7e Conférence Internationale de MOdélisation et SIMulation - MOSIM'08 - du 31 mars au 2 avril 2008 – Paris- France

« Modélisation, Optimisation et Simulation des Systèmes : Communication, Coopération et Coordination.».

UNE ARCHITECTURE D'E-LEARNING MULTI-AGENTS ADAPTATIVE

D. ROY

LGIPM école Nationale d'Ingénieurs de Metz
Ile du saulcy
57045 Metz cedex, France
roy@enim.fr

A. MENG

Faculty of Hydroelectric and Digital Enginering
Huazhong University of Science of Technology
430074 Wuhan,China
menganbo@vip.sina.com

RÉSUMÉ : *Une étude de la littérature montre que le paradigme d'E-éducation a le potentiel de révolutionner les codes de l'éducation en la rendant individuelle plutôt que basée sur des institutions, en éliminant les mesures en temps réels en faveur de la mesure des performances et résultats et en favorisant des solutions d'apprentissage sur mesure. Par conséquent, il n'est pas surprenant que l'E-éducation soit maintenant devenue plus acceptable à la société et aux académies qui offrent divers types d'environnements d'apprentissage en ligne qui permettent d'obtenir, à la demande, des programmes d'enseignement flexibles, sur mesure, indépendamment de la répartition géographique, sociale et économique. Malheureusement, il semble que toutes les promesses de l'E-Learning ne soient pas tenues et que certains freins subsistent, dus au fait que les systèmes existants sont toujours calqués sur le modèle de l'enseignement présenciel. C'est pourquoi, nous nous présentons une architecture d'E-learning suffisamment adaptative et flexible pour obtenir le meilleur des possibilités théoriques de l'E-Éducation. Pour ce faire, nous nous sommes appuyés sur le paradigme des systèmes multi-agents qui nous semble bien adapté à cette situation. Après une présentation générale de l'architecture, nous présenterons deux fonctionnalités particulières : le travail collaboratif et l'aide appariée.*

MOTS-CLÉS : *Multi-agents, Ontologie, E-Learning, Apprentissage Adaptatif*

1 INTRODUCTION

Le monde se transforme en un grand village avec le développement rapide des Technologies de l'Information et de la Communication (ICT) (Nabil *et al*., 1997). En particulier, l'avance des ICT a potentiellement provoqué un changement énergique du processus d'éducation en améliorant la manière dont l'information et la connaissance sont représentées et distribuées aux étudiants.

En tant qu'une des applications majeures d'Internet, l'e-Éducation naissante s'avère être un aspect important pour le secteur éducatif (Lennon *et al*., 2003) aussi bien que pour les entreprises en tant qu'élément d'une approche holistique de gestion de la connaissance (Hasebrook, 2001). Comme précisé dans (ADL, 2004) : « *les paradigmes et les réalisations de l'e-éducation ont apporté beaucoup d'avantages à l'éducation à distance basée sur la technologie.* »

Malheureusement, jusqu'ici le grand potentiel de l'e-Éducation est loin d'avoir été complètement exploité. Ceci est souvent expliqué par le fait que, en dépit d'impressionnants développements récents, la plupart des systèmes actuellement disponibles d'e-Éducation sont toujours moins attrayants que les méthodes d'enseignement traditionnelles, tant pour des étudiants que pour les professeurs.

En effet, les étudiants se plaignent souvent du manque d'outils flexibles à l'appui d'apprentissages personnalisés, de partage stimulant de connaissances mutuelles, d'aides appariées "juste à temps" ou des conseils efficaces et « en temps » d'enseignants.

De la perspective des enseignants, l'inconvénient principal des systèmes actuel d'e-Éducation est qu'ils tendent à exiger plus d'effort en termes d'écriture des matériaux d'étude et de préparation des exercices ou des examens que leurs contreparties classiques. La nécessité de maîtriser les outils de haute technologie et le manque des connaissances en informatique des enseignants les rendent souvent peu disposés à participer à l'activité d'enseignement en ligne.

Dans cet article, nous nous sommes concentrés sur l'approche Multi-agents comme environnement d'intégration et d'encapsulation des technologies et méthodologies développées, ainsi qu'à modéliser et implémenter plusieurs applications typiques de l'e-Éducation à différents niveaux et dans différents contextes tels que l'apprentissage individuel et collectif et la recherche d'une aide personnalisée dans l'environnement d'étude distribué.

L'objectif global de cet article est d'essayer d'établir un système flexible, redimensionnable, adaptatif et intelligent d'e-Éducation (MAGE) par l'intégration de certaines technologies de pointe telles que SMA, XML, etc. aussi bien que des théories d'enseignement et d'apprentissage bien établies.

2 ARCHITECTURE DE MAGE (Multi-AGent e-Education system)

Le modèle de référence Leaning Technology Systems Architecture (LTSA), du IEEE LTSC (IEEE, 2003) semble être une bonne référence pour construire notre système d'e-Éducation. De fait, nous commencerons par analyser ce modèle de référence puis, nous présenterons notre architecture multi-agents d'un système d'e-Éducation qui est constituée de trois types d'agents : agents Apprenant, agents Serveur et agents Contenu.

2.1 Un modèle de référence recommandé pour l'e-Éducation

Comme le montre la Figure 1, le modèle LTSA identifie quatre processus :

- le processus d'apprentissage ;
- le processus d'évaluation ;
- le processus de guidage ;
- le processus de distribution.

On y trouve aussi deux bases de données :

- informations sur les élèves ;
- cours/EEO.

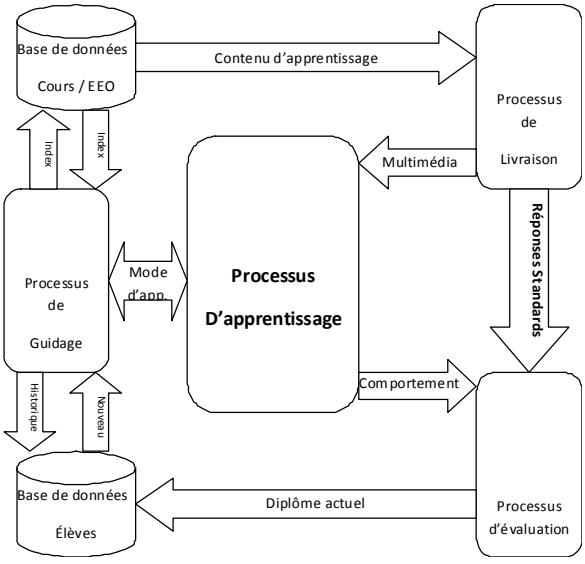

Figure 1. Un modèle recommandé de référence pour l'e-Éducation (IEEE LTSA)

De plus, dix flux d'informations relient ces composants : comportements de l'élève, multimédia, contenu d'apprentissage, index de contenu, index de recherche, historique de diplôme, diplôme actuel, réponses standards et mode d'apprentissage.

