

Tutoring et Les Systèmes Multi-Agents : Conception, Modélisation et Expériences

ABDELLAH BENNANE

CENTRE DE FORMATION DES INSPECTEURS DE L'ENSEIGNEMENT (CFIE)

RABAT, MAROC.

bennanea@yahoo.fr

RESUME. Les systèmes tuteurs sont devenus complexes dans lesquels, on peut trouver, modules de cours, modules d'exercices, simulateurs, des systèmes en ligne, hors ligne, simple utilisateur, multi-utilisateurs, Web, etc. Cette complexité incite à améliorer la conception et la modélisation et introduire de nouveaux concepts qui permettent au système tuteur d'interagir efficacement avec ses utilisateurs potentiels dans le sens d'offrir un bon service ergonomique, un bon temps de réponse et une meilleure adaptabilité. L'introduction de nouveaux concepts tels que les systèmes Multi-Agents (SMA), la technologie web dans le processus de conception et de modélisation des systèmes tuteurs va contribuer à améliorer les solutions existantes surtout pour la formation à distance et à trouver des idées innovantes aux problèmes créés par cette association (SMA, Web). Des expériences multiples ont vu le jour et la présentation de quelques projets pertinents parmi ces expériences peut être bénéfique dans le but de tirer des idées innovantes pour élaborer de nouveaux projets, un savoir faire pour les mettre en œuvre, etc.

MOTS CLES : agent, environnement, jeux stochastiques, Web, RL, SMA, tutoring

SUMMARY : The tutoring systems become complex where one can find, course modules, exercises, simulators, systems online, offline, single user, multi-user, Web, and so on. This complexity motivates to improve the design and modeling and introduce new concepts that allow the tutoring system to interact efficiently with potential users in the sense to provide good ergonomic service, an excellent response time and better adaptability. The introduction of new concepts such as multi-agent systems (MAS), web technology in the process of designing and modeling tutoring systems will assist improving the existing solutions for distance learning and to find an innovative ideas to the created problems by the association (MAS, Web). Multiple experiments have emerged and the presentation of some relevant projects among these experiences may be beneficial in order to draw some innovative ideas to develop new projects, how to implement them, and so on.

KEYWORDS: agent, environment, stochastic games, web, RL, SMA, tutoring .

1. Introduction

La technologie de communication a modelé et a remodelé notre monde comme la radio qui a suivi le télégraphe, la radio suivie par la télévision, et cette dernière suivie par les satellites et les ordinateurs. Les technologies, dans différentes générations, ont été annoncées comme une force qui peut transformer l'éducation. Dans ce sens, trois questions s'imposent : la technologie de communication a-t-elle modelée le développement de l'enseignement ouvert et à distance, et si ainsi comment ? Comment les technologies sont-elles employées ? Change-t-elle tout et maintenant ?

La dernière décennie a été une période de changement de la technologie de communication qui peut avoir des effets profonds sur l'enseignement ouvert et à distance. Les changements ont touché la technologie des satellites, le prix des produits technologiques, la disponibilité et la puissance des ordinateurs. L'expansion de la technologie Internet a accéléré et transformé la communication entre ceux qui ont accès en elle. Plusieurs de ces changements ont affecté les établissements éducatifs, chacun à sa manière. Les traitements de texte ont transformé la production des matériaux de télé-enseignement juste de la même manière qu'ils ont changé la correspondance dans le monde des affaires. L'utilisation de l'email et l'accès à la littérature en format électronique ont réduit l'isolement du personnel académique des universités du sud. L'informatique ouvre quelques possibilités spécifiques à la formation à distance. Elle peut être utilisée dans l'enseignement ou la formation pour développer un cours entier avec une variété de médias, de rendre ceci disponible sur Internet et d'employer la même technologie pour le contact entre l'étudiant et le tuteur et les étudiants entre eux ; pour la distribution des documents afin de réduire le coût et le délais des correspondances ; dans le management pour aider à gérer, internationaliser et globaliser l'enseignement à distance. La British Open University, par exemple, offre un programme de Master en enseignement à distance. Les étudiants téléchargent le matériel d'enseignement à partir du Web, contactent leur tuteur par email et sont encouragés de parler dans les téléconférences considérées comme une partie du cours. Une université virtuelle pourrait inscrire des étudiants, et en utilisant l'Internet, fournir l'enseignement aux étudiants globalement.

La technologie excite, ouvre quelques possibilités, ferme d'autres. Aujourd'hui, alors que le coût de beaucoup de technologie avait diminué nettement, beaucoup de propositions peuvent limiter l'accès, plutôt que de l'élargir, en mettant plus de coûts (charges) sur l'utilisateur. Dans les pays de sud, une grande partie des cours disponibles sur ordinateur (Net) atteindra les riches, ou ceux qui travaillent dans le secteur moderne ; mais supprimera le reste de la liste (pauvres, personnel des secteurs non modernes). Notons que ce processus touchera des femmes davantage que des hommes.

L'introduction et l'exploitation des nouvelles technologies à l'école doit être une des priorités. L'égalité des chances est à l'ordre du jour. Nous trouvons que la création d'un pôle de formation continue et à distance (PFCAD) en utilisant les NTIC pour les enseignants et les formateurs par exemple est l'un des moyens qui peut valoriser l'école en valorisant le personnel de l'éducation en lui offrant une chance de se former et de s'auto-former et d'ouvrir l'école à son environnement et au monde.

