TABLE DES MATIÈRES

Page

INTRODUCTION 1

CHAPITRE 1 PROBLÉMATIQUE.....3
1.1 Problématique générale.....3
1.2 Problématique spécifique.....5
1.3 Objectifs généraux et spécifiques de la recherche 7
1.4 Méthodologie générale.....8

CHAPITRE 2 REVUE DE LA LITTÉRATURE 9
2.1 Les environnements informatiques d'apprentissage humain (EIAH).....9
2.1.1 EIAH et EAI : quelle appellation choisir? 11
2.1.2 Impacts et avantages en enseignement 11
2.2 Les agents intelligents.....13
2.2.1 Les AI en enseignement dans les EIAH 15
2.2.2 Impacts et avantages des AI en enseignement..... 16
2.3 Notre apport à la littérature 17

CHAPITRE 3 ANALYSE ET CONCEPTION PRÉLIMINAIRES AVEC MISA 20
3.1 Présentation de MISA 20
3.2 Analyse et conception préliminaires avec MISA.....22
3.2.1 Définition du problème de formation 22
3.2.2 Identification des objets d'apprentissage..... 24
3.2.3 Définition de l'approche pédagogique..... 27
3.2.4 Définition de l'approche médiatique 32
3.3 Critique de la méthode MISA 33

CHAPITRE 4 CONCEPTION UML DE L'ENVIRONNEMENT AARTIC.....35
4.1 Le langage UML 35
4.2 Description et Architecture de l'environnement AARTIC 36
4.2.1 Description d'AARTIC..... 36
4.2.2 Architecture de l'environnement AARTIC 39
4.3 Diagramme de classe UML de l'environnement AARTIC 41
4.3.1 Modules « etudiant » et « communicationBD » 42
4.3.2 Module « *etape* » 45
4.3.3 Le module enseignant 47
4.4 Diagramme d'activité UML de l'environnement AARTIC.....48
4.4.1 L'environnement AARTIC du côté enseignant 48
4.4.2 L'environnement AARTIC du côté étudiant 50

CHAPITRE 5 CHOIX TECHNOLOGIQUES 53
5.1 Développement des agents avec JESS 53
5.2 La logique métier : technologie Java 54
5.3 Site Web AARTIC: SPIP versus développement HTML 55
5.3.1 Présentation SPIP 55
5.3.2 Inconvénients de SPIP pour AARTIC et choix d'une programmation HTML 56
5.4 Représentation des concepts théoriques du cours avec Freemind 57
5.5 Développement des questionnaires à choix multiple 58

CHAPITRE 6 RÉALISATION 60
6.1 Pré-test 60
6.2 Test de l'application AARTIC 61
6.2.1 Réalisation 61
6.2.2 Évaluation de AARTIC 65
6.3 Effort de l'enseignant 67
6.4 Impact sur l'apprentissage 70

CONCLUSION 73

ANNEXE I QUESTIONNAIRE PRÉ-TEST 76

ANNEXE II QUESTIONNAIRE D'ÉVALUATION DE L'ENVIRONNEMENT AARTIC 79

ANNEXE III JAVA ET JESS 81

APPENDICE A LES MACHINES À ENSEIGNER 83

BIBLIOGRAPHIE 86

LISTE DES TABLEAUX

Page

Tableau 2.1 Comparaison entre l'application Sphinx et AARTIC 18

Tableau 3.1 Description des différents éléments du graphe de connaissances 26

Tableau 3.2 Définition des éléments du réseau des évènements d'apprentissage 29

LISTE DES FIGURES

Page

Figure 3.1 Graphe des connaissances du système d'apprentissage AARTIC. 25

Figure 3.2 Réseau des évènements d'apprentissage. 28

Figure 3.3 Graphe du modèle médiatique. 32

Figure 4.1 Fonctionnement de l'environnement AARTIC. 38

Figure 4.2 Architecture de l'environnement AARTIC. 39

Figure 4.3 Diagramme de classe UML du module etudiant et du module communicationBD. 42

Figure 4.4 Diagramme de classe du module etape. 45

Figure 4.5 Diagramme de classe simplifié du module enseignant. 47

Figure 4.6 Diagramme d'activité lié à la gestion des exemples et des comptes étudiants. 49

Figure 4.7 Diagramme d'activité de l'exploration d'une étape dans AARTIC. 51

Figure 4.8 Diagramme d'activité du passage d'un QCM dans AARTIC. 52

Figure 6.1 Interface d'accueil du système AARTIC. 62

Figure 6.2 La page QCM du site Web AARTIC. 63

Figure 6.3 Affichage des résultats du QCM. 64

Figure 6.4 Évaluation de AARTIC par les étudiants du cours LOG625. 66

Figure 6.5 Carte conceptuelle de la première étape du laboratoire 69

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AARTIC	Apprentissage, Adaptatif, Repère, Technologie de l'information et des communications
EAI	Environnements d'apprentissage informatisé
EAO	Enseignement assisté par ordinateur
EIAH	Environnement informatique d'apprentissage humain
EIAO	Enseignement Interactif assisté par ordinateur
IA	Intelligence artificielle
JESS	Java expert system shell
JSP	Java server pages
MISA	Méthode d'ingénierie des systèmes d'apprentissage
QCM	Questionnaire à choix multiples
SQL	Structured query language
TI	Technologies de l'information
TIC	Technologies de l'information et de la communication
UML	Unified modeling language

INTRODUCTION

L'objectif principal de ce mémoire est de concevoir et de développer un environnement informatique d'apprentissage humain (ou EIAH) intitulé « AARTIC ». Intégré dans les cours de baccalauréat offerts à l'École de technologie supérieure (ÉTS), l'environnement AARTIC (acronyme pour Apprentissage, Adaptatif, Repère, et Technologies de l'Information et de la Communication) contribue à l'encadrement des étudiants durant leur apprentissage.

L'une des spécificités d'AARTIC est sa capacité d'adapter l'information présentée en fonction du niveau de connaissance de chaque étudiant utilisateur de l'environnement. Cette adaptabilité est gérée, par deux agents intelligents intégrés à AARTIC. Le premier est associé à l'étudiant ; il lui sert de guide et de conseiller tout au long de son parcours. L'autre agent est associé à l'enseignant et lui permet d'effectuer un suivi des activités et de l'avancement de chacun des étudiants dans un cours.

Notre environnement se base sur deux paradigmes d'apprentissage importants. Le premier est l'apprentissage par l'exemple en vertu duquel l'étudiant peut assimiler une notion théorique présentée dans le cours grâce à un ensemble d'exemples proposé. Conjointement à l'apprentissage par l'exemple, nous introduisons le concept d'auto-explication. En effet, grâce à un forum de discussion associé à l'environnement AARTIC, chaque étudiant peut proposer sa propre explication d'un exemple. De plus, l'utilisation des forums de discussion permet d'encourager la collaboration entre étudiants afin d'atteindre un but commun : expliquer un exemple pour maîtriser un ou des concept(s) théorique(s). Le deuxième paradigme employé dans AARTIC est l'apprentissage par étape. Une étape représente une partie du cours théorique comme elle peut représenter une phase d'un projet de laboratoire à réaliser. Nous avons utilisé l'apprentissage par étape pour guider l'étudiant dans son apprentissage et pour lui permettre de visualiser la hiérarchie et les relations entre les concepts théoriques à une étape spécifique. Afin d'avoir un suivi continu sur l'avancement et le niveau de connaissance d'un étudiant, la fin de chaque étape est marquée par un questionnaire d'évaluation.

Vu l'importance du projet, nous nous sommes limités, dans un premier temps, à un contexte d'utilisation spécifique de l'environnement AARTIC. Nous nous sommes intéressés au laboratoire du cours « introduction aux systèmes intelligents » offert à l'école de technologie supérieure (ÉTS). Il s'agit d'un cours obligatoire du programme de baccalauréat en génie logiciel et vise environ une centaine d'étudiants par année.

Le projet présenté dans ce mémoire relève d'une problématique d'actualité. En effet, l'apprentissage personnalisé représente l'un des 10 défis les plus importants en ingénierie (National Academy of Engineering, 2009). De plus, ce que nous proposons dans ce mémoire permettra de contribuer à l'amélioration de la qualité de l'enseignement grâce à un apprentissage personnalisé, collaboratif et un suivi plus soutenu de la part des enseignants.

Ce mémoire s'organise en six chapitres. Nous commençons par la spécification des facteurs déclencheurs de la parution de ce projet de mémoire. Ensuite nous regardons de plus près, l'état de l'art des EIAH et des agents intelligents à travers le chapitre 2. Dans le troisième chapitre, nous présentons l'analyse et la conception préliminaires de notre EIAH grâce à la méthode d'ingénierie des systèmes d'apprentissage « MISA ». Après, nous dévoilons l'architecture d'AARTIC et approfondissons sa conception avec les diagrammes du langage UML. Nous profitons du cinquième chapitre pour présenter les divers choix technologiques effectués pour la réalisation de l'environnement d'apprentissage. Puis, le chapitre six traitera des deux évaluations réalisées avant et après l'intégration de notre EIAH durant le laboratoire du cours « introduction aux systèmes intelligents » offert à l'ETS. Ce dernier chapitre permettra également d'analyser les résultats obtenus concernant l'importance de l'intégration de tels EIAH et l'effort fourni par l'enseignant pour préparer et insérer le contenu pédagogique (définitions, exemples, questionnaires,…) dans l'environnement informatique d'apprentissage humain. Enfin nous conclurons avec une description de nos travaux futurs.

CHAPITRE 1

PROBLÉMATIQUE

1.1 Problématique générale

Durant la réalisation de projets de laboratoire associés à des cours en ingénierie, les étudiants se retrouvent souvent face à des obstacles qui les empêchent d'avancer correctement dans leurs travaux (par exemple : compréhension déficiente de la matière, mauvaise organisation du travail,…). Tel est le cas pour le cours d'introduction aux systèmes intelligents offert au département de génie logiciel et des TI de l'ETS. D'ailleurs, à la session d'été 2008, 55% des étudiants de ce cours ont estimé que leur premier projet de laboratoire était difficile à réaliser (voir chapitre 6, la section pré-test). Il en résulte que les étudiants ne profitent pas des avantages des projets de laboratoire qui consistent à comprendre et à maîtriser les notions théoriques vues dans le cours en les appliquant dans le cadre d'un problème réel (Prince et Felder, 2006).

Il faut également considérer que chaque étudiant a son propre rythme d'apprentissage et degré de connaissance (Emmanuel, Capus et Tourigny, 2006). En effet, bien que la majorité des étudiants arrive à comprendre un concept théorique présenté dans le cours, il existe une minorité qui n'arrive pas à saisir ce même concept. De son côté, l'enseignant n'a souvent pas les moyens nécessaires, mis à part les examens et les travaux de laboratoire, pour suivre de manière continue, chacun de ses étudiants et de repérer cette minorité qui a besoin de plus d'encadrement; spécialement lorsqu'il s'agit d'un cours où le nombre d'étudiants par classe peut être supérieur à 50. Il s'agit ici bien sûr de l'un des défis rencontrés en éducation : comment adapter le discours pédagogique à des apprenants ayant différents niveaux de compétences?

Rapport-Gratuit.com

Nous remarquons aussi la réticence des étudiants face aux tâches impliquant une réelle collaboration. C'est d'ailleurs le cas pour la participation aux forums de discussions associés à certains cours. En effet, 74% des étudiants du cours mentionné plus haut ne participent pas aux forums associés à certains de leurs cours (voir Chapitre 5, section pré-test). Par conséquent, les étudiants ne profitent pas des avantages liés à la collaboration comme le partage efficace des connaissances et l'amélioration du processus d'apprentissage.

Parmi les solutions envisageables, les environnements informatiques d'apprentissage humain ou EIAH visent, justement, à palier aux problèmes mentionnés précédemment. Comme son nom l'indique, un EIAH est un environnement informatique dont la finalité est de soutenir l'apprentissage humain (Tchounikine, 2002a). D'ailleurs, l'essor important connu par les technologies de l'information et de la communication (TIC), et spécialement Internet, a engendré un nombre important d'environnements d'apprentissage dans tous les domaines et à tous les niveaux d'enseignement (du primaire à l'université).

Aussi, grâce à l'intelligence artificielle et aux agents pédagogiques, une personnalisation et une adaptabilité des EIAH ont été rendues possibles. Plusieurs environnements basés sur les agents pédagogiques existent. Certains de ces systèmes peuvent être très complexes à mettre en œuvre (comme AutoTutor(Institute for intelligent systems, 2008)) et, en conséquence, difficiles à adapter rapidement. D'autres systèmes se spécialisent dans l'observation d'activités pédagogiques basées sur l'analyse de « trace » (Loghin, Carron et Marty, 2007a). Il existe également des systèmes comme ABALS (Chen et Zhang, 2006), incitant les étudiants à participer à des activités collaboratives.

Notre recherche vise à poser les balises pour mettre en place un environnement d'apprentissage permettant un enseignement plus structuré, à base d'exemples, collaboratif et adaptatif. Elle se base sur les méthodes et les théories provenant de deux domaines complémentaires : le domaine des environnements informatiques d'apprentissage humain (EIAH) et celui de l'intelligence artificielle.

La recherche, dans son ensemble, a pour tâche de répondre aux questions suivantes:

- Comment concevoir un environnement informatique d'apprentissage humain?
- Quels outils utiliser pour développer un tel environnement?
- Sous quelle forme adapter l'apprentissage?
- Comment inciter les étudiants à être plus collaboratifs?
- Quels sont les impacts de tels environnements sur l'apprentissage?

1.2 Problématique spécifique

Dans le cadre de cette recherche, nous nous intéressons plus précisément au développement d'un système d'apprentissage pour le cours LOG625 « Introduction aux systèmes intelligents ». Ce cours est offert à l'École de technologie supérieure (ÉTS) sous la forme de trois heures et demie de cours magistral par semaine, où les concepts et notions théoriques sont présentés, et de deux heures de laboratoires par semaine pendant lesquelles des équipes de deux à quatre étudiants sont amenées à appliquer les notions théoriques vues dans le cours pour la réalisation d'un projet de laboratoire (d'une durée de 4 à 6 semaines). Dans le cadre de cette recherche nous considérons plus particulièrement le premier projet de laboratoire proposé dans le cours qui consiste à développer un système expert capable de jouer au jeu de stratégie Clue[1]. Ce projet s'articule autour de l'un des thèmes principaux du cours « Représentation des connaissances et raisonnement » (Ratté, 2008). Comme outils de

[1] http://fr.wikipedia.org/wiki/Cluedo

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

programmation, les étudiants sont amenés à se familiariser et utiliser le module JESS (Java Expert System Shell).

Parmi les points forts de ce cours, on dénote : la pertinence des projets de laboratoire par rapport au contenu et objectifs du cours LOG625, l'abondance en information (livres, site Web, articles scientifiques,..) traitant des thèmes du cours ainsi que la richesse du site Internet attaché au cours.

Cependant, on remarque aussi certaines faiblesses concernant le type d'enseignement pratiqué pour ce cours ainsi que pour d'autres cours similaires, et plus particulièrement dans les travaux de laboratoires associés à ces cours. Parmi ces faiblesses, on trouve:

- La complexité des notions théoriques présentées dans le cours. En effet, certains concepts et notions sont difficiles à assimiler et requièrent parfois plus d'attention et de temps pour l'étude de ces notions,
- Le rythme personnel des études. Chaque étudiant a son propre rythme d'étude. Certains sont bien préparés et très à l'aise avec ce genre de cours tandis que d'autres se retrouvent un peu perdus et ont du mal à suivre le rythme normal du cours vu la complexité de celui-ci,
- Le nombre élevé d'étudiants. Étant donnée la diversité des rythmes d'apprentissage, un suivi personnalisé de la part de l'enseignant est fortement recommandé. Malheureusement, l'enseignant, dans ce genre de cours, peut se retrouver devant des classes d'une quarantaine voire d'une cinquantaine d'étudiants par session (46 étudiants ont suivi ce cours à la session d'hiver 2008) et par conséquent un suivi personnalisé pour chacun de ses étudiants est très difficile,
- La désorientation et l'absence de logique de conception. Dans la réalisation de leur premier projet de laboratoire, les étudiants sont amenés à utiliser et à maîtriser un nouveau paradigme de programmation (le développement avec le moteur d'inférence JESS) qui est différent de celui qu'ils ont déjà connus. En conséquence, les étudiants ne savent pas comment ni par où commencer leur projet.

1.3 Objectifs généraux et spécifiques de la recherche

Notre recherche a pour objectif principal d'aider l'étudiant et l'enseignant, durant les projets de laboratoire associés au cours, à travers un environnement informatique d'apprentissage humain (EIAH) intitulé « AARTIC » (acronyme pour Apprentissage, Adaptatif, Repère, et Technologies de l'Information et de la Communication). AARTIC est basé sur les stratégies d'apprentissage par l'exemple et d'auto-explication associées à l'approche d'enseignement par étape (Ratté et Caron, 2002). Cet environnement permet de répondre aux faiblesses mentionnées précédemment en aidant l'étudiant à maîtriser les concepts théoriques nécessaires à la réalisation de son projet de laboratoire, tout en proposant une familiarisation rapide avec la programmation JESS.

L'approche par étape permet de mieux orienter l'étudiant tout au long de la réalisation de son projet de laboratoire. En effet, cette approche répond à un besoin actuel des étudiants dans l'organisation de leurs travaux. D'après un sondage effectué à l'École de technologie supérieure, 42,7% des étudiants planifient leur temps de travail « rarement ou jamais » (Viau, 2006). Dans AARTIC, chaque étape définit une section importante dans la réalisation du projet et comporte un ensemble de concepts théoriques (généralement présenté au préalable dans le cours) qu'il faut retenir. Pour pouvoir passer à l'étape suivante, l'étudiant doit réussir un questionnaire d'évaluation en ligne.

Le système assiste l'étudiant dans l'assimilation des concepts théoriques du cours à travers des exemples qu'il lui propose selon le niveau de connaissance de l'étudiant. Ce niveau est calculé à partir des résultats des questionnaires d'évaluation proposés dans AARTIC. L'étudiant a la possibilité de proposer sa propre explication de l'exemple qu'il vient de voir à travers le forum de discussions intégré à l'environnement d'apprentissage. De ce fait, tous les étudiants peuvent commenter l'explication proposée ou en fournir une nouvelle. Ceci toujours sous l'encadrement de l'enseignant ou de l'auxiliaire de laboratoire qui peut intervenir à tout moment dans le forum.

L'aspect adaptatif du système est assuré par deux agents pédagogiques. L'un, associé à l'étudiant, fait le suivi des différentes activités de celui-ci. L'autre, associé à l'enseignant, lui permet d'offrir un encadrement personnalisé de chacun des étudiants pris individuellement ou de la classe en entier.

AARTIC est un environnement Web, ce qui offre aux étudiants la possibilité de travailler quand ils le souhaitent, où ils le souhaitent et selon leur propre rythme.

1.4 Méthodologie générale

Afin de réaliser l'environnement AARTIC, nous avons commencé par cerner dans la littérature, la définition d'environnement d'apprentissage humain et d'agent intelligent, et identifier des exemples et des résultats de l'utilisation de ces deux éléments (EIAH et agent intelligent) sur l'apprentissage. Par la suite, nous avons déterminé le cadre de notre recherche et l'analyse des besoins grâce à la méthode d'ingénierie des systèmes d'apprentissage « MISA ». Une fois l'analyse et la conception préliminaires réalisées, nous avons préparé l'implémentation de l'environnement en produisant son architecture générale et en détaillant cette architecture avec les diagrammes du langage UML. Nous avons également sélectionné les différents outils technologiques utiles pour notre environnement. Enfin, nous avons pu évaluer l'environnement produit en le faisant tester par les étudiants du cours LOG625 concerné. Cette évaluation s'est effectuée en deux étapes. Dans un premier temps, à la session d'été 2008, nous avons pu justifier l'utilité d'un tel environnement d'apprentissage en recensant les besoins des étudiants et les difficultés qu'ils rencontrent dans le cours LOG625. Nous avons profité de ces résultats pour concevoir un EIAH qui répond à leur besoin. Dans un deuxième temps, à la session d'été 2009, les étudiants du cours LOG625 ont utilisé et évalué l'environnement que nous avons construit.

CHAPITRE 2

REVUE DE LA LITTÉRATURE

Ce chapitre se divise en trois grandes sections. D'abord, après avoir mieux défini les « environnements informatiques d'apprentissage humain », nous examinons les impacts de tels environnements sur l'apprentissage. Ensuite, nous nous intéresserons à la notion d'agent intelligent et aux avantages liés à son intégration dans un EIAH. Finalement, nous terminons ce chapitre en situant notre propre travail dans le cadre des recherches existantes dans le domaine.

