Des méta-modèles pour guider l'élicitation des connaissances en EIAH : contributions à l'enseignement de méthodes et à la personnalisation des activités

Nathalie Guin

▶ **To cite this version:**

Nathalie Guin. Des méta-modèles pour guider l'élicitation des connaissances en EIAH : contributions à l'enseignement de méthodes et à la personnalisation des activités. Environnements Informatiques pour l'Apprentissage Humain. Université Claude Bernard - Lyon 1, 2014. tel-01227436

HAL Id: tel-01227436
https://hal.science/tel-01227436

Submitted on 10 Nov 2015

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

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

Des méta-modèles pour guider l'élicitation des connaissances en EIAH : contributions à l'enseignement de méthodes et à la personnalisation des activités

Nathalie Guin
LIRIS
Université Claude Bernard – Lyon 1

Mémoire présenté pour obtenir l'Habilitation à Diriger des Recherches en Informatique

Partie 2 : Travaux de recherche

HDR soutenue le 9 décembre 2014, à l'Université Claude Bernard – Lyon 1,
devant le jury composé de :

Mme Nathalie Aussenac-Gilles	Directrice de recherche au CNRS, IRIT	Président
M. Serge Garlatti	Professeur, Télécom Bretagne	Rapporteur
Mme Monique Grandbastien	Professeur, Université de Lorraine	Rapporteur
M. Alain Mille	Professeur, Université Claude Bernard – Lyon 1	Examinateur
M. Roger Nkambou	Professeur, Université du Québec à Montréal	Rapporteur
M. Thierry Nodenot	Professeur, Université de Pau et des Pays de l'Adour	Examinateur

Table des matières

INTRODUCTION **7**

CHAPITRE 1 : PROBLÉMATIQUE GÉNÉRALE ET THÉMATIQUES DE RECHERCHE **11**

PARTIE A : ENSEIGNEMENT DE MÉTHODES **33**

INTRODUCTION 35

CHAPITRE A1 : ÉLICITER LES CONNAISSANCES D'UNE MÉTHODE DE RÉSOLUTION DE PROBLÈMES 39

CHAPITRE A2 : MODÉLISER UN PROCESSUS DE RÉSOLUTION DE PROBLÈMES MOBILISANT CONJOINTEMENT DES CLASSES ET DES CAS 53

CHAPITRE A3 : AMBRE, ENSEIGNER UNE MÉTHODE À PARTIR D'EXEMPLES 65

CHAPITRE A4 : ÉLICITER LES CONNAISSANCES PERMETTANT D'ACCOMPAGNER L'APPRENANT 81

CHAPITRE A5 : PERMETTRE À L'ENSEIGNANT D'ADAPTER L'EIAH À SES BESOINS 95

CHAPITRE A6 : VERS UN OUTIL AUTEUR POUR AMBRE 107

CONCLUSION ET PERSPECTIVES 115

PARTIE B : PERSONNALISATION DES EIAH **121**

INTRODUCTION 122

CHAPITRE B1 : ÉLABORER UN PROFIL DE L'APPRENANT À PARTIR DE L'ANALYSE DE SES TRACES 125

CHAPITRE B2 : GÉNÉRER DES EXERCICES 153

CHAPITRE B3 : PILOTER LA GÉNÉRATION D'EXERCICES À PARTIR DES PROFILS 171

CONCLUSION ET PERSPECTIVES 187

CONCLUSION ET PERSPECTIVES DE RECHERCHE **191**

RÉFÉRENCES BIBLIOGRAPHIQUES **197**

INDEX DES ILLUSTRATIONS **211**

TABLES DES MATIÈRES DÉTAILLÉE **215**

Au fermier,

au loup, à la chèvre et au chou

REMERCIEMENTS

J'écris ces quelques lignes après ma soutenance d'HDR, afin de remercier toutes les personnes qui ont d'une manière ou d'une autre rendu possible cette belle journée.

Je pense en premier lieu aux membres du jury, qui m'ont fait l'honneur de se pencher sur ce volumineux mémoire, et qui ont permis une discussion très intéressante lors de la séance de questions. Monique Grandbastien avait déjà évalué mon mémoire de thèse, et je la remercie d'avoir accepté d'évaluer de nombreuses années plus tard l'ensemble des travaux qui ont suivi. J'apprécie ses remarques toujours pertinentes et je me réjouis de partager avec elle un point de vue scientifique similaire sur notre domaine de recherche. Merci aussi à Roger Nkambou d'être venu de si loin pour nous apporter un point de vue nouveau sur des travaux qu'il ne connaissait pas forcément, ce qui nous permettra peut-être de travailler ensemble dans le futur. Merci enfin à Serge Garlatti, animateur infatigable de la communauté de recherche française en EIAH et IC, d'avoir pris du temps pour évaluer mon travail. A moi maintenant de prendre du temps pour l'aider à animer cette communauté.

