



(12) BREVET D'INVENTION

(11) N° de publication : **MA 39613 A1** (51) Cl. internationale : **G09B 5/08**

(43) Date de publication :
31.08.2018

(21) N° Dépôt :
39613

(22) Date de Dépôt :
29.12.2016

(71) Demandeur(s) :
Université Mohammed V RABAT , Avenue des Nations Unies, Agdal, bp 8007 NU, Rabat, 10000, Maroc (MA)

(72) Inventeur(s) :
ESSAIDI Mohamed ; EN-NAIMI ZOUHAIR

(74) Mandataire :
KARTIT ZAID

(54) Titre : **Système de tutorat intelligent pour l'environnement e-Learning**

(57) Abrégé : Cette invention est un système de tutorat intelligent pour l'environnement e-Learning basé sur le raisonnement dynamique incrémental et les multi-agents. Pour arriver à ce résultat nous avons adopté une approche de raisonnement à partir de cas dynamiques à base d'agents pour les situations dynamiques (situations en cours de progression) fondé sur les expériences passées. Cette approche est capable d'assurer la prédiction des situations dynamiques, en se basant sur les expériences passées similaires. Nous signalons aussi le caractère générique de cette approche élargissant son champ d'application à plusieurs domaines tels que e-service, elearning, e-commerce, etc. Dans cette approche, les expériences passées sont modélisées sous la forme des traces qui comprennent les productions et les actions résultantes de l'interaction de l'apprenant avec la plateforme d'apprentissage. Le système proposé a été analysé, conçu et modélisé suivant la méthodologie de développement de système multi-agents. Ses agents ont été développés en langage de programmation JAVA sur la plateforme SMA/JADE .Par ailleurs, l'acquisition automatique des connaissances du domaine par notre système permet d'améliorer la performance de ce dernier

Abrégé :

Cette invention est un système de tutorat intelligent pour l'environnement e-Learning basé sur le raisonnement dynamique incrémental et les multi-agents. Pour arriver à ce résultat nous avons adopté une approche de raisonnement à partir de cas dynamiques à base d'agents pour les situations dynamiques (situations en cours de progression) fondé sur les expériences passées. Cette approche est capable d'assurer la prédiction des situations dynamiques, en se basant sur les expériences passées similaires. Nous signalons aussi le caractère générique de cette approche élargissant son champ d'application à plusieurs domaines tels que e-service, e-learning, e-commerce, etc.

Dans cette approche, les expériences passées sont modélisées sous la forme des traces qui comprennent les productions et les actions résultantes de l'interaction de l'apprenant avec la plateforme d'apprentissage. Le système proposé a été analysé, conçu et modélisé suivant la méthodologie de développement de système multi-agents. Ses agents ont été développés en langage de programmation JAVA sur la plateforme SMA/JADE. Par ailleurs, l'acquisition automatique des connaissances du domaine par notre système permet d'améliorer la performance de ce dernier

Titre : Système de tutorat intelligent pour l'environnement e-Learning

Description

L'évolution continue des besoins de formation, vers plus d'efficacité, de flexibilité, d'interaction, et avec l'avènement des technologies de l'information et de communication (TIC) ont favorisé l'émergence d'outils pédagogiques informatiques tels que le e-Learning et les MOOCs, dont l'objectif est de faciliter davantage l'accès à des formations et de les rendre plus attractives et donc plus facilement assimilables par les apprenants. Le e-Learning est l'introduction des TIC dans le domaine de la formation. Tous les publics sont visés : étudiants, employés, techniciens, cadres, ainsi que tous les domaines de formation et d'apprentissage. Cette nouvelle approche et technologie de formation et apprentissage permet d'une part à l'apprenant de travailler où et quand il le souhaite, il lui permet aussi d'avancer à son rythme, de commencer une formation quand il en a besoin, et d'autre part il permet aux enseignants-tuteurs de faire le suivi d'un ou de plusieurs apprenants, de participer et d'animer des activités en ligne, de communiquer avec les apprenants à tout moment, et enfin de développer un travail collaboratif de groupe entre les apprenants.

Le e-Learning est aujourd'hui en plein essor et des avancées ont été réalisées au cours de ces dernières années dans le domaine. Ceci permet par exemple l'interopérabilité des plateformes d'apprentissage, l'évaluation des apprenants en ligne, le lien entre apprenants dans un environnement non supervisé et dispersé géographiquement, et la possibilité pour l'apprenant de choisir un horaire d'apprentissage souple. L'intégration de diverses composantes du processus d'apprentissage et l'adaptation selon le profil de l'apprenant joue un rôle important et déterminant dans le processus d'apprentissage. Malgré ce rôle déterminant de l'adaptation selon le profil de l'apprenant, il y a lieu de noter que jusqu'aujourd'hui, les fonctionnalités associées à l'adaptation d'apprentissage selon le niveau et la capacité de l'apprenant sont encore insuffisantes, voire non efficaces.

Les systèmes tutoriels intelligents (STI) sont des systèmes informatiques fournissant un enseignement et un encadrement personnalisés. Ils contiennent des modèles de contenu d'apprentissage qui précisent quoi enseigner en utilisant des bases de connaissances ainsi que des stratégies d'enseignement qui précisent comment l'enseigner [Wenger, 1987]. Les STI sont des outils conçus pour aider et faciliter la tâche d'apprentissage de l'apprenant.

Depuis quelques années, les travaux sur les Systèmes Tutoriels Intelligents s'intéressent à la conception et à la réalisation de systèmes capables de reproduire le comportement d'un tuteur intelligent capable de permettre un apprentissage adapté au rythme et aux capacités de l'apprenant. Il existe de nombreux travaux de recherche qui s'intéressent à la conception et à la réalisation de systèmes informatiques permettant d'assister l'apprenant dans son apprentissage. On trouve par exemple des agents tuteurs ou pédagogiques qui accompagnent l'apprenant en lui proposant des activités de remédiation [Frasson et al., 1998]. On trouve aussi des agents de support à la collaboration de groupe en apprentissage [Constantino et al., 2001] encourageant, quant à eux, la participation des élèves et facilitent la discussion entre