2.1 Les environnements informatiques d'apprentissage humain (EIAH)

L'appellation « environnement informatique d'apprentissage humain » (EIAH) peut être divisée en trois parties : l'environnement informatique, l'apprentissage et le caractère humain. Le terme « environnement informatique » désigne l'ensemble des composantes informatiques, de la « structure organisationnelle » (Grand dictionnaire terminologique, 2009) et des procédés fonctionnant dans le but d'offrir un certain nombre de services. L'« apprentissage» signifie l'acquisition d'une information ou d'un savoir par une personne. L'adjectif « humain » spécifié à la fin permet de souligner l'importance de l'utilisateur concerné par ces environnements. L'ensemble de tous ces termes forment l'acronyme « EIAH » signifiant les environnements au service de l'éducation à travers un outil informatique. Nous adopterons, pour la suite de ce travail, la définition de Tchounikine :

> « Un EIAH est un environnement informatique conçu dans le but de favoriser l'apprentissage humain, c'est-à-dire la construction de connaissances chez un apprenant. La conception d'un EIAH est un processus complexe, profondément pluridisciplinaire. » (Tchounikine, 2002a, p. 62)

Le terme EIAH est apparu à la fin des années 90; sa venue introduisait une nouvelle approche et de nouvelles normes (des notions comme la « réutilisation d'objets pédagogiques » ou encore des normes comme SCORM «*Sharable Content Object Reference Model*»). En réalité les EIAH découlent d'une grande lignée des dispositifs pédagogiques et d'approches pour la conception de ces dispositifs (Kreczanik, 2008) . Les premières machines à enseigner existent depuis bien longtemps : pour cela il faut remonter jusqu'au début du siècle dernier en 1926 quand Sidney L. Pressey a développé une machine proposant des questions à choix multiples et une rétroaction rapide à l'utilisateur (ITS) . Plus tard, dans les années 50 et 60 est apparue la notion d'«enseignement programmé ». Dans le courant des années 70 et avec l'essor de l'intelligence artificielle, on parlait plutôt de « EAO » ou Enseignement Assisté par Ordinateur. C'est à ce moment qu'apparaissent les premières applications éducatives de l'informatique. Dans le courant des années 80, le terme EAO s'est modernisé avec l'intégration des notions d'intelligence artificielle pour devenir « EIAO » ou Enseignement Intelligemment Assisté par Ordinateur (Tchounikine, 2002b). Grâce aux EIAO, on mettait à l'avant-plan les théories d'apprentissage. Dix ans plus tard, l'EIAO est devenu acronyme de « Enseignement **Interactif** d'Apprentissage avec Ordinateur». Lui-même devenu à la fin des années 90 EIAH ou Environnement Informatique pour l'Apprentissage Humain. Ainsi, on mettait en exergue, à la fois, l'aspect informatique et humain de ce genre de dispositifs (De Vries, 2001).

La recherche dans ce domaine, comme l'indique son historique, est étendue et comporte une revue de littérature abondante compte tenu de la pluridisciplinarité du domaine. En effet, en EIAH deux grandes disciplines se rencontrent : d'un côté les environnements informatiques qui connaissent un essor exponentiel durant ces dernières décennies, de l'autre, les sciences de l'éducation dont les théories ne cessent de changer et de se diversifier.

L'apparition de l'Internet et des TIC (Technologies de l'information et de la communication) a constitué un facteur important dans l'évolution des méthodes d'enseignements et de représentation d'éléments pédagogiques (Tchounikine, 2002b).

2.1.1 EIAH et EAI : quelle appellation choisir?

Dans la littérature, on peut trouver un autre terme que EIAH pour désigner ces environnements informatiques à but éducatif : les EAI ou les « environnements d'apprentissage informatisés » (Doré et Basque, 1999).

En réalité, les deux terminologies sont quasi-équivalentes. EIAH et EIA désignent la même chose : des environnements informatiques ayant des objectifs pédagogiques; ils sont tous les deux basés sur des théories et des notions communes(Kreczanik, 2008). La seule différence réside dans le fait que les EIA représentent une notion plus « générique» que les EIAH puisqu'ils « intègrent des artefacts informatiques non nécessairement conçus dans un objectif d'apprentissage» (Kreczanik, 2008, p. 72).

Dans le cadre de ce mémoire, nous estimons que les deux appellations sont synonymes. Nous avons choisi pour la suite de notre travail de parler d'EIAH.

2.1.2 Impacts et avantages en enseignement

En dépit des différents environnements d'apprentissage existants, des méthodes et des façons de faire qui en découlent et des théories d'apprentissage contemporaines (constructivisme, béhaviorisme, cognitivisme,…), tous les experts du domaine sont d'accord quant aux effets bénéfiques de ces environnements informatiques. Au sujet de l'environnement Sphinx réalisé

à l'université de Laval, les auteurs déclarent qu'il a « amélioré le processus d'apprentissage »[2] des étudiants (Capus et al., 2006, p. 108).

Parce qu'un EIAH peut jouer différents rôles, « outils de présentation de l'information », « outils de traitement de la connaissance », « outils de communication » (Tchounikine, 2002b, p. 236), il réussi à travers chacun de ces rôles à stimuler l'apprentissage et le développement cognitif[1] de l'apprenant :

- « Outils de présentation de l'information » (par exemple un environnement Web, une carte sémantique,…). Grâce à ces outils, l'information devient plus accessible, plus à jour, et peut être présentée sous différents formats. Ceci peut s'avérer un bon stimulant à l'imagination, à la mémorisation et à la compréhension chez l'apprenant. De plus, l'apprenant peut, selon l'outil de présentation choisi, accéder à l'information sans limite de temps, de contrainte géographique ni de limite sur la quantité de données (Aubé, 1996),
- « Outils de traitement de la connaissance » (par exemple une base de connaissances, un questionnaire en ligne,…). Ces outils permettent un meilleur encadrement de l'apprenant et un apprentissage plus personnalisé et adapté. L'apprenant peut ainsi s'auto-évaluer et obtenir des recommandations et des rétroactions plus rapidement,
- « Outils de communication » (ici on parle de deux types de communications : la communication homme/machine par exemple à travers les agents intelligents et la communication homme/homme en passant par la machine par exemple à travers les forums de discussion, les clavardages,…). La communication représente l'un des piliers de base de l'enseignement. En effet, une bonne communication, quelle que soit sa forme, favorise un bon enseignement et donc un bon apprentissage. Le rôle d'outils de communication que peut jouer un EIAH est donc un rôle clé. Grâce aux TIC, la collaboration entre apprenant, la participation active de l'apprenant, ainsi que le dialogue

---

[2] "we can conclude that the use of the Sphinx environment improved the learning process"

apprenant-enseignant devient plus facile et plus accessible. Plusieurs études ont démontré l'impact positif de l'intégration des forums de discussion en enseignement sur le développement de la pensée critique de l'étudiant. Citons, à titre d'exemple, les travaux de Jeong (Jeong, 2003) ou encore ceux de Rézeau (Rézeau, 2004) qui démontrent l'apport de l'utilisation des forums de discussion dans l'apprentissage d'une langue étrangère.

Un EIAH est un environnement dans lequel interagissent l'homme et la machine dans un objectif d'éducation ou de formation. Cette interaction peut se faire localement ou encore à travers un réseau informatique. Grâce aux différents rôles que peut jouer un EIAH, l'apprentissage devient plus guidé, plus accessible et plus personnalisé. L'apprenant réussit à communiquer facilement avec ses pairs, son tuteur mais aussi avec la machine grâce entre autres aux agents intelligents qui le composent.

Mais qu'est ce qui définit un agent intelligent? Quelle différence y a-t-il entre ce dernier et un simple programme informatique? Et quels sont les avantages que peut représenter l'intégration de tels agents en éducation? C'est ce que nous traiterons dans la section suivante.

2.2 Les agents intelligents

Les agents intelligents appartiennent au domaine de l'intelligence artificielle (IA) dont l'origine remonte au milieu des années 40 après la fin de la seconde guerre mondiale. Alan Turing fut l'un des pionniers du domaine. Vers les années 70, débute la forte évolution de l'informatique qui a engendré une progression importante des techniques et des méthodes utilisées en IA. Depuis, et jusqu'à ce jour, il est désormais difficile d'énumérer de manière succincte ce que le monde de l'intelligence artificielle nous a apporté et ce dans divers

Rapport-Gratuit.com

domaines: en commerce électronique avec les systèmes d'aide à la décision, en médecine, avec les systèmes d'aide au diagnostic, en éducation, avec les systèmes tuteurs intelligents ou les agents pédagogiques, et même en cuisine où les appareils électroménagers deviennent de plus en plus intelligents comme les réfrigérateurs « intelligents » capables de préparer automatiquement votre liste d'épicerie. Aujourd'hui, l'intelligence artificielle fait partie intégrante de notre vie quotidienne. D'ailleurs, les mots « intelligent » et intelligence deviennent de plus en plus vidés de leur sens. Il convient donc de revenir à la base.

Dans sa définition la plus simple, un agent intelligent, comme le décrivent Russel et Norvig, est une « entité » qui perçoit l'environnement à travers ses « capteurs », sélectionne la meilleure action à entreprendre et agit grâce à ses « effecteurs » (Russell et Norvig, 2006, p. 37) . Les deux auteurs déclarent qu'il existe divers types d'agents intelligents dépendamment de la nature de l'environnement dans lequel ils évoluent mais aussi du type de programmes dont l'agent dépend. Parmi ces catégories d'agents, on trouve « les agents réflexes simples » (Russell et Norvig, 2006, p. 53) fonctionnant de la forme « **si** *percept* **alors** *action* » sans tenir compte de l'historique des percepts. Les agents « fondés sur les buts » (Russell et Norvig, 2006, p. 57) forment une autre catégorie. En plus de considérer la situation actuelle, ces agents ont besoin de connaître le but à atteindre afin de choisir l'action appropriée.

Quelle est la différence entre un agent intelligent et un simple programme logiciel? La question est souvent posée. Les auteurs Agarwal, Deo et Das distinguent un agent intelligent du programme informatique en lui associant les caractéristiques suivantes : « autonome», « persistant », « réactif », « proactif », « personnalisé » et « social » (Agarwal, Deo et Das, 2004, p. 1). À cet effet, les auteurs déclarent:

> « The programs which apply a fair degree of intelligence and are able to carry out tasks independently without any supervision fall under the category of intelligent agents»(Agarwal, Deo et Das, 2004, p. 1).

Il existe plusieurs catégories d'agents intelligents. Dans le cadre de notre travail, nous nous intéressons plus particulièrement aux agents pédagogiques. Il s'agit d'agents dont la fonction est éducative ou pédagogique et dont le but est d'améliorer l'apprentissage. Aujourd'hui, les agents pédagogiques deviennent de plus en plus complexes et performants. Citons à titre d'exemple, l'agent STEVE (Rickel et Johnson, 1999), un agent personnifié fonctionnant dans un environnement virtuel de formation, ou BAGHERA (Pesty, Webber et Balacheff, 2003) qui repose sur un système distribué multi-agents où chaque agent peut jouer le rôle de tuteur, d'assistant à l'apprenant ou d'assistant à l'enseignant. Un autre agent intelligent célèbre est « AutoTutor » (Graesser et al., 2004) qui aide dans l'apprentissage des notions en mécanique newtonienne, en informatique, ou encore en raisonnement scientifique à travers un dialogue en langage naturel qu'il instaure avec l'apprenant.

2.2.1 Les AI en enseignement dans les EIAH

Dans sa définition, Tchounikine (2002b) mentionne qu'un EIAH comprend des agents humains et artificiels. Ce qu'il entend par agent artificiel est possiblement toute composante informatique y compris les agents intelligents. La littérature est abondante d'exemples d'EIAH basés sur les techniques d'intelligence artificielle en général et d'agents intelligents en particulier.

Parmi ces EIAH, on trouve l'environnement « ABALS » (Chen et Zhang, 2006) permettant un apprentissage personnalisé avec un encouragement à la collaboration entre étudiants et entre l'étudiant et son tuteur. ABALS est basé sur trois catégories d'agents intelligents :

- l'agent étudiant est associé à chaque étudiant et gère le modèle d'apprentissage de l'étudiant;
- l'agent tuteur est associé au tuteur, son rôle est d'encadrer l'apprentissage;
- l'agent d'information est chargé d'assembler et de sélectionner l'information selon les besoins de chaque utilisateur de l'environnement.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

« I-MINDS » (Soh et al., 2006) (« Intelligent Multiagent Infrastructure for Distributed Systems in Education »), réalisé dans une université japonaise, constitue un autre exemple intéressant. Une des forces de I-MINDS est la personnalisation du contenu et l'incitation à la collaboration notamment grâce aux trois agents intégrés à l'environnement :

Agent Enseignant (« Teacher Agent ») interagit avec l'enseignant. Ses fonctions :

- Récupérer les informations à partir des agents étudiants,
- Gérer le profil de l'apprenant (progrès et participation),
- Classer les questions posées par les apprenants,
- Gérer la classe.

Agent Étudiant (« Student Agent ») interagit avec l'étudiant. Ses fonctions :

- Aider l'étudiant durant son apprentissage,
- Gérer les communications entre les étudiants et entre étudiant-enseignant,
- Former des alliances avec les autres apprenants pour les tâches collaboratives.

Agent de Groupe (« Group Agent ») interagit avec le groupe. Ses fonctions :

- Former un apprentissage coopératif structuré,
- Gérer les activités de groupe.

2.2.2 Impacts et avantages des AI en enseignement

Pour une éducation efficace, Agarwal et *al.* proposent de miser sur le travail de groupe, l'adaptabilité et le feedback constructif (Agarwal, Deo et Das, 2004). Tout ceci est réalisable grâce aux agents intelligents. En effet, leur autonomie, leur aspect social et personnalisable permettent aux agents intelligents d'adapter le contenu et la communication du cours selon le profil de chaque apprenant. De plus, un agent intelligent peut encourager la collaboration et l'interactivité comme dans le cas du système ABALS (Chen et Zhang, 2006) ou I-MINDS (Soh et al., 2006).

Grâce aux agents intelligents, on a réussi à enrichir les environnements d'apprentissage dit statiques. En effet, les environnements d'apprentissage intelligent sont riches en contenus. De plus, ils peuvent évaluer le niveau de connaissance d'un apprenant pour pouvoir mieux le guider et pour s'adapter à son niveau améliorant ainsi non seulement l'apprentissage mais aussi la communication entre l'apprenant et le tuteur (Urretavizcaya-Loinaz et de Castro, 2002).

En conclusion, l'intégration des agents intelligents dans le domaine des EIAH permet de dynamiser le processus d'apprentissage et de moderniser l'éducation (Agarwal, Deo et Das, 2004). Les agents intelligents permettent d'enrichir les EIAH en adaptant la présentation de l'information au niveau de connaissance de chaque apprenant et en facilitant la communication et la collaboration entre apprenants et entre apprenant et tuteur.

2.3 Notre apport à la littérature

À travers la revue de littérature, nous avons pu observer les avantages et les grandes qualités des EIAH et des agents intelligents. D'ailleurs, la littérature est remplie d'exemples de projets et d'expériences réussies liés à l'utilisation des EIAH et à l'intégration d'agents intelligents au sein de ces environnements. Certains sont plus complexes à mettre en œuvre, comme AutoTutor (Graesser et al., 2004). D'autres, comme ABALS (Chen et Zhang, 2006), privilégient de passer par les agents dans toutes les tâches. Ce que nous proposons dans ce mémoire se positionne entre ces deux solutions. L'un des environnements qui s'apparente le plus à notre travail et qui nous a beaucoup inspiré est l'environnement « Sphinx ». Bien qu'il n'utilise pas la technologie des agents intelligents, Sphinx (Capus et al., 2006) est un EIAH adaptatif, permettant un apprentissage par l'exemple dans le cadre d'un cours d'intelligence artificielle. Réalisé à l'université Laval, Sphinx est un environnement dédié à l'apprentissage collaboratif et à partir d'exemples. Grâce au modèle de l'apprenant qu'il crée et met à jour

constamment, Sphinx est capable d'offrir une aide en temps réel. Tout comme AARTIC, Sphinx est un environnement Web permettant d'encourager les étudiants à travailler plus « en dehors des cours ». Il est également destiné à un cours d'intelligence artificielle. Le tableau 2.1 présente les différences entre notre système et Sphinx (Capus et al., 2006).

Tableau 2.1 Comparaison entre l'application Sphinx et AARTIC

Critères	**Sphinx**	**AARTIC**
Exemples et concepts	Pour tout le cours	Pour une partie du cours : destinée à la réalisation du projet de laboratoire
Sélection exemple	En fonction de l'avancement du cours	Selon le cours et **surtout** selon l'étape de développement du système intelligent (avancement du laboratoire).
Ajout des exemples	Chaque semaine (d'ailleurs les exemples sont présentés par semaine)	Par étape (on suit mieux ainsi le rythme de l'étudiant)
Guidage des étudiants	« En fonction des connaissances de l'étudiant » (Emmanuel, Capus et Tourigny, 2006, p. 212)	En fonction du niveau de connaissance de l'étudiant : obtenu à partir des résultats des tests QCM proposés par le système
Tutorat et Conseils automatiques	Sphinx utilise des « gabarit de messages »	L'agent intelligent intervient au moment opportun.

Comme le montre le tableau 2.1 AARTIC permet d'aider les étudiants dans la réalisation des laboratoires associés au cours. Les concepts théoriques sont rassemblés par étape (qui peut représenter une phase de la réalisation du laboratoire) où chaque concept est illustré par différents exemples. Ces derniers sont présentés à l'étudiant selon son niveau de connaissance. Notre environnement compte également un agent intelligent qui intervient selon des situations nécessaires afin de guider et d'orienter l'étudiant.

CHAPITRE 3

ANALYSE ET CONCEPTION PRÉLIMINAIRES AVEC MISA

Dans ce chapitre, nous définissons l'analyse et la conception préliminaires de notre environnement AARTIC grâce à la méthode MISA. Nous commençons par présenter cette méthode. Par la suite nous nous intéressons à la phase d'analyse et de conception de la méthode ce qui nous permet de définir plus en détail l'environnement d'apprentissage à développer. Nous terminons le chapitre par une évaluation et une critique de la méthode MISA.

3.1 Présentation de MISA

MISA est une méthode d'ingénierie des systèmes d'apprentissage réalisée au centre de recherche LICEF[3] (laboratoire en informatique cognitive et environnements de formation à la Télé-université). Elle permet de guider les concepteurs et les différents membres impliqués (enseignant, expert de contenu, gestionnaire, etc.) dans la réalisation d'un système d'apprentissage. MISA se base sur :

- la théorie d'apprentissage constructiviste selon laquelle l'apprentissage est considéré comme une « construction » de notre propre vision de la réalité et non une simple reproduction de celle-ci (Paquette, Crevier et Aubin, 1998, p. 15),
- le design pédagogique : il s'agit d'un processus « complexe » (Paquette, Crevier et Aubin, 1998) permettant de définir et d'utiliser les méthodes de formations dans le but d'améliorer l'apprentissage.

---

[3] http://www.licef.ca/

Paquette (Paquette, Crevier et Aubin, 1997) définit MISA comme une méthode dont l'objectif principal est de développer un système d'apprentissage défini à travers trois modèles : le modèle de connaissances permettant de préciser le contenu d'apprentissage (ce qu'il y a à apprendre), le modèle pédagogique permettant d'identifier les éléments pédagogiques à utiliser pour apprendre (comment apprendre et quels outils utiliser) et le modèle médiatique permettant de déterminer comment rendre le système d'apprentissage réalisé accessible à l'étudiant (où apprendre et par quel moyen apprendre). Grâce à MISA, on obtient à la fin « le devis d'un système d'apprentissage » (Paquette, Crevier et Aubin, 1997, p. 2).Ce devis est réalisé à travers les 5 phases de MISA : l'analyse et la conception préliminaires, l'architecture, la conception des matériels, la validation et la mise en place. Tout au long de ces phases, on passe par différentes étapes nommées « élément de documentation » ou ED. La MISA est composée, au total, de plus de 33 éléments de documentation différents.

L'une des grandes qualités de la MISA est le degré de liberté infini qu'elle accorde au concepteur d'un système d'apprentissage. Cette liberté réside dans le choix de l'approche pédagogique (apprentissage par problème; apprentissage par projet; apprentissage basé sur les cas, etc.), le choix du mode de livraison (à distance, présentiel, autoformation, etc.) ou encore dans le choix du support médiatique (CD, site Web, etc.).

Afin de cerner les besoins en apprentissage et d'identifier les objets pédagogiques nécessaires pour répondre à ces besoins, nous avons profité de la méthode MISA. Nous nous sommes intéressés, principalement, à la première phase de la méthode : la phase d'analyse et de conception préliminaires.

3.2 Analyse et conception préliminaires avec MISA

La méthode MISA permet d'identifier nos besoins et nos objectifs d'apprentissage. La phase d'analyse et de conception préliminaires de MISA, nous permet, plus précisément, de déterminer la faisabilité de notre projet de réalisation d'un environnement d'apprentissage.
Elle permet aussi de définir de façon détaillée l'environnement à développer et de faire les bons choix en matière de solution médiatiques et pédagogiques. Cette phase est composée de quatre étapes :

- Étape 1 définition du problème de formation : déterminer le contexte d'apprentissage actuel, ses faiblesses et la solution proposée,
- Étape 2 identification des objets d'apprentissage : concevoir le modèle de connaissances,
- Étape 3 définition de l'approche pédagogique : définir les orientations pédagogiques et représenter le réseau d'événement d'apprentissage,
- Étape 4 définition de l'approche médiatique : déterminer la solution médiatique sélectionnée.

Dans ce qui suit, nous décrirons la phase d'analyse et de conception préliminaires pour l'environnement AARTIC à travers ces quatre étapes.

3.2.1 Définition du problème de formation

Nous avons précédemment décrit en détail le problème de formation. Nous référons le lecteur au premier chapitre de ce mémoire à la section « problématique spécifique ».