2.2 Les systèmes Multi-agents

Cette technologie a longtemps été perçue comme une solution viable pour développer de grandes applications complexes dans des environnements dynamiques tels qu'internet ou le Web. Dans le domaine de l'e-Éducation, la technologie SMA attire de plus en plus l'attention des chercheurs et des utilisateurs, car c'est la promesse d'un nouveau paradigme pour la conceptualisation, la conception et l'implantation d'environnements d'éducation complexes et distribués dans lesquels les diverses ressources (cours en ligne, personnes, outils, etc.) sont —précisément— dynamiquement géographiquement distribuées, les comportement des systèmes sont souvent imprévisibles et leurs besoins sont toujours modifiables.

Originellement, un système multi-agent est un ensemble d'agents où chaque agent est placé dans un environnement quelconque et est capable d'interagir avec celui-ci et avec les autres agents. Aujourd'hui, le terme système multi-agents a un sens beaucoup plus général et il est maintenant utilisé pour tout type de système constitué de composants autonomes multiples. Chacun de ces composants a les caractéristiques suivantes :

- Chaque agent dispose d'informations incomplètes, d'un point de vu limité et de capacités insuffisantes pour résoudre un problème ;
- Il n'y a pas de système de contrôle global ;
- Les données sont décentralisées ;
- Les calculs sont asynchrones.

De plus, un système multi-agents doit avoir les compétences suivantes (Camacho *et al.*, 2002) : Coordination, coopération, négociation et communication.

2.2.1 Standard FIPA

La Fondation pour agents physiques intelligents (Foundation for Intelligent Physical Agents - FIPA) est une organisation internationale à but non-lucratif dont l'objectif est de promouvoir l'industrie des agents intelligents. Dans ce but, la FIPA développe des spécifications publiques qui permettent l'interopérabilité entre agents et applications basées sur les agents de manière à répondre aux besoins des applications actuelles de distribution de services qui sont dynamiques et hétérogènes (O'Brian *et al.*, 1998). À travers cet ensemble complet de spécifications, la FIPA tente de soutenir à la fois l'interopérabilité au niveau agent et plateforme.

2.3 Cadre de MAGE

La figure 2 présente le cadre de notre système multi-agent d'e-Éducation.

En fonction des fonctionnalités des agents de MAGE, nous les avons classés en trois types :

- Agents Apprenant ;
- Agents Serveur ;
- Agents Contenu.

Les agents Apprenant sont responsables de la fourniture de services spécifiques d'enseignement aux élèves, les agents Serveur sont en charge de la gestion efficace et de la maintenance de l'ensemble du système et les agents Contenu sont dédiés à la gestion, la maintenance et la création de cours et d'objet d'apprentissage.

2.3.1 Principaux Agents Apprenant

- **Agent Apprenant Assistant (AAA)**

L'AAA fournit à l'élève une interface personnalisée d'apprentissage en accord avec ses préférences et performances. Sa principale fonction est de permettre l'interaction entre les apprenants et MAGE qui peut, ainsi, réagir activement aux demandes des élèves et présenter des contenus d'enseignement sur mesure.

- **Agent Apprenant d'Évaluation (AAE)**

L'AAE est responsable de l'évaluation (comprenant pre- et post-évaluation et l'évaluation processus) des performances des élèves durant tous les processus d'apprentissage. Toutes les informations d'évaluation sont utilisées pour ajuster le modèle d'apprentissage ou aider l'agent Pédagogie à créer des stratégies ou des parcours d'apprentissage appropriés.