L'un des domaines qui a contribué à la réussite du développement de l'INTERNET est la formation à distance. Le potentiel de l'Internet permet une offre de matériaux et d'expériences riches, une possibilité et une capacité à apprendre plus de savoir et de savoir faire. Ce sont des facteurs d'une importance croissante dans un monde où la quantité d'information qui doit être apprise croît très rapidement et devient très vite obsolète.

Les apprenants d'une classe virtuelle sur l'Internet sont largement distribués géographiquement, et le nombre de participants est important. Par conséquent, les systèmes statiques et centralisés sont inadaptés et insuffisants. Dans ce sens, les caractéristiques spécifiques de l'enseignement - apprentissage permettent à l'architecture Multi-Agent d'être plus approprié à l'implémentation des systèmes d'apprentissage distribués en utilisant la technologie web. Un système multi - agent (SMA) distribué avec des agents personnalisés pour chaque apprenant est attrayant au sens de l'adaptabilité. Les classes sont de nature dynamique. Le contexte, les connaissances et les compétences des apprenants changent au fil du temps. Le matériel et peut-être les méthodes pédagogiques vont changer aussi. La finalité est de trouver une adéquation entre l'offre « pédagogique » en matière d'enseignement - apprentissage et les besoins d'un public hétérogène d'apprenants. Cela ne peut être à l'ordre du jour que si on élabore des projets innovants, pédagogiquement et techniquement.

Dans ce papier, nous allons (1) présenter quelques projets pertinents qui peuvent résumer l'état de l'art des systèmes multi-agents et le Tutoring ; (2) exposer une application (SIOF) que nous avons eu le plaisir de la concevoir et de l'implémenter afin d'essayer quelques techniques de l'apprentissage automatique pour les agents cognitifs. Les bases théoriques de l'apprentissage des SMA tels que l'apprentissage par renforcement, les jeux stochastiques et le modèle espace action jointe seront exhibées. Cette application va essayer de donner une idée sur le « comment » implémenter un agent cognitif. On termine par (3) une conclusion.

2. SMA et tutoring

2.1. Introduction

L'utilisation des techniques de modélisation a été exploitée par de nombreuses applications dans différents domaines à savoir E-Learning, les systèmes de conseil, la recherche d'assistance, hypermédia adaptatif et le commerce électronique (e-commerce). Deux problèmes ont été associés à la modélisation de l'utilisateur: l'identification des informations pertinentes à modéliser et la décision qui permet de sélectionner les méthodes à appliquer en vue d'acquérir les informations pertinentes au sujet de l'utilisateur.

Le SMA est une société organisée, composés d'agents semi - autonome qui interagissent les uns avec les autres, pour collaborer à solutionner une série de problèmes, où à réaliser une série d'objectifs individuels ou collectifs. Ces agents peuvent être homogènes ou hétérogènes et peuvent avoir des objectifs communs ou non, mais maintiendront toujours un certain degré de communication entre eux. L'utilisation des SMA dans le développement des logiciels destinés à l'enseignement est prometteuse du fait que la nature des problèmes d'enseignement et d'apprentissage est un champ d'étude, de recherche et d'expérience. La plupart des méthodes et techniques nouvelles de l'apprentissage automatique par exemple ont été essayées dans ce domaine .

Les systèmes Multi-Agents nécessitent une évolution de la manière de penser sachant que la conception scientifique et technique est fondée sur l'idée qu'un système est une entité monobloc composée de sous-systèmes bien identifiés et figés et non sur l'idée qu'un système est une population d'entités autonomes en interactions complexes.

Parmi les caractéristiques des SMA, on trouve : (1) une **organisation** sociale avec laquelle les groupes d'agents s'organisent à l'intérieur du système, selon le rôle, les caractéristiques, et les responsabilités de chacun ; (2) une **coopération** entre agents, qui provient du partage des résultats intermédiaires pour trouver une solution aux objectifs individuels, dans le même temps, contribuer à atteindre les objectifs globaux du système ; (3) une **coordination** par le biais de la communication, les agents coordonnent actions et comportements, ce qui permet aux systèmes d'éviter les situations de conflits entre les agents et d'être cohérents ; (4) le **contrôle**: est la base du mécanisme de l'implémentation de la coordination dans les SMA. Les paramètres de contrôle sont de deux types, globaux et locaux et (5) par la **communication** entre agents, les objectifs individuels ou ceux de la société ou du système dans lequel ils vivent, peuvent être atteints.

Cette approche reproduit une organisation sociale évoluée d'une société moderne pour les systèmes artificiels.

2.2. Environnements utilisant la technologie des SMA

L'utilisation de la technologie SMA dans le processus de conception et de modélisation des environnements d'enseignement a connu une évolution spectaculaire en parallèle avec la vulgarisation de la technologie Internet. L'expérience a montré qu'une grande quantité de projets a vu le jour dans le monde où les technologies d'information et de communications sont enracinées ou presque. Dans cette section, nous allons présenter quelques projets qui vont nous permettre de voir les différentes approches utilisées, les domaines d'applications et leur rapport avec les architectures classiques des systèmes d'information orientés vers l'enseignement.