En bref, notre objectif est d'apporter les améliorations suivantes :

- Maîtriser et comprendre les notions complexes du cours,
- Développer une communication plus efficace entre apprenants et entre apprenant-enseignants,
- Tirer profit de l'apprentissage par l'exemple et de l'apprentissage collaboratif avec les forums de discussion,
- Guider l'apprenant et l'encadrer de manière plus personnalisée,
- Collaborer entre étudiants pour atteindre le même objectif : expliquer un exemple.

L'environnement AARTIC vise à aider les étudiants dans la réalisation du premier projet de laboratoire associé au cours LOG625 « Introduction aux systèmes intelligents ». Le système[4] assiste les étudiants dans l'assimilation des concepts théoriques du cours nécessaires pour la réalisation de leur projet. La présentation des concepts se fait à travers des exemples présentés par le système selon le niveau de connaissance de l'étudiant. Cette adaptabilité est assurée par l'agent intelligent dit « agent apprenant » qui sert de guide-accompagnateur de chaque étudiant utilisateur du système et de l'agent enseignant qui permet à l'enseignant de voir l'avancement de chacun des étudiants. Le système n'interviendra pas dans le développement du projet de laboratoire en tant que tel mais dans la compréhension de notions théoriques du cours nécessaires pour le bon déroulement du projet. La présentation de chaque concept se fera à travers une définition mais surtout au moyen de l'exemple illustrant le concept. L'étudiant pourra alors tenter d'expliquer, à travers le forum de discussion associé au système, l'exemple qu'il vient de voir.

[4] En MISA, on a tendance à parler de système d'apprentissage qui représente en réalité le noyau de l'environnement d'apprentissage (du EIAH).

Rapport-Gratuit.com

Il s'agit d'un projet plutôt complexe pour différentes raisons. D'abord, le cours visé (LOG625) est un cours de 3 crédits donné à des étudiants en baccalauréat en génie logiciel en fin de formation (utilisateurs assez exigeants et qui n'ont pas de temps à perdre). Ensuite, le contenu du cours est assez complexe et dense. Le projet de laboratoire est perçu à priori comme un projet plutôt compliqué (connaissance en intelligence artificielle et système expert, maîtrise du langage JESS). De ce fait, plusieurs questions se posent : comment présenter les concepts? Quels exemples choisir et comment les classer? Comment développer et intégrer des agents intelligents qui sont associés au système? Nous répondrons à l'ensemble de ces questions au cours du présent chapitre et au chapitre suivant.

3.2.2 Identification des objets d'apprentissage

La représentation des connaissances (qu'elles soient de type concept, procédure ou principe) et les objectifs pédagogiques de notre système se font à travers le graphe de connaissance de MISA. Ce dernier, présenté à la figure 3.1, est l'un des graphes de bases de la méthode MISA.

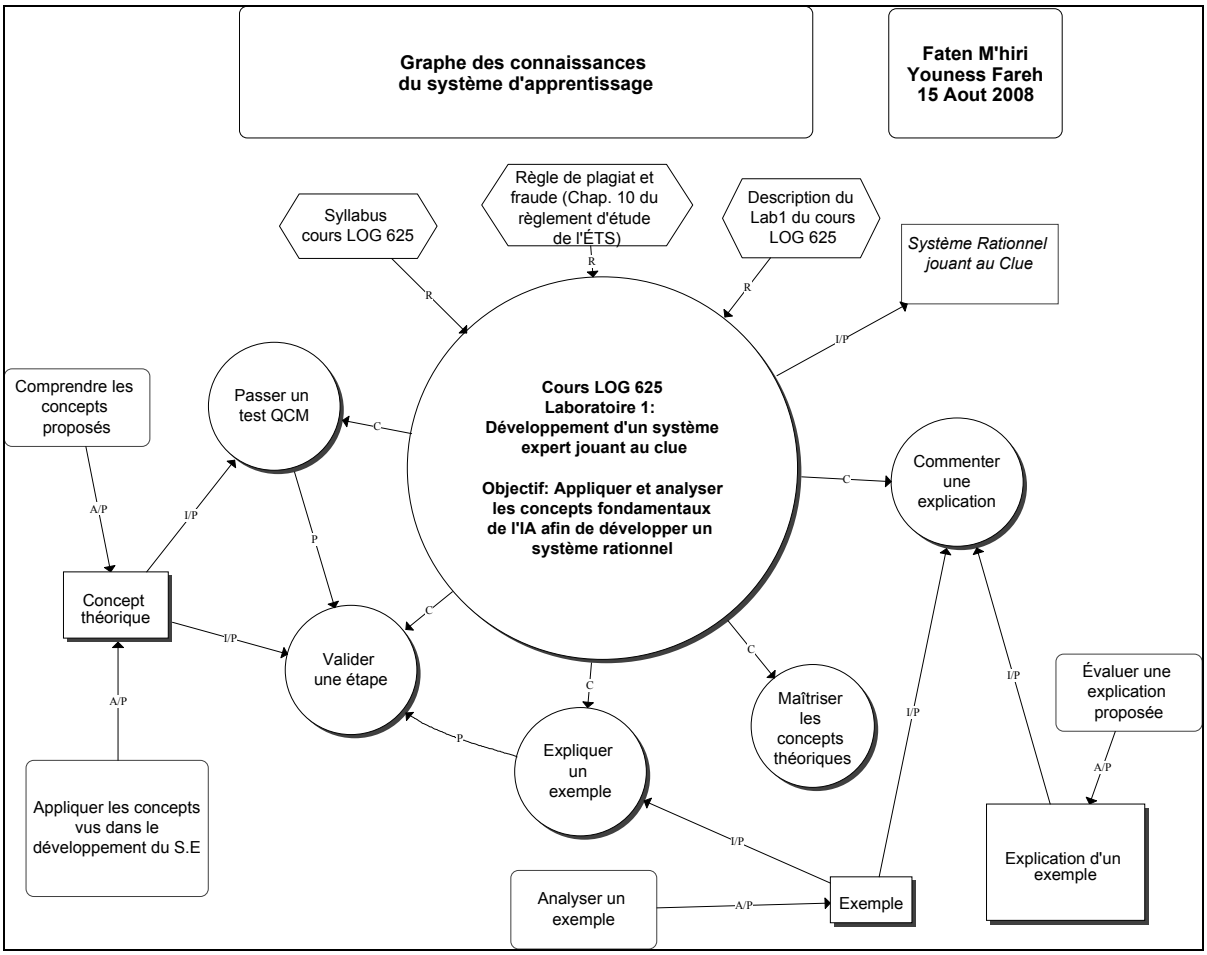

Figure 3.1 Graphe des connaissances du système d'apprentissage AARTIC.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Le tableau 3.1 permet de bien comprendre les éléments du graphe de connaissance et leur signification dans le cadre de notre projet.

Tableau 3.1 Description des différents éléments du graphe de connaissances

Symbole	**Signification**	**Définition**	**Cas de l'environnement AARTIC**
	Procédure	Une procédure décrit comment faire pour traiter les objets	Les procédures dans AARTIC : maîtriser un concept théorique, expliquer un exemple ou encore passer un test QCM
	Habilité	« connaissance » permettant de gérer (comprendre, évaluer, étudier,…) un fait	Les habilités identifiées dans AARTIC : l'habilité « analyser » ou encore « appliquer»,…
	Principe	Un principe décrit les propriétés d'un objet définissant ainsi les liens et les règles gérant les objets pédagogiques.	Parmi les principes on trouve les règles énoncées dans le syllabus du cours ou encore les règles de plagiat de l'ETS.
	Concept	Un concept permet de définir la nature d'un objet.	Dans notre cas, il s'agit de traiter avec des exemples, des concepts théoriques, etc.

3.2.3 Définition de l'approche pédagogique

Dans son plan de recherche stratégique (Décanat des études: ÉTS, 2008), l'École de technologie supérieure encourage l'innovation en enseignement universitaire et l'intégration de nouvelles pratiques de formation plus adaptée aux étudiants d'aujourd'hui. Dans le but de répondre à ces orientations, nous avons défini nos objectifs d'apprentissage de la manière suivante :

- Concevoir un système accessible à partir du site Web du cours LOG625, intégrant un forum de discussion. Ceci répond, en partie, à la volonté de l'École d'intégrer les Nouvelles Technologies de l'information et des Communications (NTIC) dans l'enseignement, afin d'améliorer la qualité de celui-ci (Décanat des études: ÉTS, 2008),
- Utiliser divers moyens technologiques permettant de personnaliser l'apprentissage. De cette façon, notre système participera à promouvoir un apprentissage adaptatif en ingénierie et améliorer les résultats des apprenants dans les cours assez complexes,
- Concevoir un système d'apprentissage qui favorise un suivi plus personnalisé de l'avancement des étudiants de la part de l'enseignant. Ainsi AARTIC participera à « Améliorer la qualité de l'enseignement et la qualité des apprentissages dans les cours et les laboratoires. » (ÉTS, 2008, p. 1),
- Profiter des avantages de l'enseignement de type inductif (Prince et Felder, 2006) selon lequel, à partir d'une observation, d'une expérience ou d'un problème à résoudre, l'apprenant ressent le besoin d'une nouvelle information, d'un nouveau principe. Plus précisément, cette approche favorise un apprentissage par problèmes plus autonome, actif et collaboratif,
- Mettre en place un système d'apprentissage générique pour qu'il puisse s'appliquer à d'autres projets de laboratoire ou cours.

Le réseau des évènements d'apprentissage (ou REA), illustré à la figure 3.2, permet d'identifier les différents événements d'apprentissage de notre système et les relations entre ces évènements. Les relations entre les évènements d'apprentissage (ou EA) comprennent entre autre la composition (la flèche avec la lettre « C » dans le graphe) et la précédence (la flèche avec la lettre « P » dans le graphe).

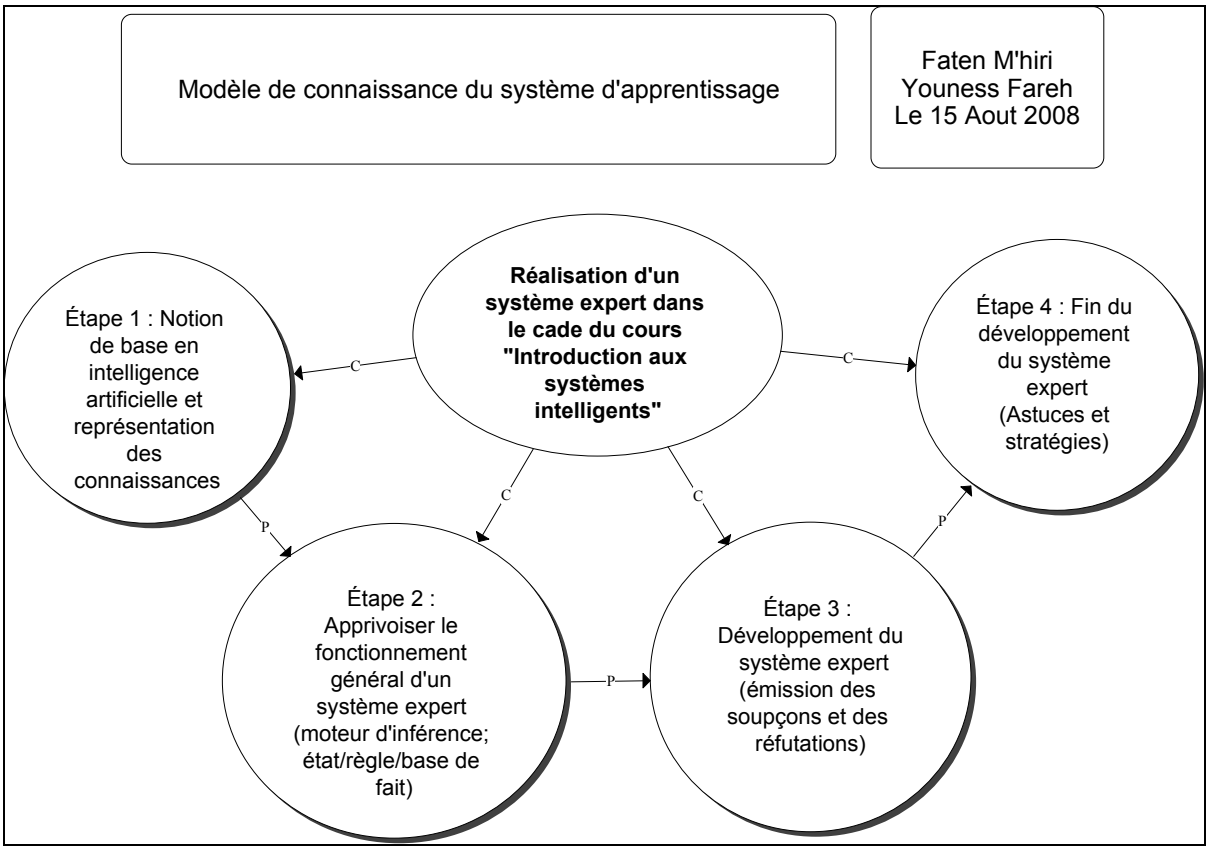

Figure 3.2 Réseau des évènements d'apprentissage.

Le tableau 3.2 présente chacun des évènements d'apprentissage (EA) représentés dans le graphe et des habilités reliées.

Tableau 3.2 Définition des éléments du réseau des évènements d'apprentissage

Titre de l'EA	**Objectifs (habilités)**	**Connaissances**
Projet de laboratoire : Réalisation d'un système expert (agent intelligent capable de jouer au Clue) dans le cadre du cours introduction aux systèmes intelligents	- Développer des habilités dans le développement d'un système expert, - Maîtriser JESS et appliquer les concepts théoriques en IA dans le développement du système expert.	Ensemble des connaissances visées par le projet de laboratoire : modéliser un agent logique, utiliser le module JESS, respecter les clauses de Horn, respecter la forme normale conjonctive. (Ratté, 2008)
Étape 1 : Notion de bases en IA et représentation des connaissances	- Se familiariser avec le système d'apprentissage, - Comprendre et analyser les exemples et les concepts de base associés, - Appliquer les concepts théoriques de base dans le développement du système expert (SE).	- Connaissances médiatiques, - Concepts théoriques de base en IA et en représentation des connaissances.

Tableau 3.3 Définition des éléments du réseau des évènements d'apprentissage (suite)

Titre de l'EA	**Objectifs (habilités)**	**Connaissances**
Étape 2 : Apprivoiser le fonctionnement général d'un système expert (moteur d'inférence; état/règle/base de fait)	- Comprendre et analyser les exemples et les concepts associés, - Appliquer les concepts théoriques dans le développement du SE.	- Concepts théoriques de bases, - Analyse et explication d'exemple, - Développement d'un agent rationnel capable d'émettre des soupçons dans le jeu Clue.
Étape 3 : Développement du système expert (émission des soupçons et des réfutations)	- Comprendre et analyser les exemples et les concepts associés, - Appliquer les concepts théoriques dans le développement du SE.	- Concepts théoriques de bases, - Analyse et explication d'exemple, - Développement d'un agent rationnel capable de réfuter les accusations des autres joueurs dans le jeu Clue.
Étape 4 : Fin du développement du système expert (Astuces et stratégies)	- Comprendre et analyser les exemples et les concepts associés, - Appliquer les concepts théorique dans le développement du SE, - intégrer les stratégies du jeu Clue.	- Concepts théoriques de bases, - Analyse et explication d'exemple, - L'agent développé peut jouer une partie entière en faisant des soupçons et des réfutations tout en appliquant des stratégies et des astuces dans le but de gagner la partie.

Une fois arrivé à ce stade de la phase d'analyse et de conception préliminaires, le scénario général de l'utilisation de notre système d'apprentissage est développé :

L'étudiant doit réaliser, en première partie de la session du cours, un projet de laboratoire. Ce projet a pour objectif de développer un agent rationnel capable de jouer au jeu de stratégie Clue (ou Cluedo). Le système d'apprentissage aide l'étudiant afin de mener à bien son projet de laboratoire. Le système va donc inciter l'étudiant à passer par différentes étapes, chaque étape est équivalente à une phase du projet et elle offre à l'apprenant une liste des concepts théoriques à voir et à maîtriser grâce aux exemples offerts par le système illustrant chaque concept. Pour passer d'une étape à l'autre, chaque apprenant doit répondre à un questionnaire proposé par le système et qui va évaluer, si oui ou non, l'apprenant maîtrise les concepts donnés à l'étape courante.

L'étudiant pourra aussi communiquer, proposer une explication pour un exemple, ou commenter l'explication d'un autre étudiant grâce au forum de discussion associé au système d'apprentissage. Le forum de discussion permet, non seulement, la communication entre les apprenants mais aussi entre l'étudiant et l'enseignant.

Afin de garantir l'utilisation du forum de discussion, nous allons lier celui-ci à la réalisation d'une activité de groupe où des équipes de 2 à 4 étudiants tentent d'expliquer le même exemple distribué par l'enseignant ou l'auxiliaire de laboratoire. Chaque équipe propose son explication sur le forum de discussion et il y aura élection pour la meilleure explication proposée. Après validation de l'enseignant/auxiliaire de laboratoire, cette explication sera associée à l'exemple sur le système.

3.2.4 Définition de l'approche médiatique

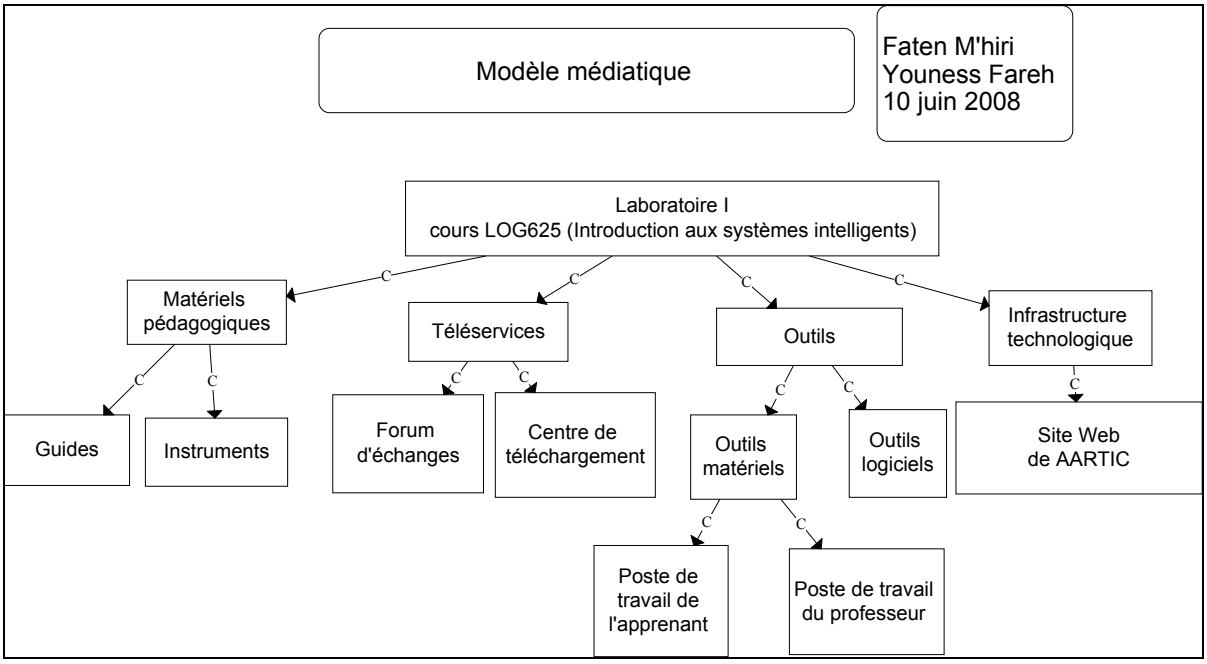

Figure 3.3 Graphe du modèle médiatique.

Le Modèle médiatique (Niveau I) du système d'apprentissage dédié au premier laboratoire du cours LOG625 et illustré à la figure 3.3, représente une vision préliminaire du support médiatique appelé à intégrer notre contenu pédagogique ainsi que les connaissances déjà détaillées dans le graphe des connaissances.

Respectant les recommandations de la méthode MISA, ce modèle se compose de quatre sections. La section « matériels pédagogiques » englobe tous les instruments comme les définitions des concepts théoriques du cours LOG625, les exemples les illustrant, le contenu des questionnaires, les guides pédagogiques tels que la description du premier laboratoire ainsi que les autres éléments documentaires. La section intitulée « télé services » contient principalement le forum qui sera consacré aux échanges entre les étudiants et aussi le centre de téléchargement des différents guides et de la documentation pertinente.

La partie « Outils » rassemble autant les outils matériels que logiciels comme l'agent intelligent et la base de données MySQL. Enfin, la section « infrastructure technologique » représente le site Web de l'environnement AARTIC qui donne accès à toutes les fonctionnalités et les éléments des sections que nous venons de citer (les choix technologiques et médiatiques sont présentés plus en détails au chapitre 5).

3.3 Critique de la méthode MISA

La méthode MISA couvre en profondeur la conception d'un système d'apprentissage grâce aux différentes phases de la méthodologie et aux divers éléments graphiques produits. Le dossier d'analyse et de conception préliminaires de MISA, nous a permis de tirer avantage d'un des points forts de MISA : évaluer la situation d'apprentissage actuelle et décrire en détail les améliorations possibles pour bonifier l'apprentissage en identifiant les besoins pédagogiques et technologiques.

