



Gestion et suivi de projets d'étudiants. Vers un système de médiation

Alain-Jérôme Fougères, Victoria Eugenia Ospina

► To cite this version:

Alain-Jérôme Fougères, Victoria Eugenia Ospina. Gestion et suivi de projets d'étudiants. Vers un système de médiation. Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie, Oct 2004, Compiègne, France. pp.156-163. edutice-00000703

HAL Id: edutice-00000703

<https://edutice.hal.science/edutice-00000703>

Submitted on 15 Nov 2004

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

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

Gestion et suivi de projets d'étudiants. Vers un système de médiation

Alain-Jérôme FOUGERES, Victoria EUGENIA OSPINA

Laboratoire M3M – Méthodes, Modèles, Métiers en Mécanique

Université de Technologie de Belfort-Montbéliard ; Rue du Château Sévenans ; 90010 Belfort ; France

{alain-jerome.fougeres,victoria.ospina}@utbm.fr

Résumé

L'utilisation de systèmes d'information coopératifs doit s'accompagner de niveaux suffisants et variés d'assistance. Les systèmes multi-agents permettent de coordonner le comportement d'agents interagissant et communiquant dans une société pour réaliser des tâches ou résoudre des problèmes. Il nous semble donc bien adaptés pour concevoir une médiation destinée aux utilisateurs de ces systèmes complexes – chaque agent représentant un niveau d'assistance. Dans l'intention d'illustrer notre approche, nous présenterons un système d'assistance intégré dans l'environnement d'apprentissage *iPédagogique* utilisé à l'Université de Technologie de Belfort-Montbéliard. L'enseignement d'unités de valeurs scientifiques ou techniques est largement fondé sur l'acquisition de connaissances conceptuelles et la validation d'un savoir-faire. Au cours d'une formation, l'apprenant et l'enseignant sont engagés dans des activités coopératives : communication, organisation, évaluation, synchronisation, réalisation, etc. Nous détaillerons plus particulièrement la gestion et le suivi de projets d'étudiants, particulièrement représentatifs de ces activités de coopération.

Mots-Clés : système de médiation, gestion de projet d'étudiants, environnement d'apprentissage coopératif.

Abstract

The use of co-operative information systems requires varied levels of assistance. Multi-agent systems allow coordinating the behavior of interacting agents, in order to realize tasks or resolve problems. Thus, they seem to us well adapted to conceive a multi-assistance, intended for complex systems users - each agent representing a level of assistance. To illustrate our approach, we present an integrated assistance system within the learning environment of *iPédagogique*. The learning of scientific or technical credits is widely based on the acquisition of conceptual knowledge and the validation of the know-how. During the learning process, the student and the teacher must be efficient, especially by overcoming organizational difficulties and by introducing implicit cooperation competences. We present the orientations of this environment then we detail the multi-assistance system conceived to facilitate the numerous cooperative tasks, especially through the project management activity.

Keywords: mediation system, students project management, co-operative learning environment.

Introduction

L'utilisation de systèmes d'information complexes, coopératifs, fortement interactifs et parfois distribués, doit s'accompagner de niveaux suffisants et variés d'assistance. L'identification et la mise en œuvre de ces niveaux d'assistance peut conduire à concevoir un véritable système de médiation dans lequel la communication homme-machine jouera un rôle majeur. Un tel système de médiation doit servir d'intermédiaire de coopération non seulement entre les utilisateurs et le système, mais aussi réciproquement. En effet, le système ne peut mener à bien les tâches qui lui sont affectées, sans la coopération des utilisateurs. Les processus dynamiques, coopératifs et autonomes nécessaires à cette interaction doivent alors intégrer une représentation des connaissances et des comportements de l'utilisateur et posséder de réelle capacité à communiquer. L'approche multi-agents offre un niveau d'abstraction adapté à cette problématique. En effet, les systèmes multi-agents (SMA) permettent de coordonner le comportement d'agents interagissant dans une société pour réaliser des tâches ou résoudre des problèmes [1].

Nous rapportons ici l'intégration expérimentale réussie d'une pédagogie par projets assistée par l'environnement *iPédagogique*. Outre le fait que cet outil appartienne à la classe des systèmes coopératifs, ce qui justifierait en soi le développement d'une aide à l'utilisateur, nous avons constaté lors de son utilisation ces deux dernières années, au département de Génie Informatique de l'Université de Technologie de Belfort-Montbéliard, qu'il était nécessaire de lui adjoindre un véritable système d'aide multi-usages et multi-utilisateurs.

Cet article est structuré comme suit : dans la section 2 nous présentons les modèles d'activités coopératives de la gestion de projets d'étudiants (GPE) et du suivi de projets d'étudiants (SPE). La section suivante est consacrée à la description des éléments de conception d'un système de médiation à base d'agents. Nous y présenterons la définition des éléments qui conduisent à une conception agent d'un système complexe, et l'architecture à trois composantes d'un système de médiation : une interface homme-machine (IHM), un système de multi-assistances (SMAs) et une base de connaissances (BC). La section 4 présente la mise en œuvre du SMA intégré à l'environnement *iPédagogique* au travers de la gestion et du suivi de projets tutorés. Finalement, en section 6, après avoir donné les éléments de conclusion sur notre approche, nous évoquons les prolongations de notre travail.

