souhaite le bienvenu et l'informe qu'il sera suivi par deux tuteurs : lui et M. Trespedago. Il lui explique les principes de la formation* (cf. section 6.3.1) *et l'encourage à le contacter en cas de besoin.*

**[C'est l'AP du tuteur qui se charge de l'envoi du message.]**

*Roger, très motivé, répond au tuteur en lui annonçant qu'il a choisi comme projet «Développement d'une application P2P » et qu'il a décidé de lancer un appel pour former un groupe.*

**[L'AP de Roger se charge de l'envoi du message. Ensuite, Roger utilise le système pour créer un projet et proposer un calendrier à respecter pour ce projet. En réponse à la requête la demandant, l'a_AUTH fournit à l'AP de Roger la liste des apprenants. L'AP se charge alors de lancer un appel à former un groupe incluant les informations sur le projet aux apprenants de cette liste (***figure VI-5***).]**

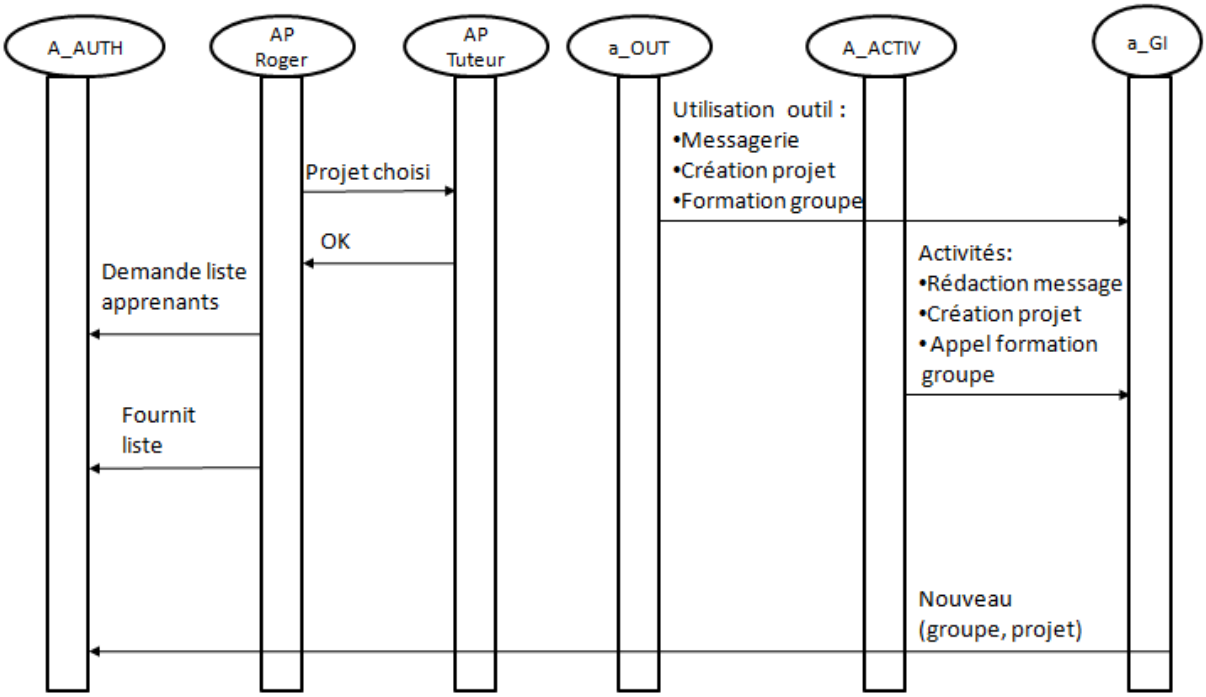


Figure VI-5 *Interaction des agents dans COLYPAN*

*Le groupe étant formé, le travail commence alors. Le tuteur envoie donc à chaque membre du nouveau groupe les documents nécessaires pour démarrer le projet et leur demande de tenir compte du calendrier prévisionnel. Ces documents contiennent des explications concernant la méthode MAETIC que doit suivre les apprenants dans la réalisation de leur projet.*

*Roger remarque qu'à chaque phase, le même rituel se répète et il doit travailler avec les apprenants de son groupe de façon collaborative pour réaliser leur tâche et rendre le délivrable au tuteur dans les délais.*

*Roger commence alors à rédiger ce document en utilisant les outils de traitement de texte du système et en discutant avec les membres connectés quand soudain, il reçoit un message privé d'un membre de son groupe, Eric. Il vient de se connecter et il a besoin d'aide. Roger lui fournit les explications dont il a besoin et l'invite à rejoindre le chat pour discuter avec ces collègues et prendre connaissance de leur point de vue.*

**[Interactions des agents :**

**- L'a_OUT envoie à l'a_GI que Roger a utilisé trois outils : outils de traitement de texte, messagerie privée, le chat.**

**- L'a_ACT envoie à l'a_GI les informations concernant les activités de Roger: la rédaction, envoi de message, discussion sur le chat. Les mêmes actions se produisentt dans le système d'Eric.**

**- L'a_GI envoie un récapitulatif à l'a_AUTH]**

*Roger met ensuite le document qu'il a rédigé à disposition de tous les membres, pour qu'ils le commentent et proposent des améliorations ce que font aussi les autres membres qui ont commencé la rédaction. Toto trouve des points de divergence entre son document et celui de Roger. Il veut en discuter avec Roger qui n'est pas connecté, Il discute alors avec les autres membres connectés. Le tuteur, connecté à ce moment rejoint la communauté pour donner son avis et répondre aux interrogations du groupe. Le groupe continue ainsi leur collaboration jusqu'à se mettre d'accord sur le contenu, un apprenant se charge alors d'envoyer le délivrable au tuteur. C'est la fin de la première phase. Le groupe de Roger a réussi à finir la première phase dans les délais. Le tuteur, à son tour, prend connaissance du document et envoie ses commentaires aux membres du groupe.*

**[Les AP du tuteur et des apprenants téléchargent le document rédigé par Roger. En effet, les documents ne s'envoient pas dans COLYPAN mais par application du principe de partage des fichiers dans le P2P, leur propriétaire les met dans un répertoire partagé et les utilisateurs viennent les chercher. Pour l'a_OUT : il y a utilisation du chat, des outils de traitement de texte, de l'outil pour poster un commentaire, de messagerie privée. Quant à l'a_ACT, il y a les informations concernant la rédaction d'un document (Roger, Toto et d'autres), discussion en utilisant le chat, envoi de message, envoi au tuteur].**

*La deuxième phase a été plus compliquée pour le groupe. Ils n'ont pas réussi à la finir à temps. L'a_AUTH leur a envoyé un avertissement. Roger en apprenant responsable, a voulu découvrir les raisons de ce retard. Il consulte alors le suivi de son groupe. Il découvre alors que le groupe est passif à cause de plusieurs membres passifs. Il tente alors de les motiver et de les encourager à collaborer pour réaliser le projet et ceci en leur envoyant des messages privés ou bien en leur demandant de l'aide. Le tuteur tente de son côté aussi de motiver ces apprenants. Roger a réussi ainsi à rendre son groupe actif (figures VI-6 et VI-7).*

**[Interactions des agents :**

- **L'a_AUTH envoie un avertissement à tous les AP du groupe**
- **L'AP de Roger demande à l'a_AUTH le suivi de son groupe, l'a_AUTH lui envoie le suivi demandé**
- **L'a_ACT récolte les informations concernant les activités suivantes : demande de consultation de suivi, envoi de message, demande d'aide et les transmet à l'a_GI**
- **L'a_OUT informe l'a_GI de l'utilisation des outils suivants : les outils de demande d'aide, la messagerie privée, le suivi]**