Il est vrai que MISA est une méthodologie complète qui permet de cerner le problème et de poser les bonnes questions tant sur l'aspect pédagogique que du côté technologique d'un système d'apprentissage. Cependant, pour développer un EIAH complet et pour passer à la phase d'implémentation, nous avons constaté que la transition de la conception MISA vers l'implémentation du système est fastidieuse. Du fait de sa densité et de sa précision, la méthodologie MISA est une méthodologie assez complexe avec beaucoup d'éléments de documentation : le risque de ne pas disposer du temps nécessaire pour la suite du développement de notre système est ainsi devenu très présent. De plus MISA ne propose pas d'outils capables de nous indiquer la structure et l'interaction des différents éléments dans le système comme le ferait un langage de conception orienté objet tel que le langage UML (Unified Modeling Language).

Rapport-Gratuit.com

Notre réflexion va dans le même sens que la constatation de Henri, Compte et Charlier:

> « La scénarisation pédagogique s'inscrit alors dans un processus de conception/production qui doit se plier à des méthodes normalisées d'ingénierie pédagogique, comme la MISA du Centre de recherche LICEF, **et** à des méthodes de développement informatique associées ou dérivées du langage Unified Modeling Language (UML) tel que le propose l'IMS Global Learning Consortium, le consortium le plus important en matière de spécification de contenus éducatifs. » (Henri, Compte et Charlier, 2007, p. 19)

C'est pour ces raisons que nous avons décidé de réaliser uniquement la phase d'analyse de MISA et de poursuivre la conception avec un langage mieux adapté. Nous avons donc choisi d'utiliser le langage de modélisation UML pour représenter les objets du système et ainsi continuer le développement de celui-ci tout en tirant profit de l'analyse et de la conception préliminaires réalisées avec MISA. Cela nous a permis de gagner un temps considérable dans l'avancement du projet et d'avoir une structure plus adaptée à la programmation.

CHAPITRE 4

CONCEPTION UML DE L'ENVIRONNEMENT AARTIC

Comme nous l'avons mentionné précédemment, lors de la conception d'un EIAH, on se retrouve devant un bon nombre d'obstacles dû à la multidisciplinarité du domaine. En effet, un EIAH relève non seulement du monde de l'éducation mais aussi de celui des systèmes informatiques. Bien que la méthode MISA nous permette de résoudre des difficultés qui relèvent de l'analyse et la conception d'un système d'apprentissage, elle demeure une méthodologie assez complexe. De plus, ses éléments graphiques ne permettent pas de représenter une solution facile à implémenter. D'où le besoin d'un langage de modélisation objet comme UML plus approprié à l'implémentation de notre système.

Dans ce chapitre, nous débutons par une brève présentation du langage UML. Ensuite, nous exposerons l'architecture et le fonctionnement de notre environnement AARTIC. Nous poursuivons en détaillant les différents modules de l'architecture AARTIC grâce aux diagrammes de classes UML. Enfin, nous présenterons la dynamique de notre environnement à travers les diagrammes d'activité d'UML.

4.1 Le langage UML

UML est un langage de modélisation graphique et textuel permettant de décrire et de comprendre les besoins et les spécifications de l'utilisateur. C'est un langage qui n'impose pas de méthode de travail particulière. Ainsi, il peut s'intégrer à n'importe quel processus de développement logiciel. UML fournit plusieurs outils permettant de représenter l'ensemble

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

des éléments du monde objet. C'est pour ces raisons qu'UML est le standard du « Object Management Group » le plus utilisé[5].

Dans le cadre des environnements informatiques pour l'apprentissage humain, on trouve diverses « spécialisations » du langage UML utilisées pour modéliser la conception d'un EIAH (Nodenot, 2007, p. 91). L'une de ces spécialisations est le langage CPM (Laforcade, 2005) ou *Cooperative Problem-based learning Metamodel*. CPM se limite à la modélisation de la phase d'analyse préliminaire pour la conception de situations d'apprentissage par problèmes.

Puisque nous avons déjà utilisé la méthodologie MISA pour l'analyse et la conception, nous nous sommes limités, dans la suite de la conception de notre système, à deux diagrammes du langage UML : un diagramme statique, le diagramme de classe et un diagramme dynamique, le diagramme d'activité.

4.2 Description et Architecture de l'environnement AARTIC

4.2.1 Description d'AARTIC

L'environnement AARTIC propose à l'étudiant des définitions des concepts théoriques du cours illustrés par différents exemples. Ceci permet à l'étudiant d'approfondir ses connaissances et de maîtriser la théorie afin de mieux l'appliquer dans le cadre de travaux pratiques et autres projets de laboratoire. AARTIC a été modélisé de manière à aider les apprenants dans l'acquisition de ces connaissances et les guider tout au long de leur réalisation de projets de laboratoire. L'environnement offre un parcours linéaire à l'étudiant grâce à l'approche par étapes conditionnée par les résultats des questionnaires.

---

[5] http://www.uml.org/

La figure 4.1 illustre bien le fonctionnement d'AARTIC. Tout au long de la réalisation de son projet de laboratoire, l'étudiant passe par différentes étapes; chaque étape correspond à une phase du projet à réaliser et offre une liste des notions théoriques à voir; chaque notion est illustrée par une définition et des exemples de différents niveaux de difficulté (débutant, intermédiaire et avancé).

Selon le niveau de connaissance de l'étudiant, un exemple lui est proposé. L'étudiant peut évidemment tenter d'expliquer chaque exemple dans un forum de discussion associé au système et ses camarades peuvent intervenir. La fin de chaque étape est marquée par la réalisation d'un test QCM (Questionnaire à Choix Multiples) et d'une activité en laboratoire. Le test QCM a pour but d'évaluer les connaissances de chaque étudiant et de mettre à jour son profil. L'activité requiert une heure de la séance de laboratoire. Durant celle-ci, des équipes d'étudiants tentent d'expliquer le même exemple et mettent leur explication en ligne sur le forum. À la fin de cette activité, toutes les équipes vont voter, à travers le système pour la meilleure des explications proposées.

Durant son utilisation de l'environnement AARTIC, l'étudiant est suivi par un agent intelligent appelé « agent apprenant ». Celui-ci permet d'analyser les activités de l'étudiant afin de mieux le guider et de l'aider (en lui proposant, les concepts théoriques à voir avant de passer un test QCM ou encore les concepts pour lesquels il a certaines lacunes,…). De son côté, l'enseignant pourra voir l'avancement de chacun de ses étudiants et avoir ainsi de l'information pertinente sur les capacités et les difficultés de la classe.

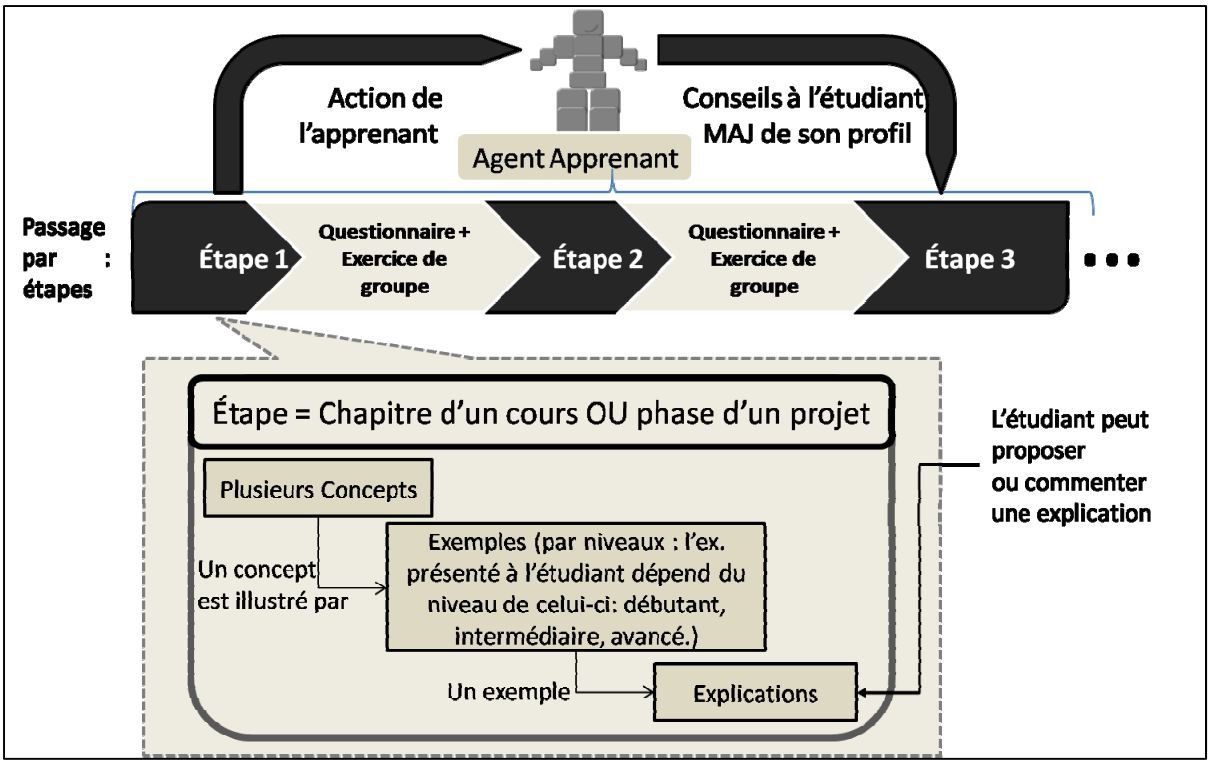

Figure 4.1 Fonctionnement de l'environnement AARTIC.

L'utilisation de l'environnement AARTIC permet l'amélioration et la stimulation de certaines habilités cognitives de l'apprenant telles que : la lecture et la compréhension, l'écriture, la pensée critique et l'apprentissage, ceci grâce aux différents outils que proposent l'environnement et qui permettent le développement de ces habilités tels que le site Web, la carte conceptuelle, les tests QCM, les forums de discussion, les agents intelligents, etc.

4.2.2 Architecture de l'environnement AARTIC

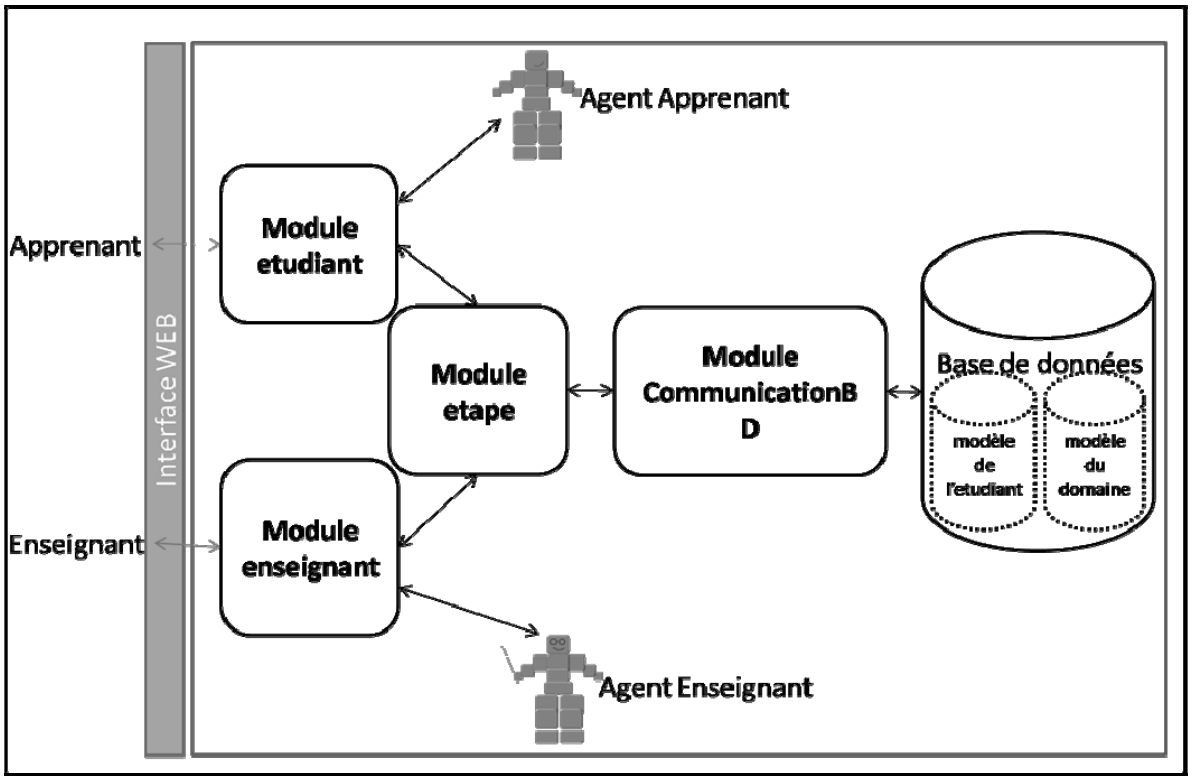

Figure 4.2 Architecture de l'environnement AARTIC.

L'architecture du système présentée à la figure 4.2 comporte les éléments suivants :

- Module *etudiant* : Associé à chaque étudiant, ce module propose diverses fonctions telles que la visualisation des exemples proposés, la participation au forum de discussion, la réalisation des tests d'évaluation. Ce module collabore avec l'Agent Apprenant pour la construction du modèle de l'apprenant, la génération de conseils, etc.

- Module *enseignant* : à travers ce module, l'enseignant peut accéder au système, ajouter, modifier ou supprimer un exemple, participer au forum, modifier un test d'évaluation, voir les statistiques concernant l'activité de la classe, etc. Ce module collabore avec l'Agent Enseignant,

- Module *etape* : comporte toutes les classes nécessaires à la définition d'une étape : les concepts théoriques, les exemples les illustrant, le questionnaire associé à la fin de chaque étape,

- Module *communicationBD* : il permet aux modules étudiant et enseignant de communiquer avec la base de données MySQL,

- Base de données : on y retrouve le modèle de l'apprenant. Certaines informations (nom, code, date naissance, pseudonyme,…) sont saisies par l'apprenant lui-même ou par l'enseignant. D'autres informations concernant le parcours et l'apprentissage de l'apprenant, par exemple son niveau d'apprentissage, sont calculées à partir des résultats des questionnaires. La base de données comporte également le « modèle du domaine » (Réty et al., 1998, p. 5). Ce modèle comporte les entités pédagogiques tels que les différentes définitions de concepts, les exemples proposés, les niveaux d'apprentissage associés à chaque exemple, etc.

- Agent Apprenant : il interagit avec l'étudiant et possède certaines fonctionnalités de tutorat. L'agent apprenant a pour tâche de :

  - accéder à toutes les informations concernant l'apprenant (nombre d'exemples expliqués (individuel et en groupe), résultats aux tests, exemples consultés,…) en vue de construire le modèle de l'apprenant : il s'agit là d'un rôle d'observation (Loghin, Carron et Marty, 2007b),
  - générer des conseils suite à un test passé : exemple et/ou concept à voir ou à revoir : ici l'Agent Apprenant joue le rôle d'un tuteur, accompagnateur.

- Agent Enseignant : l'agent enseignant récupère les données de l'agent apprenant et retourne à l'enseignant, les informations les plus pertinentes, telles que les étapes actuelles de chaque étudiant, le progrès de la classe, les concepts les plus difficiles à assimiler, etc.
- Interface WEB : Elle sert de lien entre les divers utilisateurs et le système.

4.3 Diagramme de classe UML de l'environnement AARTIC

Dans cette partie, nous décrivons de manière plus détaillée chacun des modules de l'architecture d'AARTIC. Nous commençons par explorer les modules *etudiant* et *communicationBD* et la relation entre les deux. Ensuite, nous nous intéressons au module *etape* et de ses classes. Enfin, nous définissons la structure du module *enseignant* et sa relation avec *communicationBD*.

4.3.1 Modules « etudiant » et « communicationBD »

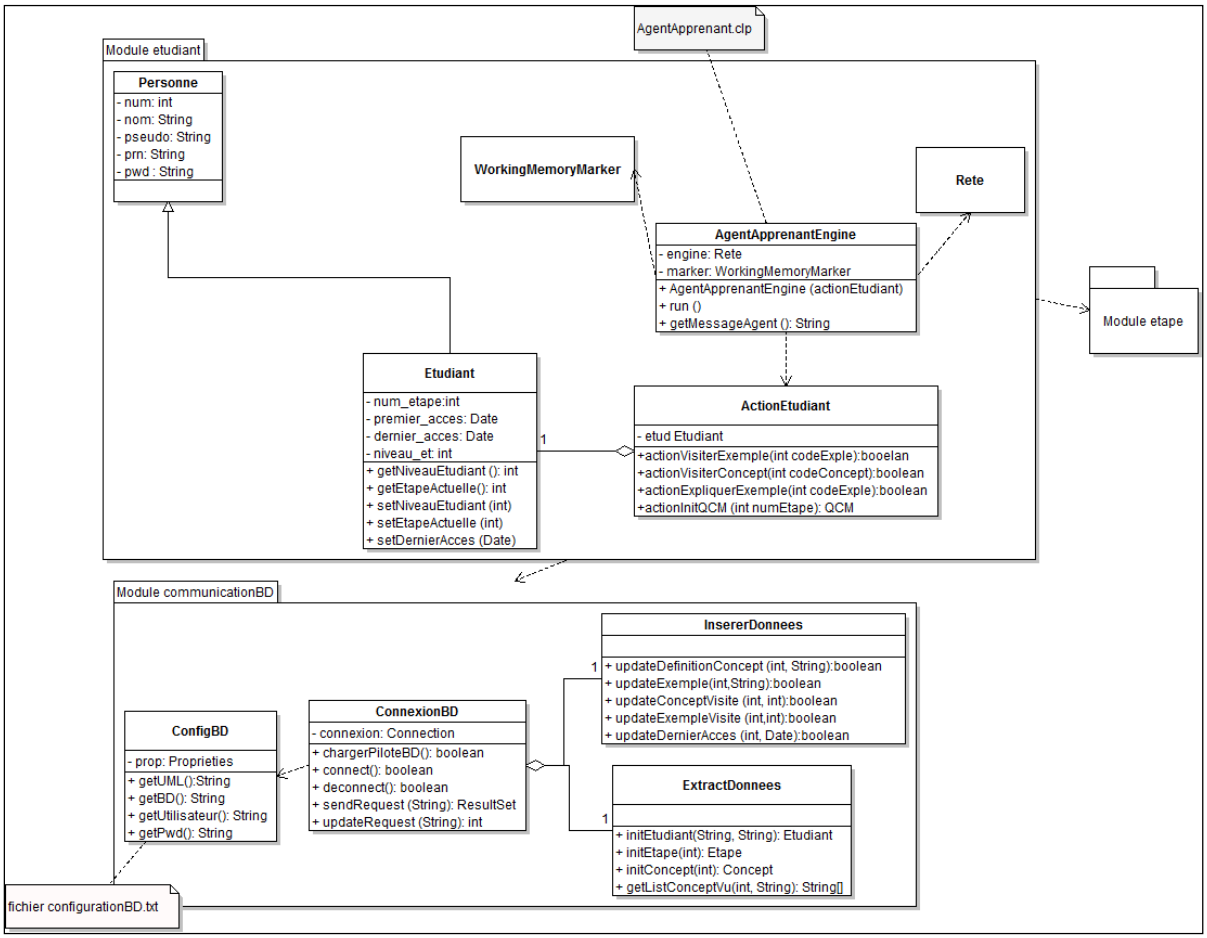

Figure 4.3 Diagramme de classe UML du module etudiant et du module communicationBD.

Le module etudiant :

Ce module comporte les informations et les fonctionnalités associées à un étudiant, utilisateur de l'environnement AARTIC. Il est formé de quatre classes principales :

Classe *Personne* : classe abstraite qui comporte tous les éléments définissant un utilisateur de l'environnement AARTIC comme numéro d'identification, nom, prénom, pseudonyme, et mot de passe pour s'authentifier sur le système;

Classe *Etudiant* : comporte les informations spécifiques à chaque étudiant (le modèle étudiant) et les méthodes pour y accéder. Les paramètres les plus importants sont *niveau_et*, qui correspond au niveau de connaissance débutant, intermédiaire ou avancé de l'étudiant (ce qui correspond respectivement aux valeurs 1,2 ou 3) *et numero_etape* qui indique l'étape actuelle à laquelle l'étudiant est arrivé;

Classe *ActionEtudiant* : regroupe les fonctions de l'étudiant sur le système. Cette classe représente le cœur du module étudiant. Elle est associée à la classe *Etudiant* et *AgentApprenantEngine*. Parmi ses méthodes, ont trouve : *actionVisiterConcept* (pour afficher la définition du concept et l'exemple sélectionné selon le niveau de l'étudiant), et la méthode *actionExpliquerExemple* (qui signale que tel exemple a été expliqué par l'étudiant). Cette classe interagit avec la classe *ExtractDonnees* du module *CommunicationBD* et la classe *Etape* du module *etape*;

Classe *AgentApprenantEngine* : Cette classe établi le lien entre le code Java et le code JESS dans le fichier *AgentApprenant.clp*. Elle crée un objet nommé *engine* de type « Rete » et un autre objet *marker* de type « WorkingMemoryMarker ». Les classes « Rete » et «WorkingMemoryMarker » appartiennent à la bibliothèque JESS (à importer dans le code Java). La méthode « *run* » de cette classe, a pour but d'exécuter les règles du fichier « *AgentApprenant.clp* ».