Je souhaitais avoir un avis issu de la communauté IC sur mes recherches en EIAH, et c'est un très grand honneur pour moi que Nathalie Aussenac-Gilles ait accepté d'évaluer mon travail et de présider mon jury. Merci également à Thierry Nodenot d'avoir accepté malgré ses lourdes charges de participer au jury et de m'avoir donné avec sa bienveillance habituelle son point de vue qui est toujours intéressant. Merci enfin à Alain Mille d'avoir participé à ce jury, merci également pour nos discussions toujours enrichissantes, et pour sa relecture de la première version de ce mémoire.

J'ai ensuite une pensée pour Jean-Marc Fouet, qui m'a fait venir à Lyon, et dont l'absence s'est cruellement fait sentir pendant toutes ces années. Merci aussi aux membres de l'équipe SILEX, passés et présents, qui m'ont aidée et soutenue. Je pense en particulier à Jean-Charles, qui a relu ce mémoire et qui n'a cessé de m'encourager avec sa bonne humeur habituelle, à Amélie qui a passé une partie de son été à effectuer une relecture minutieuse de ce texte - merci aussi pour cette amitié qui dure depuis toutes ces années. Merci enfin à Marie, qui a relu de nombreux chapitres de ce mémoire, et qui m'a surtout libéré un temps précieux dans l'encadrement des étudiants, le suivi des projets en cours, et même l'enseignement.

Le travail présenté dans ce mémoire est pour une grande part celui des jeunes chercheurs. Merci donc à Sandra, Marie, Lemya, Sébastien, Awa, Bryan, Mohamed, Baptiste, Sonia, Florian. Merci également aux collègues avec lesquels j'ai travaillé pendant toutes ces années : André pour le plaisir du travail pluridisciplinaire, Stéphanie pour nos réunions déjantées lors de la conception de AMBRE-add, Vanda avec qui c'est un plaisir de co-encadrer des thèses, et auprès de qui j'ai beaucoup appris, Pierre-Antoine pour sa curiosité scientifique sans borne et son humeur toujours égale, et Amélie pour l'acuité et l'étendue de sa vision scientifique. Je remercie enfin les partenaires industriels avec lesquels je travaille au sein de plusieurs projets, car ces collaborations permettent de mesurer l'utilité de nos travaux de recherche pour les utilisateurs finaux : les élèves et les enseignants.

Merci aussi aux collègues du département qui m'ont permis de consacrer du temps à la préparation de cette HDR en prenant le relai sur de nombreuses responsabilités.

Je remercie enfin mes parents, qui m'ont toujours soutenue, et ma petite famille, qui pâtit trop souvent du temps consacré à mon travail. Merci à Emilie, Juliette et Corentin pour leur patience, et à Olivier pour son soutien sans faille.

INTRODUCTION

L'objectif de ce mémoire est de présenter la démarche qui a sous-tendu mon[1] parcours de recherche depuis ma thèse de doctorat effectuée au Laboratoire d'Informatique de Paris 6, de décrire les résultats qui ont jalonné ce parcours ainsi que les perspectives qui pourraient orienter mes travaux futurs. En tant que Maître de Conférences en informatique à l'Université Lyon 1, mes recherches ont été conduites au sein du laboratoire LISI (Laboratoire d'Ingénierie des Systèmes d'Information), puis du laboratoire LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information – UMR 5205) à partir de sa création en 2003.

Je mène mes recherches dans le domaine des Environnements Informatiques pour l'Apprentissage Humain (EIAH), avec une approche issue du domaine de l'Intelligence Artificielle, et plus particulièrement de l'Ingénierie des Connaissances. Ces recherches ont souvent été menées dans un contexte pluridisciplinaire, en collaborant avec des chercheurs d'autres disciplines et des praticiens du domaine éducatif.

Les Environnements Informatiques pour l'Apprentissage Humain (EIAH) sont des outils destinés à favoriser l'apprentissage, dans des contextes variés, par exemple en classe avec un enseignant, ou bien à distance dans un contexte d'apprentissage ouvert et massif. Le champ de recherche portant sur la conception et l'évaluation de ces outils rassemble plusieurs disciplines : l'informatique, la psychologie cognitive, les didactiques des disciplines, les sciences de l'éducation… Mes recherches relèvent des apports de l'informatique à la question de l'ingénierie des EIAH, champ de recherche qui porte sur « les questions scientifiques liées aux concepts, méthodes, théories, techniques et technologies utiles à la conception des EIAH» (Tchounikine, 2009).