Gestion et Suivi de Projets d'Etudiants

L'activité de projet met en œuvre des compétences d'analyse, de spécification, de conception et de développement, ce qui la fait quasi-systématiquement adoptée par le corps enseignant dans sa démarche pédagogique, avec une quote-part non négligeable du temps consacré à l'UV et de l'évaluation finale.

Le concept de projet et les processus associés (développement, gestion et suivi) sont des notions beaucoup trop abstraites, aussi nous ne considérons ici que la conception de projets logiciels. Nous ne doutons pas cependant de l'extension de notre démarche à d'autres domaines techniques puisque nous nous inscrivons dans un processus de conception plus général [2]:

- < **Définition**(Besoins,Spécifications),
Elaboration(Spécifications, Architecture),
Développement(Architecture, Système) >.

Les Projets d'Etudiants

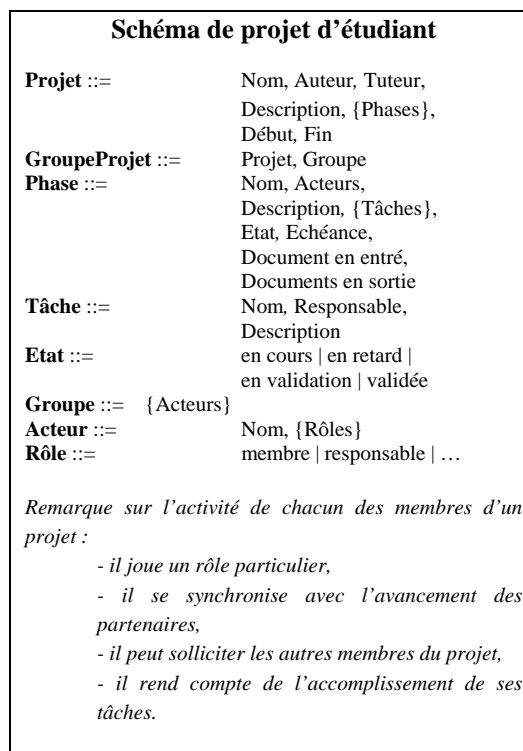


Figure 1. Définition d'un projet d'étudiant

Un projet logiciel est une démarche spécifique qui permet de structurer méthodiquement un système logiciel à venir (la DSI du CNRS vient de mettre en ligne une information très éclairante sur le sujet <http://www.dsi.cnrs.fr/conduite-projet/>). Il est défini et mis en œuvre pour élaborer une réponse au besoin de futurs utilisateurs. Il implique un objectif et des actions à entreprendre avec des ressources données. Un projet est constitué de tâches identifiées en tant que telle (rôle joué dans l'exécution d'un projet), caractérisées par un début et une fin, consommatrices de ressources et reliées entre elles par une relation d'antériorité. Ces définitions sont à

la base du modèle de projet d'étudiant que nous proposons dans la figure 1.

La Gestion de Projets d'Etudiants

Comme nous l'avons déjà évoqué, le projet est une activité qui s'inscrit dans le processus d'apprentissage et de validation d'une UV. Cependant cette activité est peu supportée par les systèmes pédagogiques [3] principalement du fait de sa complexité et de la mise en jeu de nombreux acteurs : l'administrateur, les enseignants (experts et suiveurs), les apprenants et l'outil support pédagogique.

Le processus de conception et de développement de projet logiciel peut se décomposer selon sa nature (cycle en V, spirale, prototypage rapide) en de nombreuses activités tout le long du projet (Figure 2). Ces activités sont sujettes à de nombreuses itérations (nouveaux besoins, nouvelles spécifications, revues et corrections, améliorations, tests, intégrations, planning du projet...).

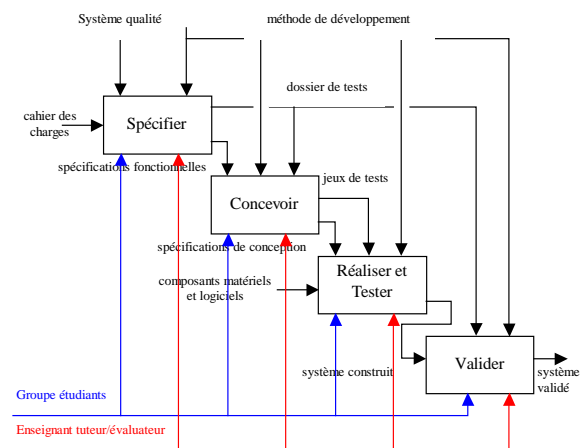


Figure 2. La GPE : une activité coopérative

La gestion de projet comporte deux fonctions : la direction de projet et la gestion de projet proprement dite. La première fonction qui s'intéresse à des décisions stratégiques ou tactiques, n'est pas celle qui est principalement évaluée dans nos formations, même si l'on peut insister pour que les étudiants à l'intérieur d'un groupe alternent les rôles de responsabilité. La seconde, par contre, traite de décisions opérationnelles, plus facilement évaluables et justifiables au gré des différentes réalisations. En effet, dans le processus de gestion de projet les étudiants déploient et peuvent rapporter des activités variées telles que :

- Structuration : identification des tâches à mener.
- Estimation : déterminer les quantités/qualités de ressources nécessaires aux différentes tâches.
- Organisation : définir les rôles, affecter les ressources, identifier les contraintes.
- Planification : maîtriser le calendrier et les risques.
- Ordonnancement : établir l'enchaînement des tâches, définir les priorités, synchroniser.
- Suivi : contrôler périodiquement l'avancement réel du projet (et des tâches) par rapport aux prévisions.