*Dans le cas où le groupe reste passif, les membres passifs doivent être éliminés du groupe et un appel à candidature sera lancé.*

*A la fin de la dernière phase, si le travail a bien été accompli, le tuteur déclare le projet réalisé. Ainsi, les membres peuvent choisir de rester grouper pour travailler sur un autre projet ou bien de supprimer ce groupe.*

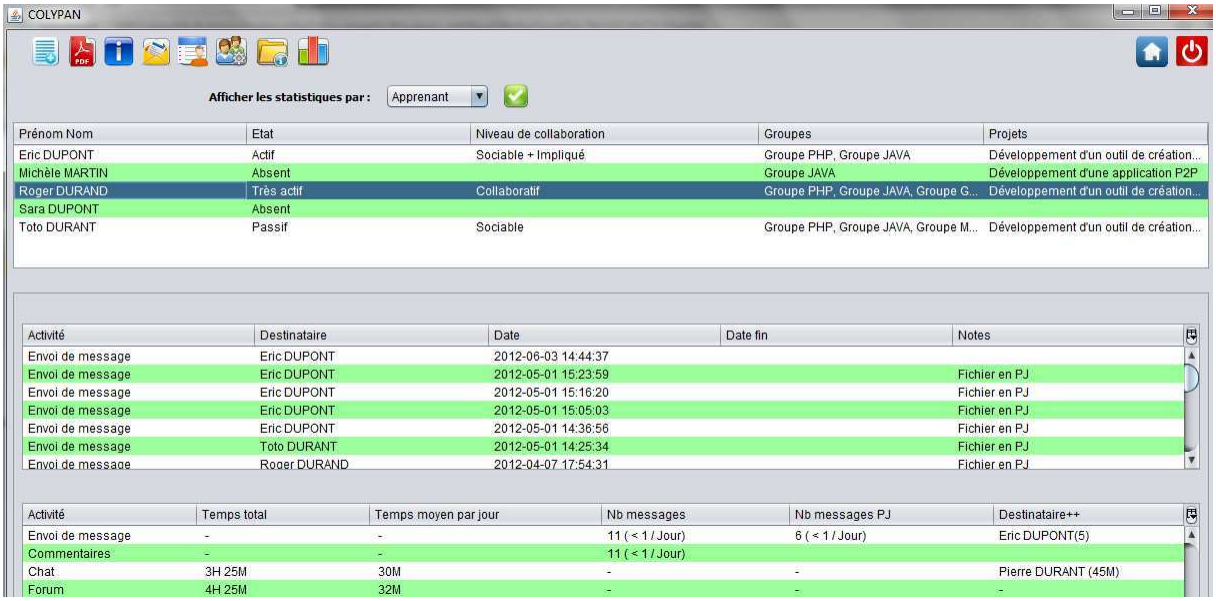


*Figure VI-6 Suivi de l'apprenant*

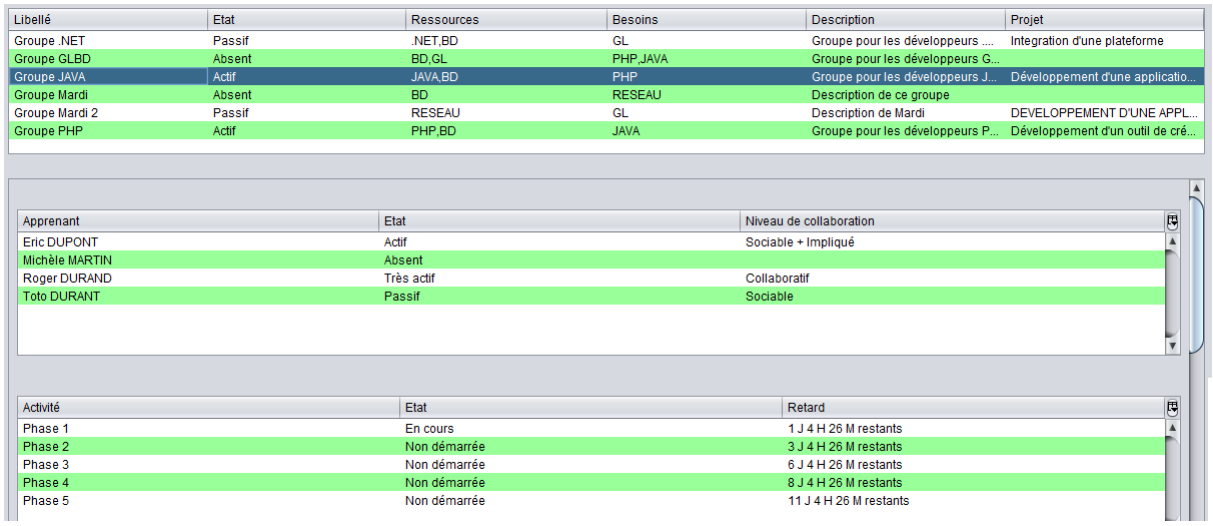


*Figure VI-7 Suivi du groupe*

## 6.4 Conclusion

Ce chapitre a présenté l'implémentation de notre système COLYPAN qui est un système d'apprentissage collaboratif qui adopte le point de vue des constructivistes en considérant que l'apprentissage se construit et que l'échange des connaissances entre les apprenants contribue à cet apprentissage. Malgré qu'il soit initialement conçu pour l'apprentissage de façon collaborative de la gestion de projet, COLYPAN peut être utilisé dans tous les domaines d'apprentissage collaboratif (langues, sciences, etc.) voire même du travail collaboratif. En effet, COLYPAN peut être utilisé dans des entreprises où des employés ont besoin de communiquer et d'échanger leurs connaissances.

Le système COLYPAN, conçu comme étant un système multi-agents à architecture P2P hybride, a été implémenté à l'aide de la plateforme Madkit. C'est la dernière phase de la méthodologie Aalaadin, qui a pour but d'expliquer le fonctionnement du système. Nous avons donc présenté l'implémentation de COLYPAN ainsi que les choix techniques. Ainsi, pour la gestion des données, COLYPAN exploite plusieurs bases de données distribuées. En effet, il y a les bases de données locales qui sont déployées sur les machines des utilisateurs. Chaque utilisateur peut accéder à celles des autres à distance si le propriétaire leur permette ; il y a aussi des bases centralisées et se trouvant sur les machines super-pairs. Ainsi, les utilisateurs ont besoin de consulter la base de données centrale à chaque fois qu'ils ont besoin de connaître les membres de leur groupe.

Nous avons décrit les agents du système, créés à l'aide des outils offerts par Madkit ainsi que leur fonctionnement. Nous avons présenté ensuite l'interface de COLYPAN et les outils de travail collaboratif dont il dispose et qui permettent aux utilisateurs de communiquer et d'échanger leurs connaissances et leurs informations. Ces outils sont essentielles pour l'apprentissage qui sans eux ne peut avoir lieu.

Nous avons détaillé aussi les fonctionnalités qu'il offre pour le suivi des activités des utilisateurs. En effet, COLYPAN permet de récolter les traces de chaque utilisateur et de sauvegarder ces traces dans la base de données locale de ce dernier. Ces traces sont ensuite analysées et stockées dans la base de données centrale pour être consultées par tous les utilisateurs (appartenant à un même groupe). Ces fonctionnalités sont très importantes dans COLYPAN puisqu'elles permettent à chaque utilisateur d'avoir des informations sur la productivité et le niveau de collaboration de chaque membre de son groupe ainsi que sur la réalisation des activités. Ces informations sont indispensables pour prévenir le retard dans la réalisation des activités mais aussi pour identifier les membres qui sont en situation d'échec et/ou qui abandonnent. En effet, s'agissant d'un apprentissage collaboratif à distance, il est du rôle du tuteur et de l'intérêt de chaque membre du groupe de veiller à ce que les membres du groupe soient productifs et collaboratifs afin de réaliser les activités et d'achever le projet.