eux. On trouve également des solutions basées sur les agents qui intègrent et cherchent à faire coopérer différents Systèmes Tuteurs Intelligents [Brusilovski 1997]. La plateforme Baghera [Webber et Pesty 2002], est un Environnement Informatique pour l'Apprentissage Humain (EIAH) distant, exploite les concepts et les méthodes de l'approche multi-agents. Baghera assiste les élèves dans leur travail de résolution d'exercice en géométrie, et ils peuvent interagir avec d'autres élèves ou enseignants. Les enseignants peuvent connaître l'état d'avancement des travaux des élèves afin d'intervenir si c'est nécessaire. SAAID [Labouidya et al., 2008], Système d'Aide à l'Apprentissage Individualisé en Autoformation à Distance, est un système qui se base sur l'approche Multi-agents dans le but de produire une assistance pour l'apprentissage dans le domaine des télécommunications et des réseaux. HeapMotiv [Derbali, 2013], est un système qui se base sur des éléments motivationnels issus de jeux vidéo et des stratégies motivationnelles d'un modèle théorique de la motivation dans l'objectif de soutenir les apprenants à atteindre des hauts niveaux de motivation, de persévérance et de performance. Enfin, Le Corre [Le Corre, 2013], propose CHRYSAOR, système tutoriel intelligent pour les environnements virtuels d'apprentissage humain. C'est un couplage entre la formation en réalité virtuelle et les Systèmes Tutoriels Intelligents. Les concepts proposés dans ce modèle permettent au formateur de modifier les rôles et les affectations (humain ou agent autonome) afin de rendre le système modulaire. CHRYSAOR a été appliqué dans une plateforme de réalité virtuelle à un instrument de diagnostic en hémostasie utilisé en milieu hospitalier.

Les STI forment un courant particulier du e-learning. Ils ont pour principe la personnalisation de l'apprentissage en s'intéressant aux besoins spécifiques de l'apprenant. Ils évaluent et diagnostiquent les problèmes et difficultés de l'apprenant en apprentissage, et fournissent l'aide nécessaire, Ainsi, les STI placent l'apprenant au centre de la situation apprentissage. Les STI offrent des capacités de raisonnement et de résolution, passant par le recueil de connaissances sur l'apprenant. Ils évaluent les connaissances acquises par l'apprenant, en comparant ses activités et les informations sur le domaine, ils proposent alors des assistances adaptées.

Le e-Learning est le support et environnement d'apprentissage pour les apprenants alors que STI est l'adaptation de l'apprentissage selon la capacité et rythme des apprenants. La figure 1 présente le lien STI/E-learning .

Il existe principalement quatre acteurs (figure 2) qui interviennent autour de la plateforme d'apprentissage, chacun avec ces rôles et niveaux d'implication : l'apprenant, le tuteur, le concepteur et les contenus.

II.4 Systèmes Tutoriels Intelligents : notre objectif

L'accompagnement et le suivi individualisé de l'apprenant ont pour objectifs l'anticipation d'un éventuel abandon de l'apprenant ou encore l'échec et la démotivation de celui-ci. Le système que nous proposons dans cette invention, permet d'analyser le parcours (trace) de l'apprenant afin d'adapter l'apprentissage selon son rythme et ses capacités.

Les traces des activités passées d'apprentissage seront la source de connaissances pour notre processus d'adaptation d'apprentissage. Elles sont stockées dans la base de cas. Les traces comprennent les productions et actions résultantes de l'interaction de l'apprenant avec la plateforme e-Learning. Après le démarrage d'une session d'apprentissage par un apprenant, ce dernier laisse ses traces sur le serveur d'apprentissage. Ces dernières peuvent être considérées comme un ensemble d'actions effectuées l'une après l'autre par l'apprenant. Cette suite d'évènements peut alors être assimilée à une séquence temporelle d'actions. Donc le problème traité est un cas cible dynamique. L'objectif s de cette invention sera donc de représenter cette situation (cas cible), prendre en compte le changement dynamique de cette situation, prévoir les évolutions possibles de cette dernière, et réagir en fonction des situations particulières (en fonction du profil des apprenants). Ceci peut être réalisé en s'appuyant sur des situations d'apprentissages antérieures effectuées dans des conditions semblables et dont on connaît les conséquences. Il s'agit alors d'utiliser le raisonnement à partir de cas pour les situations dynamiques.

Les systèmes classiques souffrent de limites dans la gestion des paramètres dynamiques et ils ne sont pas capables de détecter automatiquement l'évolution de ses paramètres ainsi que s'adapter aux changements de la situation. Ils considèrent chaque changement dans la situation en cours de traitement comme un nouveau cas cible, et par la suite ils exécutent le même cycle statique à nouveau. Or le changement concerne toujours le même cas cible, donc l'utilisation du cycle classique de raisonnement semble non adéquate, par conséquent nous avons besoin d'un cycle dynamique pour traiter un cas cible dynamique. En effet plusieurs va-et-vient entre les étapes du cycle peuvent se produire. Notre système est capable de ré-exécuter une étape antérieure du cycle suite à un changement dans le cas cible. Les apprenants laissent leurs traces en fichier Log. Ces traces permettent d'alimenter notre système via des agents d'interface, ensuite notre système procède à l'analyse de l'évolution de ces traces afin de détecter rapidement les situations de réussite ou d'échec dans lesquelles se trouvent les apprenants. Notre système permet à tout moment de prédire les prochaines tâches à exécuter par l'apprenant.

L'approche proposée dans cette invention permet d'apporter des éléments de réponses aux difficultés liées au développement et aux limites des systèmes tutoriels intelligents :

- la difficulté d'acquisition des connaissances du domaine pour les systèmes STI rend le développement de ces derniers coûteux et difficile [Diattara, 2013]. Notre approche, ne

nécessite aucune connaissances du domaine ce qui constitue un atout pour sa mise en pratique.

- selon Le Corre [Le Corre, 2013] l'une des principales limites des STI est le manque de généralité. Par exemple, les modifications de STI seront nécessaires chaque fois que nous changeons le domaine d'apprentissage. De plus, selon Mendelsohn et Dillenbourg [Mendelsohn et Dillenbourg, 1993], cité par [Croset, 2009], les techniques des STI sont difficilement applicables à des domaines moins bien structurés que la programmation, les mathématiques, ... Nous proposons une approche générique indépendante de tout domaine d'apprentissage.
- enfin dans notre approche, on n'a pas besoin d'algorithmes de calcul, en dehors de la mesure de similarité, qui est facile à mettre en application.

Notre approche permet à la plateforme e-Learning de s'adapter à la diversité des apprenants : nous proposons d'utiliser cette plateforme comme support et environnement d'apprentissage pour les apprenants. Celle-ci permet la mise à disposition des apprenants de ressources pédagogiques (textes, audio, vidéo, évaluations, ...). Elle permet aussi la communication entre les tuteurs et les apprenants (chat, forum). Mais elle n'offre pas une adaptation d'apprentissage selon les besoins des apprenants ce qui présente des faiblesses et des inconvénients majeurs par conséquent une démotivation des apprenants. Nous proposons alors notre invention comme complément à cette plateforme dans le but d'adapter l'apprentissage selon la capacité et rythme des apprenants.