Rapport-Gratuit.com

Le fichier *AgentApprenant*.*clp* : rassemble la base des règles à exécuter. Voici une liste des règles qu'on peut trouver dans le fichier :

- Interface d'accueil (à la connexion de l'étudiant) : l'agent va lancer un message d'accueil,
- Passer QCM (après que l'étudiant ait choisi de passer le QCM): l'agent va vérifier si tous les concepts importants ont été vus pas l'étudiant,
- L'agent va vérifier à la fin du test s'il a des problèmes avec certains concepts et, le cas échéant, lui conseille de revoir certains de ceux-ci,
- Choix d'un exemple selon concept/niveau (pour l'instant c'est le programme Java qui se charge de cette fonction),
- Passer à l'étape suivante : l'agent va vérifier si l'étudiant a participé à une activité de groupe, aussi s'il a passé le test QCM et il l'a réussi et s'il a expliqué au moins un exemple de l'étape courante. C'est l'agent qui confirme le passage à l'étape suivante.

Le Module communicationBD :

Permet la connexion et l'accès aux données de la base de données MySQL. Ce module est formé de 4 classes :

Classe *ConfigBD* : permet d'extraire les paramètres de connexion (nom de la base de données , son emplacement, …) à partir du fichier *ConfigurationBD.txt*;

Classe *ConnexionBD* : permet de se connecter, déconnecter et d'exécuter des requêtes SQL reliées à la BD;

Classe *ExtractDonnees* : cette classe comporte les méthodes décrivant les requêtes SQL (Structured query language) nécessaires pour avoir les données de la base de données;

Classe *InsertDonnees* : Elle permet la définition de requêtes SQL (Structured query language) relatives à la mise à jour et l'insertion de nouvelles données dans la base.

Le lien entre les modules *communicationBD* et *etudiant* se trouve surtout au niveau des classes *ExtractDonnees* et *InsertDonnees* qui sont instanciées par les classes du module *etudiant*. Le module *etudiant* fait intervenir aussi des classes du module *etape*.

4.3.2 Module « *etape* »

Ce module contient tous les éléments formant une étape du laboratoire. Comme nous l'avons précisé auparavant, une étape est constituée de plusieurs concepts théoriques, chaque concept est illustré par un ou plusieurs exemples. Aussi la fin de chaque étape est marquée par un test QCM. Comme son nom l'indique, chaque question d'un QCM est associée à différentes propositions de réponses. Une ou plusieurs de ces réponses sont correctes. La figure ci-dessous illustre bien le diagramme de classes du module *etape* :

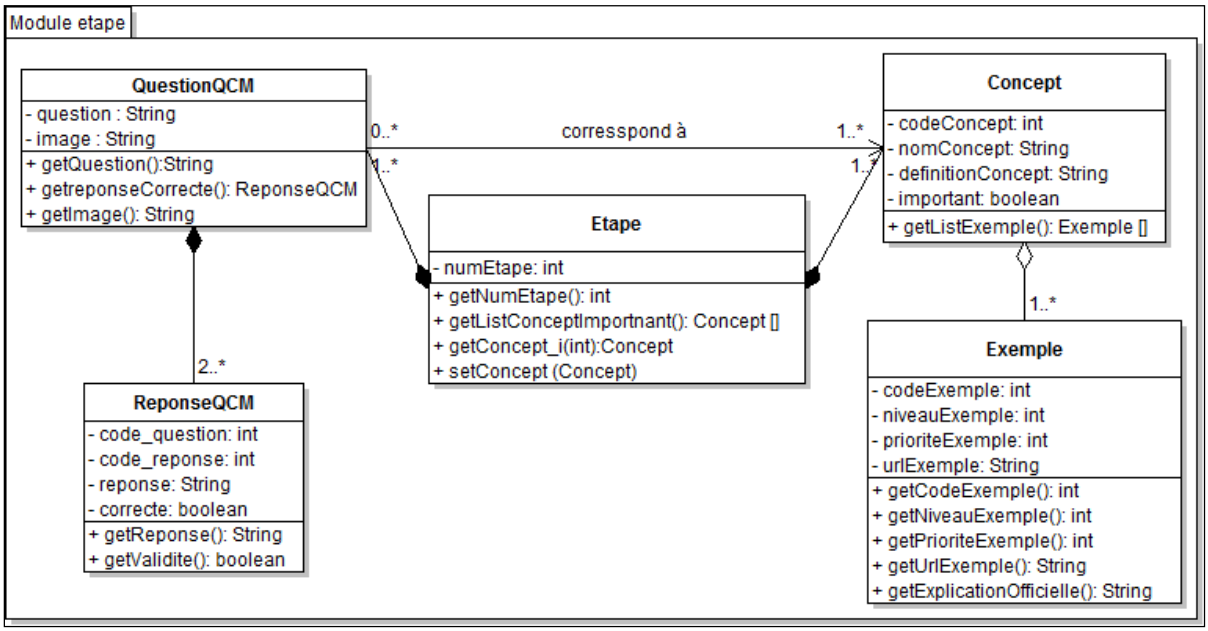

Figure 4.4 Diagramme de classe du module etape.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Classe *Etape* : contient la description de l'étape du projet. Il y a un lien de composition entre Etape et les classes *Concept* et *QuestionQCM*. La classe *Etape* comporte une méthode « *getListConceptImportant* » qui permet de sélectionner les concepts de l'étape qualifiés de « important ».

Classe *Concept* : contient tous les éléments définissant un concept théorique. Parmi ses attributs on trouve : le nom du concept, sa définition, son importance,… La classe *Concept* est composé d'objets de la classe *Exemple*.

Classe *Exemple* : dont les attributs sont : nom de l'exemple, niveau de l'exemple et priorité de l'exemple et URL (*Uniform Resource Locator*) de l'exemple (l'exemple peut être dans un fichier texte ou une image). Grâce à l'attribut *prioriteExemple*, le système peut choisir l'exemple dont la priorité est la plus grande pour illustrer un concept donné (sachant qu'un exemple peut illustrer plus d'un concept théorique).

Classe *QuestionQCM* : contient une question, son identifiant et l'URL de l'image associée à la question (dans le cas où on n'a pas besoin d'une image pour une question donnée, ce champ restera vide). Dans un questionnaire à choix multiples (QCM), une question est associée à plusieurs réponses possibles dont une seule est vraie. Conséquemment, la classe *QuestionQCM* est composée de plusieurs objets de la classe *ReponseQCM*.

Classe *ReponseQCM* : contient le corps de la réponse proposée, son identifiant et sa validité (si la réponse proposée est fausse ou correcte).

4.3.3 Le module enseignant

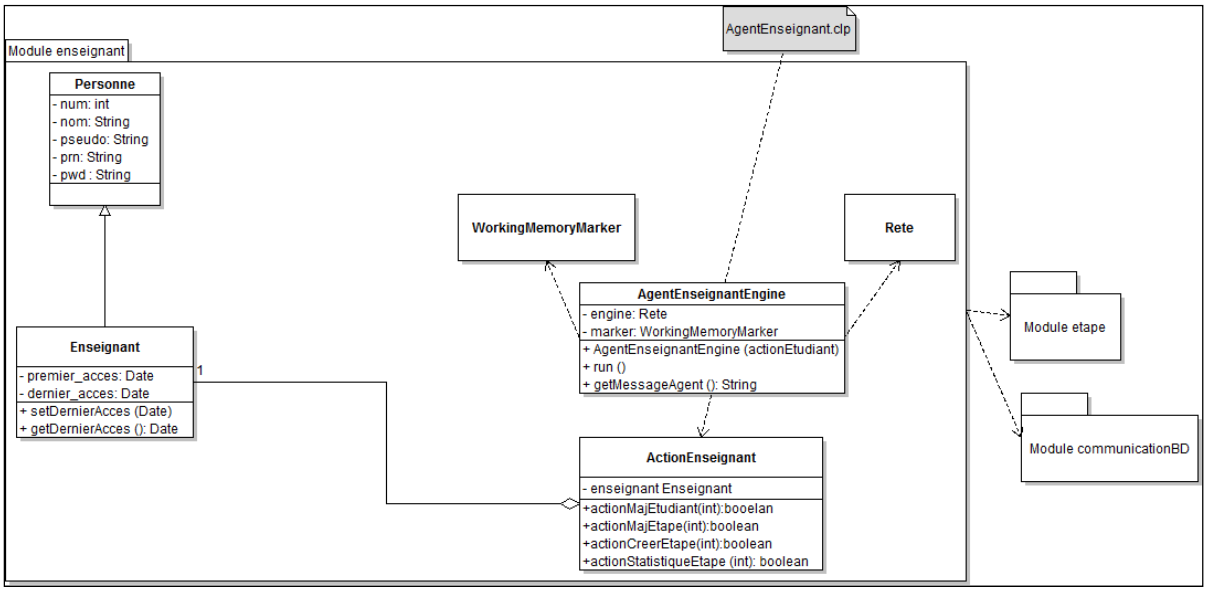

Figure 4.5 Diagramme de classe simplifié du module enseignant.

Le module *enseignant* comporte 3 classes :

Classe *Enseignant* : permet de décrire les informations de chaque enseignant (chargé du cours, auxiliaire de laboratoire) et les méthodes pour y accéder. Tout comme la classe *Etudiant*, cette classe hérite aussi de la classe abstraite *Personne*;

Classe *AgentEnseignantEngine* : fonctionne tout comme AgentApprenantEngine permet d'avoir un lien entre le code Java et le code JESS dans le fichier *AgentEnseignant.clp*;

Classe *ActionEnseignant* : regroupe les fonctionnalités associées à l'enseignant telle que la mise à jour du profil d'un étudiant, la modification d'une étape ou encore la visualisation des statistiques liés à l'étape. Cette classe interagit avec la classe *ExtractDonnees* et *InsererDonnees* du module CommunicationBD.

Le module enseignant est, non seulement en lien avec celui de *communicationBD*, mais aussi avec *etudiant* (notamment la classe Etudiant) et le module *etape*.

4.4 Diagramme d'activité UML de l'environnement AARTIC

4.4.1 L'environnement AARTIC du côté enseignant

Dans l'environnement AARTIC, l'enseignant (ou le chargé de cours ou l'auxiliaire de laboratoire) a toute la liberté possible pour créer, mettre à jour ou supprimer une étape, les concepts et les exemples associés (représentés dans le diagramme d'activité de la figure 4.6 ci-dessous). Il peut en faire de même pour la gestion des comptes des étudiants. En effet, dans AARTIC l'enseignant est lui-même considéré administrateur du système.
De plus, l'étudiant peut avoir un aperçu des performances de la classe et de chaque étudiant suivant les fonctionnalités suivantes :

- Accès aux statistiques relatives au niveau de la classe à chaque étape du laboratoire,
- Accès aux statistiques relatives à la note moyenne de la classe par concept théorique (le concept théorique doit être associé à une question du QCM); grâce à ces données l'enseignant pourra reconnaître les concepts les plus difficiles à assimiler par les étudiants et pourra ainsi modifier la définition de celui-ci et ajouter des exemples plus simples,
- Accès aux statistiques concernant la connexion au site,
- Accès au profil de chacun des étudiants.

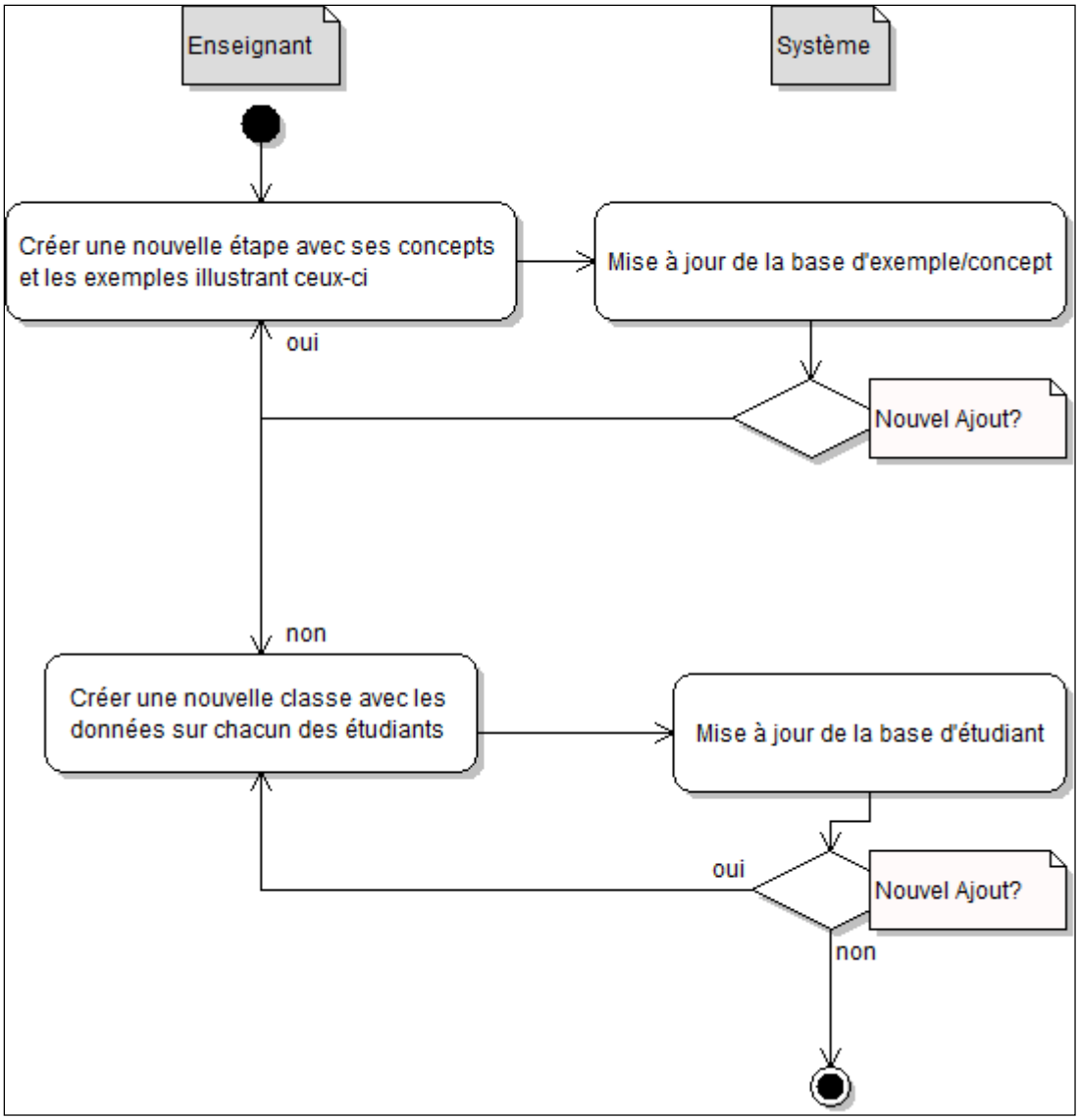

Figure 4.6 Diagramme d'activité lié à la gestion des exemples et des comptes étudiants.

4.4.2 L'environnement AARTIC du côté étudiant

Muni de son nom d'utilisateur et de son mot de passe, chaque étudiant du cours peut se connecter sur le site d'AARTIC. Une fois authentifié, il peut faire les activités suivantes :

- Accès à l'étape courante (à laquelle il est arrivé) et les étapes qui la précédent et voir les concepts associés,
- Accès à chaque concept et à l'exemple proposé par le système (en tenant compte du niveau de connaissance de l'étudiant). Il peut toujours voir les autres exemples associés au même concept,
- Accès au forum de discussion pour proposer une explication d'un exemple, commenter une explication déjà proposée, poser une question,…,
- Accès au test QCM associé à l'étape courante,
- Accès à son profil et à toutes les informations le concernant,
- Accès à la page d'aide pour voir les questions fréquemment posées concernant le fonctionnement du site Web.
-

Dans ce qui suit, les diagrammes d'activité (figure 4.7 et figure 4.8) liés à l'étudiant décrivant le scénario d'utilisation d'AARTIC.

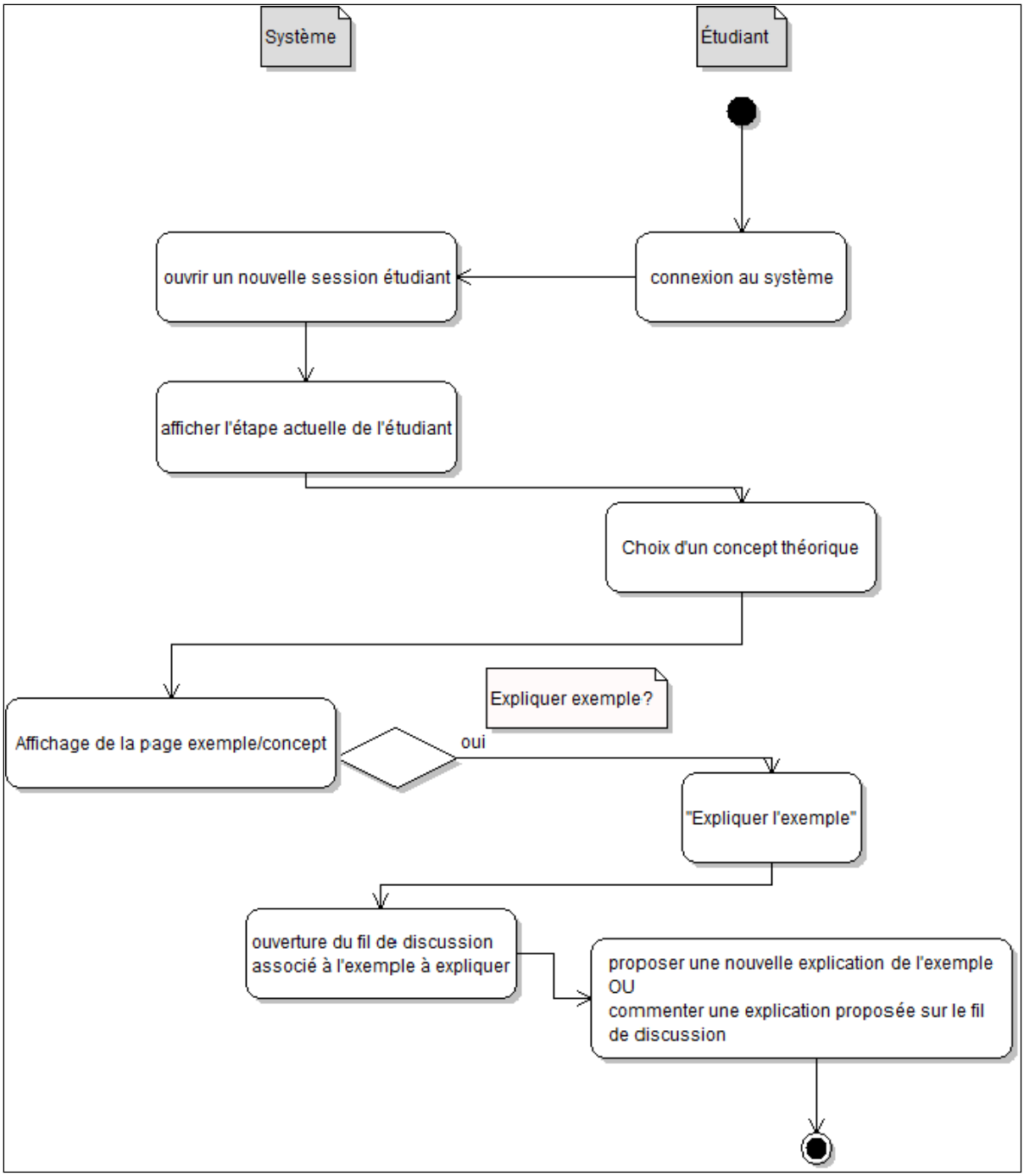

Figure 4.7 Diagramme d'activité de l'exploration d'une étape dans AARTIC.