Au sein de ce champ de recherche, je m'intéresse à la manière de représenter au sein de l'environnement informatique les connaissances qui lui permettront d'accompagner l'apprenant au cours de son apprentissage. Il me semble en effet fondamental d'une part de représenter explicitement les connaissances enjeu de l'apprentissage et d'autre part de mettre en œuvre des mécanismes permettant au système informatique qui interagit avec l'apprenant de s'appuyer sur ces connaissances afin de susciter leur apprentissage chez l'élève. Les connaissances enjeu de l'apprentissage ne sont pas les seules qui sont nécessaires au système informatique : des méta-connaissances permettent d'exploiter ces connaissances pour accompagner au mieux l'apprenant, et il faut donc aussi les représenter dans le système. Doter ainsi le système informatique de capacités de raisonnement sur des connaissances explicites et partageables me semble un préalable à de nombreuses recherches actuelles en informatique dans le domaine des EIAH, portant sur des problématiques que je ne traite pas moi-même, et qui visent à susciter des interactions entre apprenants, ou entre des apprenants et un enseignant, à motiver l'apprenant en lui proposant des activités ludiques ou impliquant des interfaces innovantes, et ce dans un contexte qui peut être mobile.

J'ai donc restreint mes recherches à ces EIAH que j'appellerai *centrés connaissances*, dans lesquels l'ensemble des interactions entre le système et l'apprenant sont construites sur une

[1] Les travaux de recherche décrits dans ce mémoire ont souvent été réalisés en collaboration avec d'autres

représentation explicite des connaissances enjeu de l'apprentissage. De ce fait, j'ai été amenée à collaborer avec des chercheurs en psychologie cognitive, qui étudient les mécanismes susceptibles de déclencher l'apprentissage chez le sujet humain, et également avec des chercheurs en didactiques des disciplines, du fait de l'expertise qu'ils apportent sur les connaissances enjeu de l'apprentissage.

J'ai de plus considéré dans mes travaux uniquement des situations d'apprentissage individuel, où un apprenant interagit avec un système informatique autour d'une tâche de résolution de problèmes, sans interactions instrumentées avec d'autres apprenants. Dans ces situations d'apprentissage, l'enseignant est entièrement impliqué, d'abord comme partie prenante de la conception des outils destinés aux élèves, mais aussi car une grande partie de mes recherches ces dernières années a eu comme objectif de fournir à l'enseignant des outils lui permettant de définir des EIAH ou des ressources pédagogiques adaptés à ses besoins.

Mes recherches se sont focalisées autour de deux grandes thématiques de recherche : l'enseignement de méthodes de résolution de problèmes, et la personnalisation des EIAH.

Les travaux sur l'enseignement de méthodes ont été menés dans le cadre du projet AMBRE, projet au long cours que j'ai commencé suite à mes travaux de thèse de doctorat, et qui se poursuit actuellement. Une méthode de résolution de problèmes dans un domaine donné consiste à réfléchir sur l'énoncé du problème avant d'entamer sa résolution, pour identifier de quelle classe relève le problème, afin de choisir une technique de résolution adaptée au problème. L'objectif du projet AMBRE est la conception d'EIAH destinés à enseigner de telles méthodes.

Les travaux sur la personnalisation des EIAH se sont focalisés sur l'adaptation à l'apprenant des activités qui lui sont proposées au sein d'un EIAH. La principale question de recherche est donc de choisir ou de construire une activité pédagogique répondant au besoin de l'apprenant. L'objectif de ces recherches est de permettre à l'enseignant de piloter ce processus en lui proposant des outils informatiques destinés à l'assister dans cette tâche. Deux questions de recherche sont issues de la question de l'affectation d'une activité à un apprenant : celle de l'élaboration d'un profil de l'apprenant et celle de la construction d'activités pédagogiques. L'élaboration d'un profil de l'apprenant consiste à extraire de ses traces d'interaction avec l'environnement des éléments informant sur son degré de maîtrise des connaissances en jeu dans l'activité. La construction d'activités pédagogiques consiste à choisir ou générer des activités à proposer à l'apprenant à partir de contraintes fixées en fonction de son profil.

Le point commun à ces deux thématiques de recherche dans le cadre de mes travaux est la question de l'élicitation des connaissances nécessaires à l'EIAH pour accompagner l'utilisateur. Cette problématique de recherche peut être considérée selon deux points de vue : celui de l'humain à qui il faut fournir des modèles et des langages de représentation lui permettant d'expliciter les connaissances et de les représenter d'une façon compréhensible pour le système informatique, et celui du système informatique qui doit acquérir auprès de l'humain les connaissances qui sont nécessaires pour qu'il puisse fournir un support adéquat à ses utilisateurs. Même s'il s'agit de la même problématique, mes recherches ont évolué au cours des années d'un point de vue initialement centré sur la représentation des connaissances de l'expert au sein du système informatique, à un point vue actuellement plus

centré sur les modèles et les outils permettant l'acquisition des connaissances auprès de l'expert.