Le prototype de cette invention est composé de deux grandes parties complémentaires :

- **Application multi-agents** : la programmation du système multi-agents. Le système proposé présente aussi des interfaces de visualisation pour qu'on puisse suivre et étudier l'évolution des agents, analyser leurs comportements et apporter les réglages nécessaires.
- **Application Web** : c'est un portail web, interface entre ce système et les différents utilisateurs (apprenants, tuteurs et administrateur).

L'architecture générale de ce système est présentée par la figure 3. Celui-là permet de rechercher les anciennes situations similaires à la situation en cours. L'architecture de ce système est constituée d'un ensemble d'agents en relations les uns aux autres, assurant chacun des tâches et rôles spécifiques. En ce qui concerne la réactivité et la cognition, les agents déployés auront des stimuli réactifs (transfert de messages, calcul de similarité entre des traces suite à la réception d'un message) ainsi que des stratégies de décisions. Chacun de ces agents, cherche à coopérer et collaborer avec les autres pour prédire une nouvelle situation. Ces agents seront aussi capables de s'adapter et d'apprendre de leur environnement, en particulier, ils apprennent et visent à construire une base des cas afin de bien coordonner leurs décisions. Pour agentifier cette approche, plusieurs contraintes ont été prises en considération

:

- chaque étape du cycle doit être associée à au moins un agent ;
- plusieurs agents d'interfaces ont été définis pour communiquer avec la plateforme sources des traces ;
- un agent est dédié au calcul de similarité entre le cas cible et les cas sources ;
- des agents de coordinations et vérifications ont été définis dans cette architecture ;
- Cette architecture contient trois couches d'abstraction, organisée selon les étapes du système ;
- pour la première couche de ce système et pour chaque utilisateur un agent et un seul est chargé de gérer les traces de l'utilisateur.

Le système doit représenter dynamiquement et en temps réel la trace cible, l'analyser et la rapprocher à des traces passées stockées dans une base de cas afin de prédire la trace cible. La trace prédite sera ajoutée à la base de cas afin d'enrichir la mémoire du système. Le rapprochement se base sur des valeurs de similarités stockées dans des tableaux de similarités. Ces tableaux contiennent les valeurs de similarités entre la trace cible et chacune des traces passées. Cette notion de similarité consiste en un nombre, représentant à quel point la trace cible et les traces passées sont semblables. Elles sont calculées par les agents du système. La plateforme générique proposée est présentée sur la Figure 3. Celle-ci peut être interfacée, (ou alimentée), avec n'importe quelle source de traces.

Le rôle des agents de la couche d'élaboration est de gérer les informations venant de l'environnement est de représenter dynamiquement et en temps réel les informations décrivant la situation courante. Cette couche est alimentée par les traces laissées par un utilisateur en interaction avec une plateforme.

La couche de remémoration contient les agents permettant la comparaison entre la situation en cours et les situations passées stockées dans la mémoire du système.

Le rôle des agents de la couche d'adaptation est de prédire la situation en cours par l'adaptation des expériences passées choisies par la couche de remémoration ainsi que de vérifier et d'évaluer le choix de ces expériences.

- Couche d'élaboration

La couche d'élaboration contient les trois agents suivants : Agent générateur, Agent de requête et Agent-Traces-C1,

Agent Générateur: Le but de cet agent est de créer et de mettre à jour les agents de traces existants (un nouvel agent trace est créé si aucun des agents existants ne s'intéresse à la dernière trace interceptée par l'agent générateur, sinon, l'agent qui détient cette trace se met à jour). Au départ, l'agent Générateur est en état d'écoute, il attend les traces de l'agent de requête, ensuite l'agent Générateur vérifie s'il s'agit d'une nouvelle trace et dans ce cas l'agent Générateur crée un nouvel Agents-Traces-C1. Dans le cas contraire l'agents-traces-C1 concerné par la trace sera mis à jour.

Agent de requête : Le rôle de l'Agent de Requête est d'établir le lien entre le système et l'environnement. Il alimente le système en informations, recueillies à partir des fichiers de traces (sous forme des fichiers Log.). L'un des buts de cet agent est de vérifier s'il y a un changement dans le fichier traces. Au début, l'agent de requête est en état d'écoute, il attend les demandes de traces de l'agent Générateur ou de l'agent analyseur, ensuite l'agent cherche les traces dans le fichier Log du serveur. Enfin, les traces seront envoyées à l'agent Générateur.

L'agent Traces-C1 : Les traces transmises au système par l'agent de requête, sont envoyées à l'agent Générateur. Ce dernier se charge de la création des Agent-Traces-C1 (Agent traces de la couche 1). Au début, l'Agent-Traces-C1 est en état d'écoute, il attend les traces de l'agent Générateur : si ces dernières concernent l'un des agents-traces-C1 alors ce dernier sera mis à jour, sinon un nouvel agent-Traces-C1 sera créé.

III.1.2 Couche de remémoration et d'Apprentissage

La couche de remémoration contient quatre agents : l'agent Analyseur, l'agent ILCSS, l'agent-Traces-C2 et l'agent d'apprentissage.

Agent ILCSS : Le rôle de cet agent est d'évaluer continûment la similarité entre la situation en cours et les différents scénarios stockés dans la mémoire en se basant sur la mesure de similarité ILCSS. L'agent ILCSS stocke les valeurs de similarités calculées entre la situation en cours et les scénarios passés dans la table de similarités. Il est chargé de réévaluer ces valeurs à chaque fois qu'il y a changement dans la situation en cours ou suite à la demande des autres agents.

Agent Analyseur : Le but de cet agent est de s'assurer et de vérifier s'il y a changement dans les agents des traces de la deuxième couche suite à l'arrivée de nouvelles informations, ceci d'une manière périodique. L'agent Analyseur demande à l'agent de requête de la première couche de vérifier s'il y a changement dans le fichier trace. Dans ce cas l'agent analyseur demande à l'agent ILCSS de mettre à jour la table de similarités du cas cible (pour chaque nouveau cas cible le système crée une nouvelle table de similarités afin de stocker tous les valeurs de similarités entre ce nouveau cas cible et les traces passées).