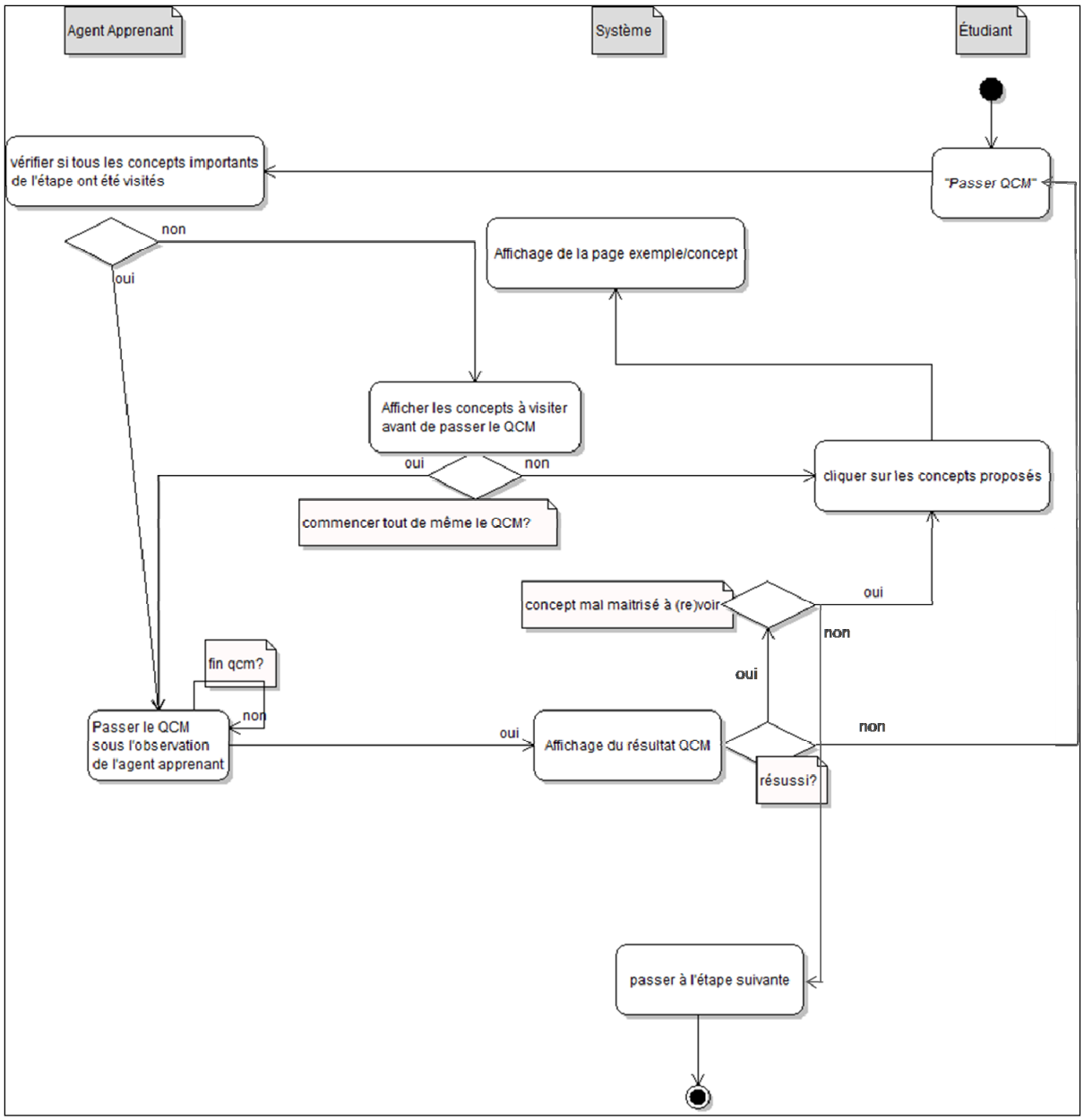

Figure 4.8 Diagramme d'activité du passage d'un QCM dans AARTIC.

CHAPITRE 5

CHOIX TECHNOLOGIQUES

L'objectif de ce chapitre est d'expliquer et de justifier les outils technologiques que nous avons sélectionnés pour la réalisation de l'environnement AARTIC. D'abord, nous présentons le moteur d'inférence JESS choisi pour le développement des agents intelligents. Par la suite, nous expliquons notre choix de la technologie Java concernant le développement du noyau du système. Après quoi, nous discutons notre choix pour le développement du site Web. Nous présentons ensuite l'outil utilisé pour hiérarchiser les concepts théoriques du cours exposés sur le site Web. Enfin, nous justifions notre choix en ce qui concerne le développement des questionnaires à choix multiples. Nous concluons le chapitre en exposant ces différents outils et les relations entre eux pour construire l'environnement informatique d'apprentissage humain AARTIC.

5.1 Développement des agents avec JESS

JESS[6] (Java Expert System Shell) est un "moteur d'inférence et un langage de script" entièrement écrit en Java. Inspiré du logiciel CLIPS développé par la NASA, JESS a vu le jour dans les laboratoires Sandia National en Californie, à la fin des années 90. Depuis presque 20 ans, JESS a été utilisé pour le développement de systèmes experts dans divers domaines : que ce soit en médecine, en cryptographie, en enseignement ou encore pour des jeux. Son moteur d'inférence emploie l'algorithme RETE[7] pour le traitement des règles.

---

[6] http://www.jessrules.com/jess/index.shtml

[7] http://fr.wikipedia.org/wiki/Algorithme_de_Rete

Rapport-Gratuit.com

Sa compatibilité avec Java permet d'établir facilement un lien entre un programme Java et un autre écrit en JESS. En effet, d'un côté on peut manipuler et accéder à des objets Java à partir des règles JESS et de l'autre on peut faire appel à l'expertise d'un agent intelligent à partir d'un programme Java.

5.2 La logique métier : technologie Java

Puisque les technologies Java et JESS sont complémentaires, nous avons choisi d'utiliser la technologie Java. Ainsi, toute la logique métier et toutes les classes nécessaires au fonctionnement de l'environnement AARTIC ont été programmées en Java. Nous avons également profité de la technologie JSP/Servlet proposée par Java pour la gestion de l'affichage des données à l'utilisateur final (l'étudiant).

Java est un langage de programmation orienté objet et multi plateformes qui possède une très large bibliothèque de fonctions. C'est également un langage très apprécié dans le développement Web puisque ses applications sont exécutables sur n'importe quelle machine indépendamment de son système d'exploitation. Aujourd'hui, ses différentes qualités largement reconnues (sa flexibilité, sa portabilité ou encore sa sécurité) font de Java la technologie idéale pour l'informatique de réseau, si bien que jusqu'à présent, elle équipe plus de 4,5 milliards de périphériques (Sun Microsystems).

JSP et Servlet :

Avec les JSP (Java Server Pages) et les servlets nous avons pu respecter une architecture MVC (Modèle-Vue-Contrôleur) pour le développement d'AARTIC et de ce fait rendre notre environnement plus générique et pouvant dès lors s'appliquer à différents cours offerts à l'ÉTS.

5.3 Site Web AARTIC: SPIP versus développement HTML

5.3.1 Présentation SPIP

SPIP est un système de publication pour l'Internet permettant de gérer rapidement et facilement un site Web. Grâce à la simplicité d'installation et d'utilisation, SPIP est à la porté de toute personne, quelles que soient ses connaissances en informatique ou en gestion de site Web et quelle que soit la langue du site Web (jusqu'à aujourd'hui, dans sa version 2.0.9, SPIP est offert en 30 langues différentes). Il peut être employé pour tout site Web quel que soit sa nature : universitaire, institutionnel ou propriétaire. De plus, SPIP est connu pour son « fonctionnement collaboratif » puisque l'édition du site Web peut être effectuée en groupe et ne se limite pas à une seule personne (Perline, 2004). Ce système de publication ne cesse de s'améliorer grâce à son importante communauté de développeurs.

On remarque également une utilisation accrue de SPIP dans un contexte pédagogique (d'ailleurs SPIP supporte le format SCORM) : en France plus de 300 sites éducatifs utilisent la technologie SPIP[8].

Il est vrai que SPIP offre plusieurs services intéressants dans la création et la gestion d'un site Web, notamment dans un contexte d'enseignement. Cependant, il reste encore quelques lacunes et inconvénients à l'utilisation de ce système de gestion de contenu tel qu'on l'a constaté durant notre recherche. C'est ce que nous allons décrire dans la section suivante.

---

[8] http://spip-edu.edres74.net/

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

5.3.2 Inconvénients de SPIP pour AARTIC et choix d'une programmation HTML

Comme nous l'avons décrit précédemment, l'adaptabilité de notre EIAH se situe principalement au niveau des agents intelligents intégrés. Nous avons choisi JESS pour le développement de ces agents. Puisque JESS est compatible avec Java, nous avons développé la couche métier de notre environnement AARTIC en Java. Or SPIP est fondé sur le langage PHP (*Hypertext Preprocessor*), ce qui donne lieu à la question : Est-il possible de faire fonctionner SPIP et Java ensemble?

Il est vrai qu'il existe différentes solutions pour faire fonctionner un programme Java à travers un environnement SPIP. Parmi celles-ci, deux solutions se distinguent:

- Un applet Java peut fonctionner dans un site SPIP (Fareh, 2008) : Pour notre cas, cette solution n'est pas envisageable puisque toutes les règles de gestion sont dans des packages Java et qu'on ne peut se limiter à un simple applet,
- Utiliser un pont Php/Java, comme celui proposé par la Zend Platform[9] : cette solution risque de prendre beaucoup de temps de développement (or on est limité par le temps, si on veut faire une évaluation de notre environnement pas les étudiants du cours LOG625). De plus, ce genre de passerelles entre php et java est assez reconnu pour leur manque de fiabilité et de stabilité.

Finalement, et après une longue réflexion, nous avons décidé de développer la première version de notre environnement sans avoir recours à SPIP. Nous avons préféré développer la suite de notre site Web AARTIC en HTML tout en utilisant la technologie Servlet/JSP et donc d'utiliser le serveur Web Tomcat d'Apache.

---

[9] http://files.zend.com/help/Zend-Studio-Eclipse/java_bridge.htm

5.4 Représentation des concepts théoriques du cours avec Freemind

Pour représenter les concepts théoriques du cours à chaque étape, nous avons utilisé les cartes conceptuelles pour hiérarchiser et organiser ceux-ci. Nous avons utilisé dans cette carte des codes de couleur et des styles de police (gras, italique) pour déterminer les concepts visités par l'étudiant et identifier les concepts théoriques les plus importants de l'étape courante.

Une carte conceptuelle est un graphique illustrant des idées, des connaissances ou des concepts et les liens sémantiques entre ces éléments sous forme de réseaux ou d'arborescences (Drissi, Talbi et Kabbaj, 2006). En enseignement, les cartes conceptuelles facilitent et améliorent l'apprentissage : elles représentent un outil d'organisation des connaissances, de travail collaboratif, mais aussi elles sont un moyen de navigation dans des dispositifs hypertextes ou encore d'évaluation.

Nous avons donc choisi le logiciel FreeMind[10] pour la création de la carte conceptuelle représentant les concepts théoriques du cours dans chaque étape et les liens entre eux. Il s'agit d'un logiciel libre écrit en java. L'un des avantages de FreeMind est qu'il permet d'exporter la carte conceptuelle réalisée sous format HTML. De plus, les éléments de la carte peuvent être représentés sous formes d'hyperliens vers des pages Web ou des fichiers.

---

[10] http://freemind.sourceforge.net/wiki/index.php/Main_Page

5.5 Développement des questionnaires à choix multiple

Un QCM ou questionnaire à choix multiples est un questionnaire pour lequel on fournit, pour chaque question posée, différentes propositions de réponse dont une ou certaines de ces propositions constituent les réponses valides. Les questionnaires à choix multiples représentent une partie importante de notre environnement puisque le passage de l'étudiant d'une étape à l'autre dépend du résultat obtenu à ces questionnaires. De plus, ce résultat nous permet de calculer le niveau de connaissance des étudiants. Grâce à quoi, l'agent est en mesure de mieux guider l'étudiant durant son apprentissage, le système pouvant ainsi adapter le contenu présenté à cet étudiant selon son profil.

Étant donné l'importance des QCM, le choix d'un outil approprié pour les générer s'est avéré difficile. En effet, aujourd'hui on trouve plusieurs logiciels permettant de mettre en ligne des QCM : tels que JQuiz de Hotpotatoes[11] ou Prométhée[12] ou encore Questionmark[13]. L'inconvénient avec ce genre d'outils est que même si certains d'entre eux sont performants, gratuits et offres des fonctionnalités intéressantes (comme JQuiz avec Hotpotatoes), ils ne permettent pas d'avoir un suivi détaillé de chaque réponses des étudiants et ils ne permettent pas non plus l'enregistrement des résultats des étudiants. Les seuls outils de création de QCM (comme Questionmark) proposant un rapport détaillé sur la performance de l'étudiant qu'on ait trouvé sont payants.

---

[11] http://hotpot.uvic.ca/

[12] http://promethee.eu.org/index.php

[13] http://www.questionmark.com/fra/

La solution que nous avons choisie a été d'implémenter un module QCM en Java : composé principalement de deux classes Java, une pour les questions et les concepts théoriques qui s'y rapportent et une autre pour les propositions de réponses et leur validité. De cette manière, il est possible d'enregistrer les résultats des QCM directement sur la BD et l'agent apprenant peut collaborer avec ce module pour faire un suivi de la performance de cet apprenant.

En conclusion, AARTIC est un environnement Web de type EIAH développé en Java. Pour gérer l'interface utilisateur nous avons préféré utiliser la technologie JSP/Servlet plutôt que système gestionnaire de contenu SPIP. Pour cela, l'environnement nécessite le serveur Web Tomcat (version 5.5) pour gérer les JSP et les servlet.

CHAPITRE 6

RÉALISATION

Ce chapitre décrit les résultats de l'évaluation de l'environnement AARTIC. Premièrement, nous montrons l'intérêt d'intégrer un tel environnement grâce à la pré-évaluation que nous avons effectuée. Par la suite, nous dévoilons les résultats de l'évaluation de l'environnement par les étudiants du cours LOG625. Nous nous intéressons à la troisième section à l'estimation de l'effort que l'enseignant doit fournir pour préparer le contenu de notre environnement AARTIC. Enfin, nous synthétisons les impacts estimés et attendus d'un environnement d'apprentissage comme le nôtre.

6.1 Pré-test

Afin d'évaluer l'importance d'une éventuelle intégration d'un système d'aide à l'apprentissage dans ce cours, nous avons présenté le 25 Juin 2008 (à la session d'été), un questionnaire (rempli de manière anonyme) à toute la classe du cours LOG625 « Introduction aux systèmes intelligents » (Questionnaire 1 se trouve à l'annexe A). Nous avons expliqué en détail le but de la recherche et l'objectif du questionnaire à tous les étudiants avant de le distribuer.

Le questionnaire nous a permis d'évaluer la complexité du premier laboratoire à réaliser dans le cadre de ce cours et de valider l'intérêt de l'apprentissage par l'exemple. Voici les résultats que nous avons obtenus. Premièrement, nous avons réussi à évaluer le degré de complexité du premier laboratoire du cours LOG625 selon les étudiants de ce cours. Plus de 55% des étudiants trouvent que le projet est « complexe ». D'ailleurs, presque 2 étudiants sur 5 déclarent avoir passé de 20 à 50 heures de travail **individuel** alors que laboratoire se déroule en groupe de 2 à 4 étudiants et que le temps estimé pour le travail ne dépasserait pas 40 heures pour **tout le groupe**.

En ce qui concerne les exemples, la majorité des étudiants (74%) est d'accord pour dire que les exemples présentés sont très « utiles » pour la réalisation du laboratoire et que plus d'exemples aiderait davantage (84% des étudiants).

De plus, ce questionnaire nous a permis de constater le manque de participation des étudiants aux forums de discussion associés à leurs cours (74% des étudiants déclarent ne jamais participer à ces forums). Enfin, l'une des quelques propositions supplémentaires que nous avons reçues des étudiants est le besoin de « préciser les objectifs de chaque séance de laboratoire ».

6.2 Test de l'application AARTIC

6.2.1 Réalisation

Au final, nous avons pu développer une première version d'AARTIC. Cette version manque encore certaines fonctionnalités (particulièrement celles liées à la partie enseignant qui est en cours de réalisation). Mais nous avons, au final, pu produire un site ergonomique et fonctionnel. Les figures 6.1, 6.2 et 6.3 illustrent le résultat.

1. Aperçu de la page d'accueil = page de l'étape courante:

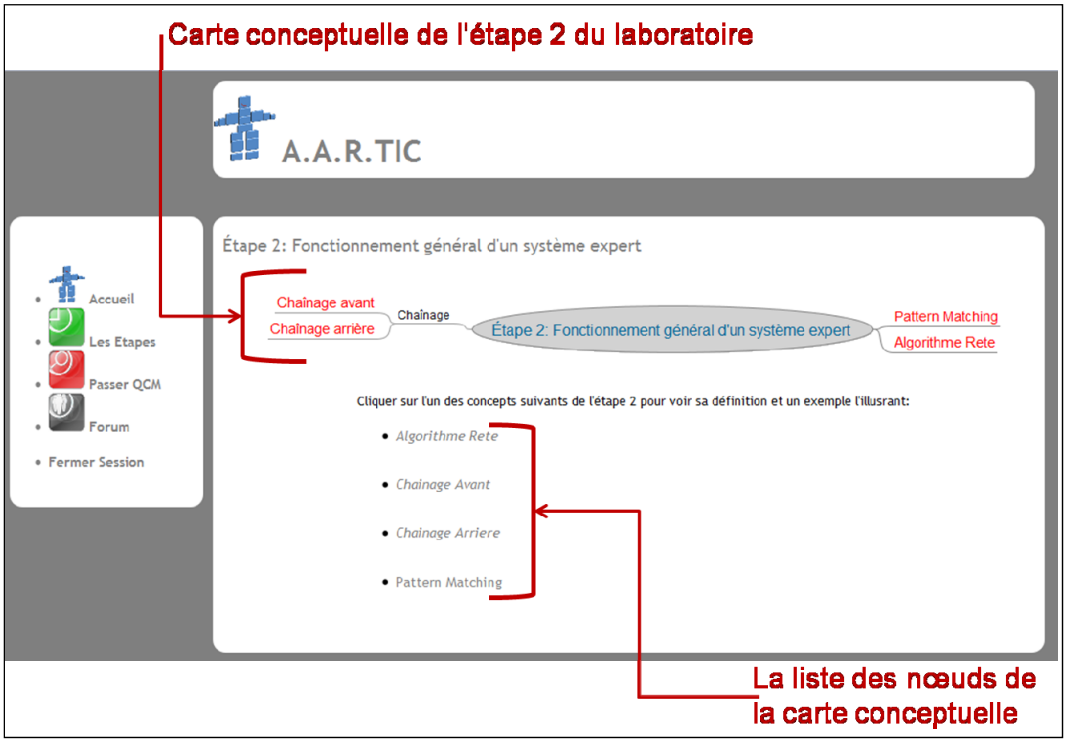

Figure 6.1 Interface d'accueil du système AARTIC.

Ci-dessus, l'interface d'accueil d'AARTIC. Elle est constituée de deux barres de navigations (celle du haut et celle de droite) qui sont fixes pour toutes les pages du site permettant de parcourir ses différentes rubriques. La section du milieu est celle qui varie d'une page à l'autre. La page d'accueil représente l'étape actuelle à laquelle l'étudiant est arrivé. La figure 6.1 présente le cas d'un étudiant arrivé à l'étape 2 du laboratoire. Les différents concepts théorique de l'étape sont représentés à travers la carte conceptuelle crée avec FreeMind. Le seul inconvénient avec les cartes FreeMind est qu'elles ne sont pas cliquables. Cet outil permet, cependant, de générer les différents nœuds de la carte, organisés de manière hiérarchique, sous forme de liste cliquable (d'où la liste des concepts théoriques qu'on voit à l'interface d'accueil).

2. Aperçu de la page QCM :

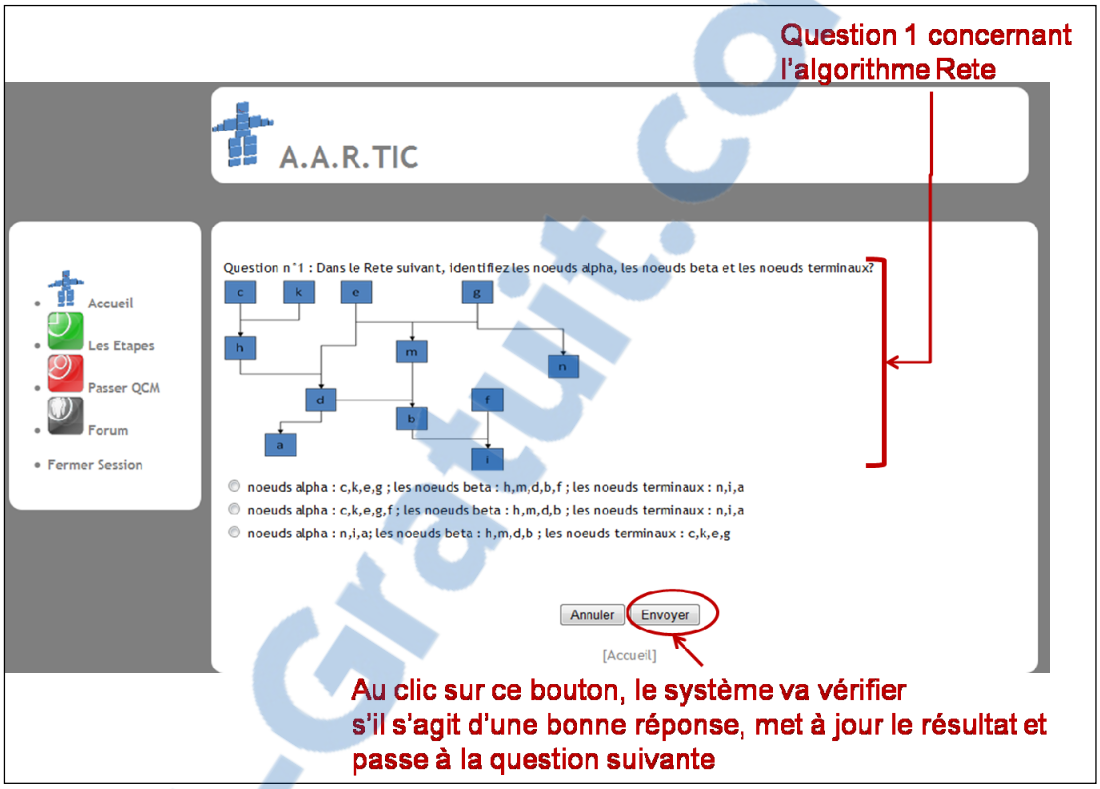

Figure 6.2 La page QCM du site Web AARTIC.