Nous avons terminé ce chapitre en proposant un scénario qui décrit en détail comment utiliser COLYPAN ainsi que le déroulement de l'apprentissage. Ce scénario a permis de tester le bon fonctionnement du système. En effet, nous n'avons pas pu le tester grandeur nature mais c'est dans nos perspectives à court terme.

# Chapitre VII. Conclusion et perspectives

## 7.1 Conclusion

L'origine de cette thèse est un intérêt pour les systèmes complexes et ouverts ce qui est le cas des systèmes multi-agents, des systèmes P2P ainsi que des systèmes d'apprentissage collaboratif en ligne. Les SMA se caractérisent par les interactions, l'autonomie et la distribution des agents. Cette distribution implique une distribution du contrôle et des connaissances: chaque agent possède une connaissance parcellaire et tous coopèrent pour un objectif global. Ainsi, les SMA s'adaptent bien à la conception des systèmes d'apprentissage à distance où la distribution des données, du contrôle et de l'expertise s'impose. Un autre avantage des systèmes multi-agents qui est très utile pour le système d'apprentissage est le fait de pouvoir ajouter des nouveaux agents et même changer le comportement de certains agents sans affecter toute la structure du système ce qui permet un développement incrémental du système. Les systèmes P2P permettent de relier les agents et d'établir la communication entre eux. L'utilisation des systèmes P2P apporte plusieurs avantages. En effet, cela permet d'éviter les goulots d'étranglement des ressources, de rendre le système flexible et tolérant aux pannes de serveur. Cependant, ces systèmes posent aussi des problèmes à cause de leur nature dynamique. Ainsi, nous nous sommes intéressés aux problèmes de coordination d'agents dans le contexte de systèmes P2P parce que sans coordination, les interactions d'agents peuvent devenir nuisibles et la communauté peut dégénérer rapidement en une collection chaotique d'individus. Nous présentons donc dans la section suivante nos contributions au déploiement des systèmes multi-agents sur une architecture P2P mais aussi à l'amélioration des conditions d'apprentissage collaboratif en ligne.

## 7.2 Contributions de nos travaux de thèse

La coordination d'actions d'agents est une question clé pour les systèmes multi-agents. Ainsi, plusieurs mécanismes de coordination ont été définis comme par exemple, la planification distribuée, l'anticipation, la formation de coalition. Cependant, ces mécanismes ne sont pas adaptés au contexte dynamique des systèmes P2P. Nous avons présenté quelques travaux proposant des méthodes de formation de coalition pour le contexte de P2P. Cependant, une étude récente sur l'effet de l'identité de groupe sur la formation des coalitions [Tremewan, 2010] a été faite. Le résultat principal de cette étude est que les joueurs, formant une coalition et jouant ensemble, qui appartiennent au même groupe ont plus de chance de gagner que ceux qui ne font pas partie du même groupe et ceci montre que la coordination est plus facile entre les joueurs du même groupe. Cette étude vient réconforter notre idée qui est de proposer une méthode de formation de GROUPE comme solution pour la coordination d'actions d'agents dans une application P2P. En effet dans

cette solution, nous essayons de reproduire ce qui se passe dans la réalité, en rassemblant des agents qui vont bien ensemble afin de mettre en commun leurs compétences et ainsi réaliser un but global. Ces agents auront ainsi le temps de se connaître et d'avoir une appréciation des autres agents avant que le groupe, atteignant une certaine taille, ne se divise pour former des nouveaux groupes et ceci selon les préférences des agents qui ont une idée sur les compétences mais aussi sur la sociabilité (niveau de collaboration et d'interaction) des autres agents.

Nous avons ensuite étudié quelques plateformes et travaux sur l'e-Learning et nous avons constaté qu'aucun des travaux présentés ne s'intéressent au suivi de l'apprentissage dans un environnement P2P. En effet, les travaux s'intéressant au suivi des activités des apprenants n'offrent qu'une réponse partielle. Cependant, ces outils et plateformes tels qu'ils sont ne suffisent plus pour satisfaire les nouvelles exigences que pose l'utilisation du modèle P2P dans un contexte d'apprentissage collaboratif à distance.

En effet, nous nous sommes intéressés aux nouveaux rôles de l'enseignant, des apprenants et du groupe dans le domaine d'apprentissage collaboratif à distance. Ces nouveaux rôles impliquent des besoins spécifiques que nous avons identifiés afin de trouver les fonctionnalités à ajouter à l'outil informatique pour les satisfaire. Ces fonctionnalités ont donc trois buts. Le premier est de faciliter la tâche du tuteur en satisfaisant ses besoins et en lui proposant d'apprécier les activités des apprenants et de suivre leur progression dans l'apprentissage ainsi que leur niveau de collaboration. Le deuxième est d'éviter l'isolement des apprenants et de les motiver et le dernier est d'encourager le travail collaboratif et l'utilisation des outils le permettant tels que le forum, le chat, la messagerie instantanée. Nous sommes convaincus que c'est la combinaison de tous ces buts qui améliore la situation d'apprentissage collaboratif à distance et permet sa réussite.

Bien que ce que nous proposons soit convenable pour tous les domaines d'apprentissage collaboratif voire du travail collaboratif, nous avons choisi de prendre comme application l'apprentissage collaboratif de gestion de projet selon la méthode MAETIC. Nous avons donc adopté une démarche génie logiciel pour l'analyse, la conception et le développement de notre système COLYPAN. Ainsi, nous avons choisi la méthodologie Aalaadin qui repose sur le modèle Agent/Groupe/Rôle qui a l'avantage d'assurer la généricité, ce qui nous a permis d'implémenter nous-mêmes les comportements attendus par le rôle et de décrire explicitement les rôles que les agents sont susceptibles d'utiliser. Nous l'avons ensuite implémentée avec la plateforme Madkit qui est une plateforme reposant sur le modèle AGR. le choix de cette plateforme était motivé par le fait que Madkit est particulièrement destinée à des SMA fondés sur des critères organisationnels et qu'elle fonctionne en mode distribuée de manière transparente à partir d'une architecture P2P sans nécessiter de serveur dédié.

Ainsi, COLYPAN est un système d'apprentissage collaboratif de gestion de projet qui implémente la méthode MAETIC et où l'apprentissage se fait par projet et en groupe. COLYPAN implémente aussi la méthode de formation de groupe que nous avons proposé et permet de coordonner les actions d'agents en tenant compte de la nature dynamique des

systèmes P2P. Le tuteur a pour rôle de guider les apprenants et de faciliter leur apprentissage, le groupe a pour rôle de motiver les apprenants, de créer un lien social qui empêche le sentiment d'isolement. Quant aux apprenants, ils apprennent en interagissant avec les autres. Cependant, COLYPAN est aussi un système multi-agents à architecture P2P où des agents autonomes interagissent et communiquent afin de réaliser un but commun c'est-à-dire satisfaire les exigences des utilisateurs (apprenant ou tuteur).

## 7.3 Perspectives

Cette thèse a traité trois domaines qui constituent chacun un domaine très vaste. Ainsi, notre travail constitue une première étape pour répondre aux problèmes de coordination des systèmes multi-agents. En effet, pour cette problématique, il est intéressant d'étudier les autres mécanismes de coordination (négociation, résolution distribuée des problèmes) et de les adapter au contexte spécifique de P2P.