Le modèle objet du développement de projets logiciels que nous présentons ci-dessous a été conçu pour faciliter la coopération entre enseignants et étudiants dans le suivi et la gestion des projets, puis pour guider la conception d'un système de médiation pour cette activité :

- Le diagramme de classes (Figure 3) reprend les éléments du modèle de projets d'étudiants et intègre les phases du processus en cascade du développement logiciel : choix de projet, analyse, conception, codage et validation.
- Le modèle de tâche, initialement défini pour établir un modèle de connaissance sur le développement logiciel, permet de concevoir les diagrammes d'activité associés à chacune des phases de la GPE (Figure 4). A titre d'illustration, le Tableau 1 décrit la phase d'analyse des besoins pour un projet, conduisant à la rédaction du cahier des charges et à sa validation par l'enseignant tuteur du groupe d'étudiants.

Buts	Ecrire un cahier des charges (CdC) à partir d'une expression de besoins
Description	Définir les objectifs et les limites du projet
Acteurs	L'équipe projet (responsable, rédacteurs, interviewers)
Sous tâches	<ul style="list-style-type: none"> • Etudier la faisabilité • Identifier les services attendus • fixer les objectifs • répartir les rôles • rédiger le CdC • remettre le CdC
Délais	10 % par défaut (+ 5 % si choix tardif)
Validation	Validation (ou rejet) et commentaires de l'enseignant tuteur sur le CdC
Documents en entrée	Description des besoins (énoncé du projet)
Documents en sortie	Cahier des charges
Tâches tuteur	<ul style="list-style-type: none"> • Mettre à disposition les documents nécessaires à la rédaction du CdC • Lire le CdC • Valider le CdC : validation ou rejet, commentaires

Tableau 1. Fiche de définition de la première tâche à mener dans la phase 1 : « Rédiger le cahier des charges »

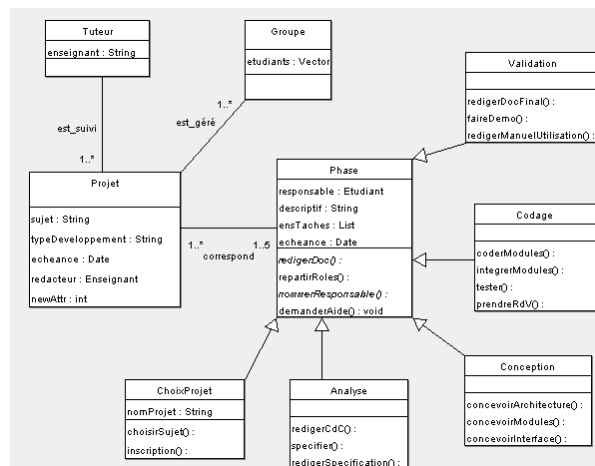


Figure 3. Diagramme de classes de la GPE

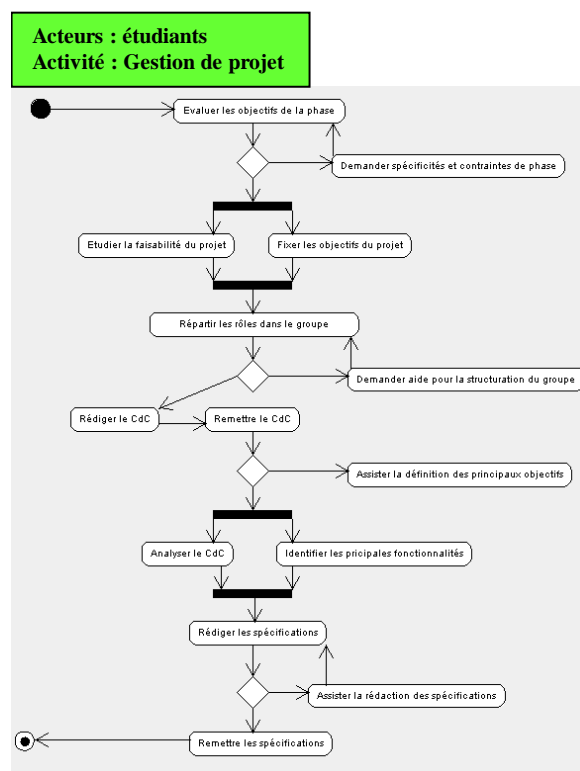


Figure 4. Diagramme d'activité de la GPE (phase 1)

Le Suivi de Projets d'Etudiants

Pour la structuration et l'évaluation des projets d'étudiants nous proposons un processus de suivi des projets. En effet, dans l'activité coopérative de gestion de projet les différents acteurs interagissent entre eux de multiple manière : ils s'adaptent continuellement, coopèrent, communiquent et négocient. En plus d'être interactifs et multi-partenaires, les procédés pédagogiques déployés dans la gestion de projets comportent de nombreuses données et relations, et sont évolutifs. Le processus de SPE doit permettre :

- Aux enseignants suiveurs (ou tuteurs) d'aider les étudiants dans leur démarche, d'apprécier la complexité des projets proposés aux étudiants et

d'évaluer plus finement le travail réalisé par le groupe de projet.