1) **BAGHERA**[1] : L'approche du projet BAGHERA repose sur: les trois principes suivants (1) la conception des environnements éducatifs considère la collaboration entre les humains et les agents artificiels comme un principe fondamental; (2). Les connaissances de l'apprenant admettent une diversité de conceptions, dont le critère de base de la pertinence n'est pas conforme à certaines connaissances de référence, mais leur efficacité dans certaines sphères de la pratique; (3). L'éducation est le résultat d'un processus complexe émergent. Il ne peut être le résultat de l'action d'une stratégie isolée ou un objectif accompli d'un agent isolé. La notion de conception est utilisée pour représenter les connaissances de l'étudiant à des domaines de pratique. En substance, il considère que l'éducation, comme un phénomène complexe, peut émerger des interactions entre les agents ayant des capacités différentes et complémentaires. Il s'agit d'une communauté éducative d'agents qui interagissent entre eux en fonction de leurs compétences afin de coopérer et mener à bien les tâches éducatives collectivement. L'objectif fonctionnel est de construire une plate forme souple et adaptable pour la formation à distance où chaque étudiant est assisté par trois types d'agents: **compagnon** de l'élève, **médiateur** et **tuteur**. De même, l'enseignant est assisté par deux types d'agents: l'interface, et l'assistant. La plate-forme BAGHERA est un système multi-agents ouvert: le nombre d'agents dans la société augmente ou diminue en fonction du nombre d'utilisateurs connectés. Par exemple, à un moment précis, étant donné un nombre n d'étudiants et un nombre m d'enseignants connectés, le nombre d'agents artificiels est $3n + 2m$. Il s'agit d'une remarque importante, puisque le nombre de connexions n'est pas limitée et le nombre d'agents n'est pas figé dans la société. Le comportement dynamique des agents peut être observé.

2) **ALLEGRO** [2] est un environnement intelligent qui permet d'offrir un apprentissage individualisé et à la manière de CSCL (Computer Supported Collaborative Learning). L'utilisation de SAM offre à ALLEGRO l'autonomie, la souplesse et l'adaptabilité. ALLEGRO repose sur trois théories de l'apprentissage: comportementalisme, cognitivisme et l'historico-social. Dans l'environnement ALLEGRO, on trouve six d'agents: (1) tuteur qui se charge d'orienter le processus d'apprentissage, décide de l'action pédagogique à faire,

comment et quand. (2) modèle de l'apprenant qui permet de maintenir le modèle de l'apprentissage de l'apprenant. (3) interface: c'est le pont entre l'utilisateur et les agents artificiels. (4) expert: il administre le contenu objet spécifique de l'enseignement - apprentissage. (5) Diagnostic: il est chargé de sélectionner et de classer le niveau de connaissances de l'apprenant (6) collaboration: à la demande de l'agent tuteur, il cherche d'autres apprenants qui sont intéressés par le même sujet dans le but d'établir des communications synchrones ou asynchrones au sens collaboratif¹ du terme.

3) **MATHEMA [3]**: propose une architecture multi-agents qui intègre les différents formalismes afin de faciliter la tâche des enseignants pour développer le contenu d'un système tuteur et en même temps fournir la capacité d'adaptation et de flexibilité dans la présentation. La technologie des SMA a été d'une grande aide dans la réduction de la distance entre le système idéal et ce que les systèmes puissent être réellement. En d'autres termes, les SMA permettent de simplifier la modélisation et la structuration des tâches grâce à la répartition, entre les différents agents tels que les modèles du domaine et de l'apprenant. L'architecture proposée est basée sur un modèle conceptuel, appelé MATHEMA, qui offre un contenu orienté méthodologie pour planifier et la présentation du domaine et les stratégies pédagogiques. Le système est destiné à enseigner la structure de données dans la discipline de l'automatisation et de contrôle à des futurs ingénieurs à l'université de Santa Catarina, au Brésil.

4) **MACES [4]** (Multi-Agent Architecture for a Collaborative Educational System): Il s'agit d'un système destiné au travail collaboratif pour l'enseignement à distance. Son architecture est composée d'agents humains (apprenant et tuteurs) et cinq types d'agents artificiels au sujet de (diagnostic, médiation, collaboration, social et sémiotique).

5) **AMPLIA [5]** (Intelligent Probabilistic Multi-Agent Learning Environment): c'est un environnement ou une ressource supplémentaire pour la formation des étudiants en médecine. Les utilisateurs (étudiants, enseignants et applications) sont représentés par des agents autonomes qui font partie d'une organisation sociale fondée sur les objectifs tels que communication, coopération et négociation. AMPLIA a comme agents artificiels: étudiant, domaine, médiateur.

6) **MAS-PLANG [6]**: c'est environnement qui a l'intention d'offrir des caractéristiques d'adaptation qui reposent sur les styles d'apprentissage pour soutenir l'enseignement à distance à travers le Web. L'environnement est composé de deux niveaux d'agents: ceux du niveau supérieur (Programmable Agent SONIA, Synthetic SMIT, Monitors and Surfing), et ceux du niveau inférieur (agent didactique, Utilisateur).

7) **JADE [7]** (Java Agent framework for Distance learning Environments). Il offre un ensemble de ressources pour faciliter l'élaboration et la mise en œuvre des environnements de calcul afin de les utiliser comme instruments pour la formation à distance. Les agents n'ont pas de mobilité, ils sont: Gestionnaires du contenu, Exercices, exemples, Interactions, modèle de l'étudiant et modèle et communication.