Présentation du plan du mémoire

Le premier chapitre de ce mémoire présente les points communs à l'ensemble de mes travaux de recherche. Il précise les caractéristiques des EIAH considérés, présente plus précisément les deux thématiques de recherche que sont l'enseignement de méthodes et la personnalisation des EIAH, et décrit mon approche scientifique concernant la problématique de l'élicitation des connaissances présente au sein de ces deux thématiques.

La partie A présente ensuite un ensemble de chapitres relevant de la thématique de l'enseignement de méthodes. Après une introduction qui présente l'ensemble de ces travaux et leurs articulations, le chapitre A1 décrit comment modéliser une méthode de résolution de problèmes de manière à ce qu'un système informatique puisse l'appliquer pour résoudre des problèmes. Le chapitre A2 discute du lien entre deux types de connaissances utilisées en résolution de problèmes : les connaissances abstraites, liées aux classes de problèmes, et les connaissances contextualisées, liées aux cas. Le chapitre A3 présente le processus choisi dans le cadre du projet AMBRE pour faire acquérir des méthodes de résolution de problèmes, qui s'appuie justement sur la construction de connaissances abstraites de type classe à partir de connaissances contextualisées liées au raisonnement à partir de cas. Il décrit aussi comment nous avons appliqué ce principe pour concevoir des EIAH AMBRE et comment ces EIAH ont été évalués. Le chapitre A4 décrit les modèles de connaissances utilisés dans les EIAH AMBRE afin d'accompagner l'apprenant dans son processus d'apprentissage de la méthode. Le chapitre A5 présente les outils proposés à l'enseignant pour qu'il puisse adapter un EIAH AMBRE à ses besoins et aborde en particulier la problématique de la génération de problèmes. Enfin, le chapitre A6 décrit les travaux actuels sur la conception d'un outil auteur pour les EIAH AMBRE. La partie A se termine par un bilan de mes travaux sur l'enseignement de méthodes, assorti de quelques perspectives, et motive également les travaux sur la personnalisation décrits dans la partie B.

La partie B décrit les travaux relevant de la thématique de la personnalisation des EIAH. Après une introduction qui présente les différentes questions de recherche issues de cette problématique, le chapitre B1 aborde la question de l'élaboration de profils d'apprenants à partir de l'analyse de leurs traces d'interactions. Le chapitre B2 présente les modèles et outils proposés pour une génération semi-automatique d'activités pédagogiques. Le chapitre B3 décrit ensuite comment nous modélisons les stratégies de personnalisation des enseignants afin de piloter automatiquement la génération d'activités adaptées au profil de l'apprenant. La partie B se termine par un bilan des contributions sur la personnalisation des EIAH et présente quelques perspectives de recherche.

Enfin, une conclusion présente une vue synthétique des travaux de recherche menés et discute de l'évolution possible de ces travaux dans les années à venir.

CHAPITRE 1
PROBLÉMATIQUE GÉNÉRALE ET THÉMATIQUES DE RECHERCHE

Résumé

Ce chapitre présente les points communs à l'ensemble de mes travaux de recherche.

Il précise les caractéristiques des EIAH considérés : des EIAH centrés connaissances où l'apprenant est dans une situation d'apprentissage individuelle, dans laquelle il est invité à résoudre des problèmes

Il présente également les principales questions de recherche abordées dans les deux thématiques de recherche que sont l'enseignement de méthodes et la personnalisation des EIAH.

Il décrit enfin l'approche scientifique choisie pour traiter la problématique de l'élicitation des connaissances présente au sein de ces deux thématiques. Cette approche est fondée sur des méta-modèles permettant de guider l'élicitation des connaissances et des moteurs de raisonnement capables d'exploiter les modèles de connaissance ainsi produits.