En plus, l'apprentissage à distance est un champ pluridisciplinaire allant des sciences humaines et sociales aux sciences dures, dans lequel des utilisateurs au profil hétérogènes interagissent. Nous pensons qu'il sera intéressant dans le domaine d'apprentissage collaboratif de proposer des méthodes pour l'analyse des contenus de chat et des forums.

Cependant à court terme, nous nous intéressons à l'expérimentation de notre système COLYPAN. En effet, nous avons fait des tests de validation mais nous n'avons pas pu faire des expérimentations réelles et à grande échelle. Toutefois, il est important de vérifier à quel point nos propositions ont réussi d'abord à faciliter la tâche du tuteur, à combattre ensuite l'isolement de l'apprenant et à le motiver et finalement à encourager l'apprentissage collaboratif.

# Liste des figures

*Figure II-1 Les différentes étapes et les modèles de MaSE* ..... 29

*Figure II-2 Le modèle organisationnel* ..... 33

*Figure II-3 Les concepts du modèle Aalaadin* ..... 34

*Figure II-4 Les diagrammes de structure organisationnelle dans Aalaadin* ..... 35

*Figure II-5 Le processus de développement de Gaia et les modèles manipulés* ..... 37

*Figure III-1 Architecture de la plateforme JXTA* ..... 54

*Figure III-2 Architecture de la plateforme JADE* ..... 55

*Figure III-3 Interface de Madkit* ..... 56

*Figure III-4 Architecture de la plateforme Madkit* ..... 57

Figure IV-1 Apports et inconvénients du P2P ..... 75

*Figure V-1 Besoins de l'apprenant* ..... 85

*Figure V-2 Cycle pédagogique de MAETIC* ..... 89

*Figure V-3 Structure organisationnelle de COLYPAN* ..... 95

*Figure V-4 Déploiement des agents* ..... 96

*Figure V-5 Notification des retardataires* ..... 97

*Figure V-6 Communication inter et intra groupes* ..... 98

*Figure VI-1 Les agents dans COLYPAN* ..... 103

*Figure VI-2 Interface de COLYPAN* ..... 104

*Figure VI-3 Appel pour former un groupe* ..... 106

*Figure VI-4 Rejoindre un groupe* ..... 106

Figure VI-5 *Interaction des agents dans COLYPAN* ..... 109

*Figure VI-6 Suivi de l'apprenant* ..... 111

*Figure VI-7 Suivi du groupe* ..... 111

# Liste des tableaux

Tableau II-1 Comparatif des différentes méthodes ..... 40

Tableau III-1comparatif des plateformes ..... 60

Tableau IV-1Comparatif de l'existant ..... 77

Tableau V-1 Rôles de l’enseignant : classique vs à distance ..... 82

Tableau V-2 Rôles et besoins du tuteur ..... 83

Tableau V-3 Exemple de LB et LP des agents ..... 86

# Bibliographie


[Adams et Hamm, 1990] D. M. Adams et M. E. Hamm, « Cooperative learning; critical thinking and collaborations across the curriculum ». Springfield Ill., U.S.A.: C. C. Thomas, 1990.

[Aknine, 2007] S. Aknine et L. Arantes, « Scalable Coalition Formation Method for Large-Scale Systems ». IAT'07- IEEE/ACM International Conference on Intelligent Agent Technology, pages: 520-523, 2007.

[Arcangeli et Leriche, 2006] J. P. Arcangeli et S. Leriche, « Un framework à composants et agents pour les applications répartis ». Dans : Revue des sciences et Technologies de l'information, série L'Objet, Lavoisier, Vol. 12, N. 4, p. 103- 132, décembre 2006.

[Basque, 1999] J. Basque, « Inforoute et technologie éducative à l'aube de l'an 2000 ». Actes du XIIe Colloque du CIPTE, Télé-université, 1999.

[Bellefemine, 2004] F. Bellefemine, G. Caire, T. Trucco et G. Rimassa, « JADE programmer's guide (JADE 3.2) ». Rapport, TILAB S.p.A, 2004.

[Belmonte, 2006] M.V. Belmonte, R. Conejo, M. Dıaz et J.L. Perez-de-la-Cruz, «Coalition formation in P2P file sharing systems». R. Marin et al. (Eds.): CAEPIA 2005, LNAI 4177, pp. 153–162, 2006.

[Bond, 1990] A. H. Bond, « Distributed Decision Making in Organization ». IEEE Transactions on systems, man and cybernetics Conference, November 1990.

[Bonner, 1988] J. Bonner, « Implications of cognitive theory for instructional design-Revisited ». Educational Communications and Technology Journal, 36 (1), 3-14, 1988.

[Breu et al, 2005] S. Breu, F. Lehner et H. Nösekabel, « LL2 (Learners Learn 2gether), a P2P-based e-Learning System ». Microlearning International Conference, Innsbruck, Autriche, juin 2005.

[Cabani et al, 2010] A. Cabani, W. Alshabi et M. Itimi, « Organisation du partage des ressources dans un réseau P2P par la coopération des pairs ». Proceedings du 8ème Conférence Internationale de Modélisation et SIMulation (MOSIM'10), Hammamet, Tunisie, mai 2010.

[Caillou et al, 2001] P. Caillou, S. Aknine, et S. Pinson, « Méthode de formation et de restructuration dynamique de coalitions d'agents fondée sur l'optimum de Pareto ». Dans Proc. 9èmes Journées Francophones d'Intelligence Artificielles et Systèmes Multi-Agents (JFIAD), 2001.

[Cardon, 1998] A. Cardon, « Modélisation des systèmes adaptatifs par agents : vers une analyse conception orientée agents », rapport technique 1998.011, lip6, université Paris VI, 1998.

[Cavalier, Klein et Cavalier, 1995] J.C. Cavalier, J.D. Klein et F. J. Cavalier, « Effects of cooperative learning on performance, attitude, and group behaviors in a technical team environment ». ETR&D, 43(3), 61-71. Cerratto, R. & Belisle, C. (1995). Reframing learning in CSCL Environments, 1995.

[Chaib-draa, 1995] B. Chaib-draa, « Industrial applications of distributed AI ». Communications of the ACM, 38(11):49-53, 1995.

[Chaib-draa, 1996] B. Chaib-draa, « Interaction between agents in routine, familiar and unfamiliar situations ». International Journal of Intelligent and Cooperative Information Systems, 1(5):7-20, 1996.

[Chaib-draa et al, 2001] B. Chaib-draa, I. Jarras et B. Moulin, « Systèmes multiagents : Principes généraux et applications », Editions J.P. Briot, Y. Demazeau, Agent et systèmes multiagents, Hermès, 2001.

[Chaib-draa et Demolombe, 2002] B Chaib-draa et R. Demolombe, Dans : Information-Interaction-Intelligence, 2002.

[Charif, 2007] Y. Charif, « Chorégraphie dynamique de services basée sur la coordination d'agents introspectifs ». Thèse de doctorat, Université Pierre et Marie Curie, 2007.

[Chasseneuil, 2000] Collectif de Chasseneuil 2000. Conférence de consensus Formations Ouvertes et à Distance. L'accompagnement pédagogique et organisationnel. Paris, 27, 28 & 29 mars 2000.

[Chevrier, 1992] V. Chevrier, « Coordination et Structuration des échanges par négociation dans les systèmes multi-agents ». Journée Systèmes multi-agents PRC-GRD. Intelligence Artificielle, Nancy, Décembre 1992.

[Cohen, 2003] B. Cohen, « Incentives Build Robustness in BitTorrent ». Proceedings of the 1st Worshop on Economics of Peer-to-Peer Systems, Berkeley, USA, 2003.