Agent-Traces-C2: Ces agents contiennent les mêmes informations et données que celles des Agents-Traces-C1. L'agent-Traces-C2 diffère par une abstraction de la trace en cours, décrit originellement par l'Agent-Traces-C1, pour rendre les séquences comparables aux traces passées stockées dans la mémoire. Pour le paramètre temps d'actions, il s'agit d'exprimer les variations entre deux mesures successives plutôt que leurs valeurs absolues. De plus, on applique un filtrage pour éliminer les données supplémentaires telles que l'adresse IP du l'ordinateur de l'utilisateur et nom de l'utilisateur dans les traces initiales de l'Agent-Traces-C1.

Agent d'Apprentissage : Cet agent permet d'ajouter la nouvelle trace dans la base des cas.

- **Couche d'adaptation et de décision**

Cette couche contient les trois agents suivants : agent Evalueur, Agent-Traces-C3 et l'Agent d'Adaptation.

Agent évaluateur : Le but de cet agent est d'évaluer la solution proposée et de s'assurer que la similarité entre la situation en cours et les expériences passées choisies par la couche de remémoration est suffisante, sinon cet agent demande l'intervention de l'agent ILCSS.

Agent-Traces-C3 : Le rôle de l'agent-Traces-C3 est de réaliser une liaison dynamique entre la trace cible et les traces passées similaires. Cette liaison est assurée par un pointeur qui contient la liste des expériences passées similaires à la situation en cours à un instant donné. Un pointeur est un objet variable contenant la liste des traces similaires à la trace cible. Il évolue au cours du temps, c'est-à-dire qui permet d'accéder à d'autres objets. L'avantage d'un pointeur c'est que la liste qu'il contient n'est pas statique et elle change d'une manière dynamique suite au changement des agents de traces, ceci en se basant sur la similarité calculée entre la situation en cours et les expériences passées. Donc, il se peut qu'une telle expérience passée soit dans la liste à un moment donné et qu'après un certain temps l'expérience perde sa place dans la liste des cas similaires au cas en cours de progression.

Agent d'Adaptation: Le rôle de l'agent d'adaptation est de sélectionner une ou plusieurs traces similaires à la trace en cours. Ceci en se basant sur la liste des traces similaires de l'agent-Traces-C3. Après le choix d'une ou plusieurs traces similaires, l'agent propose une réponse adaptée à la situation en cours.

- **Structure de cas**

Les traces des utilisateurs (les cas), désignent les informations qu'un dispositif numérique enregistre sur l'activité, la production, et toute action résultante de l'interaction de ses utilisateurs. Dans notre contexte les traces de l'utilisateur (TA) sont représentées par une collection des traits sémantiques $TA=U TS$. Un trait sémantique (TS) constitue les données élémentaires atomiques d'information de la situation [Boukachour, 2002]. Il est représenté par un triplé : $TS = (\text{objet}, (\text{qualification}, \text{valeur}) +)$. On note objet = O, qualification = Q et Valeur=V, $TS = (O,(Q,V)+)$. Les traces de l'utilisateur forment une séquence multidimensionnelle. Donc une trace peut être vue comme une liste de vecteurs ou une matrice multidimensionnelle.

Soit A une trace de dimensions $n \times d$, alors A peut s'écrire comme suit :

$$A = ((O_{A,1}, (Q_{A,1,1}, V_{A,1,1}), \dots, (Q_{A,1,d}, V_{A,1,d}), (O_{A,2}, (Q_{A,2,1}, V_{A,2,1}), \dots, (Q_{A,2,d}, V_{A,2,d}), \dots, (O_{A,n}, (Q_{A,n,1}, V_{A,n,1}), \dots, (Q_{A,n,d}, V_{A,n,d}))) \quad (1)$$

Dans notre système un cas cible ou trace cible représente une trace en progression d'un utilisateur. Une trace A à l'instant $t=t_i$ est une trace multidimensionnelle $(A)_{t=t_i}$ de la forme suivante :

$O_{A,1}$	$O_{A,2}$...	O_{A,t_i}
$(Q_{A,1,1}, V_{A,1,1})$	$(Q_{A,2,1}, V_{A,2,1})$...	$(Q_{A,n,1}, V_{A,t_i,1})$

...
$(Q_{A,1,d}, V_{A,1,d})$	$(Q_{A,2,d}, V_{A,2,d})$...	$(Q_{A,t_i,d}, V_{A,t_i,d})$

Les cas sources sont des traces des anciens utilisateurs et peuvent être représentées par la formule (1).

- Base de cas

La performance d'un système est mesurée par son temps de réponse nécessaire pour proposer une solution à un problème donné. Elle dépend de l'organisation de la mémoire et de la performance de la méthode de remémoration. Dans un système à mémoire plate, les cas sont rangés séquentiellement dans une simple liste. L'un des avantages de ce type d'organisation est que le processus d'apprentissage de nouveaux cas est simple, il s'agit d'un simple ajout du nouveau cas résolu à la mémoire. Cependant, sa grande limitation est que le temps de recherche augmente linéairement avec la taille de la base de cas. Pour ces raisons, nous avons opté pour une organisation hiérarchique pour notre base des cas.

Dans le système proposé, les traces passées sont stockées dans une structure hiérarchique contenant plusieurs niveaux. Le bas niveau est une simple mémoire plate organisée en groupes de cas similaires. Le haut niveau contient des index ou des clusters représentant les caractéristiques communes d'un groupe de cas similaires stockés dans le bas niveau. Dans notre base de cas chaque index est composé de deux parties : une partie statique contenant les objets et une autre dynamique contenant les qualifications : la composante comportementale d'un utilisateur à un instant donné. Cette composante comportementale référence un instant précis dans la production des traces de l'utilisateur. Elle regroupe l'ensemble d'activité, de production, et toute action résultante de l'interaction de l'utilisateur avec le serveur à un instant donné. La base de cas du système est représentée par la figure 4.

Parmi les étapes clefs citons la remémoration des cas similaires au cas cible et la mesure de similarité utilisée. Cependant, ces mesures proposées et utilisées dans les systèmes dynamiques ne permettent pas de comparer un cas cible en cours de progression avec des cas sources de la base des cas. Nous proposons une mesure de similarité capable de surmonter les limites des autres mesures. Dans la section suivante nous décrivons notre proposition de mesure de similarité, qui est une extension de la mesure de la plus longue sous-séquence commune (*Longest Common SubSequence LCSS*), notre mesure est intitulée *ILCSS (Inverse Longest Common SubSequence LCSS)*.