8) **MathTutor [8]**: Dans le cas du système MathTutor, le domaine des connaissances (structure de données) est divisée en deux contextes (théorique et pratique), et chacun de ces contextes est élaboré en deux parcours (abstraction procédurale et abstraction de données). Par conséquent, le tutoriel de la société d'agents est composé de quatre agents, chacun responsable d'un sous domaine: TP (théorie de l'abstraction procédurale); PP (pratiques de l'abstraction procédurale); TD (théorie de l'abstraction des données); PD (pratique de l'abstraction des données). Chaque sous - domaine (TP, PP, PD TD) est organisé en un ou plusieurs curricula. Chaque curriculum consiste en une série d'unités pédagogiques, et chaque unité pédagogique est associée à un ensemble de problèmes. Les connaissances latérales comprennent: architecture des ordinateurs, cours des langages de programmation, en particulier le langage Scheme, la complexité d'analyse, les techniques de génie logiciel, entre autres. Le formalisme adopté pour le modèle du domaine est une base de données structurée et inspirée du constructivisme et de la théorie des connaissances sociales de Vygotsky². L'idée est de permettre à l'apprenant d'acquérir et de construire des savoirs par le biais de l'interaction avec le système tuteur, qui est conçu dans le but de renforcer la participation active de l'apprenant dans le processus de l'apprentissage. Pour atteindre ce but, l'interaction est basée sur la coopération des activités de résolution de problèmes en combinant l'apprentissage par la pratique et l'apprenant en étant informé.

9) **ADIS [9]** est un exemple, conçu comme un outil pédagogique pour donner cours sur les structures de données. ADIS a la capacité d'afficher graphiquement des structures de données sur l'écran d'ordinateur ainsi que des graphiques permettant la manipulation de la structure de données créée. Un didacticiel incorporant des exercices, où les étudiants peuvent apprendre visuellement des algorithmes de base des structures de données. ADIS est entièrement mis en œuvre en Java, tout le tuteur réside à une applet Java qui est téléchargée et exécutée sur la machine cliente. Le modèle de l'élève réside sur le serveur. Le même élève peut accéder à

¹ Pour plus d'information sur le travail collaboratif, voir l'URL suivant: http://fr.wikipedia.org/wiki/Travail_collaboratif

² [http://fr.wikipedia.org/wiki/Constructivisme_\(psychologie\)](http://fr.wikipedia.org/wiki/Constructivisme_(psychologie))

l'instruction à des moments et endroits différents. ADIS et MathTutor partage l'utilisation des navigateurs Internet et un *modèle de l'élève centralisé*.

10) **RoboTA [10]** : l'architecture RoboTA est une colonie d'agents. Il a été élaboré par besoin afin de créer des laboratoires virtuels. L'un des prototypes, CyclePad par exemple, permet d'apprendre l'ingénierie de la thermodynamique. Les apprenants ont trouvé le design du système motivant. Ce critère de motivation les a permis d'apprendre les fondamentaux plus profondément. Les auteurs se sont vite rendus compte que le déplacement du modèle d'un agent simple au modèle d'une colonie d'agents (où certains agents spécialisés à fournir des services d'infrastructure pour le reste) permet au système de soutenir de multiples projets, avec seulement une petite quantité supplémentaire de complexité. L'un des plus importants aspects consistait à créer un système qui peut facilement être étendu. Dans le but de développer RoboTA utile et utilisable pour différents types d'applications, les concepteurs ont divisé le système en un serveur central et en agents pour application spécifique. Le coeur de RoboTA est système client - serveur, avec un serveur central, le PostOffice, et de multiples clients, ou agents. RoboTA utilise pour communiquer les TCP / IP pour envoyer et recevoir des messages entre le PostOffice et ses agents. Chaque composant a son propre port sur lequel il reçoit les messages entrants. Toute communication avec l'utilisateur passera par le PostOffice. Les utilisateurs ne pourront jamais communiquer directement avec tous ses agents. L'une des caractéristiques essentielles de la conception de RoboTA est qu'il devrait être facile à étendre les fonctionnalités du système. Cela permet à RoboTA d'être utilisé pour une grande variété d'objectifs. Ajouter un nouvel agent exige deux étapes fondamentales: (1) la rédaction d'une règle qui permet d'identifier l'agents approprié suite à un message entrant ; (2) l'écriture d'un agent qui traite le message e-mail, et génère un résultat pour l'utilisateur. Lors de la rédaction d'un nouvel agent, il existe trois étapes principales: - lecture du contenu du message reçu ligne par ligne ; traitement de l'entrée ; et la génération d'un contenu comme réponse à l'utilisateur. RoboTA est un projet nord-américain, une architecture d'une colonie d'agents visant à soutenir les tâches pédagogiques. Deux avantages essentiels fournis par cette architecture (1) la mise en commun de fonctions de communication par le biais du PosteOffice et offrir un Toolkit RoboTA afin de simplifier la création de nouveaux agents pour d'autres cours, et faciliter la fabrication plus de cours. (2) La capacité d'accueillir des agents de différentes machines RoboTA a un intérêt pratique qui peut être critique étant donné les contraintes budgétaires de nombreux établissements d'enseignement.

11) Le **MCOE**, un autre projet brésilien, utilise aussi la technologie multi-agents, mais en plus des agents réactifs, on utilise également des agents cognitifs, à l'instar du système MathTutor. L'approche SMA est utilisée pour traiter les fonctionnalités associées à l'architecture traditionnelle des modules des systèmes tuteurs. Le domaine est modélisé à l'aide d'un agent réactif utilisant les techniques de la machine automatique. L'élève et le tuteur sont modélisés avec des agents cognitifs au sens de l'architecture BDI (Belief / croyance, Désir, Intention)³. Un agent médiateur a été ajouté dans le but d'étendre l'architecture précédente dans le but de gérer l'énorme quantité d'informations générée au cours de l'interaction entre les agents réactifs et cognitifs.