[Coleman et al, 1994] D. Coleman, P. Arnold, S. Bodoff, C. Dollin, H. Gilchrist, F. Hayes, et P. Jeremaes, « *Object-Oriented Development: The Fusion Method* », Prentice Hall, 1994.

[Collinot et Drogoul, 1998] A. Collinot et A. Drogoul, « La méthode de conception multi-agent CASSIOPEE : application à la robotique collective ». *Revue d'intelligence artificielle*, 12(1) :125–147, 1998

[Collinot et Drogoul, A. Collinot et A. Drogoul, « Agent Oriented Design of a Soccer

1996] Robot Team ». *Dans :* M. TOKORO, éditeur : *Second International Conference on Multi-Agent Systems (ICMAS'96), Nara, Japan*, pages 41–47. AAAI Press, 1996.

[Cossentino et Potts, 2002] M. Cossentino et C. Potts, « A CASE tool supported methodology for the design of multiagent systems ». *Dans: The Proceedings of the International Conference on Software Engineering Research and Practice (SERP'02)*, 2002.

[Cossentino et al, 2005] M. Cossentino, S. Gaglio, L. Sabatucci, and V. Seidita. « The PASSI and Agile PASSI MAS Meta-Models Compared with a Unifying Proposal». Dans: *Proceedings of the CEEMAS' 05 Conference*, pages 183–192., Budapest, Hungary, septembre 2005.

[Deloach, 2001] S. Deloach, « Analysis and design using MaSE and agentTool ». The 12th Midwest Artificial Intelligence and Cognitive Science Conference, 2001.

[Deloach, 2002] S. Deloach, « *Modeling organizational rules in the multiagent systems methodology* ». Dans: Proceedings of the 15th Canadian Conference on Artificial Intelligence Calgary, Alberta, Canada, May 27-29, 2002.

[Deloach et al, 2001] S. A. Deloach, M. F. Wood, et C. H. Sparkman, « Multi-agent Systems Engineering ». The International Journal of Software Engineering and Knowledge Engineering, 2001, Volume 11 no. 3, pp. 231-258.

[Demazeau, 1995] Y. Demazeau, « From Interactions to Collective Behaviour in Agent-Based Systems ». Dans: Proceedings of the 1st European Conference on Cognitive Science. Saint-Malo, France, 1995.

[Dewey, 1922] J. Dewey, « L'école et l'enfant » (2 ed.). Neuchâtel: Delachaux et Niestlé, 1922.

[Dillenbourg, 1999] P. Dillenbourg, « Collaborative-learning: Cognitive and Computational Approaches ». Pergamon Press, 1999.

[Di Vesta et Rieber, 1987] F. J. Di Vesta et L. P. Rieber, « Characteristics of cognitive engineering: The next generation of instructional systems ». Educational Communications and Technology Journal, 35 (4), 213-230, 1987.

[Doniec et al, 2006] A. Doniec, R. Mandiau, S. Piechowiak et S. Espié, « L'anticipation comme modèle d'interaction : application à la coordination multi-agent en simulation ». 15ème congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle, 2006.

[Drogoul, 2000] A. Drogoul, « Systèmes multi-agents situés ». Mémoire

d'habilitation à diriger des recherches, Université Paris 6, 2000.

[Duffy et Cunningham, 1996] T. M. Duffy et D. J. Cunningham, « Constructivism: Implications for the design and delivery of instruction ». Dans: D. H. Jonassen (Ed.), Handbook of Research for Educational Communications and Technology. New York: Macmillan Library Reference/AECT, 170-198, 1996.

[Durfee et Lesser, 1989] E. H. Durfee et V. R. Lesser, « Negotiating task decomposition and allocation using partial global planning ». Dans: Distributed Artificial Intelligence Volume II, pages 229-244. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA, 1989.

[Durfee, 2001] E.H. Durfee, Distributed Problem Solving and Planning. Pages 118–149, 2001.

[El-Fallah et al, 1996] A. El-Fallah Seghrouni et S. Haddad, « A coordination algorithm for multiagent planning ». Dans: Proceedings of International Conference on Multiagent Systems-1996 (ICMAS'96), AAAI, Press, Kyoto, Japan, 1996.

[Feenberg, 1989] A. Feenberg, « The written world». Dans: Mindweave: Communication, Computers and Distance Education, Robin Mason and Anthony Kaye (eds). Oxford, U.K.: Pergamon Press, 1987.

[Ferber, 2009] J. Ferber, « MadKit pas à pas : Démarrage et prise en main du logiciel MadKit ». Manuel d'utilisation de MadKit, Université de Montpellier II, avril 2009.

[Ferber, 2003] J. FERBER, O. GUTKNECHT et F.MICHEL, « From Agents to Organizations: An Organizational View of Multi-agent Systems ». Dans: P. GIORGINI, J. P. MÜLLER et J. ODELL, éditeurs: Agent-Oriented Software Engineering IV, 4th International Workshop, AOSE 2003, Melbourne, Australia, July 15, 2003, Revised Papers, volume 2935 de Lecture Notes in Computer Science (LNCS), pages 214–230. Springer-Verlag, 2003.

[Ferber, 1999] J. Ferber, « Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence ». Addison Wesley, London, 1999.

[Ferber, 1998] J. FERBER et O. GUTKNECHT, « Aalaadin: a meta-model for the analysis and design of organizations in multi-agent systems ». Dans : DEMAZEAU, Y., éditeur : 3rd International Conference on Multi-Agent Systems, pages 128–135, Paris, 1998. IEEE.

[Ferber, 1995] J. FERBER, « Les systèmes multi-agents : Vers une intelligence collective ». InterEditions, 1995.

[Ferber, 1989] J. Ferber, « Objets et Agents : Une étude des structures e représentation et communication en intelligence artificielle ». Thèse de doctorat, 1989.

[Finkle et Torp, 1994] S. L. Finkle et L. L. Torp, « Introductory documents». Illinois: IMSA Center for Problem-Based Learning, Illinois Mathematics and Science Academy, 1994

[Fleury, 1994] M. FLEURY, « Implications de certains principes de design pour le concepteur de systèmes multimédias interactifs ». Éducatechnologiques, 1 (3), 63-90, 1994

[Fougères, 2010] A.J. Fougères, "Vers un Système de médiation pour les systèmes coopératifs ». Mémoire d'habilitation à diriger des recherches, Université de Technologie de Belfort-Montbéliard, 2010.

[Fung, 1996] P. Fung, « Issues in Project-Based Distance Learning in Computer Science». Journal of Distance Education/Revue de l'enseignement à distance, 1996.

[Gardner, 1993] H. Gardner, « Histoire de la révolution cognitive: la nouvelle science de l'esprit ». Paris, 1993.

[Gasser, 1990] L. Gasser, «Social Conception of Knowledge and Action». Technical Report ACT-IA-355-90, MCC, Octobre 1990.

[Genin, 2010] T. Genin, « Stratégies de formation de coalitions dans les systèmes multi-agents ». Thèse de doctorat, Université Pierre et Marie Curie, 2010.

[Glisse, 2005] B. Glisse, « Schémas d'interactions dans les systèmes multi-agents ». 7ème Rencontres de Théo Quant, Besançon, Janvier 2005.

[Greer et al, 2001] J. Greer, G. McCalla, J. Vassileva, R. Deters, S. Bull et L. Kettel, «Lessons Learned in Deploying a MultiAgent Learning Support System: The IHelp Experience». Dans: Artificial Intelligence in Education, AIED'2001, p. 410421, San Antonio, Texas, USA, 2001.