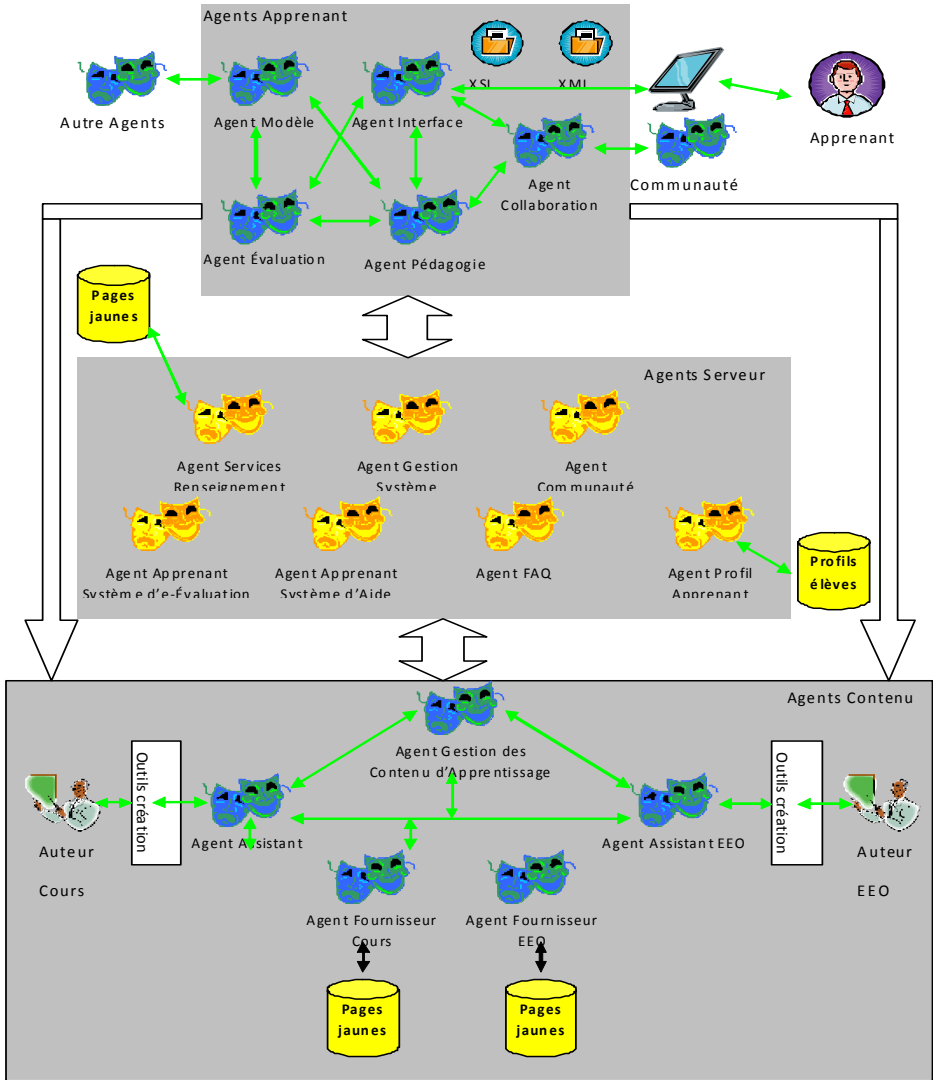

Figure 2. Système Multi-agent d'e-Éducation (MAGE)

- **Agent Apprenant Pédagogie (AAP)**

L'AAP prend en charge la création de stratégies d'apprentissages sur mesure ainsi que de parcours appropriés.

- **Agent Apprenant Collaboration (AAC)**

L'AAC peut aider l'élève à créer un environnement d'apprentissage collectif et former un groupe d'apprentissage.

- **Agent Apprenant Système d'Aide Appariée (AASA)**

Ce sous-système est responsable de la fourniture d'aide.

- **Agent Foire Aux Questions (AFAQ)**

L'AFAQ est responsable de la fourniture de réponse aux questions courantes des élèves.

- **Agent Apprenant Système d'Aide Appariée (AASA)**

Ce sous-système est responsable de la fourniture d'aide.

- **Agent Apprenant Système d'e-Évaluation (AASE)**

Ce sous-système fournit tous les services impliqués dans la réalisation d'un e-Examen.

2.3.2 Principaux Agents Serveur

Les agents Serveur peuvent être classés en deux types : agents gestion Système et agents service d'apprentissage. Le premier groupe est responsable de la gestion et de la maintenance de MAGE lui-même alors que l'autre groupe fournit des services d'apprentissage aux agents du coté client (côté élèves).

2.3.2.1 Agents Gestion Système

Ce groupe est composé de quatre types d'agents. Ils sont tous conçus de manière à être conformes aux spécifications FIPA.

- **Agent Gestion Système (AGS)**

L'AGS fournit un service d'annuaire, c'est à dire qu'il contient un répertoire des identifiants agent et s'assure que chaque agent a un nom unique dans le système. Cet agent gère aussi la création, la destruction, la suspension, la reprise et la migration des agents dans MAGE

- **Agent de TéléSurveillance (ATS)**

L'ATS offre une interface graphique dédiée à l'administration de la plateforme. Il retourne l'état courant des agents de la plateforme et propose de nombreux outils pour demander des actions administratives de l'AGS et pour débuguer ou tester les applications.

- **Agent Services Renseignements (ASR)**

L'ASR fournit un service de renseignement type "pages jaunes" dans le sens où les agents peuvent enregistrer leurs services auprès de l'ARS ce qui permettra à d'autres agents de trouver plus facilement le service dont ils auront besoin pour remplir leurs objectifs.

- **Agent Canaux de Communications (ACC)**

L'ACC est un agent qui utilise les informations fournies par l'AGS pour contrôler tous les échanges de messages dans la plateforme MAGE.

2.3.2.2 Agents Service d'Apprentissage

- **Agent Questions-Réponses (AQR)**

L'AQR est responsable de la gestion des questions-réponses venues des agents coté élèves.

- **Agent communauté (AC)**

L'AC gère les environnements d'apprentissage collectifs. Dans MAGE, tous les AAC sont sous la supervision de l'AC.

- **Agent Profil Apprenant (APL)**

L'APL est responsable de la gestion et de la maintenance de la base de données des profils de chaque élève.

2.3.3 Agents Contenu

D'une manière générale, les agents Contenu sont physiquement placés du côté serveur. Cependant, les agents Interface Cours, les agents Assistant de Cours et les agents EEO peuvent se déplacer ou être téléchargés du côté client. De fait, les auteurs de cours ou « d'objets d'apprentissage » peuvent les développer où qu'ils se trouvent. On peut noter que, comme MAGE est un système distribué point-à-point (peer-to-peer) avec un environnement commun, les frontières entre clients et serveurs sont quelque peu floues dans de nombreux cas.

- **Agent Interface Cours (AIC)**

L'AIC sert d'interface utilisateur graphique (GUI) pour les auteurs de cours. Il fournit un outil de création qui facilite le développement cohérent de cours composés d'EEO ou de matière nouvelle.

- **Agent Assistant Cours (AAC)**

L'AAC aide les auteurs de cours à accomplir des tâches spécifiques telles que la recherche d'EEOs existants, la souscription aux services de LCMA, la négociation avec les auteurs d'EEOs, etc.

- **Agent Interface EEO (AIEEO)**

Semblable à l'AIC, l'AIEEO fournit un environnement spécifique de développement des EEO.

- **Agent Assistant EEO (AAEEO)**

Semblable à l'AAC, l'AAEEO réalise des tâches spécifiques.

- **Agent Fournisseur EEO (AFEEO)**

L'AFEEO est responsable des communications avec les différentes bases de données, EEO, contenu, etc.

- **Agent Fournisseur Cours (AFC)**

L'AFC est en charge de la communication avec la base de données cours.

- **Agent Gestion des Contenus d'Apprentissage (AGCA)**

L'AGCA est responsable de la gestion des cours et des « objets d'apprentissage » ainsi que la maintenance des

fichiers métadonnées XML qui incluent la description métadonnée des cours, des EEO, etc. Cet agent fournit aussi un service « pages jaunes ».

À l'aide de l'architecture MAGE, il est possible aux élèves d'apprendre suivant de nombreux scénarii différents tels que le professorat intelligent, l'apprentissage encadré, l'apprentissage collectif, etc.