INTRODUCTION 12

1 DOMAINE D'ÉTUDE : DE QUELS EIAH PARLONS-NOUS ? 12

1.1 QUELLES SITUATIONS PÉDAGOGIQUES ? 12
1.2 QUEL RÔLE POUR L'ENSEIGNANT ? 13
1.3 DES EIAH CENTRÉS CONNAISSANCES 14
1.4 SYNTHÈSE 15

2 THÉMATIQUES DE RECHERCHE 15

2.1 ENSEIGNEMENT DE MÉTHODES 15
2.2 PERSONNALISATION DES EIAH 17
2.3 LIEN ENTRE LES DEUX THÉMATIQUES DE RECHERCHE 20

3 PROBLÉMATIQUE GÉNÉRALE : L'ÉLICITATION DES CONNAISSANCES 21

4 APPROCHE SCIENTIFIQUE 25

4.1 S'APPUYER SUR DES MÉTA-MODÈLES POUR GUIDER L'ÉLICITATION DES CONNAISSANCES 25
4.2 QUELLE ASSISTANCE À CE PROCESSUS D'ÉLICITATION DES CONNAISSANCES ? 26
4.3 POSITIONNEMENT 26
4.4 DÉMARCHE DE RECHERCHE 28

SYNTHÈSE 30

Introduction

Comme je l'ai précisé en introduction, ce mémoire est articulé autour de deux parties : la partie A présentant les travaux réalisés autour de l'enseignement de méthodes dans le cadre du projet AMBRE, et la partie B décrivant les travaux sur la thématique de la personnalisation des EIAH. L'objectif de ce premier chapitre est de montrer les points communs à l'ensemble des travaux menés dans ces deux thématiques de recherche.

La première section précise les caractéristiques des EIAH sur lesquels je travaille afin de définir le domaine de validité de mes travaux : quelles situations pédagogiques sont abordées, quel est le rôle de l'enseignant, et quelle place prennent les connaissances. La section 2 présente les questions de recherche abordées au sein des deux thématiques : l'enseignement de méthodes et la personnalisation des EIAH. La section 3 montre ensuite en quoi la problématique de l'élicitation des connaissances est sous-jacente à l'ensemble de ces questions de recherche. Enfin, la section 4 décrit l'approche que j'adopte pour répondre à cette problématique, et qui sera ensuite illustrée dans l'ensemble des chapitres suivant.

1 Domaine d'étude : de quels EIAH parlons-nous ?

L'objectif de cette section est de définir le domaine de validité de mes travaux, en précisant la nature des situations pédagogiques que je souhaite accompagner grâce à des EIAH, ainsi que les principales caractéristiques de ces EIAH.

1.1 Quelles situations pédagogiques ?

Quels apprenants ?

Les travaux présentés dans ce mémoire ont été conduits pour des apprenants de niveaux variés. Nous n'avons pas appliqué ces travaux à un contexte où les apprenants sont des adultes en situation de formation professionnelle, même si certains travaux comme ceux des projets AMBRE (cf. partie A) ou CLAIRE (cf. chapitre B2) pourraient s'y appliquer, et nous ne nous sommes pas intéressés non plus aux spécificités des apprenants en situation de handicap, même si certains travaux comme ceux relatifs à la personnalisation pourraient s'y appliquer.

Dans certains projets comme le projet AMBRE, les apprenants sont jeunes (à partir de 7 ans), ce qui nécessite d'adapter le vocabulaire utilisé dans l'interface de l'EIAH, mais par exemple dans le projet CLAIRE le public visé est plutôt constitué d'étudiants.

Quels contextes ?

Dans les travaux de recherche que j'ai menés, je n'ai pas étudié de situations d'apprentissage collaboratives. Les résultats des travaux conduits s'appliquent donc à des situations d'apprentissage individuelles, mais souvent accompagnées par un enseignant. Ces activités peuvent avoir lieu sur un environnement informatique au sein de la classe, en présence de l'enseignant (comme dans le projet AMBRE (cf. partie A) ou le projet « Cartographie des savoirs » (cf. chapitre B1)), ou bien à distance lors d'une activité en autonomie (comme dans le projet CLAIRE (cf. chapitre B2) ou le projet « Cartographie des savoirs »).

Si les premiers développements d'EIAH dans le cadre du projet AMBRE ont donné lieu à des logiciels de type « client lourd », l'ensemble des développements en cours sont effectués dans le cadre d'applications web.

Quelles activités pour l'apprenant ?

Les recherches en psychologie cognitive s'accordent pour dire que l'apprentissage est indissociable de la notion de résolution de problèmes. C'est avant tout en résolvant des problèmes que l'élève est amené à structurer ou restructurer ses représentations et acquérir ainsi de nouvelles connaissances (Alamargot, 2001; Anderson, 1983; Piaget, 1975). En effet, selon le constructivisme, l'apprentissage consiste à construire en mémoire des structures mentales (schèmes) qui sont convoqués lors de la réalisation d'une activité de type résolution de problèmes. Ces schèmes sont complétés et adaptés en fonction des situations de résolution de problèmes rencontrées.

C'est pourquoi, si l'on met de côté les environnements destinés à l'enseignant, les EIAH résultants de mes travaux proposent à l'élève des activités de type *résolution de problèmes*.