2.3. Des idées innovantes

La plupart des projets présentés sont des plate-forme. Chacune d'elles sert, d'une part, à éditer les modules d'enseignement où on trouve des agents spécialisés pour l'assurances des auteurs afin d'accomplir cette tâche, de l'autre, un générateur de module (cours) destiné aux apprenants. Cette approche permet évidemment de rendre facile la production de logiciels pédagogiques à moindre coûts [11].

D'un projet à l'autre, les architectures diffèrent, des agents isolés, système à agents et systèmes multi-agents. On trouve des agents réactifs fonctionnels, isolés et indépendants, comme on trouve des agents cognitifs qui permettent d'assister les apprenants dans leur apprentissage en tenant compte de leurs caractéristiques individuelles. Les modèles de l'architecture classique des systèmes tuteurs sont peu respectés, tels que le modèle du domaine, le modèle de l'élève et le module pédagogique.

Nous soulignons qu'il y a des idées innovantes tels que (1) le modèle de l'élève est centralisé et peut stoker toutes les interactions de tout bord afin qu'elles soient exploitées par d'autres agents tels que l'agent pédagogique [12] et (2) la création dynamique d'agents nouveaux suite à chaque nouvelle connexion dans le but de personnaliser et individualiser l'interaction client - système.

On constate que c'est un champ d'étude et de recherche récent, très vaste mais peu d'applications réelles et de réutilisation.

Dans la section suivante, on va présenter une application que nous avons développée. Elle s'intéresse particulièrement sur la question de la conception et de l'implémentation d'agents cognitifs. On présente l'architecture du système SIOF, des éléments de l'apprentissage par renforcement (RL), puis RL et SMA. Mon souhait que la présentation de cette application soit utile et un plus à côté des systèmes présentés en haut (§2.2.).

3. Système d'information orienté vers la formation (SIOF)

3.1. Présentation de l'architecture du système

³ Pour plus d'information sur BDI, voir http://fr.wikipedia.org/wiki/Mod%C3%A8le_cognitif

Le système tuteur peut avoir quatre agents, le premier se charge du modèle de domaine (Environnement), le deuxième représente le module pédagogique (Agentpeda), le troisième représentant le modèle de l'élève (Elevemodele) et le quatrième se charge du module de communication (Communication). Nous étudions ici et maintenant que la partie tutoring du système.

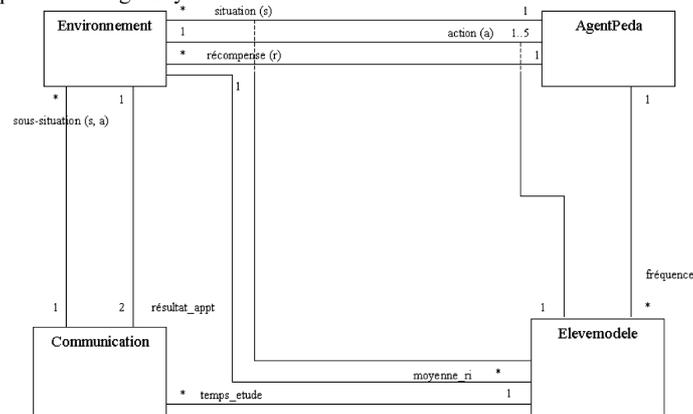


Fig. 1 : Architecture du SIOF

L'environnement envoie une situation (état) à l'agent pédagogique qui agit sur cet état (situation) par le choix d'une action et suite à cette opération l'environnement réagit en envoyant une récompense (positive ou négative) à l'agent pédagogique suivant la qualité du choix de l'action. La qualité de l'action de l'agent est déterminée en fonction du résultat de l'apprentissage de l'élève (succès, échec). Si la sous-situation identifiée par (s, a) est apprise par succès, alors une récompense positive est envoyée à l'agent, sinon une récompense négative [12].

La situation courante, l'action, la récompense et la situation suivante sont envoyées continuellement au modèle de l'élève afin de construire le parcours d'étude de chaque acteur. Le parcours ne sera enregistré que si l'acteur termine l'étude de la séquence pédagogique, sinon il sera ignoré.

3.2. Génération dynamique d'agents pédagogiques

L'enseignement et l'apprentissage des apprenants peut être n'importe où et n'importe comment, synchrone ou asynchrone. Notre but est que les caractéristiques particulières des apprenants tels que le niveau d'étude entre autres doivent être prises en charge par le système tuteur via des agents spécialisés afin de produire des parcours pédagogiques adaptatifs et individualisés.

Dans cette optique, le système tuteurs doit avoir, d'une part, un agent qui se charge du modèle de l'élève et d'une façon centralisé. Rien n'échappe en matière d'interaction entre le système - en particulier l'agent modèle de l'élève - et ses utilisateurs. De l'autre, un processus de génération dynamique d'agents pédagogiques se déclenche à chaque fois qu'un apprenant se connectant au système, et un agent nouveau se crée automatiquement. L'apprenant va être assisté par un agent qui lui individualise et personnalise son parcours d'apprentissage en tenant compte de ses caractéristiques.