Le système MAGE dans son intégralité représente un temps de spécification et de modélisation plus que conséquent. Aussi, après avoir posés les bases de l'architecture et donné ses spécifications de niveau supérieur, avons nous décidé de nous intéresser plus particulièrement à des fonctions spécifiques de MAGE, que nous allons modéliser et maquetter dans les chapitres suivants.

3 CADRES D'APPRENTISSAGE ACTIF ET ADAPTATIF BASÉS SUR LE SMA

3.1 Introduction

Puisque la personnalisation et l'adaptation font parties de nos spécifications, nous devons prendre en compte un certain nombre de problèmes, tels que la génération dynamique de parcours d'apprentissage, la distribution d'objets d'apprentissage (EEO) sur mesure qui correspondent à l'état des connaissances de l'élève et à ses préférences d'études, la recherche de ressources d'aide (conseillers, matériel, applications) pour un élève lorsqu'il rencontre des difficultés sur un certain domaine ou sujet et la création d'environnements d'apprentissages collectifs permettant un apprentissage constructiviste (Spiro *et al.*, 1992) et qui peuvent être créés à l'initiative d'un élève pour un objectif particulier.

Notre spécificité en ce domaine, par rapport aux autres systèmes adaptatifs d'apprentissage, est que nous faisons la distinction entre deux mécanismes : l'apprentissage adaptatif individuel (AAI) et l'apprentissage adaptatif collectif (AAC). Dans MAGE, l'AAI indique une situation où le système aide à choisir un parcours d'enseignement adaptatif qui consiste en une suite de domaines conceptuels et aux objets d'apprentissage associés, alors que l'AAC signifie que le système cherche des élèves appariés dans le réseau distribué afin de faciliter soit une session d'aide, soit un apprentissage collectif en construisant dynamiquement des groupes d'apprentissage en accord avec les préférences de l'élève.

3.1.1 Définition des objets d'apprentissage

Pour être plus précis, nous avons choisi d'utiliser le terme d'E-Éducation Objet (EEO) dans ce document. Voici notre définition des EEOs :

Un E-Éducation Objet (EEO) est un composant réutilisable, modifiable, redimensionnable, héritable, polymorphe et multi-objectif qui encapsule des matériaux bruts (contenus, exercices, tests) bien organisés, aussi bien que l'interface permettant d'y accéder.

Cette définition est dérivée des concepts de la Programmation Orientée Objet (POO) et se réfère aux modèles Advanced Distributed Learning de SCORM (ADL, 2001) et RLO de Cisco (Wiley, 2000). Ce triple héritage semble donner une manière efficace de description et construction des EEO.

Bien entendu, les problèmes de propriété intellectuelle doivent être pris en compte lorsqu'un EEO doit être modifié ou réutilisé par d'autres que l'auteur original. Une approche possible de cette question est de conditionner la réutilisation à la permission du détenteur du copyright et/ou imaginer un système de rétribution (droits d'auteur).

3.2 Modélisation des domaines

La représentation des domaines de connaissances est le point de départ pour implanter un mécanisme flexible et adaptatif. Le modèle de domaine que nous proposons dans ce document est composé de trois niveaux : niveau ontologie, niveau méta modèle et niveau objets d'apprentissage (voir Figure 3). Nous définissons explicitement le modèle de domaine comme :

Dmodel={concepts abstraits, liens sémantiques, métadonnées, objets d'apprentissage).

De fait, le niveau ontologies est représenté par un ensemble de concepts/sujet abstraits et des liens sémantiques plutôt que le matériel d'apprentissage actuel.

Dans le modèle proposé, nous identifions plusieurs liens sémantiques entre les concepts domaine (DC). De plus, pour augmenter l'adaptabilité et la flexibilité, les suppositions suivantes sur les objets d'apprentissage doivent être posées :

- Définir différents types de supports (texte, images, audio, vidéo, etc.).
- Différencier les types d'objets d'apprentissage (contenu, exercices, contrôle, etc.).
- Adapter les objets d'apprentissage aux caractéristiques de l'élève (langue, accessibilité, style d'apprentissage, etc.).
- Adapter les objets d'apprentissage à un concept spécifique à un domaine particulier.

3.3 Apprentissage adaptatif individuel

3.3.1 Architecture agent

Dans l'architecture proposée, il y a une fonctionnalité distincte qui la différencie des autres cadres d'apprentissage. D'après la Figure 3, il est clair que les apprentissages individuel et collectif ne sont pas isolés,

en fait, ils sont même combinés dans un cadre commun d'apprentissage qui permet de contribuer aux expériences d'apprentissage personnalisées tant individuelles que collectives.

L'*Agent Génération de Présentations* (AGP) est responsable de la génération d'un ensemble d'objets d'apprentissage (EEO) qui expliquent (enseignent) le concept cible définit par l'élève. Cette architecture introduit aussi l'espace collectif d'apprentissage impliquant le système d'aide appariée et la formation de groupes d'apprentissages. Cette architecture fournit de nombreuses possibilités à l'élève pour choisir la prochaine stratégie d'apprentissage, s'il n'est pas satisfait des contenus d'apprentissage recommandés par le système.

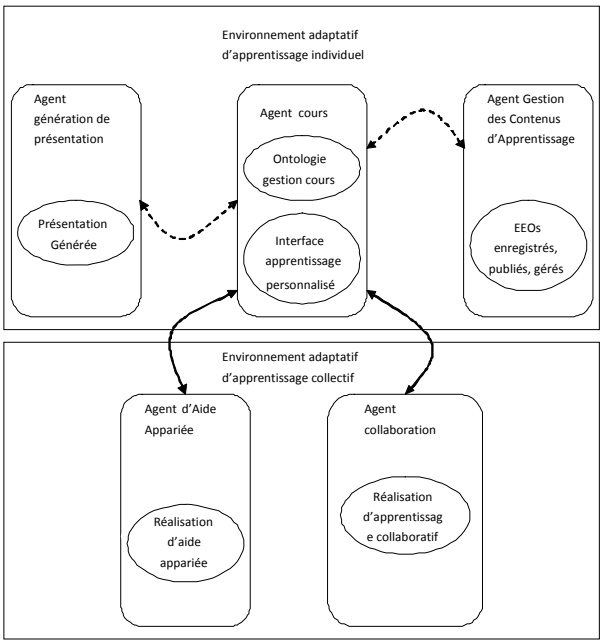

Figure 3. Architecture d'apprentissage adaptative

Cette architecture est aussi suffisamment adaptative pour traiter le problème de l'apprentissage adaptatif. En particulier, l'adjonction d'espaces d'apprentissage collectifs compense, par certains cotés, le manque de contacts directs entre les élèves et les professeurs.