Au moment où l'étudiant clique sur le lien « Passer QCM » (en haut à droite de la figure 6.2), l'agent apprenant vérifie si l'étudiant a bien visité tous les concepts importants de l'étape courante. Si l'un des concepts n'a pas encore été exploré par l'étudiant, l'agent apprenant lui suggère de le voir avant de passer le QCM. À ce moment, l'étudiant peut choisir de voir la définition et l'exemple du concept proposé ou de passer le QCM.

La figure 6.2 illustre une question proposée lors du QCM. Dans le cas présenté, la question est accompagnée d'une figure et de trois propositions de réponses.

Lorsque l'étudiant a choisi une réponse et qu'il a cliqué sur le bouton « Envoyer », le système traite celle-ci et passe à la question suivante jusqu'à la fin du questionnaire.

Rapport-Gratuit.com

3. Aperçu de l'affichage des résultats du QCM avec les recommandations de l'agent apprenant :

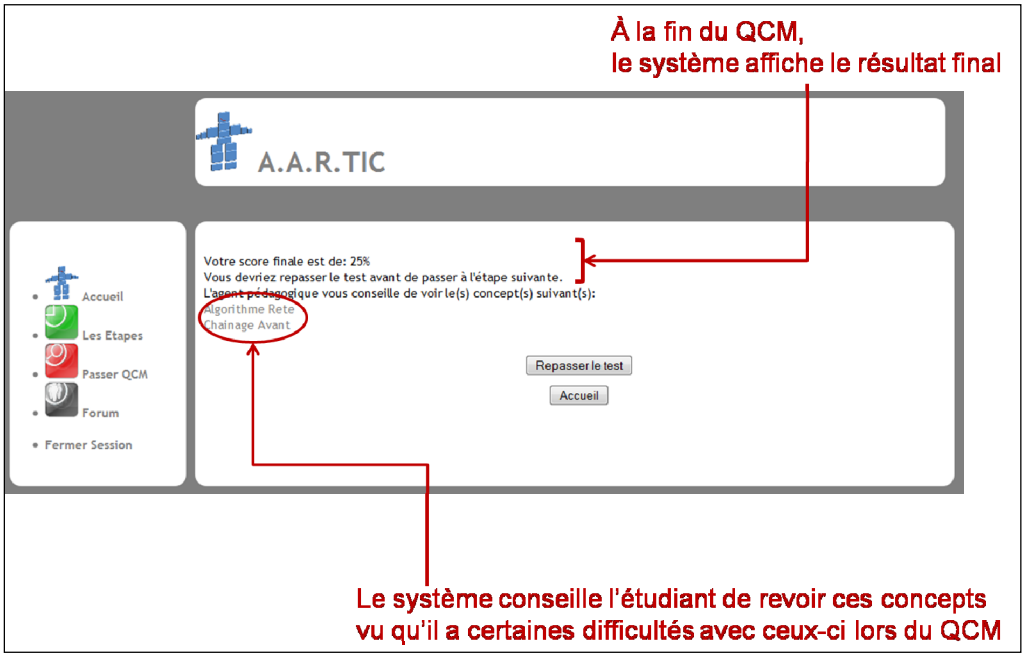

Figure 6.3 Affichage des résultats du QCM.

La figure 6.3 montre la fin d'un QCM et l'affichage du résultat final. Ici l'étudiant n'a pas réussi le questionnaire (sa note est inférieure à 50%). Le système ne lui permet pas de passer à l'étape suivante. De plus, l'agent apprenant a noté certaines difficultés rencontrées par l'étudiant lors du QCM par rapport aux concepts théoriques « Algorithme Rete » et « Chaînage Avant », l'agent va donc lui recommander de bien assimiler les concepts suivants (l'étudiant peut soit revoir ces concepts et les exemples reliés sur le site, en cliquant directement sur les concepts proposés soit demander de l'aide à travers le forum de discussion, en allant à la rubrique forum en bas à droite).

Si l'étudiant réussit son évaluation (il obtient une note supérieure à 50%), un bouton « Passer à l'étape suivante » s'affiche, lui permettant ainsi de passer à l'étape suivante. Si toutefois l'agent apprenant remarque certaines difficultés avec un ou plusieurs concepts théoriques, même si l'étudiant a réussi son QCM, l'agent va lui proposer de visiter ces concepts. L'étudiant pourra toujours passer à l'étape suivante en cliquant sur le bouton éponyme ou en cliquant sur le lien « Accueil ».

6.2.2 Évaluation de AARTIC

Nous avons testé l'environnement AARTIC et nous l'avons fait évaluer par les étudiants du cours LOG625 offert à la session d'été 2009. Les étudiants sont à la moitié de leur projet de laboratoire (ils déposent leurs travaux dans trois semaines). Ils ont donc atteint l'étape 2 du réseau d'évènement d'apprentissage, « l'apprivoisement du fonctionnement général d'un système expert » (voir le chapitre 3). Ils ont pu profiter de la première version de l'environnement AARTIC durant la séance de laboratoire d'une durée de 3 heures (sans forums de discussion, ni activité de groupe).

Suite à quelques difficultés rencontrées avec l'équipe technique de l'ÉTS pour héberger notre travail sur le serveur du département génie logiciel et technologie de l'information, nous avons décidé de tester AARTIC en permettant aux étudiants de se connecter à travers le réseau local sur l'un des postes de travail où AARTIC a été installé.

Le déroulement de l'expérience se passe comme suit. Tout d'abord, et après authentification sur le site d'AARTIC, l'étudiant passe un premier questionnaire permettant ainsi au système d'évaluer le niveau de l'étudiant. Une fois le test terminé, l'étudiant peut prendre tout le temps pour explorer l'étape 2, les concepts et les exemples associés. À la fin, l'étudiant peut passer le deuxième questionnaire afin de vérifier si l'étudiant a bien assimilé les concepts de

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

l'étape qu'il vient de voir. À la fin de cet exercice, nous avons demandé à tous les étudiants de bien vouloir répondre au questionnaire d'évaluation d'AARTIC (Annexe B : questionnaire 2), voici les résultats que nous avons obtenus.

De manière qualitative, les résultats sont encourageants. Concernant l'apparence générale du site, la majorité (80%) des étudiants estime que l'aspect est bon à excellent. En une seule séance de laboratoire, 60% des étudiants trouvent que l'apport de AARTIC à la réalisation de leurs laboratoires et à la compréhension des concepts théoriques est moyennement important à important (voir Figure 6.4). De plus, tous estiment que les exemples présentés dans AARTIC sont pertinents. Enfin, pour ce qui est du besoin de fonctionnalités supplémentaires sur AARTIC, 8 étudiants sur 9 estiment qu'il n'y a pas besoin d'autres services ou informations à ajouter.

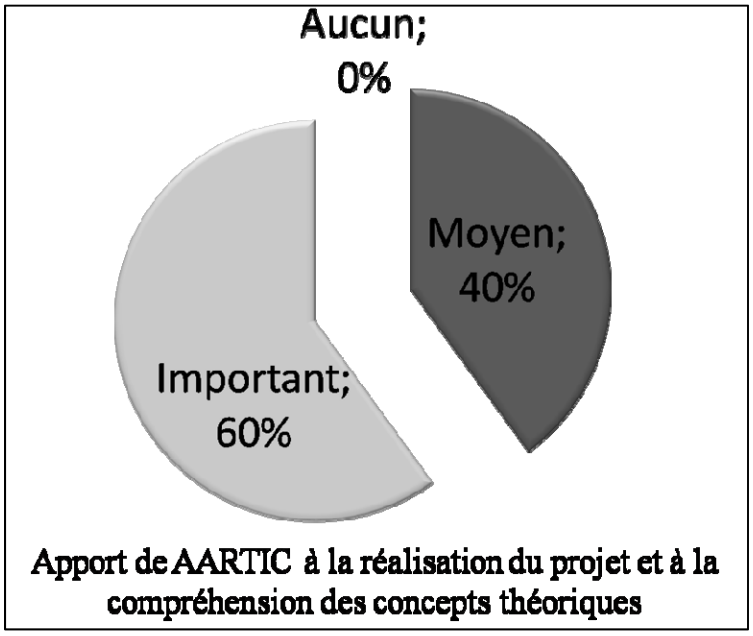

Figure 6.4 Évaluation de AARTIC par les étudiants du cours LOG625.

Il est clair que d'après notre propre conception de l'environnement, l'étudiant devrait avoir tout le temps qu'il veut pour explorer chaque étape. Dans un scénario « idéal » d'évaluation de l'environnement AARTIC, l'étudiant aurait pu expliquer quelques exemples présentés dans le système et collaborer davantage avec les autres membres de sa classe. Seulement, et pour des raisons qui nous dépassent (limites techniques et limites de temps), nous avons dû tester l'environnement dans une limite de temps très courte (seulement 3 heures) et sans pouvoir profiter du forum de discussion (et donc aucune possibilité d'expliquer un exemple).

6.3 Effort de l'enseignant

Pour utiliser le système, l'enseignant doit préparer les éléments à inclure dans le système tels que les concepts théoriques à présenter, leur définition et les exemples classés par niveau. D'abord, il nous faut préciser que nous avons couvert uniquement les étapes 1 et 2 du réseau des évènements d'apprentissage (voir chapitre 3 MISA), ce qui correspond aux deux étapes les plus chargées en concepts théoriques. Nous avons travaillé de la manière suivante.

Les concepts et définitions :

Munis du contenu de chaque séance de cours présenté en LOG625, nous avons commencé par sélectionner les concepts les plus pertinents pour la réalisation du laboratoire et les définitions provenant du contenu du cours. Ensuite, et avec l'aide du professeur chargé du cours, nous avons classé ces concepts à l'étape 1 et 2 (selon aussi à quel moment du cours ces concepts ont été présentés) : cette phase a nécessité de 3 à 4 heures pour le parcours du contenus du cours présenté (en rapport aux deux étapes du REA, nous nous sommes limités aux 4 séances de cours données). Il a fallu une heure de plus pour pouvoir valider ces informations avec le chargé du cours.

Les exemples :

Nous avons sélectionné les exemples du contenus du cours LOG625 et les avons associés aux concepts concernés : 2 à 3 journées de travail avec la validation de l'enseignant.

À chaque exemple, nous avons associé trois paramètres :

- Le concept théorique auquel il est associé,
- Le niveau de l'exemple : s'il s'agit d'un exemple de niveau 1, 2 ou 3 soit un niveau débutant, intermédiaire ou avancé,
- La priorité de l'exemple pour illustrer le concept : un concept est illustré par plusieurs exemples de différents niveaux. Pour chaque couple concept/niveau on peut avoir plusieurs exemples : le système détermine lequel de ces exemples est le plus intéressants à afficher en premier lieu pour l'étudiant grâce à la valeur de la variable « *prioriteExemple* ».

Le plus difficile à faire était de trouver au moins 3 exemples de niveaux différents pour associer à chacun des concepts du cours que nous avions sélectionnés (au minimum, un exemple par niveau : débutant, avancé et intermédiaire). Pour cela, il a fallu chercher dans d'autres livres connus (Giarratano et Riley, 1989) en intelligence artificielle pour ensuite les adapter.

Au total, nous avons assemblé plus de 24 concepts différents pour l'étape 1 et 5 concepts pertinents pour l'étape 2.

L'organisation des concepts :

La troisième tâche consistait à rassembler les concepts pour chaque étape et de les ordonner dans une carte conceptuelle de type FreeMind. Ce travail a nécessité environ 2h pour chaque carte avec la validation du chargé de cours.

Voici en exemple, la carte conceptuelle que nous avons réalisée pour l'étape 1 :

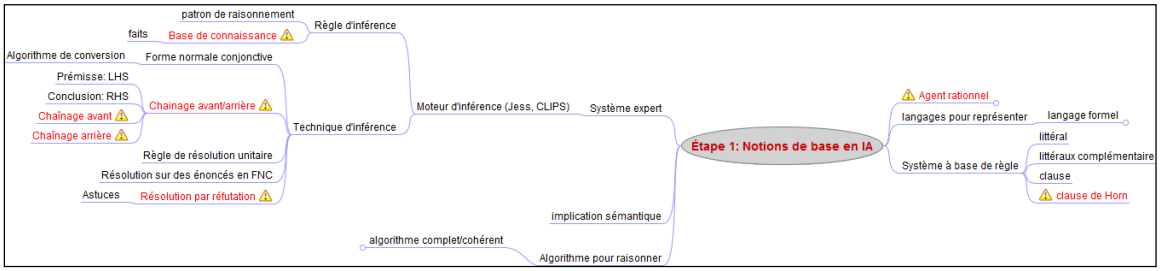

Figure 6.5 Carte conceptuelle de la première étape du laboratoire « Notions de base en IA ».

À noter que ce qui est en rouge avec des points d'exclamation représente les concepts les plus essentiels à l'étape qu'on qualifie d'« importants ».

<u>Préparation des QCM :</u>

À un stade initial, nous avons sélectionné uniquement les concepts importants. Nous avons essayé de trouver des exercices ou des exemples concernant 1 ou deux de ces concepts et nous avons essayé de reformuler ces données en une question à choix multiples.

<u>Insertion dans la base de données :</u>

Puisque la partie enseignant n'a pas encore été entièrement développée, nous avons dû faire entrer toutes les données directement sur la base de données MySQL gérant le contenu du cours. Cette tâche a été facilitée grâce à l'interface graphique intuitive proposée par le gestionnaire de base de données *MySQL administrator*[14].

---

[14] http://dev.mysql.com/downloads/gui-tools/5.0.html

6.4 Impact sur l'apprentissage

Comme le déclare Tchounikine (2002b), l'évaluation de l'apprentissage n'est pas une tâche facile, cela dépend d'un certain nombre de variables qu'on ne peut contrôler (comme par exemple la psychologie de l'apprenant). Seulement, il existe un bon nombre d'études qui prouvent que peu importe les variables, l'intégration d'un EIAH peut représenter une amélioration de l'apprentissage et la manière de présenter la matière du cours.

Les effets liés aux environnements informatiques d'apprentissage humain ont été précédemment discutés dans le premier chapitre de ce mémoire. En effet, plusieurs expériences menées au sujet de l'intégration des EIAH en enseignement, ont eu un réel succès et ont prouvé l'impact positif que peuvent apporter ces environnements informatiques sur l'apprentissage s'ils sont utilisés convenablement. Citons, à titre d'exemple, l'environnement Sphinx (Capus et al., 2006) où les notes d'évaluation des étudiants se sont considérablement améliorées une fois l'environnement intégré au cours. D'ailleurs, le succès de Sphinx a été tel que le département d'informatique de l'université Laval a souhaité intégrer la même « formule » dans d'autres cours (Capus, Potvin et Tourigny, 2002).

De notre côté, nous prévoyons obtenir les mêmes résultats avec l'utilisation de AARTIC et avoir un impact positif sur l'apprentissage et le développement des habilités cognitives des étudiants. Parmi les facteurs du développement cognitif, on trouve l'interaction avec son environnement physique et l'interaction avec son environnement social (travail de groupe, forum de discussion,…). AARTIC propose différents outils liés à ces facteurs qui permettent de développer ces habilités tels que des outils de :

- navigation : grâce au site Web développé et à la carte conceptuelle représentant les liens entre les différents concepts théoriques d'une étape,
- autogestion : l'étudiant est capable d'évaluer son propre apprentissage à travers les différents questionnaires qu'on lui propose. À travers son profil, l'étudiant peut voir les progrès qu'il fait tout au long de son utilisation du système,

- production, communication et collaboration : les forums de discussions intégrés à l'environnement représentent non seulement un outil d'écriture mais aussi de partage et de coopération,
- information : notre système présente les données du cours grâce à une base de données alimentées des différentes définitions des concepts théoriques du cours et des exemples illustrant ces concepts. De plus, l'agent apprenant joue un rôle informatif important grâce aux conseils et données retournées aux étudiants.

Grâce aux différents outils proposés par l'environnement AARTIC, nous pouvons déduire qu'un EIAH comme celui que nous présentons a un impact positif sur le développement cognitif de l'étudiant. On peut donc s'assurer de l'apport positif lié à l'utilisation de AARTIC dans l'assimilation des concepts théoriques en général et dans la réalisation des laboratoires en génie en particulier.

Ce chapitre a permis de présenter les besoins exprimés par les étudiants dans le cours LOG625 « Introduction aux systèmes intelligents». Nous avons tenu compte de ces besoins dans la réalisation de notre projet. Pour finir, nous avons pu présenter le résultat final (l'environnement Web AARTIC) aux étudiants du cours LOG625 qui l'ont testé et évalué. En conclusion, les résultats obtenus sont très encourageants. Les étudiants ont apprécié la présentation des concepts théoriques illustrés par des exemples adaptés à leur niveau. Ils ont également adopté facilement le fonctionnement du système et son organisation. De plus la majorité des étudiants a bien accueilli l'aspect visuel de l'environnement AARTIC. De ce fait, nous pouvons qualifier notre environnement d'ergonomique, une caractéristique importante dans l'évaluation d'environnement Web. Bien que nous n'avons pas pu évaluer AARTIC pendant une session de cours entière, nous restons optimiste quand à son apport à l'étudiant dans la compréhension des concepts du cours et dans l'aide à la réalisation des laboratoires grâce à son aspect adaptatif et aux différents outils que l'environnement propose.

Enfin, il reste certaines améliorations et fonctionnalités à ajouter. Nous sommes actuellement en train de développer la partie enseignant pour gérer et afficher les statistiques et les différentes données concernant la classe. L'agent apprenant pourrait être encore plus performant notamment lors de l'évaluation des connaissances de l'étudiant et de son encadrement. De plus, l'augmentation de la quantité de contenu serait plus appréciée, l'environnement pourrait alors proposer plus d'exemples et ce pour tous les niveaux de connaissance (débutant, intermédiaire, avancé).

CONCLUSION

L'environnement AARTIC est un environnement intelligent d'apprentissage humain (EIAH) permettant de résoudre les problèmes rencontrés dans les cours d'ingénierie à contenus complexe. Grâce à une base importante d'exemples, AARTIC aide l'étudiant dans l'assimilation des concepts du cours à travers des exemples proposés selon le niveau de cet étudiant. L'aspect adaptatif du système est assuré par deux agents pédagogiques: l'un, associé à l'étudiant, fait le suivi des différentes activités de celui-ci. L'autre, associé à l'enseignant, permet un suivi personnalisé de chacun des étudiants et de la classe en entier. Notre environnement met aussi l'accent sur la collaboration entre les apprenants et entre les apprenants et l'enseignant grâce à un forum de discussion. Toutes les composantes de l'environnement AARTIC permettent de moderniser et de dynamiser l'apprentissage et de stimuler développement cognitif chez l'apprenant.

AARTIC a été conçu grâce à la méthode MISA pour cerner les besoins et les objectifs d'apprentissage et au langage UML associé à MISA pour l'identification et la conception des composants de l'environnement.

AARTIC est un environnement Web qui permet aux étudiants de travailler quand ils le souhaitent, où ils le souhaitent et selon leur propre rythme. Dans un premier temps, nous avons limité le contexte d'utilisation d'AARTIC au cours de baccalauréat LOG625 « Introduction aux systèmes intelligents » qui se donne à l'ÉTS. Dans ce cours, les étudiants sont amenés à réaliser un projet de laboratoire qui consiste à développer un système expert capable de jouer au Clue. L'environnement AARTIC assiste les étudiants dans la réalisation de ce projet.

Rapport-Gratuit.com

Durant l'utilisation d'AARTIC, l'étudiant est amené à passer par différentes étapes guidé par le système. Une étape représente une phase du projet. À chaque étape, un ensemble de concepts est présenté : le système propose à l'étudiant l'exemple illustrant un concept donné compte tenu de son niveau de connaissance. L'étudiant peut, par la suite, proposer une explication de l'exemple sur un fil de discussion dédié du forum. Pour aller d'une étape à la suivante, l'étudiant doit passer un questionnaire. Ceci permet d'évaluer l'étudiant et de mettre à jour son niveau de connaissance.

Nous avons pu tester l'environnement AARTIC lors d'une séance de laboratoire du cours LOG625. Les réactions des étudiants par rapport à l'environnement sont très encourageantes. Les diverses caractéristiques telles que la représentation des concepts théoriques illustrés par différents exemples ou encore le fonctionnement général du système AARTIC ont été très appréciées.

Bien que nous n'avons pas réussi à évaluer l'impact de l'utilisation d'AARTIC sur l'apprentissage durant une session entière, nous sommes optimiste quand à son apport dans l'orientation et l'aide à la compréhension du cours. Également, l'un des avantages d'AARTIC est qu'il permet de « forcer » l'enseignant à associer des exemples aux concepts théoriques du cours. Ceci contribuera à l'amélioration de l'enseignement et incite à la motivation des étudiants. De plus, l'utilisation d'un EIAH comme AARTIC contribuera à guider les étudiants dans la réalisation des laboratoires tout en s'assurant qu'ils maîtrisent tous les concepts théoriques du cours. D'un point de vue pédagogique, AARTIC permet également de faciliter la tâche de suivi personnalisé à l'enseignant.

La version actuelle d'AARTIC est fonctionnelle et peut être intégrée à n'importe quel cours d'ingénierie nécessitant ce type d'environnement d'aide à l'apprentissage. Cela dit, certaines améliorations et fonctionnalités sont à ajouter. Parmi ces améliorations, il y a le développement de la partie enseignant et l'amélioration des performances de l'agent apprenant en lui permettant d'étudier d'autres informations à considérer provenant du profil de l'étudiant telles que la méthode d'apprentissage de l'étudiant ou encore le temps passé à la lecture d'un exemple.