[Guéraud et al, 2007] V. Guéraud, S. Michelet et J-M Adam, « Suivi de classe à distance : propositions génériques et expérimentation en électricité ». Environnements Informatiques pour l'Apprentissage Humain, Lausanne 2007.

[Guessoum, 1996] Z. Guessoum, « Un environnement de développement et de conception de systèmes multi-agents ». Thèse de doctorat. Université Paris6, France, 1996.

[Guessoum et al, 2002] Article collectif du GDI asa, « Une comparaison des plates-formes

multi-agents. In Systèmes multi-agents : des Théories Organisationnelles aux Applications Industrielles ». René Mandiau, Emmanuelle Grislin- Le Strugeon, André Péninou (eds.), Hermès, pp. 207- 240, mai 2002.

[Guessoum, 2003] Z. Guessoum, « Modèles et architectures d'agents et de systèmes multi-agents adaptatifs ». Rapport de soutenance de HDR, Université Pierre et Marie Curie, 2003.

[Gueye, 2005] O. Gueye, « Instrumentation des activités du tuteur en ligne le cas de CROISIERES, dispositif de Formation Ouverte et à Distance en langues ». Thèse de doctorat. Université du Maine, 2005.

[Gutknecht, 1999] O. Gutknecht et J. Ferber, « Vers une méthodologie organisationnelle de conception ». Dans : M.-P. Gleizes et P. Marcenac, éditeurs : Ingénierie des systèmes multi-agents – Actes des 7èmes JFIADSMA, pages 93–104. Hermès, 1999. 20

[Gutknecht et Ferber, 2000] O. Gutknecht et J. Ferber, « Madkit : une architecture de plate-forme multi-agent générique ». Rapport de recherche, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRM), Université de Montpellier II, 2000.

[Guyoumard et al, 1999] D. Guyomard, J-C. Soulie, P. Marcenac et M. Larue, « Mise en place d'un système multi-agents destiné à la simulation de dynamiques comportementales spatiales environnement /ressource. appliqué à l'espadon Xiphias Gladius dans le Sud-Ouest de l'Océan Indien ». Acte du 4ème Forum de l'Association Française d'Halieumétrie Les Espaces de l'Halieutique. Rennes. Editions de l'IRD, juin 1999.

[Heecheol et al, 2000] J. Heecheol, C. Petrie et M. Cutkosky, « JATLite: A Java Agent Infrastructure with Message Routing ». Internet Computing, Mars 2000.

[Henri et Lundgren-Cayrol, 1997] F. Henri et K. Lundgren-Cayrol, « Apprentissage collaboratif à distance, téléconférence et télédiscussion ». Montréal : Centre de recherche LICEF, Télé-université, 1997.

[Henri et Lundgren-Cayrol, 1998] F. Henri et K. Lundgren-Cayrol, « Apprentissage collaboratif et nouvelles technologies ». Centre de recherche LICEF, Télé-université, 1998

[Henri et Lundgren-Cayrol, 2001] F. Henri et K. Lundgren-Cayrol, « Apprentissage collaboratif à distance : pour comprendre et concevoir les environnements d'apprentissage virtuels ». Sainte-Foy (Québec, Canada) : Presses de l'Université du Québec, 181p, 2001.

[Hikolo, 2003] A.M. Hikolo , « Analyse, conception, spécification et développement d'un système multi-agents pour le soutien des activités en formation à distance ». Thèse de doctorat. Université de Franche-Comté, France, 2003.

[Honebein, Duffy et Fishman, 1993] P. C. Honebein, T. M. Duffy et B. J. Fishman: « Constructivism and the design of learning environments: Context and authentic activities for learning». Dans T. M. Duffy, J. Lowick, et D. H. Jonassen (Eds.), Constructive Learning. Berlin: Springer-Verlag, 87-108, 1993

[Houdé et al, 1998] O. Houdé, et al. Vocabulaire des sciences cognitives. Paris: Presses Universitaires de France, 1998.

[Jarras et Chaib-Draa, 2002] I. Jarras et B. Chaib-Draa, « Aperçu sur les systèmes multi-agents ». Série Scientifique, Montréal, 2002.

[Jennings, 1996] N. R. Jennings, « Coordination Techniques for Distributed Artificial Intelligence». Foundations of Distributed Artificial Intelligence, Editions G. M. P. O'Hare et N. R. Jennings, Wiley, 1996.

[Jennings, 1998] N. R. Jennings, K. Sycara et M. Wooldridg, « A Roadmap of Agent Research and Development». Dans: "*Autonomous Agents and Multi-Agent Systems Journal*", Kluwer Academic Publishers, Boston, 1998.

[Jennings, 1999] N. R. Jennings, « Agent-Based Computing: Promise and Perils ». Dans: Ijcai'99, pages 1429–1436, Springer-Verlag, Heidelberg, Stockholm, 31 juillet–6 août 1999.

[Jennings et Wooldridge, 1998] N. R. Jennings et M. Wooldridge, « Applications of Intelligent Agents». Dans: Agent Technology: Foundations, Applications, and Markets (eds. N. R. Jennings and M. Wooldridge) Springer-Verlag: Berlin, Germany. pp. 3-28, 1998.

[Johnson et Thomas, 1994] S. D. Johnson et R. G. Thomas, « Implications of cognitive science for instructional design in technology education». Journal of Technologies Studies, 20(1), 33-45, 1994.

[Jonassen, 1990] D. H. Jonassen, « Toward a constructivist view of instructional design». Educational Technology, 30(9), 32-34, 1990.

[Jonassen, 1991] D. H. Jonassen, « Objectivism versus constructivism: Do we need a new philosophical paradigm? ». Educational Technology Research & Development, 39(3), 5-14, 1991.

[Jonassen, 1994] D. H. Jonassen, « Toward a constructivist design model». Educational Technology, 34(4), 34-37, 1994.

[Khandaker et al, 2011] N. Khandaker, L. -K. Kiat, L. D. Miller et H. Jiang, « Lessons Learned from comprehensive deployments of multi-agent CSCL applications I-MINDS and classroom wiki». IEEE Transactions on Learning Technologies, Volume 4, Number 1 (TLT4(1)), pages: 47-58, Janvier 2011.

[Kinny, 1996] D. Kinny, M. Georgeff et A. Rao, « A Methodology and Modelling Technique for Systems of BDI agents». Dans: W. Van de VELDE et J. W. PERRAM, éditeurs : Agents Breaking Away : Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a MultiAgent World, volume 1038 de Lecture Notes in Artificial Intelligence (LNAI), pages 51–71. Springer-Verlag, 1996.

[Klein, 2009] F. Klein, « Contrôle d'un système multi-agents réactif par modélisation et apprentissage de sa dynamique globale ». Rapport de thèse, Université de Nancy 2, Décembre 2009.

[Labrou et Finin, 1997] Y. Labrou et T. Finin, « A Proposal for a new KQML Specification ». Rapport technique, 1997.

[Lajoie, 2007] Gilles Lajoie, « Recherches en modélisation urbaine ». Rapport de soutenance de HDR, Université de La Réunion, 2007.

[Lave et Wenger, 1991] J. Lave et E. Wenger, « Situated Learning: Legitimate Peripheral Participation". Cambridge: Cambridge University Press, 1991.

[Le Bars, 2003] M. Le Bars, « Un simulateur multi-agent pour l'aide à la décision d'un collectif: Application à la gestion d'une ressource limitée agro-environnementale ». Thèse de doctorat, Université Paris IX-Dauphine, mai 2003.

[Lebow, 1993] D. Lebow, « Constructivist values for instructional systems design: Five principles toward a new mindset ». Educational technology Research & Development, 41(3), 4-16, 1993.