3.3.2 Génération automatique de parcours d'apprentissage

Une Présentation (PR) est une liste d'EEOs fournie à un élève afin de remplir ses besoins d'apprentissage au mieux. Pour obtenir une PR, l'apprenant doit définir une liste de concepts cibles (CC), en fonction de ses objectifs. L'AGP enverra alors une requête à l'Agent Modèle Apprenant (AMA) basée sur les préférences d'apprentissage de l'élève et l'état de ses connaissances. Partant de ces entrées, l'AGP construira une Présentation, en fait une liste d'EEOs qui satisfait à tous les CC et qui prend en compte l'état actuel de l'apprenant (état des connaissances et préférences d'apprentissage) (Figure 4).

Dans l'algorithme de génération de présentation, la première ligne vérifie si l'ontologie qui représente la structure du cours est cohérente. Ceci, afin d'éviter des boucles inutiles dans les opérations 3 et 4. Une Ontologie est cohérente lorsque, par exemple, $x<=y$ et $y<=z$ et $z=<x$. La vérification est aisément réalisée par la recherche d'un cycle dans le graphe orienté qui représente l'ontologie. Pour chaque concept cible, la ligne 3.a collecte récursivement tous les concepts domaine (DC) nécessaires à son apprentissage. Dans la ligne 3.b, un DC est ajouté à la liste de concepts élémentaires (AtomicList) s'il n'a pas de composant fils. La ligne 4 linéarise la liste de concept élémentaire juste construite.

Input: Concepts Cibles: CC liste, KS: État des Connaissances, LP: Préférences Apprentissage.
Output: EEO liste.
1. Vérification de la cohérence de l'ontologie de cours.
2. Q:= CC, AtomicList:= Nil, PR:= Nil.
3. **pour** chaque x ∈ Q tel que ¬connu(x, KS), **faire**:
   a. **Si** [*EstPartDe*(y,x) **ou** *EstRequisPar*(x,y) **ou** *EstRéférencéPar*(x,y) **ou** *EstTestéPar*(x,y)] **et** y ∉ Q, **Alors** insérer y dans Q.
   b. **Si** ¬∃ y tel que *EstPartDe*(y,x), **alors** insérer x dans AtomicList.
4. AtomicList:= Tri(AtomicList, CC).
5. **Pour** chaque x ∈ AtomicList **faire**:
   a. Créer LOList liste des EEOs *e* tel que *EstEnseignéPar*(x, e) et *Cohérent*(e, LP).
   b. BLO:= *Choisir_le_meilleur* (LOList,LP).
   c. Insérer BLO dans *PR*.
   d. **Pour** chaque x' AtomicList tel que *EstEnseignéPar*(x', BLO), effacer x' de AtomicList.
6. Retourner *PR*.

Figure 4. Algorithme de génération de Présentation

Dans l'algorithme de génération de présentation, la première ligne vérifie si l'ontologie qui représente la structure du cours est cohérente. Ceci, afin d'éviter des boucles inutiles dans les opérations 3 et 4. Une Ontologie est cohérente lorsque, par exemple, $x<=y$ et $y<=z$ et $z=<x$. La vérification est aisément réalisée par la recherche d'un cycle dans le graphe orienté qui représente l'ontologie. Pour chaque concept cible, la ligne 3.a collecte récursivement tous les concepts domaine (DC) nécessaires à son apprentissage. Dans la ligne 3.b, un DC est ajouté à la liste de concepts élémentaires (AtomicList) s'il n'a pas de composant fils. La ligne 4 linéarise la liste de concept élémentaire juste construite.
En ligne 5, pour chaque DC *x* de la liste élémentaire, est sélectionné l'EEO le plus approprié parmi tous ceux capable d'expliquer *x* (LOlist(x)). Il est important de noter que ce choix est local à LOlist(x). En effet, un

objet d'apprentissage peut être lié à deux ou plus concepts de domaine, par exemple un même EEO peut expliquer à la fois les *dérivées* et les *intégrales*. De fait, si en ligne 5.d, la fonction : *Choisir_le_Meilleur(Dérivées, LP)* renvoie cet EEO particulier, alors *Intégrales* sera exclu en 5.d.

L'optimisation locale peut sembler restrictive, mais il s'agit ici d'un choix assumé, car une optimisation globale mènerait à une explosion combinatoire, alors que seuls de rares EEOs sont reliés à plus de un DC.

Pour finir, les fonctions *Connu*, *Cohérent* et *Choisir_le_Meilleur* sont facilement réalisables par comparaison des DCs ou des EEOs avec l'état de connaissance ou les préférences d'apprentissage.

3.4 Apprentissage adaptatif collectif

3.4.1 Introduction

Un des besoins de base de l'éducation dans le futur est de pouvoir préparer les apprenants à une participation à une société d'information en réseau où la connaissance sera la ressource la plus critique du point de vue du développement social et économique. Les interactions sociales apparaissent comme particulièrement importantes pour la réalisation, la construction et la compréhension partagée de la connaissance basées sur une négociation sociale de points de vue et d'opinions. Hiltz (1984) souligne ce fait lorsqu'elle affirme que « le processus social de développement de la compréhension partagée par l'interaction est la voie *naturelle* de l'apprentissage ».

Dans les deux sections suivantes, nous nous concentrerons sur deux approches visant à faciliter l'apprentissage constructiviste.

3.4.2 Modélisation de l'aide appariée

Pour illustrer les fonctionnalités du système d'aide appariée (SAA), nous allons utiliser un scenario d'exemple.

Imaginons qu'un apprenant travaillant sur un problème de programmation dans un cours d'informatique rencontre toujours des problèmes vis-à-vis d'un certain concept présenté par le système. Il peut alors déléguer la tâche de recherche d'aide à son Agent Assistant Personnel. Celui-ci essaie de trouver d'autres agents (qu'ils soient d'applications ou d'autres agents personnels) qui offrent des ressources d'informations en relation avec la demande d'aide.