- Mesure et tableaux de similarités

Notre objectif est d'étendre l'approche LCSS à la comparaison de séquences temporelles, multidimensionnelles, hétérogènes et dynamiques, c'est-à-dire au moins l'une de ces séquences est en cours de progression. La mesure de similarité $ILCSS_{\delta,N}$ [Zouhair, 2015], [En-Naimi, 2013], que nous proposons diffère de celle de LCSS [Vlachos et al., 2002] à plusieurs niveaux :

1. La mesure de similarité LCSS permet de comparer deux séquences statiques : elle commence cette comparaison en commençant d'abord par la comparaison des derniers

points. Dans notre contexte, nous avons besoin de commencer la comparaison des premiers vecteurs des séquences temporelles et dans le même ordre sur l'axe du temps, afin de comparer la similarité entre une trace en cours de progression et des traces passées ;

2. Notre proposition $ILCSS_{\delta,N}$ mesure de manière continue la similarité entre les cas sources et le cas cible, en utilisant une formule récursive c'est-à-dire on réutilise les résultats trouver à l'instant $t=t_i$ dans les calculs pour l'instant $t=t_{i+1}$: la mesure entre la trace cible et les traces sources à l'instant $t=t_{i+1}$ dépend des résultats à l'instant $t=t_i$;
3. Contrainte temporelle supplémentaire est nécessaire pour éviter certains cas particuliers, tels que le cas où une grande partie des vecteurs de trace en cours de progression est similaire à un ensemble de ceux d'une trace passée, sans recouvrement, et dans le même ordre sur l'axe du temps.

Dans ce qui suit, nous présentons notre proposition de mesure de similarité dynamique $ILCSS_{\delta,N}$.

Soient A et B deux traces de dimensions $n \times d$ et $m \times d$ respectivement :

$$A = ((O_{A,1}, (Q_{A,1,1}, V_{A,1,1}), \dots, (Q_{A,1,d}, V_{A,1,d}), (O_{A,2}, (Q_{A,2,1}, V_{A,2,1}), \dots, (Q_{A,2,d}, V_{A,2,d})), \dots, (O_{A,n}, (Q_{A,n,1}, V_{A,n,1}), \dots, (Q_{A,n,d}, V_{A,n,d}))) ;$$

Et

$$B = ((O_{B,1}, (Q_{B,1,1}, V_{B,1,1}), \dots, (Q_{B,1,d}, V_{B,1,d}), (O_{B,2}, (Q_{B,2,1}, V_{B,2,1}), \dots, (Q_{B,2,d}, V_{B,2,d})), \dots, (O_{B,m}, (Q_{B,m,1}, V_{B,m,1}), \dots, (Q_{B,m,d}, V_{B,m,d}))) ;$$

Pour la trace A, soit $Tail(A)$ la trace :

$$Tail(A) = (O_{A,2}, (Q_{A,2,1}, V_{A,2,1}), \dots, (Q_{A,2,d}, V_{A,2,d})), \dots, (O_{A,n}, (Q_{A,n,1}, V_{A,n,1}), \dots, (Q_{A,n,d}, V_{A,n,d}))) ;$$

$Tail(A)$ est égal à la trace A privée de son premier vecteur.

Le but est de compter le nombre de couples de vecteurs considérés comme similaires lorsqu'on parcourt les deux traces comparées.

La similarité entre deux vecteurs $(V_{A,i,1}, V_{A,i,2}, \dots, V_{A,i,d})$ et $(V_{B,j,1}, V_{B,j,2}, \dots, V_{B,j,d})$ est déterminée selon un seuil δ : un vecteur $(V_{A,i,1}, V_{A,i,2}, \dots, V_{A,i,d})$ d'une trace A est considéré similaire à un vecteur $(V_{B,j,1}, V_{B,j,2}, \dots, V_{B,j,d})$ d'une autre trace B si

$$|V_{A,i,k} - V_{B,j,k}| \leq \delta \text{ pour chaque } k = 1, \dots, d.$$

On définit aussi un entier N qui sert à contrôler l'écart de temps maximum entre deux vecteurs de chacune des traces pour qu'ils puissent être similaires.

Soient les deux traces (séquences temporelles) (A, B), un réel δ et un nombre entier N. On définit la mesure de similarité entre les deux traces, qu'on note $ILCSS_{\delta,N}(A,B)$, selon le processus récursif suivant : on initialise le processus par les deux premiers vecteurs des traces (A, B) comparées. Si l'une des deux traces est vide alors la mesure de similarité est nulle et le processus s'arrête. Sinon, si les premiers vecteurs des traces sont similaires, la mesure de similarité vaut 1 plus la similarité calculée entre les deux traces privées de leurs premiers vecteurs ($Tail(A)$, $Tail(B)$). Dans le cas où les deux premiers vecteurs des traces A et B ne sont pas similaires, alors la mesure de similarité est égale au maximum de la similarité entre une trace et l'autre privée de son premier vecteur. La formule suivante présente le principe de mesure de similarité entre les deux traces A et B.

$$ILCSS_{\delta,N}(A, B) = \begin{cases} 0 \text{ si } A \text{ ou } B \text{ est vide} \\ 1 + ILCSS_{\delta,N}(\text{Tail}(A), \text{Tail}(B)) \text{ si } O_{A,i} = O_{B,j}, Q_{A,i,k} = Q_{B,j,k}, \\ |V_{A,i,k} - V_{B,j,k}| \leq \delta, \text{ avec } 1 \leq i \leq n, 1 \leq j \leq m, |i - j| \leq N \text{ et } k = 1 \dots d. \\ \max\{ILCSS_{\delta,N}(\text{Tail}(A), B), ILCSS_{\delta,N}(A, \text{Tail}(B))\} \text{ sinon} \end{cases}$$

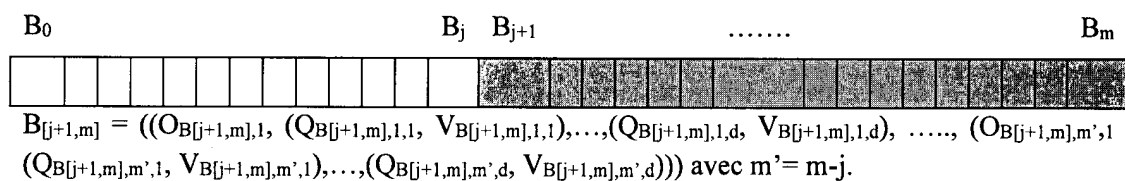
D'une part, à l'instant $t=t_{i+1}$ le système IDCBR-MAS-ITS récupère les traces $(A)_{t=t_{i+1}}$ enregistrées dans le serveur entre les deux instants t_i et t_{i+1} et on note $(A)_{t=t_{i+1}} = A_{[t_i, t_{i+1}]}$ voir figure ci-dessous.