[Leriche, 2006] S. Leriche, « Architectures à composants et agents pour la conception d'applications réparties adaptables ». Thèse de doctorat, Université Toulouse III, 2006.

[Linard, 2000] M. Linard, « L'autonomie de l'apprenant et les TIC ». Dans : 2ème journée réseaux humains / réseaux technologiques : Présence à distance. Poitiers, 2000.

[Lopes et Botelho, 2008] A. L. Lopes et L. M. Botelho, « Improving Multi-Agent Based Resource Coordination in Peer-to-Peer Networks ». Dans : Journal of Networks, Volume 3, n° 2, pages 38-47, février 2008.

[Mahdi et Attia, 2010] H. Mahdi et S. Attia, « Developing and Implementing a Multi-Agent System for Collaborative E-learning ». Dans : E-learning,

Maja Jakobovic (eds.), ISBN : 978-953-7619-95-4, InTech, 2010.

[Mansour, 2007] S. Mansour, « Un modèle de gestion distribuée de groupes ouverts et dynamiques d'agents mobiles ». Thèse de doctorat. Université de Pau et des Pays de l'Adour, 2007.

[Marchat, 2001] H. Marchat, « Kit de conduite de projet ». Paris: Organization editions, 2001.

[Mari et al, 2008] M. Mari, A. Poggi, M. Tomaiuolo et P. Turci, «Enhancing Multi-Agent Systems with Peer-to-Peer and Service-Oriented Technologies». Dans: Jung, Michel, Ricci & Petta (eds.): AT2AI-6 Working Notes, From Agent Theory to Agent Implementation, 6th International Workshop, AAMAS 2008, Estoril, Portugal, 13 mai 2008.

[Mazyad et al, 2008] H. Mazyad, I. Tnazefti-Kerkeni et H. Ajroud, «Multi-Agent System Architecture for Managing the Coordination in Peer-To-Peer Systems », 3rd International Conference on Information & Communication Technologies: from Theory to Applications (ICTTA'08), Damas, Syrie, Avril 2008.

[Mazyad et al, 2010a] H. Mazyad, I. Tnazefti-Kerkeni et B. Talon, « Multi-agents Approach for Modeling a Collaborative Learning System ». The Ninth IASTED International Conference on IASTED Web-based Education (WBE 2010), Sharm el-Sheikh, Egypte, Mars 2010.

[Mazyad et al, 2010b] H. Mazyad et I. Tnazefti-Kerkeni, « Colypan: A peer-to-peer Architecture for Collabarative Learning System ». Eighth German Conference on Multi Agents System Technologies (MATES'10), Leipzig, Allemagne, septembre 2010.

[Mazyad et al, 2010c] H. Mazyad et I. Tnazefti-Kerkeni, « An opened Agent-oriented System for Collabarative Learning ». 7th International Conference on Multimedia & Network Information Systems (MISSI), Wroclaw, Poland, septembre 2010.

[Merrill, Jones et Li, 1992] M. D. Merrill, M. K. Jones, et Z. Li, « Instructional transaction theory: Classes of transaction ». Educational Technology, 32(6), 12-26, 1992.

[Michel et al, 2005] F. Michel, G. Daniel, J. Ferber et D. Phan, « Projet intégration Moduleco/Madkit: premiers résultats ». Joint Conference on Multi-Agent Modelling for Environmental Management, SMAGET 05, Bourg-Saint-Maurice, Les Arcs, FRANCE, mars, 2005.

[Milgrom et al, 2001] E. Milgrom, P. Chainho, Y. Deville, R. Evans, P. Kearney et P. Massonet, « MESSAGE - Methodology for Engineering Systems of

Software Agents (Final Guidelines for the Identification of Relevant Problem Areas where Agent Technology is Appropriate». Technical report, EURESCOM, 2001.

[Milojicic et al, 2002] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins et Z. Xu, « Peer-to-Peer Computing ». HP Laboratories Palo Alto, report HPL-2002-57.

[Moulin et Chaib-Draa, 1996] B. Moulin et B. Chaib-Draa, « An overview of distributed artificial intelligence ». Dans: G.M.P.O'Hare and N.R. Jennings, editors, Foundations of Distributed AI, pages 3-54. John Wiley & Sons: Chichester, England, 1996.

[Muller, 2000] J.–P. Muller, « Modélisation organisationnelle en systèmes multi-agents ». Dans : Septième école d'été de l'ARCo, Bonas, 2000.

[Nwana et al, 1996] H. Nwana, L. Lee and N. Jennings, « Coordination in software agent systems », BT Laboratories internal report, (1994).

[Odell et al, 2000] J. Odell, P. Van Dyke et B. Bauer, « 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, 2000.

[Oubahssi, 2005] L. Oubahssi, « Conception de plates-formes logicielles pour la formation à distance, présentant des propriétés d'adaptabilité à différentes catégories d'usagers et d'interopérabilité avec d'autres environnements logiciels ». Thèse de doctorat. Université de Paris V, 2005.

[Paquette et al, 1997] G. Paquette, C. Aubin et F. Crevier, « Design and Implementation of Interactive TeleLearning Scenarios ». 18th World Conference of the International Council on Distance Education, Pennsylvania State University, June 2 - 6, 1997.

[Pesty et al, 2003] S. Pesty, C. Webber et N. Balacheff, « Baghera: une architecture multi-agents pour l'apprentissage humain ». Cognitique, (P. Aniorte, S. Gouarderes eds), Cepadeus Edition, Toulouse, 2003.

[Pommier et Bourdon, 2009] H. Pommier et François Bourdon, «Agents mobiles et réseaux pair-à-pair: Vers une gestion sécurisée de l'information répartie». Revue d'intelligence artificielle, ISSN 0992-499X, volume 23, n° 5-6, pages 697-718, 2009.

[Rao et al, 1995] A. S. Rao et M. P. Georgeff, « BDI-agents: from theory to practice », Proceedings of the First Intl. Conference on Multiagent Systems (San Francisco), 1995.

[Resnick, 1987] L. B. Resnick, « Education and learning to think ». Washington, D. C.: Academy Press, 1987.

[Ripeanu et al, 2002] M. Ripeanu, A. Iamnitchi et I. Foster, « Mapping the Gnutella Network ». Internet Computing, IEEE, Vol.6, Issue 1, p. 50-57, 2002.

[Shehory et al, 1998] O. Shehory et S. Kraus, « Methods for Task Allocation via Agent Coalition Formation ». Artificial Intelligence, 101(1–2):165–200, 1998.

[Smith, 1988] R. G Smith, « the Contract-Net protocol: high-level communication and control in a distributed problem solving ». Dans: Distributed Artificial Intelligence, Morgan Kauffmann Publishers, 1988.

[Soh et al, 2010] L.-K. Soh, N. Khandaker et H. Jiang, « I-MINDS: A Multiagent System for Intelligent Computer-Supported Collaborative Learning and Classroom Management». Dans: Internainal Journal of Artificial Intelligence in Education, volume 18, number 2, pages: 119-151, 2008.

[Spiro et al, 1991] R. J. Spiro, P. J. Feltovich, M. J. Jacobson et R. L. Coulson, « Cognitive Flexibility, Constructivism, and Hypertext: Random Access Instruction for Advanced Knowledge Acquisition ». Dans: ill-Structured Domains. Educational Technology, 31(5), 24-33, 1991.

[Sycara, 1998] K. P. Sycara, « Multi-agents Systems ». AI magazine Volume, 19(2), 1998.