- Aux étudiants de mesurer l'état réel du projet, de comparer le réel avec les prévisions, d'élaborer des actions correctrices (éventuellement restreindre les objectifs), de structurer la conception et de produire une synthèse.

Lors du colloque TICE 2002 [4], nous avons présenté une première approche de la gestion de projets tutorés basé sur une synchronisation de fragments de procédés coopératifs. Nous l'avons étendu en spécifiant l'activité de suivi et en y joignant la composante de médiation que nous décrirons dans la section suivante.

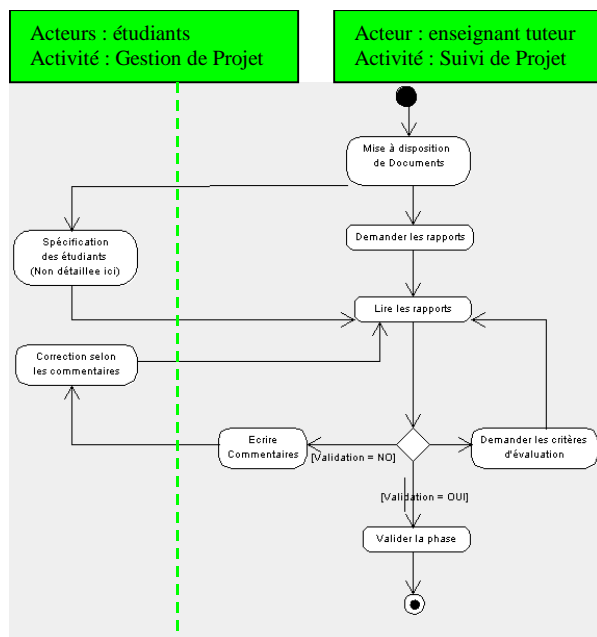


Figure 5. Diagramme d'activité de la SPE (phase 1)

Système de Médiation à Base d'Agents

L'objectif d'un système de médiation est de faciliter l'utilisation d'un système complexe (distribué ou coopératif par exemple), en proposant un ensemble ergonomique, coopératif et intelligent d'aides et de conseils aux utilisateurs, familiarisés ou non à ce type de système. Les systèmes de médiation sont conçus pour, d'une part, aider les utilisateurs individuels et les groupes à résoudre leurs problèmes, d'autre part, prendre en charge certaines fonctionnalités de l'application. La signification du terme de médiation correspond à la notion d'intermédiaire de coopération bilatérale entre au moins deux types d'acteurs : des utilisateurs et un système. Il apparaît donc que la médiation est un processus flexible qui s'applique à toutes sortes de conflits et de situations de coopération [5]. Dans un contexte d'usage croissant de systèmes coopératifs, il devient nécessaire de concevoir des systèmes de médiation qui aident les utilisateurs à exprimer leurs besoins et à découvrir l'ensemble de tous les services offerts par le système. Dans l'absolu, un utilisateur disposant d'un système de médiation peut se dispenser

d'appréhender la complexité d'un système ou d'une application.

Le processus de médiation établit donc un lien entre les acteurs qui doivent agir ensemble pour atteindre un consensus sur une situation ou une tâche commune, en assumant notamment les tâches suivantes :

- Faciliter la communication entre une application et ses utilisateurs (dans le cas des tâches coopératives, par exemple).
- Assister l'usage d'une application (interactions homme/machine), partagée ou non.
- Faciliter la découverte des fonctionnalités offertes par l'application.

Les Hypothèses de Conception

Les systèmes de médiations sont des systèmes indépendants qui servent d'interface entre l'homme et l'application afin d'enrichir leur relation. Ils peuvent se décomposer en trois sous-systèmes : un Système de Multi-Assistance, une Interface Homme-Machine et une Base de Connaissances, comprenant entre autres des connaissances du domaine et des utilisateurs (Figure 6). Dans notre démarche de conception de tels systèmes nous avons retenu 3 hypothèses [6]:

- Hypothèse 1 : l'assistance adaptée à l'utilisation d'un système complexe correspond en fait à une multi-assistance (ie, un assistant pour chaque type d'utilisation).
- Hypothèse 2 : le système de multi-assistance doit être indépendant de la partie applicative de l'outil et de son interface.
- Hypothèse 3 : le système de multi-assistance se construit de façon adéquate sous la forme d'un système multi-agents.

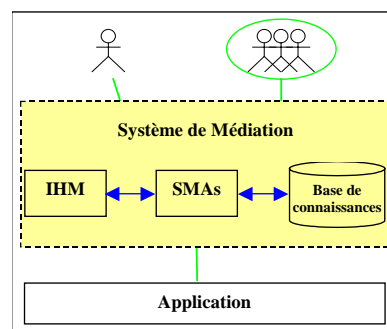


Figure 6. Principaux composants d'un système de médiation

Des Agents pour la Coopération

Le principal intérêt des systèmes multi-agents réside dans la distribution des agents, entités autonomes communicantes, réactives et compétentes (expertise, savoirs-faire). Les interactions entre agents facilitent le travail collectif : chaque agent possède une connaissance parcelaire et tous coopèrent pour un objectif global. La modélisation de tels agents consiste à définir leur

architecture, les connaissances adaptées à leurs activités et les modes de communication.