Les agents, en effet, partagent une taxonomie commune pour indexer les ressources d'informations, basée sur les sujets/concepts enseignés.

Pour trouver l'aide appariée la plus appropriée, l'Agent Assistant Apprenant doit requérir un agent spécial nommé agent « Matchmaker » pour réaliser cette correspondance. S'il n'existe pas de ressource électronique appropriée à la question de l'apprenant, le Matchmaker créera une liste classée des apprenants en ligne qui peuvent apporter des informations sur le concept cherché. Le Matchmaker envoie alors cette liste à l'Agent Assistant Personnel de l'apprenant demandeur d'aide. L'Agent Assistant Personnel commence alors les négociations avec chaque agent personnel des aides potentielles de la liste, essayant de trouver une aide à un prix acceptable.

Dès que le processus de négociation a réussi, l'agent de l'aide potentielle prévient son apprenant et demande s'il veut bien apporter son aide. Dans la négative, l'agent personnel doit négocier avec les autres agents de la liste des aides suggérées. Dans l'affirmative, un canal de communication est ouvert entre les deux utilisateurs (par exemple un simple outil de chat), et la session d'aide débute.

L'architecture (Figure 5) du sous-système dédié à ces opérations est composée de plusieurs agents qui coopèrent et négocient. Ces agents partagent des ontologies et un langage de communication commun (ACL). Chacun d'eux gère des ressources spécifiques à l'utilisateur ou à l'application qu'ils représentent.

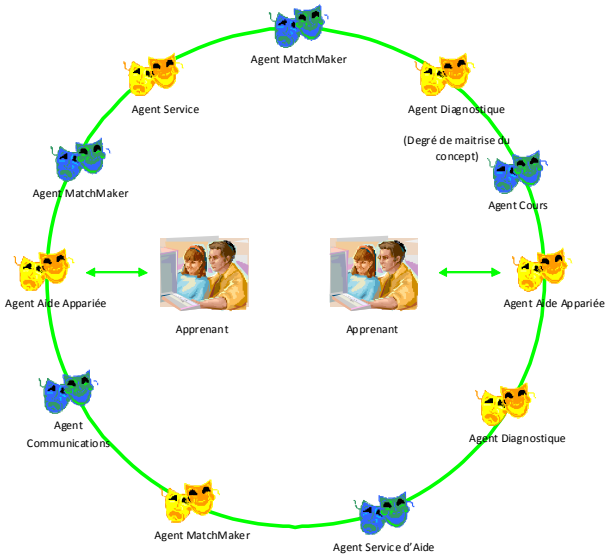

Figure 5. Architecture du système d'aide appariée (SAA)

Les principaux agents impliqués dans cette architecture sont décrit ci-dessous :

- **Agent d'Aide Appariée, (Agent AA)**, dans le contexte du système d'aide appariée, il maintient des modèles d'apprenants partiels qui contiennent leurs caractéristiques basiques.
- **Agent Cours** est une application importante munie d'une interface graphique qui représente le cours correspondent à la taxonomie du domaine enseigné.

- **Agent Service d'Enregistrement (Agent SE)** remplit une seule tâche, à savoir enregistrer les informations de service d'aide dans l'Agent Service d'Aide (voir ci-après).
- **Agents Diagnostique** (agents représentant des modules de tests, questionnaires etc.) représentent un type spécial d'application qui créé des modèles d'apprenant pour un objectif particulier.
- **Agents Service d'Aide (Agent SA)** sont une extension de l'agent DF de la FIPA (voir O'Brian 1998) qui a la capacité de publier et maintenir tout service.
- **Agents MatchMaker** agissent comme des courtiers spéciaux qui sont responsables de la bonne correspondance dans des situations spécifiques (localiser l'aide appariée qui est la plus compétente sur un certain domaine ou celle qui a un style d'apprentissage compatible, etc.).

3.4.3 Approche de modélisation dans le système d'aide appariée

La plus grande partie de la communication et du raisonnement concernant les apprenants est distribuée entre les agents du système. De fait, la modélisation des apprenants et l'adaptation est tout aussi fragmentée et locale. Dans le système d'aide appariée, il existe plusieurs situations en rapport à l'approche de modélisation multi-agents et multi-apprenants, à savoir :

- **Apprenant modélisant des Agents** : l'agent d'aide appariée (en tant que représentant de l'apprenant) peut être modélisé par son propriétaire qui peut définir comment sera son agent et ses règles de participation à une négociation.
- **Agents modélisant d'autres Agent**s : pour mieux négocier, l'agent d'aide appariée (en tant qu'agent personnel) doit être capable de prédire le prochain mouvement de son homonyme adverse. Mouvement qui dépend de la stratégie et des préférences de l'opposant.
- **Utilisateurs modélisant d'autres Utilisateurs** : un apprenant peut instruire un agent sur les autres utilisateurs, soit en lui fournissant une évaluation sous une forme quelconque interprétable par l'agent, soit en fixant explicitement des valeurs pour certaines propriétés dans le modèle d'autres utilisateurs (ou de leur agents).
- **Agents modélisant les Utilisateurs** : cette situation est proche du processus traditionnel d'un système modélisant un utilisateur.

Pour montrer clairement le processus interactif d'aide appariée proposé, nous utilisons un diagramme de séquence UML pour illustrer un scénario classique dans lequel un apprenant A peut choisir d'apprendre un ensemble de concepts domaine dans un espace individuel d'apprentissage adaptatif.

Lorsque le système détecte qu'un concept cible (complet ou partiel) visé par l'apprenant A est maîtrisé, il enregistre A automatiquement comme capable de fournir le service d'aide correspondant via l'*Agent Service d'Aide*. Bien sûr, l'apprenant A peut, à loisir, spécifier les limites, conditions et stratégies de ce service.