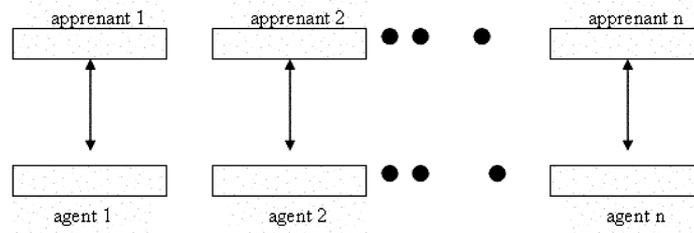
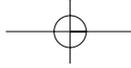


Fig. 2 : Apprenant – et génération d'agents

Centralisation du modèle de l'apprenant et la génération dynamique d'agents pédagogiques sont la clé principale de l'adaptabilité du système tuteur à ses utilisateurs potentiels.

3.3. Apprentissage par renforcement (RL)

L'apprentissage par renforcement (RL) est l'étude de la façon dont les animaux et les systèmes artificiels peuvent apprendre à optimiser leur comportement face aux récompenses et aux punitions. On a développé des algorithmes de RL qui sont étroitement liés aux méthodes de la programmation dynamique, qui est une approche générale du contrôle optimal [13]. On a observé des phénomènes de RL dans des études psychologiques du comportement animal, dans des investigations neurobiologique, etc. [14]. Une façon avec laquelle les animaux



acquièrent des comportements complexes est par l'apprentissage pour obtenir des récompenses et pour éviter des punitions. Pour ce type d'apprentissage, la théorie de RL est un modèle de calcul formel.

Le paradigme de l'apprentissage par renforcement standard, un agent est connecté à son environnement par perception et action. Un agent apprenant (un animal, un robot, etc.) observe à plusieurs reprises l'état de son environnement et choisit ensuite et exécute une action. L'exécution de l'action change l'état du « monde » et l'agent obtient une récompense numérique immédiate en conséquence. Les recettes positives sont appelées « récompenses » et les recettes négatives sont appelées « punitions ».

L'apprentissage par renforcement se compose d'un ensemble de concepts et d'algorithmes. RL ne se définit pas par une certaine classe d'algorithmes mais par le problème qu'il cherche à résoudre, celui du contrôle optimal.

RL est traditionnellement définie dans le cadre des processus de décision de Markov (MDP). Bellman a fondé la théorie de MDPs [15][16] par l'unification des travaux précédents au sujet de l'analyse séquentielle, des fonctions de décision statistique [17], et les modèles des jeux dynamiques pour deux personnes [18]. Un MDP est un quadruplet (S, A, P, R) tels que S est l'ensemble des états, A est l'ensemble des actions, et P et R sont respectivement les distributions de probabilités et de récompenses.

Comme l'agent interagit avec son environnement, la PDM décrit une trajectoire dans l'ensemble des états. Cette trajectoire dépend des actions choisies par l'agent. Ainsi, toute l'histoire d'un MDP est une séquence de paires (état, action) qui est composée des états qui ont été visités et des actions antérieures qui ont été retenues. Pour résumer, les algorithmes de RL apprennent une politique de contrôle optimale à partir d'une base de données d'actions réciproques (interactions).

En conséquence, RL se distingue aux autres paradigmes de l'apprentissage par trois grandes caractéristiques:

- La définition implicite d'un objectif par le biais des renforcements.
- L'aspect temporel de la tâche : une décision peut avoir un impact à long terme, à la fois sur la dynamique des systèmes et sur les récompenses acquises.
- Le protocole de l'apprentissage par essais et erreurs qui contraste avec l'apprentissage supervisé et non supervisé.

3.4. Méthodes de recherche des politiques optimales pour simple agent

Les approches de l'apprentissage par renforcement peuvent être divisées en deux méthodes, indirecte et directe. La première construit un modèle interne du processus qui est contrôlé et ensuite calcule la stratégie optimale basée sur ce modèle, alors que la seconde (directe) n'a pas besoin d'un modèle, mais apprend directement à partir des transitions observées.

Il est souvent mentionné que les méthodes d'apprentissage par renforcement indirectes apprennent plus vite que les méthodes directes, car les méthodes indirectes peuvent réutiliser les connaissances stockées dans les modèles internes, alors que les méthodes directes interagissent intensivement avec le processus de contrôle. L'inconvénient des méthodes indirectes est le fait de stocker un modèle interne. Les exigences de stockage d'un tel modèle augmentent quand le nombre de variables des états s'amplifie. Là encore, l'architecture des approximations peuvent être utilisées pour réduire les exigences de stockage [21].

Le terme « programmation dynamique » (DP) désigne un ensemble d'algorithmes qui peut être utilisé pour calculer les stratégies optimales des problèmes de la décision de Markov. Deux algorithmes de base sont connus, policy iteration et value iteration. Policy iteration calcule Q^{*} et la politique optimale π^{*} , alors que value iteration calcule le vecteur V^{*} .

L'apprentissage par renforcement tente de rapprocher la programmation dynamique, où les observations faites sur le processus de contrôle sont utilisées à la place d'un modèle. Un compromis entre l'apprentissage par renforcement et la programmation dynamique est donnée par les méthodes indirectes de RL. Les méthodes indirectes de RL construisent un modèle interne du processus et appliquent la PD à ce modèle. En revanche, les méthodes directes de RL n'ont pas besoin de modèle, mais apprennent directement à partir des observations.

3.5. Structure générale de l'algorithme Model-Based RL (Méthode Indirecte)

Nous allons présenter la structure générale de l'algorithme qui permet de calculer le modèle de l'environnement (P, R), puis la politique optimale via le calcul des fonctions Q*.