Eléments de Modélisation

La définition que nous avons retenue pour que les agents aient des comportements adaptés aux tâches qu'ils réalisent, est adaptée du modèle à 3 niveaux de l'opérateur [7]: comportement réflexe, comportement à base de règles, comportement à base de connaissances avec interprétation, décision et plan. Nous l'avons interprété comme modèle de processus de nos agents.

Def Agent ::= <Communication, Perception, Buts/intentions, Décision, Contrôle, Interprétation, Connaissances/Mémoire, Actions/Réactions>.

Architecture d'un Agent

En 2003 [8], nous avons proposé une architecture d'agent respectant les propriétés précédentes. Celle-ci (Figure 7), inspirée de la théorie de la modularité de J. Fodor et pour laquelle nous avons proposé un modèle réseaux de Petri, est composée de cinq modules gérant les connaissances, la perception, la communication, le contrôle et le raisonnement de l'agent.

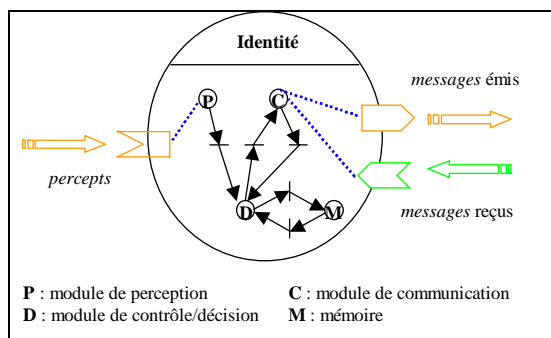


Figure 7. Architecture modulaire d'un agent

Des Agents Communicants

La communication est le principal mécanisme de coopération entre agents. Pour échanger des informations, solliciter des services ou dialoguer, les agents que nous définissons, expriment leurs intentions selon un langage proche de KQML [9], dérivé de la théorie des actes de langage de J. Searle. Le format retenu (défini par le quintuplet <intention, émetteur, récepteur, langage, message>) permet de représenter le contexte, l'intention et le message.

A titre d'illustration, la Figure 8 présente un échange entre 2 agents (l'*agentIHM* et l'*assistantGP* que nous décrivons dans l'étude de cas ci-après), suite à une demande de conseil formulée par un étudiant lors d'une phase de développement de son projet.

Des Agents Coopérants

Les systèmes de travail coopératifs sont constitués de composants distribués, hétérogènes et autonomes. Les systèmes développés en intelligence artificielle distribuée, notamment les SMA, sont donc bien adaptés.

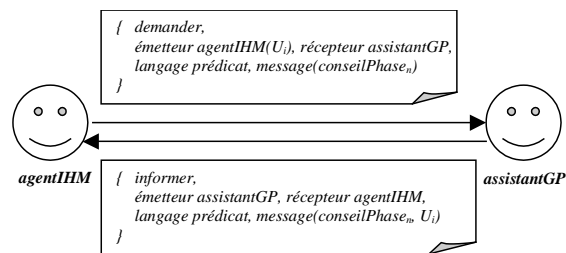


Figure 8. Exemple d'échange coopératif entre 2 agents

L'apport potentiel des agents dans un système coopératif concerne :

- la prise en charge d'actions répétitives et la délégation de tâches sans intérêt pour l'utilisateur,
- la prise de décision par compréhension du contexte d'utilisation (pertinence),
- la personnalisation de l'information (préférences, buts et capacités de l'utilisateur),
- une interactivité plus naturelle (modalités, forme et présentation),
- l'adéquation aux systèmes en réseau, et plus particulièrement les systèmes coopératifs.

Les comportements individuels et coopératifs des agents sont variés : initialisations, planification des actions, émission et réception de documents ou de messages, recherche d'information, supervision de procédures. Chacun de ces services correspond à la mise en œuvre de compétences d'un agent.

Le Processus d'Agentification

UML est devenu une référence pour la modélisation objet de systèmes d'informations. Certaines spécificités des agents (autonomie notamment), ne permettent pas de les substituer à de simples objets. Des modifications majeures sont donc à apporter à UML pour que ce langage puisse s'adapter au contexte des systèmes d'information orientés agents. Même si les résultats actuels ne sont pas entièrement satisfaisants, nous avons choisi d'utiliser AUML [10]. En conformité avec notre modèle de SMA, nous utilisons les diagrammes d'activités et de classes pour décrire le comportement et les propriétés d'un Agent, les diagrammes de contexte et de cas d'utilisation pour décrire très sommairement l'Environnement, les diagrammes de collaboration ou de séquence pour décrire les Interactions, les diagrammes de collaboration et de classes (structurées en packages) pour décrire l'Organisation. Ce processus a été suivi pour concevoir le système de médiation présenté dans la section suivante [11].

Etude de Cas : Projets d'Etudiants Tutorés dans iPédagogique

Nous avons présenté la gestion de projets d'étudiants et leurs suivis comme des activités complexes, coopératives et peu instrumentalisées. Après avoir conçu un environnement pédagogique offrant ces fonctionnalités, c'est tout naturellement que nous nous sommes interrogés sur le juste niveau d'assistance à proposer pour faciliter son utilisation. L'environnement

s'est alors révélé un terrain d'expérimentation idéal pour la conception d'un système de médiation tel que nous l'avons décrit dans la section précédente.