Lorsqu'un autre apprenant (B) ne parvient pas à comprendre certain concepts au terme d'une période définie d'apprentissage, il peut obtenir de l'aide dans l'espace collectif d'apprentissage. Dans ce cas, il lui est possible de lancer la tâche de recherche d'aide appariée via son *Agent d'Aide Appariée*. Dès que la liste des aides possibles présents en ligne est renvoyée, cet agent sélectionnera le plus approprié. Les étapes détaillées de ce processus sont explicitées ci-dessous (Figure 6) :

Étape 1 : l'apprenant A demande à l'Agent Cours d'afficher le plan du cours spécifié constitué des concepts domaine. Dès que l'apprenant aura fixé les concepts cibles visés, l'agent cours enverra une requête à l'agent modèle apprenant pour définir son état de connaissances et ses préférences d'apprentissage. En fonction de ces données d'entrée, l'agent cours pourra générer un parcours d'apprentissage adapté (tous les concepts nécessaire pour expliquer les concepts cibles) et les EEOs associés.

Étape 2 : lorsque le moment arrive, ou si l'apprenant A décide d'anticiper l'échéance, ses performances seront validées par un test. L'agent cours demandera alors à l'agent diagnostique de créer le test et d'enregistrer les résultats (Meng *et al.*, 2007).

Étapes 3 et 4 : dès que les concepts testés ont été validés par un seuil fixé, l'agent cours demande à l'agent service d'enregistrement d'inscrire le service d'aide avec l'agent service d'aide.

Étape 5 : l'apprenant B veut apprendre un concept cible quelconque via le système d'aide appariée. Il exécute donc son agent d'aide appariée.

Étape 6 : l'agent d'aide appariée envoie les informations de correspondance nécessaire à l'agent MatchMaker pour lancer la recherche de l'aide appropriée.

Étape 7 : l'agent MachMaker demande à l'agent service d'aide de localiser les aides appariées qui remplissent les conditions de correspondance.

Étape 8 : l'agent service d'aide renvoie à l'agent d'aide apparié de B une liste d'aides appariées bien informées sur le domaine demandé et qui satisfont aux autres conditions spécifiques.

Étape 9 : l'agent d'aide apparié de B commence à négocier avec son homonyme de A (en supposant que A fasse bien parti des candidats potentiel). Lorsque cette

négociation se conclut avec succès, chaque agent en avertit son apprenant respectif. Dans le cas contraire, l'agent d'aide appariée de B continuera à négocier avec d'autres agents d'autres aides potentielles.

Étape 10 et 11 : les agents d'aide apparié de l'aidé et de l'aideur informent leur apprenant des résultats de la négociation.

Étape 12 et 13 : si la négociation fut un succès, les deux apprenants exécutent leur outil de communication (chat, e-mail, téléphone, etc.) et démarrent la session d'aide.

Étape 14 : lorsque la session d'aide est terminée, l'apprenant B peut évaluer les performances de son aide appariée.

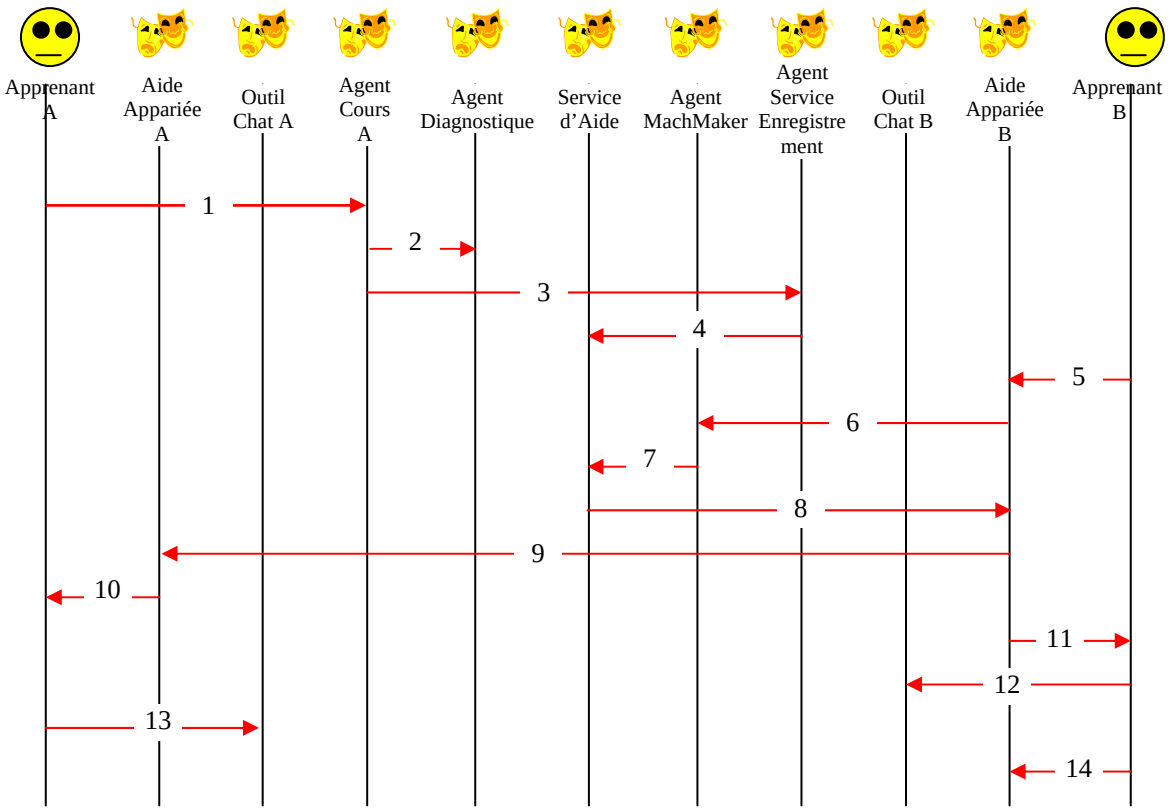

Figure 6. Diagramme de séquence UML de l'aide appariée

3.5 Modélisation de la formation de groupes d'apprenants

Cette section présente une architecture qui établit dynamiquement des groupes de collaboration pour des apprenants individuels partageant les mêmes objectifs d'apprentissage. Dans cette architecture, les apprenants individuels peuvent établir un profil de collaboration.

3.5.1 Formation de groupe

D'un point de vue technologique, la collaboration entre les utilisateurs d'un environnement d'apprentissage en ligne dépend de deux tâches qui sont la définition du groupe et l'établissement de sessions (synchrones ou asynchrones) de communication. De fait, le cadre de collaboration doit, au moins, fournir les outils permettant de réaliser ces deux opérations.