Donc $(A)_{t=t_{i+1}}$ est un bloc de traces des vecteurs et elle peut s'écrire :

$$A_{[t_i, t_{i+1}]} = ((O_{A[t_i, t_{i+1}], 1}, (Q_{A[t_i, t_{i+1}], 1, 1}, V_{A[t_i, t_{i+1}], 1, 1}), \dots, (Q_{A[t_i, t_{i+1}], 1, d}, V_{A[t_i, t_{i+1}], 1, d}), \dots, (O_{A[t_i, t_{i+1}], n'}, (Q_{A[t_i, t_{i+1}], n', 1}, V_{A[t_i, t_{i+1}], n', 1}), \dots, (Q_{A[t_i, t_{i+1}], n', d}, V_{A[t_i, t_{i+1}], n', d})))$$

D'autre part, à l'instant $t=t_{i+1}$ il reste que le bloc $B_{[j+1, m]}$ de la trace source B , bloc des traces non encore comparé avec la trace cible, voir figure ci-dessous :



$$B_{[j+1, m]} = ((O_{B[j+1, m], 1}, (Q_{B[j+1, m], 1, 1}, V_{B[j+1, m], 1, 1}), \dots, (Q_{B[j+1, m], 1, d}, V_{B[j+1, m], 1, d}), \dots, (O_{B[j+1, m], m'}, (Q_{B[j+1, m], m', 1}, V_{B[j+1, m], m', 1}), \dots, (Q_{B[j+1, m], m', d}, V_{B[j+1, m], m', d}))) \text{ avec } m' = m - j.$$

La mesure entre la trace cible A et la trace source B à l'instant $t=t_{i+1}$ dépend des résultats à l'instant $t=t_i$ via la formule réursive suivante :

$$(ILCSS_{\delta,N,t_{i+1}}(A, B))_{t=t_{i+1}} = (ILCSS_{\delta,N,t_i}(A, B))_{t=t_i} + ILCSS_{\delta,N,t_{i+1}}(A_{[t_i, t_{i+1}]}, B_{[j+1, m]})$$

Avec :

$$ILCSS_{\delta,N,t_{i+1}}(A_{[t_i, t_{i+1}]}, B_{[j+1, m]}) = \begin{cases} 0 \text{ si } A_{[t_i, t_{i+1}]} \text{ ou } B_{[j+1, m]} \text{ est vide.} \\ 1 + ILCSS_{\delta,N,t_{i+1}}(\text{Tail}(A_{[t_i, t_{i+1}]}) , \text{Tail}(B_{[j+1, m]})) \text{ si } O_{A_{[t_i, t_{i+1}]}, i'} = O_{B_{[j+1, m]}, j'}, \\ Q_{A_{[t_i, t_{i+1}]}, i', k'} = Q_{B_{[j+1, m]}, j', k'}, |V_{A_{[t_i, t_{i+1}]}, i', k'} - V_{B_{[j+1, m]}, j', k'}| \leq \delta \\ , \text{ avec } 1 \leq i' \leq n', 1 \leq j' \leq m', m' = m - j, |i' - j'| \leq N \text{ et } k' = 1 \dots d. \\ \max\{ILCSS_{\delta,N,t_{i+1}}(\text{Tail}(A_{[t_i, t_{i+1}]}) , B_{[j+1, m]}), ILCSS_{\delta,N,t_{i+1}}(A_{[t_i, t_{i+1}]}, \text{Tail}(B_{[j+1, m]}))\} \text{ sinon.} \end{cases}$$

L'indice du dernier élément commun entre les deux traces $(A)_{t=t_{i+1}}$ et B à l'instant $t=t_{i+1}$ est obtenu via l'algorithme suivant :

On initialise j à 0, puis on utilise la formule suivante :

$$\{j = j' \text{ si } O_{A_{[t_i, t_{i+1}]}, i'} = O_{B_{[j+1, m]}, j'}, Q_{A_{[t_i, t_{i+1}]}, i', k'} = Q_{B_{[j+1, m]}, j', k'}, |V_{A_{[t_i, t_{i+1}]}, i', k'} - V_{B_{[j+1, m]}, j', k'}| \leq \delta, \text{ avec } 1 \leq i' \leq n', 1 \leq j' \leq m', |i' - j'| \leq N, j' > j \text{ et } k' = 1 \dots d. \}$$

La similarité $\text{sim}_{\delta,N,t_i}(A, B)$ entre les deux traces A et B à l'instant t_i , est ensuite normalisé par la plus petite longueur des deux traces comparées, de façon à ce que la mesure de similarité varie entre 0 et 1 :

$$\text{Sim}_{\delta,N,t_i}(A, B) = 1 - \frac{\text{ILCSS}_{\delta,N,t_i}(A, B)}{\min(n, m)}$$

$A \approx B$ se lit les deux traces A et B sont similaires.

Le point fort de notre système IDCBR-MAS-ITS est que le processus de calcul de similarité ne s'arrête pas s'il y a un changement dans la trace de l'utilisateur et on ne refait pas le calcul depuis le début. Les similarités calculées sont sauvegardées dans les tableaux de similarités, détaillé dans la section suivante.

- Tableaux des similarités

Notre plateforme IDCBR-MAS-ITS récupère les traces via les agents d'interfaces d'une manière permanente, ensuite ces dernières sont traitées par notre système. Le tableau suivant présente les valeurs de similarités calculées par les agents IDCBR-MAS-ITS en temps réel, entre la trace cible et les traces passées. Nous signalons que pour chaque nouveau cas cible le système crée une nouvelle table de similarités afin de stocker tous les valeurs de similarités entre ce nouveau cas cible et les traces passées. Les valeurs de similarités sont mises à jour automatiquement chaque fois qu'il y a changement dans le fichier trace.

Similarité ILCSS	Score élaboration	Score remémoration	Score adaptation	TD
---------------------	----------------------	--------------------	------------------	----

Tableau 1 : Similarités entre trace cible et traces passées

- Trace de l'apprenant

Dans notre cette invention on distingue deux types de cas : traces en cours de comparaison (cas cible) et traces passées enregistrées dans la base de cas (cas sources). Les traces des apprenants (les cas cible et sources), désignent les informations que Moodle enregistre sur l'activité, la production, et toute action résultante de l'interaction de ses apprenants. Ces traces (TA) sont représentées par une collection des traits sémantiques : triplet : TS = (objet, (qualification, valeur) +). Nous utilisons le langage XML pour formater les TS qui alimentent le système.