Présentation d'iPédagogique

iPédagogique est une plate-forme auteur pour l'enseignement d'unités de valeurs scientifiques et techniques dont la pédagogie est orientée projet [12]. Le premier objectif de cette plate-forme est d'améliorer la relation pédagogique et d'accroître l'autonomie des étudiants. Cela concerne le support pédagogique des UV et les supports électroniques (cours, TD et TP) utilisables lors des séances en présentiel et disponibles par ailleurs (auto-apprentissage). Le second objectif de cette plate-forme est d'offrir une assistance aux étudiants, centrée sur deux activités : la réalisation des TP interactifs et la gestion des projets tutorés [4].

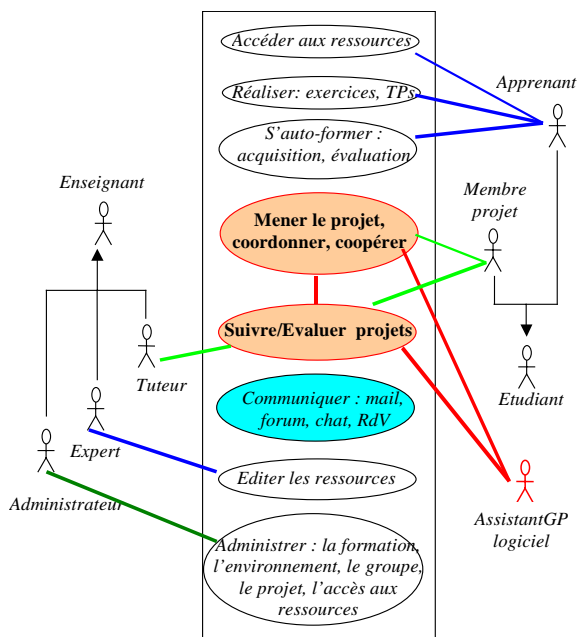


Figure 9. Cadres d'utilisation d'iPédagogique

La plate-forme offre également un support organisationnel à l'enseignant responsable d'une UV. Elle propose des fonctionnalités d'édition de séquence d'enseignement, incluant son organisation et des sessions projets qu'elle met en œuvre. Elle permet aussi la diffusion à tous les acteurs (étudiants, groupe de projets et équipe pédagogique) des informations pédagogiques et administratives.

Le contexte d'utilisation d'iPédagogique se répartit suivant quatre missions pédagogiques : *enseigner*, *apprendre*, *réaliser* et *interagir*. Chacune de ces missions se subdivise. Ainsi, l'interaction va de la communication à la coopération en passant par la collaboration et la négociation. La figure 9 présente les différents rôles que les utilisateurs peuvent remplir et les fonctionnalités qui leur sont proposées.

Cette plate-forme nous l'avons voulue ouverte, c'est-à-dire extensible dans les schémas pédagogiques à mettre en œuvre, adaptable à différents types de matières, et configurable par un utilisateur néophyte. La mise en œuvre nécessite des moyens logiciels et

matériels très abordables (MySQL ou easyPhp, cgi/php, html et javascript). Cela autorise une intégration facilitée de l'outil configuré au sein d'une autre plate-forme (WebCT par exemple).

Le Système d'Assistance Intégré à iPédagogique

Comme nous l'avons présenté dans la Figure 6, le système de multi-assistance est la composante centrale du système de médiation. Ce système, conçu selon une approche multi-agents, est constitué de 5 agents d'assistance (Figure 10) :

- Un assistant de gestion d'UV.
- Un assistant pour la gestion des projets étudiants.
- Un assistant de profil utilisateur
- Un assistant pour l'utilisation des formulaires
- Un assistant tutoriel.

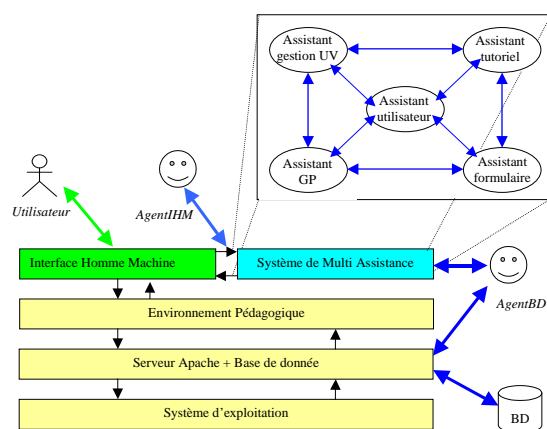


Figure 10. Architecture du système iPédagogique avec visualisation des agents d'interfaces

Remarque : le nombre de 5 agents n'est pas arrêté ; il correspond seulement à notre stade expérimental. L'étude des services rendus par l'environnement, schématisés par le diagramme de la Figure 9, nous a permis d'identifier 5 classes d'utilisation, et d'y associer systématiquement un assistant logiciel.