[Talon et Leclet, 2008] B. Talon et D. Leclet, « Dispositif pédagogique pour un apprentissage de savoir-faire », Revue Internationale des technologies en pédagogie universitaire, Volume 5, N2, pp 58-74, 2008.

[Teutsh et al, 2004] P. Teutsch, J. Bourdet et O. Gueye, « Perception de la situation d'apprentissage par le tuteur en ligne ». Dans : Conférence TICE 2004, pp.59-66, Compiègne, 2004.

[Tremewan, 2010] J. Tremewan, « Group Identity and Coalition Formation: Experiments in a three-player divide the dollar Game ». CEPREMAP Working Papers (Docweb) 1020, CEPREMAP, Juin, 2010.

[Treuil et al, 1997] J. P.Treuil et C. Mullon, « Expérimentation sur mondes artificiels : pour une réflexion méthodologique ». Editions Scientifiques et Médicales ELSEVIER, 1997.

[Unesco, 2004] Unesco, « Technologies de l'information et de la communication en éducation : un programme d'enseignement et un cadre pour la formation continue des enseignants ». Organisation des nations unies pour l'éducation la science et la culture, division de l'enseignement supérieur, Unesco 2004.

[Vassileva et al, 2003] J. Vassileva, G. McCalla et J. Greer, « MultiAgent MultiUser Modeling ». User Modeling and UserAdapted Interaction, Special Issue on user Modelling and Intelligent Agents, Vol. 3:(1), p.179210, 2003.

[Vauvert, 2002] G. Vauvert et A. El Fallah-Seghrouchni, « Formal Specification of Opinions Applied to the Consensus Problem". Dans: Proceeding of the 8th Iberoamerican Conference on Artificial Intelligence. (IBERAMIA'02), 2002.

[Vernadet et Azena, 1992] F. Vernadet et P. Azena, « Prototypage de systèmes d'agents communicants ». Journée Systèmes Multi-Agents PRC-GRD. Intelligence Artificielle, Nancy, Décembre 1992.

[Von Martial, 1990] F. von Martial, «Interactions among autonomous planning agents ». Dans Decentralized AI - Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-89). Elsevier Science Publishers B.V.: Amsterdam, Pays Bas, 1990.

[Watson, 1930] J.B. Watson, « Behaviorism». University of Chicago Press, 1930.

[Wilson, 1992] B.G. Wilson, « The Post modern Paradigm». Constructivism and instructional design: Some Personal reflections, 1992.

[Wilson et al, 2004] S. Wilson, B. Olivier, S. Jeyes, A. Powell et T. Franklin, « A Technical Framework to Support e-Learning ». Technical report, JCLT, 2004.

[Winn et Snyder, 1996] W. Winn, et D. Snyder, « Cognitive perspectives in psychology ». Dans: D. H. Jonassen (Ed.), Handbook of Research for Educational Communications and Technology (pp. 112-142). New York: Macmillan Library References/AECT, 1996.

[Wooldridge, 2000] M. J. Wooldridge, « The Computational Complexity of Agent Design Problems ». Dans: Fourth International Conference on MultiAgent Systems, icmas'2000, Boston, Massachusetts (USA), 10–12 juillet 2000.

[Wooldridge, 2001] M. Wooldridge, P. Ciancarini, « Agent-oriented software engineering: the state of the art ». Editeurs: P. Ciancarini and M. Wooldridge, Agent-Oriented Software Engineering. Springer-

Verlag Lecture Notes dans AI Volume, Janvier 2001.

[Wooldridge, 2002] M. Wooldridge, « An Introduction to Multiagent Systems ». Editeurs: John Wiley & Sons Ltd, England, ISBN 047149691X, 2002.

**Une Approche Multi-agents à Architecture P2P pour l'Apprentissage Collaboratif**

**Résumé** Les Systèmes multi-agents ou SMA proposent une approche originale de conception de systèmes intelligents et coopératifs. Ils se caractérisent par la distribution du contrôle global du système et par la présence d'agents autonomes évoluant dans un environnement partagé et dynamique. De plus, il existe plusieurs interdépendances entre les buts des agents, leurs capacités et les ressources qu'ils utilisent, donc afin d'éviter d'éventuels conflits, de favoriser la synergie des activités des agents et de partager les ressources de l'environnement commun, il est important que les agents coordonnent leurs actions.

Nous nous sommes intéressés, dans le cadre de cette thèse, au déploiement des systèmes multi-agents sur une architecture pair à pair (Peer-to-Peer ou P2P) et ceci dans le but d'établir la communication entre ces agents et les relier entre eux. Cependant, du fait de la nature dynamique des systèmes P2P où chaque pair peut apparaître et disparaître à tout moment, des nouveaux problèmes se posent pour la coordination d'agents nécessitant des mécanismes de coordination adaptés au contexte spécifique des P2P. Pour cela, nous avons proposé une méthode de formation de groupe comme solution à ces problèmes. Nous nous sommes intéressés ensuite à l'application des systèmes multi-agents à architecture P2P au domaine d'apprentissage collaboratif en ligne où des apprenants contribuent aux apprentissages du groupe, et en retour, le groupe contribue à ceux des apprenants et c'est la cohérence du collectif qui permet d'atteindre l'objectif. Cependant, l'apprentissage collaboratif à distance implique des nouveaux rôles pour l'enseignant ainsi que pour les apprenants. Il est donc essentiel de définir ces rôles pour identifier les besoins qui en découlent pour pouvoir intégrer à l'outil informatique des fonctionnalités afin de satisfaire ces besoins. En effet, il est essentiel de fournir aux enseignants et apprenants la possibilité d'avoir des informations sur la progression de leur apprentissage ainsi que sur les niveaux de collaboration et de sociabilité de chaque apprenant et du groupe. Enfin, nous avons proposé, comme application de nos travaux, un système appelé COLYPAN (COllaborative Learning sYstem for Project mANagment) conçu pour l'apprentissage à distance et de façon collaborative de la gestion des projets.

**Mots clés :** SMA, P2P, Coordination, Formation de groupes, Apprentissage collaboratif

**A Multi-agent Approach based on P2P Architecture for Collaborative Learning**

**Abstract** Multi-Agents systems (MAS) propose an original approach to design intelligent and cooperative systems. They are characterized by the distribution of the overall system control and the presence of autonomous agents operating in a shared and dynamic environment. In addition, there are many interdependencies between: agents' goals, their abilities and used resources. So, in order to avoid possible conflicts, promote synergy of agents activities and share resources of the common environment, it is important that the agents coordinate their actions.

We are interested, in the context of this thesis, in the deployment of multi-agents systems on Peer-to-Peer (P2P) networks in order to establish communication between these agents. However, because of the dynamic nature of P2P systems where each peer may appear and disappear at any time, new problems arise concerning the coordination of agents. Thus, coordination mechanisms adapted to the specific context of P2P are required. For that, we have proposed a group formation method to solve these problems. Then, we were interested in the usage of multi-agents systems with P2P architecture in the field of collaborative e-learning. In such applications, each learner contributes in the learning process of the group, and in return, the group contributes in the learning process of its members. The consistency of the whole group allows to achieve the goal. However, collaborative e-learning implies new roles for teachers as well as for learners. It is therefore essential to define these roles in order to identify the users needs and integrate, in the platform, the functionalities that allows us to satisfy such needs. Indeed, it is essential to provide teachers and learners with the opportunity to obtain information about the progress of their learning processes as well as the level of collaboration and sociability of each learner in the group. Finally, we have proposed, as an application of our work, a system called COLYPAN (COllaborative Learning sYstem for Project management) designed for the collaborative e-Learning project management.

**Keywords**: MAS, P2P, Coordination, Group formation, Collaborative learning