Revendications :

1. Système de tutorat intelligent pour les plateformes e-Learning caractérisé en ce qu'il est basé sur le raisonnement dynamique incrémental et les systèmes multi-agents indépendamment de l'objet de la matière enseignée.
2. Système de tutorat intelligent Selon la revendication 1, capable d'assurer la prédiction des situations dynamiques, en se basant sur les expériences passées similaires.
3. Système de tutorat intelligent, selon les revendications 1.2, qui modélise les expériences passées sous la forme des traces qui comprennent les productions et les actions résultantes de l'interaction de l'apprenant avec la plateforme d'apprentissage.
3. Système de tutorat intelligent selon les revendications 1.2.3, dont les expériences passées sont modélisées sous forme de traces qui comprennent les productions et les actions résultantes de l'interaction de l'apprenant avec la plateforme d'apprentissage.
4. Système de tutorat intelligent, selon les revendications précédentes, l'acquisition automatique des connaissances du domaine par notre système permet d'améliorer la performance de ce dernier
5. Système de tutorat intelligent, Selon la revendication 1, peut être appliquée dans d'autres services en ligne tels que le e-commerce et e-gov

Annexes

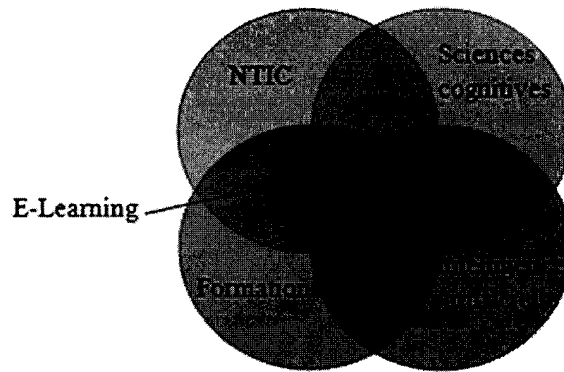


Figure 1 : STI/E-learning

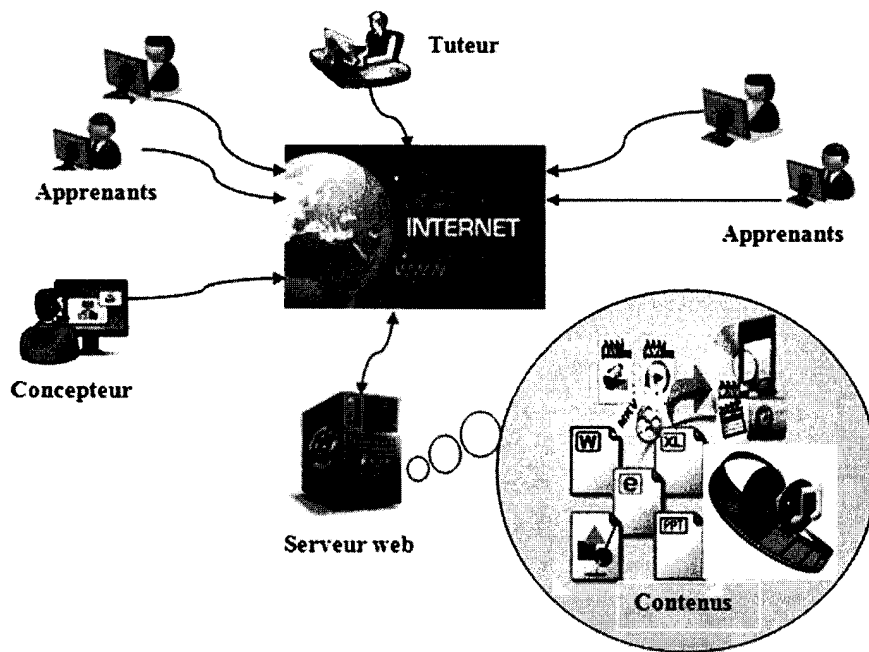


Figure 2 : Principaux acteurs d'une plateforme e-learning

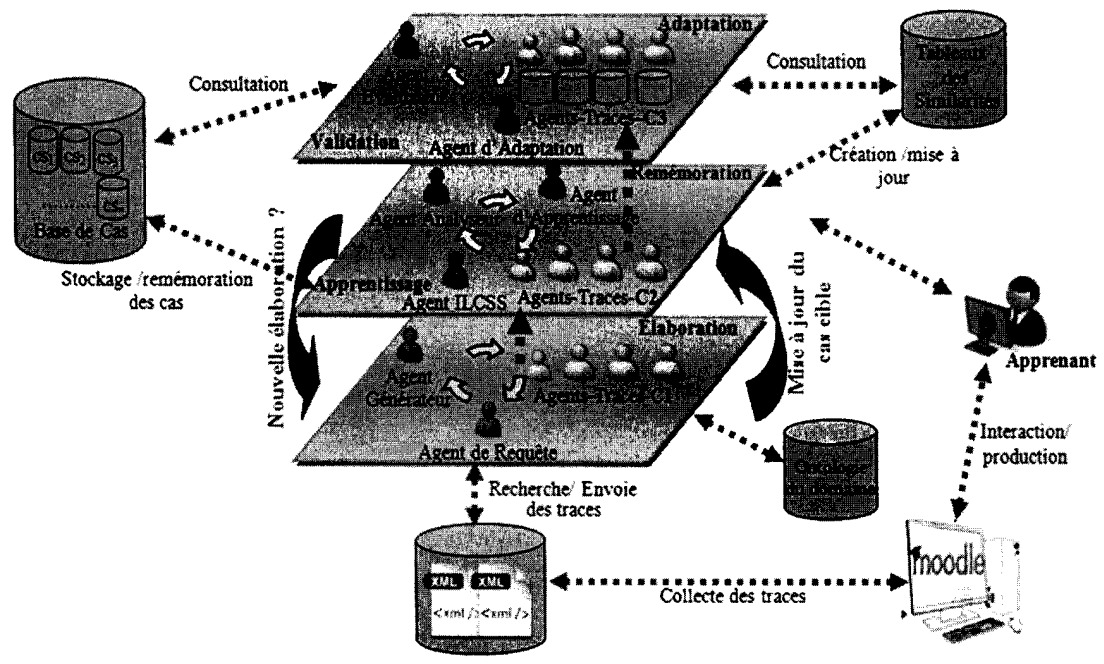


Figure 3 : Architecture générale du système IDCBR-MAS-ITS

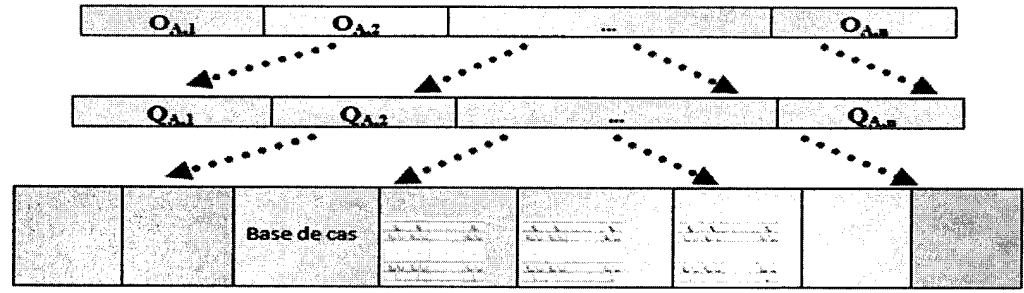


Figure 4 : Structure globale de la base de cas du système.