L'Assistant de Gestion de Projet

En cours de semestre, l'assistant de gestion de projets (*assistantGP*) simplifie la communication, la gestion et le suivi des projets :

- pour l'enseignant responsable, en l'assistant dans les tâches de diffusion des sujets, de contrôle des inscriptions aux projets, de suivi des plannings, de suivi des groupes, etc ;
- pour les groupes de projets étudiants, en les assistant dans les tâches d'inscriptions aux projets, de respect des plannings, de répartition des rôles, de remise de documents, de prises de rendez-vous, de rappel des contraintes de phases, etc ;
- pour les enseignants tuteurs de projets, en les assistant dans les tâches de suivi des plannings des

groupes de projets, de réception de documents, de prises de rendez-vous, etc.

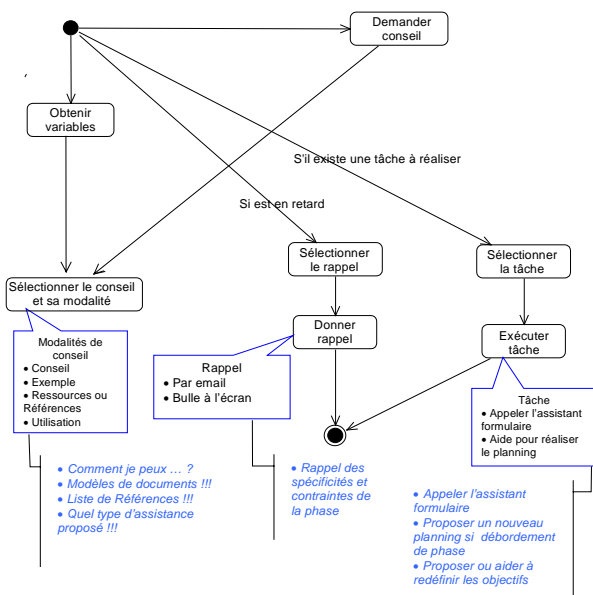


Figure 11. Comportement de l'assistantGP

Le modèle d'activité de cet assistant (comportement d'assistance et coopération entre assistants) est fourni dans la Figure 11. Pour illustrer le comportement coopératif de l'assistant de gestion de projet envers un groupe d'étudiants, nous présentons ci-dessous un exemple sur chacun de ses deux modes d'action :

- Le mode intervention : la Figure 12 présente la réaction de l'assistantGP faisant suite au retard d'un groupe de projet lors de la réalisation de la phase d'analyse - celle-ci doit normalement aboutir à la remise d'un cahier des charges et d'un document de spécification à l'enseignant suiveur du groupe. L'interaction sera assurée avec l'étudiant responsable de la phase en question.

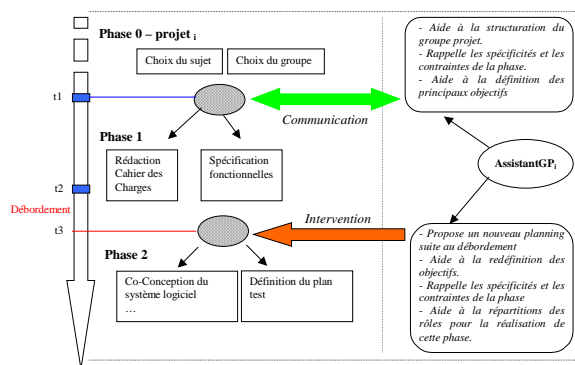


Figure 12. Intervention de l'assistantGP, suite à un débordement de phase

- Le mode informationnel (ici un rappel) : la Figure 13 fournit un exemple de conseil affiché par le système à la connexion d'un étudiant impliqué dans un projet. L'IHM envoie une information de connexion au SMAs (login de l'étudiant). L'assistantUtilisateur

envoie alors une requête à l'agentBD pour obtenir les données de l'utilisateur, dont celles concernant l'avancement de projet. Sur la requête de l'assistantUtilisateur, l'assistantGP fait la sélection d'un conseil parmi ceux qu'il n'a pas encore fournis, la transmet à l'assistantUtilisateur qui l'envoie en guise de réponse à l'agentIHM. Ce dernier peut alors retourner la réponse à l'IHM qui se charge de l'afficher.

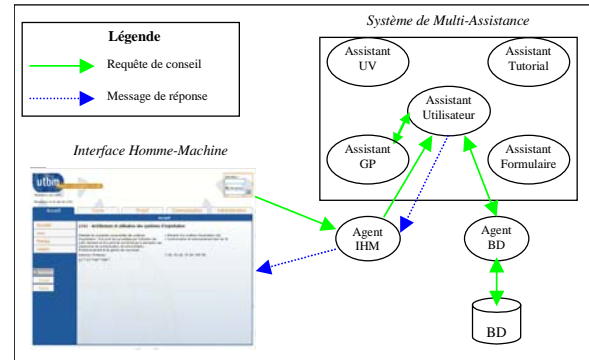


Figure 13. Procédure de conseil de gestion de projet, activée lors de la connexion

Les deux captures d'écran de la Figure 14 présentent l'interface de consultation et de définition permettant d'assurer le suivi et la gestion de projets pour les différents acteurs.