1.2 Quel rôle pour l'enseignant ?

Le rôle des EIAH que mes travaux contribuent à créer n'est bien sûr pas de remplacer l'enseignant, mais de lui fournir des outils destinés à accompagner son enseignement. Il peut s'agir d'environnements qu'il fait utiliser à ses élèves en classe, et dont il accompagne l'utilisation, ou d'applications que ses élèves ou étudiants sont invités à utiliser en complément de son enseignement, pour s'entraîner, s'évaluer, remédier à des difficultés ou approfondir certains sujets. Un enseignant auteur peut également concevoir des ressources pédagogiques pour des étudiants avec qui il n'a jamais de contact direct, et qui travaillent beaucoup en autonomie, comme dans le cadre des MOOCs (Massive Open Online Courses). Pour que les EIAH soient effectivement utilisés en classe, il faut qu'ils soient adaptés aux besoins et aux pratiques de chaque enseignant. Une première réponse à cette problématique consiste à associer des enseignants à la conception de ces EIAH, favorisant ainsi leur acceptabilité par leurs collègues (Leroux, 2002) (Bruillard et al., 2000). Cela ne garantit pas pour autant de répondre au besoin individuel de chaque enseignant, et c'est pourquoi il est également important de permettre à chaque enseignant d'achever cette conception (Grandbastien, 1999) grâce à un module qui lui est destiné et *via* lequel il peut paramétrer l'environnement destiné à ses élèves, mais aussi choisir ou créer les activités qu'ils vont effectuer. Afin de favoriser la réingénierie des EIAH, il faut aussi lui fournir des outils qui lui permettent d'observer et d'analyser l'activité des apprenants avec l'EIAH (Choquet, 2007).

Afin d'aller encore plus loin dans la possibilité donnée à l'enseignant d'agir sur les activités proposées à ses élèves, je travaille depuis quelques années sur des outils auteurs pour les enseignants. Les utilisateurs ciblés par ces outils auteur sont des enseignants bien familiarisés avec l'outil informatique mais n'ayant pas de compétence en programmation. L'enjeu est alors d'assister ces enseignants dans leur tâche en automatisant une partie du processus de création d'activités pédagogiques. Pour que le système puisse jouer ce rôle, il faut qu'il dispose des connaissances nécessaires pour mettre en œuvre cette automatisation, ce qui implique de fournir aux enseignants des outils leur permettant d'éliciter ces connaissances.

Si la nécessité d'impliquer les enseignants dans la conception des EIAH est maintenant largement partagée (Choquet, 2007; Nodenot, 2005), il me semble que les travaux sur les outils auteurs laissent trop souvent de côté l'aspect *ingénierie des connaissances* de l'ingénierie des EIAH. Nodenot (Nodenot, 2005) souligne que très peu d'EIAH issus de la recherche sont utilisés sur le terrain. Il cite par exemple Cabrigéomètre (Baulac, 1990) ou Aplusix (Bouhineau & Nicaud, 2006) pour les travaux français, et Geometry Tutor (Anderson, Boyle, & Yost, 1985) à l'international. Il explique que la faible utilisation des EIAH est due au coût élevé de conception de ces outils, qui ne permet pas aux enseignants de construire de tels EIAH adaptés à leurs besoins. Il présente alors les outils de type gestion de contenus (Learning Management Systems) comme une alternative. Même si je suis d'accord avec l'analyse effectuée concernant la faible utilisation des EIAH issus de la recherche, il me semble qu'il n'est pas pertinent de renoncer à des outils centrés connaissances pour privilégier une création rapide de ressources et de modalités de travail collaboratives. En effet, je défends comme Grandbastien (Grandbastien, 1999) qu'il est possible d'ouvrir à l'enseignant des EIAH comme ceux cités précédemment (ce que nous avons par exemple réalisé avec AMBRE-enseignant), mais aussi qu'il est possible de concevoir des outils auteurs qui permettent à l'enseignant d'éliciter les connaissances du domaine et les connaissances pédagogiques qui lui permettront de concevoir des EIAH *centrés connaissances*.

1.3 Des EIAH centrés connaissances

B. Woolf (Woolf, 2009, chap. 2) caractérise les EIAH selon quatre dimensions, soutenant qu'un EIAH efficace devrait être centré connaissances (knowledge-centered), centré apprenant (student-centered), centré évaluation (assessment-centered) et centré communauté (community-centered). Selon elle, un EIAH centré connaissances doit être capable de raisonner sur les connaissances du domaine, de savoir ce que les apprenants ont besoin de savoir et ce qu'ils feront des connaissances après les avoir acquises. Ainsi il doit posséder une représentation structurée de la connaissance enjeu de l'apprentissage, de manière à pouvoir définir des priorités et construire du matériel pédagogique adapté à l'apprenant.