Il serait possible d'améliorer le déroulement des questionnaires à choix multiples proposés dans le système. En effet, nous pourrions associer à chaque question un degré de difficulté et une limite de temps maximale pour répondre à chaque question afin d'estimer la difficulté qu'éprouve un étudiant pour répondre à une question proposée. À long terme, nous voulons profiter de la taxonomie SOLO « Structure of Observed Learning Outcome » (Biggs et Collis, 1982) pour évaluer l'étudiant. Cette taxonomie décrit les niveaux de complexité dans la compréhension d'un sujet et permet une meilleure analyse et évaluation des connaissances d'un apprenant.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

ANNEXE I

QUESTIONNAIRE PRÉ-TEST

Nous développons actuellement un système d'aide à l'apprentissage qui assistera les étudiants du cours LOG625 dans la réalisation de leur projet de laboratoire. Et dans le but d'évaluer l'apport du futur système, nous avons besoin de votre aide.

Comme vous êtes étudiant du cours LOG625- Introduction aux systèmes intelligents à l'été 2008, vous avez été amené à réaliser un projet de laboratoire visant à développer un agent rationnel capable de jouer au jeu Cluedo. Nous aimerons avoir votre avis concernant la réalisation de ce projet.

A- Réalisation du projet :

A.1. Combien d'heures avez-vous passé dans la réalisation de ce projet?

☐	Moins de 20h
☐	Entre 20h et 50h
☐	Plus de 50h

A.2. Sur une échelle de 0 à 5, comment évaluerez-vous la complexité de ce projet de laboratoire?

0 : ne s'applique pas ; 1 : très facile à réaliser ; 5 : très complexe

0	1	2	3	4	5
☐	☐	☐	☐	☐	☐

A.3. Sur une échelle de 0 à 5, trouvez-vous que les exemples donnés dans le cours vous ont aidé dans la réalisation de votre projet de laboratoire?

0 : ne s'applique pas ; 1 : aucune aide; 5 : très utiles

0	1	2	3	4	5
☐	☐	☐	☐	☐	☐

B- Concepts théoriques :

B.1. Sur une échelle de 0 à 5, pensez-vous que les exemples donnés dans le cours ont été suffisants pour assimiler les concepts vus dans le cours?

0 : ne s'applique par, 1 : exemples insuffisants, 5 : exemples suffisants

0	1	2	3	4	5
☐	☐	☐	☐	☐	☐

B.2. Comment faites-vous pour bien comprendre les exemples de code en JESS proposés dans le cours LOG625?

☐	J'essaie de comprendre le fonctionnement général, sans trop m'attarder à chaque ligne de code
☐	J'analyse chaque règle du code, sa prémisse (LHS- Left Hand Side) et sa conclusion (RHS- Right Hand Side)
☐	Je teste le code, pour voir les faits initiaux et les changements dans la base des faits
☐	Autres, précisez : _____ _____

B.3. Pensez-vous que plus d'exemples aurait pu vous aider davantage dans la compréhension de certains concepts?

☐	oui
☐	non
☐	Je ne sais pas

C-Forums de discussions :

C.1. Est-ce que vous participez, généralement, dans les forums de discussions associés à certains cours?

☐	Oui
☐	Non

C.2. Si oui, à quelle fréquence visitez-vous ces forums?

☐	Tous les jours
☐	Une fois par semaine
☐	Rarement

Commentaires :

Si vous avez d'autres commentaires ou suggestion qui pourront vous aider dans la réalisation de projet de ce genre, il nous fera plaisir d'en tenir compte :

_____

_____

_____

Merci de votre collaboration
Votre aide est grandement appréciée

****

ANNEXE II

QUESTIONNAIRE D'ÉVALUATION DE L'ENVIRONNEMENT AARTIC

Durant la première partie de la session d'été 2009 et dans le cadre du cours LOG625 – Introduction aux systèmes intelligents, vous avez été amenés à réaliser un projet de laboratoire visant à développer un agent rationnel capable de jouer au jeu Cluedo. Durant ce projet, vous avez pu tester notre système d'aide à l'apprentissage « AARTIC ». On souhaiterait avoir votre avis concernant ce système.

A- Aspect général du système :

A.1.Sur une échelle de 0 à 5 (0 : mauvaise ; 5 : excellente), comment trouvez-vous l'apparence générale du système?

0	1	2	3	4	5
☐	☐	☐	☐	☐	☐

A.2. Sur une échelle de 0 à 5, à combien estimez-vous l'apport de ce système par rapport à la réalisation de votre projet et à la compréhension des concepts théoriques en général?

0 : ne s'applique par, 1 : aucun apport, 5 : apport très important

0	1	2	3	4	5
☐	☐	☐	☐	☐	☐

B- Évaluation des exemples présentés :

B.1. Sur une échelle de 0 à 5, à combien évaluez-vous la pertinence des exemples présentés dans le cours (s'ils illustrent réellement les concepts liés)?

0 : ne s'applique par, 1 : peu pertinent, 5 : très pertinent

0	1	2	3	4	5
☐	☐	☐	☐	☐	☐

B.2. Pensez-vous qu'il faudrait ajouter plus d'exemples pour illustrer les concepts du cours?

☐	Oui
☐	Non

Si oui lesquels (citez les concepts qui manquent d'exemples):

_____

_____

B.3. Trouvez-vous que d'autres services ou informations sont manquants et devraient être ajoutés sur le site?

☐	oui
☐	non

Si oui, lesquels :

_____

_____

Commentaires :

Si vous avez d'autres commentaires ou suggestions, il nous fera plaisir d'en tenir compte :

Merci de votre collaboration

****

ANNEXE III

JAVA ET JESS

Ci-dessous le code source de la classe : AgentApprenantEngine. Il s'agit de la classe qui « communique » avec le fichier AgentApprenant.clp pour l'exécution des règles de l'agent apprenant dans l'environnement AARTIC. Voici un scénario général de la fonctionnalité de cette classe :

1/ Remise de l'engin (l'objet *engine*) à son état initial grâce à la fonction *reset*,
2/ Récupération des données avec la méthode *LoadDonnees,*
3/ Exécution des règles stockées dans le fichier AgentApprenant.clp avec la méthode *run*.
La classe présentée ci-dessous est adapté d'un exemple présenté dans la documentation officielle de JESS sur son site officiel (Friedman-Hill, 2007).

```
/*************************************************************
Projet : AARTIC
Nom : AgentApprenantEngine.java
Description : la classe AgentApprenantEngine permet d'utiliser le JESS à partir du code Java
CODE EMPRUNTÉ: la classe AgentApprenantEngine est inspirée de l'exemple donné dans la documentation de JESS (http://herzberg.ca.sandia.gov/docs/70/embedding.html)
Date Création: 01/12/2008
Dernière Mise à jour: 16/04/2009
*************************************************************/
import java.util.Iterator;
import jess.*;
import modele.*;
public class AgentApprenantEngine {
   //Les attributs de la classe
    private Rete engine;
    private WorkingMemoryMarker marker;
  //les méthodes
    public AgentApprenantEngine(ActionEtudiant actionetudiant) throws JessException {
        // Créer un moteur d'inférence JESS
        /*Le moteur d'inférence JESS implémente l'algorithme RETE (algorithme qui se base sur l'appariement de formes ou «pattern matching»)*/
        engine = new Rete();
        engine.reset();
```

```
        // appel aux règles de fichier .clp
      /*Appel au fichier .clp où se trouvent toutes les règles
nécessaires au moteur d'inférence*/

        engine.batch("AgentApprenant.clp");
        engine.add(actionetudiant.getEtudiant());
    }
    private void loadDonnees(Etudiant etud,Etape etape) throws JessException {
              // ajouter les faits nécessaires
              engine.add(etud);
              engine.add(etape);
    }

/*La méthode run permet d'exécuter les règles du fichier
AgentApprenant.clp*/
 public void  run(Etudiant etud,Etape etape) throws JessException {

       // insérer les données
        loadDonnees(etud,etape);
        // executer les règles jess
        engine.run();
    }
}
```

APPENDICE A

LES MACHINES À ENSEIGNER

«I believe that the motion picture is destined to revolutionize our educational system and that in a few years it will supplant largely, if not entirely, the use of textbooks... »
1922, Thomas Edison[15]

1809: Brevet de la première machine à enseigner nommée la "mode of teaching reading" par H. Chard aux états unis,
1909: l'utilisation du phonographe pour enseigner dans une école du Milwaukee (États-Unis),
1913: Le psychologiste Edward Thorndike propose un « livre mécanique » où on ne peut passer à la page suivante sans qu'on ait réussi les exercices de la page courante
1924: La première machine « moderne » à enseigner développée par Sidney Pressey. Elle permet de corriger automatiquement un questionnaire à choix multiple.
1933: La première télé éducative à l'université de l'Iowa.

Années 50 à 60 : L'enseignement programmé

1958: La machine de Skinner permet d'évaluer la réponse d'un étudiant entré dans la machine. Elle compare la réponse de l'étudient à la réponse valide.
1959: La machine de Norman Crowder qui améliore celle de Skinner en mettant en avant l'aspect adaptatif.
1960: L'apparition de l'enseignement assisté par ordinateur. Les tâches effectuées mécaniquement sur les anciennes machines à enseigner sont automatisées.

---

[15] http://www.cs.brown.edu/courses/cs092/2000/cs92.cuban86.html

Rapport-Gratuit.com

SOCRATES : système réalisé à l'université de l'Illinois. Il s'agit d'un des premiers systèmes capable de dialoguer avec l'apprenant.

PLATO : système de l'université de l'Illinois pour la diffusion de tutoriels avec les graphiques.

1966: Le premier ordinateur pour l'enseignement assisté : IBM 1500

LOGO : langage de programmation adapté à l'enseignement.

Années 70 : l'enseignement assisté par ordinateur

1970: SCHOLAR : le premier système tuteur intelligent jouant le rôle d'un tuteur humain en communicant avec l'apprentissage en langage naturel.

1974: SOPHIE : pur apprendre les règles et procédures pour déterminer les problèmes dans les circuits électriques.

1979: GUIDON : utilisant un système expert pur enseigner comment diagnostiquer et traiter certaines infections sanguines.

Années 80 : l'enseignement intelligemment assisté par ordinateur

1987 : M/EU (Mind Extension University) Il s'agit d'une chaine câblée offrant des programmes d'enseignement aux états unis

Années 90 : l'enseignement interactif assisté par ordinateur

1994 : L'entreprise Lotus lance sa plateforme d'apprentissage le "*Lotus Learning Management System"* et sa classe virtuelle «*Lotus Virtual Classroom* ». L'entreprise fut rachetée par IBM quelques années plus tard.

1994/95 : CALCampus.com : Il s'agit de la première école en ligne. Ceci a inspiré le lancement des premières méthodes d'enseignement à distance.

1997: CourseInfo développe ILN « *Interactive Learning Network* ». C'est un réseau d'apprentissage interactif utilisé dans des grandes institutions comme l'université de Cornell, et l'école de médecine de Yale. Il s'agit du premier système *e-Learning* en son genre.
1997 : Développement du système tuteur intelligent Autotutor

Fin des années 90 : environnement informatique d'apprentissage humain

1997: le lancement de la première version de la plateforme américaine d'apprentissage en ligne WebCT.
2001: le lancement du dispositif de formation ouvert et à distance Moodle.
2002, 2003,…2009: grâce à l'évolution de l'informatique et à l'expansion de l'Internet, on ne peut compter les diverses plateformes, logiciel et outils technologiques dédiés à l'enseignement.

Sources : (Antoniadis et al., 2009) et (Bordeleau, 1999)

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

BIBLIOGRAPHIE

Agarwal, R, A Deo et S Das. 2004. « Intelligent agents in E-learning ». *ACM SIGSOFT Software Engineering Notes,* vol. 29, n° 2, p. 1-1.

Antoniadis, Georges, Sylviane Granger, Olivier Kraif, Claude Ponton et Virginie Zampa. 2009. « NLP and CALL: integration is working ». en ligne. Bruxelles: N. Kubler. <http://w3.u-grenoble3.fr/lidilem/labo/file/TALC07.NLPandCall.pdf>. Consulté le 5 novembre 2009.

Aubé, M. 1996. « Sur l'autoroute électronique, les voyages formeront-ils la jeunesse ». *Vie pédagogique,* vol. 98, p. 36–39.

Biggs, John, et Kevin Collis. 1982. *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*. New York: Academic Press.

Bordeleau, Pierre. 1999. « Technologies éducatives : déjà une longue histoire ». en ligne. les cahiers du millionaires. <http://www.millenaire3.com/uploads/tx_ressm3/textes_bordeleau.pdf>.

Capus, Laurence , Frédéric Curvat, Olivier Leclair et Nicole Tourigny. 2006. « A Web environment to encourage students to do exercises outside the classroom: A case study ». *Educational Technology & Society,* vol. 9, n° 3, p. 173-181.

Capus, Laurence, Benoît Potvin et Nicole Tourigny. 2002. « Retour d'expérience sur l'enseignement de l'intelligence artificielle: vers un environnement d'apprentissage Web ». en ligne. Sainte-Foy (Québec): TICE. <http://edutice.archives-ouvertes.fr/docs/00/02/60/64/PDF/Capus.pdf>. Consulté le 12 Août 2009.

Chen, Tianyun, et Jianping Zhang. 2006. « An agent-based adaptive learning system (ABALS) ». In *First International Conference on Innovative Computing, Information and Control, 2006. ICICIC '06.* . Vol. 3, p. 378-381. Beijing

De Vries, E. 2001. « Les logiciels d'apprentissage: panoplie ou éventail ». *Revue Française de pédagogie,* vol. 137, p. 105-116.

Décanat des études: ÉTS. 2008. *École de technologie supérieur : plan de recherche stratégique*. Montréal: école de technologie supérieure, 7 p.

Doré, Sylvie, et Josianne Basque. 1999. « Les environnements d'apprentissage informatisés ». *Télé-université et École de technologie supérieure,* vol. 2, p. 15.

Drissi, M'hammed, Mohamed Talbi et Mohamed Kabbaj. 2006. « La formation à distance, un système complexe et compliqué ». en ligne. Association Enseignement Public & Informatique <http://www.epi.asso.fr/revue/articles/a0609b.htm>. Consulté le 5 Septembre 2009.

Emmanuel, Jalil, Laurence Capus et Nicole Tourigny. 2006. « Une interface plus intelligente pour SPHINX, un système d'aprentissage humain à partir d'exemples ». In *Proceedings of the 18th International Conferenceof the Association Francophone d'Interaction Homme-Machine*. p. 209-213. Montreal, Canada: ACM New York, USA.

Fareh, Youness. 2008. « Conception et réalisation d'un environnement d'apprentissage informatisé pour l'enseignement des échecs dans les écoles primaires de Montréal ». Montréal, École de technologie supérieure.

Friedman-Hill, Ernest. 2007. *Embedding Jess in a Java Application*. en ligne. <http://herzberg.ca.sandia.gov/docs/70/index.html >.

Giarratano, Joseph C., et Gary Riley. 1989. *Expert systems: principles and programming*. Brooks/Cole Publishing Co. Pacific Grove, CA, USA.

Graesser, AC, S Lu, GT Jackson, HH Mitchell, M Ventura, A Olney et MM Louwerse. 2004. « AutoTutor: A tutor with dialogue in natural language ». *Behavior Research Methods Instruments and Computers,* vol. 36, n° 2, p. 180-192.

Grand dictionnaire terminologique. 2009. *Office québécois de la langue française : grand dictionnaire terminologique* en ligne. <http://www.granddictionnaire.com/btml/fra/r_motclef/index800_1.asp>. Consulté le 20 Août 2009.

Henri, F, C Compte et B Charlier. 2007. « La scénarisation pédagogique dans tous ses débats… ». *Revue internationale des technologies en pédagogie universitaire,* vol. 4, n° 2, p. 14-24.

Institute for intelligent systems. 2008. « AutoTutor: What is AutoTutor ». In *AutoTutor*. En ligne. <http://www.autotutor.org/what/what.htm>. Consulté le 13 Octobre 2009.

ITS. *Le monde IA pour tous: Intelligent Tutoring Systems*. En ligne. <http://tecfa.unige.ch/staf/staf-i/gorga/staf18/projet/projet6/tm/edu-intell.html>. Consulté le 30 Janvier 2009.

Jeong, AC. 2003. « The sequential analysis of group interaction and critical thinking in online threaded discussions ». *American Journal of Distance Education,* vol. 17, n° 1, p. 25-43.

Kreczanik, Thomas. 2008. « Conception et appropriation des dispositifs d'information pédagogiques hypertextuels ». Thèse de doctorat en Sciences de l'Information et de la Communication en ligne, Lyon, Université Jean-Moulin Lyon 3, 459 p. <http://tel.archives-ouvertes.fr/docs/00/34/22/86/PDF/2008.10.23_These_VFINALE_LYON3_SCD.pdf> . Consulté le 17 Août 2009.

Laforcade, Pierre. 2005. « Approche par transformation de modèles pour la conception d'EIAH: Illustration entre les langages CPM et IMS-LD ». *Environnements Informatiques pour l'Apprentissage Humain* (Montpellier (France)).

Loghin, GC, T Carron et JC Marty. 2007a. « Apporter de la flexibilité dans l'observation d'une activité pédagogique ».

Loghin, GC, T Carron et JC Marty. 2007b. « Apporter de la flexibilité dans l'observation d'une activité pédagogique ». *Environnements Informatiques pour l'Apprentissage Humain (EIAH 2007)*, p. 83-94.

National Academy of Engineering. 2009. *Grand Chalenges for Engineering*. en ligne. <http://www.engineeringchallenges.org/cms/8996/9127.aspx>. Consulté le 12 Juillet 2009.

Nodenot, T. 2007. « Scénarisation pédagogique et modèles conceptuels d'un EIAH: que peuvent apporter les langages visuels? ». *International Journal of Technologies in Higher Education,* vol. 4, p. 2.

Paquette, Gilbert , Françoise Crevier et Claire Aubin. 1997. « Méthode d'ingénierie d'un système d'apprentissage (MISA) ». *Revue Informations In Cognito,* vol. 8, p. 37-52.

Paquette, Gilbert , Françoise Crevier et Claire Aubin. 1998. *Introduction à la méthode d'ingénierie d'un système d'apprentissage (MISA)*. Montréal: Centre de recherche LICEF, 34 p.

Perline. 2004. *SPIP système de publication pour Internet, créez votre site Web communautaire*. Coll. « Dossier Micro Application ». 467 p.

Pesty, S, C Webber et N Balacheff. 2003. « Baghera: une architecture multi-agents pour l'apprentissage humain ». *Cognitique, Toulouse, Cepadeus Edition*, p. 204-214.

Prince, Michael J., et Richard M. Felder. 2006. « Inductive teaching and learning methods: Definitions, comparisons,and research bases ». *Journal of Engineering Education,* vol. 95, n° 2, p. 123–38.

Ratté, Sylvie. 2008. *LOG625: Introduction aux systèmes intelligents: plan de cours*. Montréal: École de technologie supérieure, 5 p.

Ratté, Sylvie, et Jocelyne Caron. 2002. « On using the web as a collaboration space in the context of an industrial simulation ». *Proceedings of the 7th annual conference on Innovation and technology in computer science education* (Aarhus, Denmark).

Réty, JH, JC Martin, C Pelachaud et N Bensimon. 1998. « Coopération entre un hypermédia adaptatif éducatif et un agent pédagogique ». *Revue. Volume,* vol. 1, n° 1.

Rézeau, J. 2004. « L'apport original du forum de discussion de WebCT dans un cours d'anglais appliqué à l'archéologie ». *Les Cahiers de l'Acedle,* vol. 1.

Rickel, J, et WL Johnson. 1999. « Animated agents for procedural training in virtual reality: Perception, cognition, and motor control ». *Applied Artificial Intelligence,* vol. 13, n° 4, p. 343-382.

Russell, SJ, et P Norvig. 2006. *Intelligence artificielle:[avec près de 400 exercices]*. Pearson Education, 1184 p.

Soh, LK, N Khandaker, X Liu et H Jiang. 2006. « A computer-supported cooperative learning system with multiagent intelligence ». In., p. 1556-1563. ACM New York, NY, USA.

Sun Microsystems. *Java: S'informer sur la technologie Java*. en ligne. <http://www.java.com/fr/about/>. Consulté le 16 novembre 2009.

Tchounikine, Pierre. 2002a. « Pour une ingénierie des Environnements Informatiques pour l'Apprentissage Humain ». *Revue I3 information–interaction–intelligence,* vol. 2, n° 1, p. 59-95.

Tchounikine, Pierre. 2002b. « Quelques éléments sur la conception et l'ingénierie des EIAH ». *Actes des deuxièmes assises nationales du GdR I3 : Conception et ingénierie des EIAH*. p. 233-246.

Urretavizcaya-Loinaz, M, et IF de Castro. 2002. « Artificial intelligence and education: an overview ». *Artificial Intelligence: Technology with a Future. The European Online Magazine for the IT Professional,* vol. 3, n° 5, p. 53-58.

Viau, R. 2006. *Le profil d'apprentissage des étudiantes et des étudiants de l'École de technologie supérieure de Montréal*. Coll. « Rapport de l'ETS ». Montréal (Qc): Décanat à la formation : École de technologie supérieure