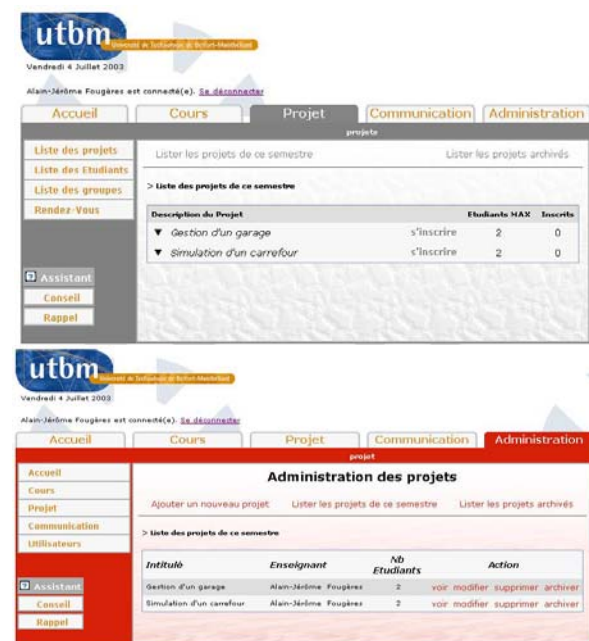


Figure 14. Consultation et administration de projets par l'enseignant responsable.

Conclusion

Nous venons de présenter la définition et la conception d'un système de médiation pour un environnement pédagogique qui allie complexité (distribution des niveaux d'assistance en fonction des usages) et clarté de

présentation puisque l'assistance est conçue comme un véritable système, connecté à la couche applicative de l'environnement et à l'IHM. Nous tenons à insister sur le fait que l'objectif de cette n'est pas de se substituer aux utilisateurs, mais bien de les aider à évoluer et à coopérer dans leurs différentes tâches. La gestion de projets d'étudiants proposée par l'outil *iPédagogique* a été expérimentée pendant 3 ans. Après le plébiscite accordé par les différents utilisateurs (étudiants, enseignants, administrateur) et l'observation de leurs modes d'utilisation, c'est le développement d'une assistance qui s'est imposée. Plutôt que de travailler sur la seule perspective d'une assistance à la gestion de projets, une réflexion plus générale, centrée sur l'ensemble des cas d'utilisation et de coopération, identifiés lors de la modélisation de la première version de l'outil, a guidé la conception du SMAs.

Nous nous sommes plus particulièrement focalisé sur la description du SMAs. La réalisation de celui-ci a été particulièrement illustrée avec l'assistant de gestion des projets d'étudiants (objectif du prototype développé), mais notre souci de généricité rend le modèle utilisable pour l'ensemble de l'assistance d'*iPédagogique*, telle que présentée dans la section 4.1. La suite du projet va consister en une étape de tests en présence d'étudiants (dans le cadre d'une UV d'Intelligence Artificielle) tout au long du semestre d'automne 2004, afin de rendre opérationnelle la nouvelle version d'*iPédagogique* (SMAs et nouvelle IHM) pour la rentrée universitaire de printemps 2005. Au-delà des résultats attendus pour ce projet, les perspectives de ce travail rentrent dans l'élaboration d'une méthodologie de conception de système de médiation adapté aux systèmes complexes de types distribués et coopératifs. Ceci s'étend à la définition d'une architecture de SMAs, à la résolution des problèmes de communication et de coopération entre les composants de cette architecture, à la spécification des niveaux d'interaction réciproque intervenant dans la coopération entre les utilisateurs et le système, et à l'acquisition des connaissances nécessaires à la mise en œuvre efficace et pertinente de l'assistance dans le domaine cible.

Références

[1] Ferber, J. 1997. Les systèmes multi-agents : un aperçu général. *Technique et Science Informatiques*, 16(8), 979-1012.

[2] Matta, N., Ribière, M. et Corby, O. (1999). Définition d'un modèle de mémoire de projet. Rapport Technique INRIA n°3720, INRIA.

[3] Tchounikine, P. 2002. Pour une ingénierie des Environnements Informatiques pour l'Apprentissage Humain. *Revue Information-Interaction-Intelligence*, Cépaduès Edition, 2(1).

[4] Fougères, A.-J., Canalda, P. 2002. *iPédagogique* : un environnement intégrant la gestion assistée de projets d'étudiants. Colloque TICE 2002, Lyon.

[5] Giraldo G., Reynaud C. 2002. Vers l'automatisation de la construction de systèmes de médiation pour le commerce électronique, Journées Scientifiques Web sémantique, 10-11 octobre.

[6] Ospina, V. E., Fougères, A.-J. 2003. Un système d'assistance dans un environnement coopératif d'apprentissage. CITE'03 (Coopération Innovation et Technologies), Troyes, 3-4 décembre.

[7] Rasmussen, J. 1983. Skills, rules, and knowledge ; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13, 257-266.

[8] Fougères, A.-J. 2003. Des agents communicants pour simuler et détecter des épidémies. *Ingénierie des Systèmes d'Information*, Hermès, 8(1), 91-112.

[9] Finin, T., Fritzson, R., McKay, D., McEntire, R. 1994. KQML as an agent communication language. *Proceedings of CIKM'94*, ACM Press.

[10] Odell, J., Partunak, H.V.D., Bauer, B. 2000. Extending UML for agents. *Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence*, Austin, Texas.

[11] Ospina, V. E. 2003. Assistance dans les EIAH : une gestion assistée de projets d'étudiants. Mémoire de DEA de l'Université de Technologie de Troyes.

[12] George, S., Leroux, P. 2001. Un environnement support de projets collectifs entre apprenants : SPLACH. *Sciences et techniques éducatives*, Hermès, 8(1-2), 49-60.