J'utiliserai ce terme d'*EIAH centré connaissances* pour caractériser les EIAH sur lesquels je travaille, pour lesquels la question de recherche principale est la représentation des connaissances qui permettent à l'EIAH d'assurer l'ensemble des fonctionnalités que l'on en attend. Il s'agit en effet de représenter *explicitement* les connaissances enjeu de l'apprentissage, c'est-à-dire les connaissances du domaine permettant la résolution de problèmes. Ces connaissances ne sont pas les connaissances d'un expert dans le domaine, mais les connaissances telles qu'on voudrait qu'elles fonctionnent chez l'apprenant à l'issue de l'apprentissage (Corbett, Koedinger, & Anderson, 1997 ; Rogalski, 1994). Je m'inscris ainsi dans la même démarche que les *Model Tracing Tutors* (Anderson, Corbett, Koedinger, & Pelletier, 1995). Afin de pouvoir accompagner l'élève dans son apprentissage, il est également nécessaire de représenter explicitement des méta-connaissances qui décrivent comment exploiter les connaissances enjeu de l'apprentissage pour fournir à l'apprenant de l'aide, diagnostiquer ses réponses, construire des rétroactions pertinentes, élaborer un modèle de ses compétences, choisir une stratégie pédagogique et construire des activités personnalisées (Grandbastien, 1999).

Ces connaissances doivent être représentées grâce à des modèles computationnels qui permettent au système de raisonner sur les connaissances décrites. Ainsi, l'ensemble des

interactions entre l'EIAH et l'apprenant sont construites sur ces connaissances explicites. Une modification au sein de l'un de ces modèles de connaissances entraine alors une modification du comportement de l'EIAH.

1.4 Synthèse

Afin de cerner le domaine de validité de mes travaux, j'ai dans cette section caractérisé les EIAH issus de mes travaux : il s'agit d'EIAH où l'apprenant est dans une situation d'apprentissage individuelle, dans laquelle il est invité à résoudre des problèmes, soit en classe accompagné par l'enseignant, soit en autonomie, les activités pédagogiques ayant été préparées par l'enseignant. Cet enseignant est largement impliqué dans la conception de l'EIAH, *via* le paramétrage de l'application, la définition d'activités pédagogiques ou la création des activités de l'EIAH à l'aide d'un outil auteur. Ces EIAH sont *centrés connaissances*, c'est-à-dire que l'ensemble des fonctionnalités de l'EIAH s'appuie sur une représentation explicite des connaissances enjeu de l'apprentissage et des connaissances pédagogiques.

2 Thématiques de recherche

Cette section présente les deux grandes thématiques de recherche dans lesquelles mes travaux ont pris place : l'enseignement de méthodes et la personnalisation des EIAH. Pour chacune d'elles je définis la problématique et les questions de recherche qui en découlent.

2.1 Enseignement de méthodes

Enseigner une méthode de résolution de problèmes, dans un domaine donné, consiste à apprendre à l'apprenant à raisonner sur l'énoncé du problème avant d'entamer sa résolution (Rogalski, 1994) (Schoenfeld, 1985). Une méthode est constituée de méta-connaissances de résolution (Pitrat, 1990) qui permettent de reconnaître la classe du problème avant de le résoudre, afin de pouvoir choisir une technique de résolution adaptée aux problèmes de cette classe. Dans un domaine donné, une méthode est issue d'études en didactique de la discipline concernée.

Mes travaux sur l'enseignement de méthodes ont été conduits dans le cadre du projet AMBRE (Apprentissage de Méthodes Basé sur le Raisonnement à partir de l'Expérience). Afin de pouvoir concevoir des EIAH destinés à enseigner des méthodes, plusieurs questions de recherche se sont posées. Je les présente ci-dessous en décrivant brièvement les réponses que nous y avons apportées et en indiquant les chapitres du mémoire présentant ces contributions.

Comment représenter les connaissances d'une méthode pour permettre à un système d'appliquer cette méthode pour résoudre des problèmes ?

L'objectif du projet AMBRE étant de concevoir des EIAH destinés à enseigner des méthodes, la première question de recherche posée est celle de la conception de systèmes à base de connaissances capables d'appliquer de telles méthodes pour résoudre des problèmes. J'ai étudié cette problématique pendant ma thèse de doctorat qui a donné lieu à la conception et à l'implémentation de l'architecture SYRCLAD (SYstème de Résolution de problèmes basé sur une CLAssification du Domaine). Cette architecture est constituée d'une part de méta-modèles de connaissances qui permettent d'exprimer pour un domaine donné les modèles

de connaissances nécessaires à l'application de la méthode. Elle comporte d'autre part un moteur de raisonnement capable d'exploiter les modèles de connaissances conformes aux méta-modèles. On obtient ainsi dans chaque domaine étudié un résolveur de problèmes du domaine qui applique la méthode explicitée par les modèles de connaissances.