Notation :

- ✓ $C(s, a, s')$: le nombre de transition de l'état (s) à l'état (s') après l'exécution de l'action (a) ;
- ✓ $C(s, a)$: le nombre de fois l'agent a exécuté l'action (a) à l'état (s) ;
- ✓ $r(s, a, s')$: la somme des récompenses reçues par l'agent après avoir exécuté l'action (a) dans l'état (s) pour transiter à l'état (s').
- ✓ θ est un infiniment petit, et γ est un paramètre compris entre 0 et 1.
- ✓ $\theta = 0.0000000001$.
- ✓ $\gamma = 0.9$.

Algorithme :

1. Recueillir M interactions ($s_t, a_t, r_{t+1}, s_{t+1}$) ;



2. Calculer $C(s, a, s')$;
3. Calculer $C(s, a)$;
4. Calculer $r(s, a, s')$;
5. $P(s, a, s') = C(s, a, s') / C(s, a)$;
6. $R(s, a, s') = r(s, a, s') / C(s, a, s')$;
7. Initialiser $Q^*(s, a)$ à 0, pour tout $(s, a) \in S \times A$
8. Répéter
9. $\Delta = 0$
10. pour $s = 1$ à $|S|$ faire
11. pour $a = 1$ à $|A|$ faire
12. $Q^* = Q^*(s, a)$;
13. $Q^*(s, a) = \sum_{s'} P(s, a, s') [R(s, a, s') + \gamma \cdot \max_{a'} Q^*(s', a')]$;
14. $\Delta = \max(\Delta, \text{abs}(Q^* - Q^*(s, a)))$;
15. a suivant;
16. s suivant;
17. Jusqu'à ($\Delta < \theta$)

3.6. RL et SMA

Dans la technique de l'espace action jointe, l'apprentissage se passe dans l'espace résultat du produit de l'ensemble des états S , et une collection des ensembles d'action : A_1, \dots, A_n (un ensemble pour chaque agent). La distribution des probabilités de transition est définie de: $S \times A_1 \dots \times A_n \times S \rightarrow [0, 1]$ telle que pour chaque tuple (s, a_1, \dots, a_n, s') , on associe une probabilité et à chaque agent reçoit une récompense associée, définie par la fonction de récompense $R: S \times A_1 \dots \times A_n \rightarrow \mathbb{R}$. C'est le modèle sous-jacent des jeux stochastiques, aussi appelés Jeux de Markov [19]. Dans cette approche, on est sûr que l'influence d'un agent sur tous les autres agents peut être modélisé de manière à ce que la propriété de Markov tient toujours. Ceci, combinée avec une solution unique, tels que la notion d'équilibre de Stackelberg bootstrap, est généralement réalisé dans les techniques de RL [20].

Dans l'apprentissage par renforcement, le comportement d'un agent simple est spécifié par une fonction (politique). Dans les jeux stochastiques, les politiques sont séparées pour chaque agent et l'objectif de chaque agent est de trouver une politique d'équilibre, c'est-à-dire, une politique qui a la meilleure réponse vis-à-vis des politiques « adversaires ».

Les agents indépendants de RL essaient d'optimiser leur comportement sans aucune forme de communication avec les autres agents. Ils utilisent seulement le feedback reçu de l'environnement. Ces agents indépendants peuvent utiliser les algorithmes traditionnels de RL, élaborés, dans le cas stationnaire, pour un simple agent. Soulignons que le feedback provenant de l'environnement est généralement dépendant d'une combinaison d'actions prises par multiple agents, et non seulement par un seul agent. Dans ce cas, la propriété de Markov ne détermine plus, et le problème devient dépendant et non stationnaire.

Dans les environnements multi-agents, si le comportement des autres agents converge, c'est-à-dire, la distribution de sélection des actions devient stationnaire à la limite, la règle Q-learning (mise à jour) converge vers la fonction optimale Q^* avec une probabilité un.

Dans de nombreuses applications réelles où les techniques de RL sont utilisées dans les SMA, les techniques RL pour un simple agent sont directement appliquées. Bien que certaines hypothèses qui sous-tendent ces modèles d'apprentissage sont violés, les méthodes marchent étonnamment et dans de nombreux cas [20].

4. Conclusion

SMA est un champs d'étude et de recherche en pleine expansion où ses applications se multiplient dans divers domaines, y compris l'enseignement et la formation. Théoriquement, les SMA sont liés à la fois à la théorie des jeux et à l'apprentissage par renforcement, associant leur modèle et leur technique.

Dans le domaine de l'enseignement, les SMA essaient de suivre le même itinéraire qu'à parcouru les systèmes tuteurs intelligents (ITS) en matière de problématique tels que la modélisation de l'élève, le fameux problème des stratégies pédagogiques et le développement des logiciels destinés à l'enseignement à moindre coût. L'association des SMA et la technologie Web dans des applications ont suscité une nouvelle problématique, celle des connexions au système par multi-utilisateurs à la fois, donc comment l'utilisateur (l'élève) va être pris en charge ?.

De notre part, nous avons élaboré une architecture d'un système tuteur à quatre agents, en plus la création dynamique d'agent à la demande de toute nouvelle connexion. Nous soulignons qu'il y a des idées nouvelles telles que (1) le modèle de l'élève est centralisé et peut stocker toutes les interactions de tout bord afin qu'elles soient exploitées par d'autres agents tels que l'agent pédagogique et (2) la création dynamique des agents nouveaux suite à chaque nouvelle connexion dans le but de personnaliser et individualiser l'interaction client – système.