**RAPPORT DE RECHERCHE
AVEC OPINION SUR LA BREVETABILITE**
(Conformément aux articles 43 et 43.2 de la loi 17-97 relative à la
protection de la propriété industrielle telle que modifiée et
complétée par la loi 23-13)

Renseignements relatifs à la demande	
N° de la demande : 39613	Date de dépôt : 29/12/2016
Déposant : Université Mohammed V RABAT	
Intitulé de l'invention : Système de tutorat intelligent pour l'environnement e-Learning	
Le présent document est le rapport de recherche avec opinion sur la brevetabilité établi par l'OMPIC conformément aux articles 43 et 43.2, et notifié au déposant conformément à l'article 43.1 de la loi 17-97 relative à la protection de la propriété industrielle telle que modifiée et complétée par la loi 23-13.	
Les documents brevets cités dans le rapport de recherche sont téléchargeables à partir du site http://worldwide.espacenet.com , et les documents non brevets sont joints au présent document, s'il y en a lieu.	
Le présent rapport contient des indications relatives aux éléments suivants :	
Partie 1 : Considérations générales	
<input checked="" type="checkbox"/> Cadre 1 : Base du présent rapport <input type="checkbox"/> Cadre 2 : Priorité <input type="checkbox"/> Cadre 3 : Titre et/ou Abrégé tel qu'ils sont définitivement arrêtés	
Partie 2 : Rapport de recherche	
Partie 3 : Opinion sur la brevetabilité	
<input type="checkbox"/> Cadre 4 : Remarques de clarté <input checked="" type="checkbox"/> Cadre 5 : Déclaration motivée quant à la Nouveauté, l'Activité Inventive et l'Application Industrielle <input type="checkbox"/> Cadre 6 : Observations à propos de certaines revendications dont aucune recherche significative n'a pu être effectuée <input type="checkbox"/> Cadre 7 : Défaut d'unité d'invention	
Examineur: I.Oubiyi	Date d'établissement du rapport : 16/02/2017
Téléphone: 212 5 22 58 64 14/00	

Partie 1 : Considérations générales		
<p><i>Cadre 1 : base du présent rapport</i></p> <p>Les pièces suivantes de la demande servent de base à l'établissement du présent rapport :</p> <ul style="list-style-type: none"> • <u>Description</u> 11 Pages • <u>Revendications</u> 5 • <u>Planches de dessin</u> 2 Pages 		
Partie 2 : Rapport de recherche		
<p>Classement de l'objet de la demande : CIB : G09B 5/08</p>		
<p>Bases de données électroniques consultées au cours de la recherche : EPOQUE, Orbit</p>		
Catégorie*	Documents cités avec, le cas échéant, l'indication des passages pertinents	N° des revendications visées
X	Sujet de la thèse de doctorat : « Raisonement à partir de cas dynamique multi-agents – Application à un système de tuteur intelligent » ; Abdelhamid Zouhair ; soutenue le 20-10-2014	1-5
A	WO2016112858 ; 21-06-2016; Huawei Technologies Co., Ltd.	1-5
A	CN102610129 A ; 25-07-2012 ; JIANGSU MODERN ENTPR INFORMATIZATION APPLIC & SUPPORT SOFTWARE ENGINEERING TECHNOLOGY RES & DEV CT	1-5
<p>*Catégories spéciales de documents cités :</p> <p>-« X » document particulièrement pertinent ; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément</p> <p>-« Y » document particulièrement pertinent ; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier</p> <p>-« A » document définissant l'état général de la technique, non considéré comme particulièrement pertinent</p> <p>-« P » documents intercalaires ; Les documents dont la date de publication est située entre la date de dépôt de la demande examinée et la date de priorité revendiquée ou la priorité la plus ancienne s'il y en a plusieurs</p> <p>-« E » Éventuelles demandes de brevet interférentes. Tout document de brevet ayant une date de dépôt ou de priorité antérieure à la date de dépôt de la demande faisant l'objet de la recherche (et non à la date de priorité), mais publié postérieurement à cette date et dont le contenu constituerait un état de la technique pertinent pour la nouveauté</p>		

Partie 3 : Opinion sur la brevetabilité*Cadre 5 : Déclaration motivée quant à la Nouveauté, l'Activité Inventive et l'Application Industrielle*

Nouveauté (N)	Revendications aucune	Oui
	Revendications 1-5	Non
Activité inventive (AI)	Revendications aucune	Oui
	Revendications 1-5	Non
Possibilité d'application Industrielle (PAI)	Revendications 1-5	Oui
	Revendications aucune	Non

Il est fait référence aux documents suivants. Les numéros d'ordre qui leur sont attribués ci-après seront utilisés dans toute la suite de la procédure

D1 : <http://www.theses.fr/fr/182861279>

1. Nouveauté (N) et Activité inventive (AI)

Le document D1 divulgue un système de tutorat pour les plateformes e-Learning basée sur le raisonnement dynamique incrémental et les systèmes multi-agents indépendamment de l'objet de la matière enseignée (voir résumé).

Par conséquent, l'objet de la revendication 1 n'est pas nouveau au sens de l'article 26 de la loi 17-97 telle que modifiée et complétée par la loi 23-13.

Les revendications dépendantes 2-5 ne semblent pas contenir des caractéristiques supplémentaires, en matière de nouveauté, en étant combinées avec les caractéristiques techniques de la revendication indépendante 1 auxquelles lesdites revendications dépendantes sont liées. Par conséquent, l'objet desdites revendications n'est pas nouveau au sens de l'article 26 de la loi 17-97 telle que modifiée et complétée par la loi 23-13.

2. Possibilité d'application industrielle (PAI) :

L'objet de la présente invention est susceptible d'application industrielle au sens de l'article 29 de la loi 17-97 telle que modifiée et complétée par la loi 23-13, parce qu'il présente une utilité déterminée, probante et crédible.