Dans un environnement d'apprentissage en ligne où le professeur peut ne pas être présent, l'apprenant ignorant qui sont les autres élèves et ce qu'ils étudient, la définition d'un groupe d'étude peut devenir assez complexe. Le problème clé du cadre proposé, qui vise à faciliter l'établissement de paramètres de collaboration dans un tel environnement, est le concept de groupe : un ensemble dynamique d'apprenants qui sont réunis ensembles pour discuter sur un concept ou un sujet particulier.

Pour faciliter les groupes d'apprentissage dans ce contexte, nous proposons la définition d'un profil de collaboration de l'apprenant. Le profil du groupe est composé d'une liste de conditions exprimant l'objectif à atteindre et, optionnellement, le degré de connaissance (exprimé par une valeur réelle entre 0 et 1) requis par les membres du groupe sur le concept considéré. Ce profil de groupe est utilisé par le cadre d'apprentissage collaboratif pour trouver les utilisateurs dans l'environnement virtuel dont les profils individuels satisfont l'ensemble de conditions proposées. Lorsqu'un apprenant reçoit une liste de membres potentiels du groupe qui satisfont au profil de groupe proposé, la phase de recherche prend fin et le système démarre la phase d'invitation.

En plus de permettre la création de nouveaux groupes, le système doit fournir des fonctionnalités pour contrôler les activités et la participation des membres du groupe.

3.5.2 Architecture agents

Il est clair que le modèle proposé est naturellement implanté avec les SMA puisque les tâches liées à la recherche et à l'invitation de collaborateurs appariés sont aisément réalisables par des agents. Les principaux d'entre eux sont :

- **L'Agent Groupe** réalise la tâche de recherche considérant un profil de groupe donné et procède à l'invitation des partenaires sélectionnés ou même demande l'inclusion dans un groupe existant. Cet agent est aussi responsable de la vérification de la correspondance entre un groupe spécifié et les profils individuels des apprenants.
- **L'Agent Collaborateur**, au nom de son apprenant, est responsable de la réception des spécifications d'objectifs des utilisateurs et réalise les actions pour atteindre ces objectifs.
- **L'Agent Conseil** remplit la plupart des tâches en relation avec le contrôle de la connaissance du groupe et de ses membres.

4 CONCLUSION

Dans cet article, nous avons donc décrit l'architecture d'un système d'E-Learning flexible et adaptatif qui doit nous permettre d'approcher les fonctionnalités théoriques extraordinaires de l'E-Éducation. Pour ce faire, nous nous sommes appuyés sur le paradigme des multi-agents qui, intrinsèquement, nous apportent les capacités de flexibilité et d'adaptabilité.

L'architecture, dans son ensemble, étant bien trop longue à spécifier en détail, nous nous sommes attaché à définir quelques fonctionnalités intéressantes qu'elle est en mesure d'apporter. Pour ce faire, nous présentons, contrairement aux systèmes existants :

- Une conception d'un algorithme de génération de parcours d'apprentissage sur mesures, tant sur le fond que sur la forme, composés de concepts permettant d'atteindre l'objectif fixé par l'apprenant vis-à-vis de l'état de ses connaissances et de ses préférences d'apprentissage ;

- Une résolution du problème de recherche des ressources appropriées telles qu'une aide appariée (« peer to peer ») ou un professeur dans l'espace distribué d'apprentissage via un système d'aide astucieux ;

- Un modèle flexible et innovant de l'apprentissage collaborative, en accord avec les principes du cognitivisme. De plus, ce modèle rend aisé pour les professeurs l'organisation et le suivi des activités de groupe.

Une maquette de cette architecture, capable de remplir ces fonctionnalités et d'autres, comme la génération automatique d'examens, a été développée en utilisant la plateforme multi-agent JADE. Les résultats des tests effectués sur cette maquette sont plus qu'encourageants et démontrent que l'architecture fonctionne, même s'ils furent réalisés par « simulation » et non sur un public enseignants/apprenants réel. Dans l'avenir, il serait intéressant de finaliser le développement en intégrant toutes les fonctionnalités et, bien sûr, d'avoir l'opportunité de lancer un test en grandeur réelle.

5 RÉFÉRENCES

ADL Advanced Distributed Learning, 2004, SCORM, 2004. Last visit: 2004-03-26, *URL: http://www.adlnet.org/index.cfm?fuseaction=scormabt*

Camacho Fernández, D., 2002, Recuperación e Integración de Información, *disponible en Web para la Resolución de Problemas, Universidad Carlos III de Madrid.*

Hasebrook, J.P., 2001, Learning in the Learning Organisation. *In journal J.UCS, vol. 7, pp. 472-487.*

Hiltz Starr Roxanne, 1984, *Online Communities: A Case Study of the Office of the Future*, Norwood NJ, Ablex.

IEEE, 2003, *IEEE Standard for Learning Technology-Learning Technology Systems Architecture (LTSA)*, ANSI, doc#: IEEE Std 1481.1-2003

Lennon, and J., Maurer, H., 2003, *Why it is Difficult to Introduce e-Learning into Schools And Some New Solutions.* In journal *J.UCS*, vol. 9, pp. 1244-1257.

Meng A., Ye L., Roy D., Padilla P., 2007, *Genetic Algorithm based multi-agent system applied to test generation.* Computer & Education 49(2007) pp 1205-1223. Elsevier eds.

Nabil, A., Awerbuch, B., Slonim, J. Wegner, P. & Yesha, Y., 1997, *Globalizing business, education, and culture through the Internet*. Communications of the ACM, 40(2).

O'Brien, P and Nicol, R, 1998, *FIPA - Towards a Standard for Software Agents.* In: BT Technology Journal, Vol.16:3, pages 51-59.

Spiro, R. J., Feltovich, P. J., Jacobson, M. J. & Coulson, R. L., 1992, *Cognitive flexibility, constructivism, and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domain.* In T. Duffy & D. Jonassen (Eds.) *Constructivist and the technology of instruction*, Hillsdale, N.J.: Lawwrence Erlbaum Association Publishers, 57-75.

Wiley, David A., 2000, *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy.* In D.A. Wiley (Ed.). *The Instructional Use of Learning Objects* [on-line]. Available: http://reusability.org/ read/.