Bibliographie

- [1] Webber, C., Bergia, L., Pesty, S., Balacheff, N. « The Baghera project: a multi-agent architecture for human learning », Proceedings of the Workshop Multi-Agent Architectures for Distributed Learning Environments, AIED2001, San Antonio, TX, USA. pp. 12-17, 2001. <http://julita.usask.ca/mable/webber.pdf>.
- [2] Rosa M. Viccari, Demetrio A. Ovalle, Jovani A. Jim'enez; ALLEGRO: Teaching/Learning Multi-Agent Environment using Instructional Planning and Cases- Based Reasoning (CBR). <http://www.clei.cl/cleiej/papers/v10i1p4.pdf>
- [3] Adriana Postal, Eliane Pozzebon, Luciana B. Frigo, Guilherme Bittencourt, Janette Cardoso; MATHTUTOR: A MULTI-AGENT INTELLIGENT TUTORING SYSTEM. <http://w3.univ-tlse1.fr/irit/soc/perso/cardoso/cardoso-aiai.pdf>
- [4] Patricia Augustin Jaques, Everton Bocca, Rosa Maria Vicari; Considering Student's Emotions in Computational Educational Systems; XIV Simpósio Brasileiro de Informática na Educação - NCE - IM/UFRJ 2003; <http://www.nce.ufrj.br/sbie2003/publicacoes/paper54.pdf>
- [5] Louise Jeanty de Seixas, Cecília Dias Flores, Rosa Maria Vicari, Marcelo Ladeira; An Architecture for an Intelligent Learning Environment with a constructivist approach; http://www.inf.ufrgs.br/~dflores/publicacoes/SEAMED_ITS2002.pdf
- [6] Clara-Inés Peña, Jose-L. Marzo Josep-Lluís de la Rosa; Intelligent Agents in a Teaching and Learning Environment on the Web, 2001; <http://bcds.udg.es/papers/icalt94.pdf>
- [7] Ricardo Azambuja Silveira, Eduardo Rodrigues Gomes; FIPA COMPLIANT PEDAGOGICAL AGENTS IN DISTRIBUTED INTELLIGENT LEARNING ENVIRONMENTS. <http://minerva.ufpel.edu.br/~ergomes/trabalhos/E-society2002.pdf>
- [8] Adriana Postal, Eliane Pozzebon, Luciana B. Frigo, Guilherme Bittencourt, Janette Cardoso; MATHTUTOR: A MULTI-AGENT INTELLIGENT TUTORING SYSTEMS. 2004. <http://w3.univ-tlse1.fr/irit/soc/perso/cardoso/cardoso-aiai.pdf>
- [9] Kai Warendorf; ADIS - An Animated Data Structure Intelligent Tutoring System or Putting an Interactive Tutor on the WWW'; 1997. http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Warendorf/Warendorf.html
- [10] Sven E. Kuehne Kenneth D. Forbus; RoboTA: An Agent Colony Architecture for Supporting Education; http://www.eecs.northwestern.edu/docs/techreports/2002_TR/nwu-cs-02-10.pdf
- [11] BENNANE Abdellah; L'apprentissage par renforcement et la conception d'un système tuteur adaptatif ; 7ème colloque Africain sur la recherche en informatique (22-25 Nov. 2004) (CARI'04) ; <http://www.cari-info.org>
- [12] BENNANE Abdellah; Agent d'apprentissage et la gestion du processus pédagogique ; 8ème colloque Africain sur la recherche en informatique (6-9 Nov. 2006) (CARI'06) ; http://www.cari-info.org/prog_cari.php
- [13] [Sutton98]. Sutton R., Barto A. G. Reinforcement Learning, A Introduction; A Bradford Book, 1998.
- [14] [DAYAN2001] : Peter Dayan & Christopher JCH Watkins. Reinforcement Learning. Encyclopedia of Cognitive Science, 2001; <http://www.gatsby.ucl.ac.uk/~dayan/papers/dw01.pdf>
- [15] R. Bellman. Dynamic Programming. Princeton University Press, 1957.
- [16] R. Bellman. A Markovian decision process. Journal of Mathematics and Mechanics, 6:679-684, 1957.
- [17] A. Wald. Statistical Decision Functions. John Wiley, 1950.
- [18] L.S. Shapley. Stochastic games. In Proc. of the National Academy of Sciences of the United States of America, volume 39, pages 1095-1100, 1953.
- [19] KARL TUYLS and ANN NOWÉ; Evolutionary game theory and multi-agent reinforcement learning; *The Knowledge Engineering Review*; Vol 20(01):63-90, 2005. Cambridge University Press Printed in the United Kingdom. <http://www.cs.unimaas.nl/k.tuyls/publications/b2hd-TuylsKER05.html>
- [20] Könönen, V: Multiagent Reinforcement Learning in Markov Games: Asymmetric and Symmetric Approaches. PhD thesis, Helsinki University of Technology, Helsinki, Finland, 2004. <http://www.cis.hut.fi/kononen/>
- [21] *Martin Appl*; Model-Based Reinforcement Learning in Continuous Environments, PhD thesis, December 2000, Technical University of Munich. <http://www.igi.tugraz.at/ril-toolbox/papers/ContinuousStateLearning/FuzzyModelBasedLearning.pdf>