Ce travail est décrit dans le chapitre A1 du mémoire.

Comment résoudre des problèmes en utilisant à la fois des connaissances issues d'une méthode et des connaissances plus contextualisées ?

La littérature en psychologie cognitive montre que les experts en résolution de problème se distinguent des novices par la classification qu'ils donnent des problèmes, les experts classant les problèmes en fonction des techniques qu'ils vont pouvoir appliquer pour les résoudre (Chi, Feltovich, & Glaser, 1981). Les experts utilisent donc des méthodes, mais mobilisent également des connaissances plus contextuelles (S. Blessing & Ross, 1996). Nous avons étudié avec André Didierjean les différents mécanismes associés à l'utilisation de ces deux types de connaissances (les classes et les cas), afin d'élaborer une modélisation psychologiquement valide des processus mis en jeu. Ce travail a donné lieu à l'architecture CASCADE (CAS et Connaissances Abstraites : une Double Expertise), qui définit des méta-modèles de connaissances ainsi que le moteur de raisonnement permettant dans un domaine donné d'exploiter les modèles de connaissances conformes aux méta-modèles. On obtient ainsi pour chaque domaine un résolveur de problèmes capable de mobiliser conjointement deux types de connaissances complémentaires, pour une résolution de problèmes plus efficace et psychologiquement valide.

Ce travail est décrit dans le chapitre A2 du mémoire.

Comment enseigner une méthode de résolution de problèmes ?

Une chose est de disposer d'un système capable d'appliquer une méthode de résolution de problèmes, une autre est de proposer des activités à l'apprenant qui vont lui permettre d'acquérir cette méthode. En m'appuyant sur des travaux de psychologie cognitive (B. Ross & Kennedy, 1990) montrant que les élèves construisent des schémas à partir de la comparaison d'exemples et de la résolution de problèmes par analogie, j'ai proposé le cycle AMBRE, inspiré par le Raisonnement à Partir de Cas (RàPC), qui invite l'apprenant à adapter la résolution d'un problème-type qu'il a déjà étudié pour résoudre un nouveau problème.

Le chapitre A3 de ce mémoire décrit plus précisément le cycle AMBRE et la manière dont nous l'avons mis en œuvre pour réaliser l'EIAH AMBRE-add appliqué au domaine des problèmes additifs à l'école primaire. Je décris également dans ce chapitre les expérimentations menées dans le cadre de la thèse de Sandra Nogry afin d'évaluer l'impact du cycle AMBRE sur l'apprentissage de la méthode.

Comment diagnostiquer les réponses de l'apprenant, et lui fournir aide et explications sur ses erreurs ?

Un système qui résout les problèmes d'un domaine en appliquant une méthode ne dispose pas pour autant des connaissances lui permettant de diagnostiquer les réponses de l'apprenant, ni de lui fournir des explications sur ses erreurs. L'architecture CHAMADE (arCHitecture pour l'Apprentissage de Méthodes permettant Aide, Diagnostic et Explications) définit des méta-modèles de connaissances qui complètent les bases de connaissance de l'architecture SYCLAD, en permettant la modélisation de connaissances de

diagnostic, de connaissances permettant de fournir de l'aide à l'apprenant, et de connaissances permettant de générer des explications adaptées aux réponses de l'élève.

Ce travail est décrit dans le chapitre A4 du mémoire.

Comment permettre à l'enseignant d'agir sur l'EIAH ? Comment générer des problèmes ?

Suite aux contributions issues des questions de recherche présentées ci-dessus, nous avions la possibilité de concevoir des EIAH AMBRE pour enseigner des méthodes et accompagner l'élève dans son apprentissage, ce qui a en particulier amené au développement de l'EIAH AMBRE-add. Afin de favoriser l'utilisation de tels EIAH en classe, nous avons étudié comment permettre à l'enseignant d'agir sur l'EIAH destiné à ses élèves, afin qu'il puisse l'adapter à ses besoins, et conçu le logiciel AMBRE-enseignant. Nous avons en particulier étudié la question de la génération semi-automatique des problèmes, pour que le système génère automatiquement des problèmes à poser aux apprenants à partir de contraintes sur ces problèmes fixées par l'enseignant. Ce travail a donné lieu à l'architecture GenAMBRE, qui décrit des méta-modèles de connaissances nécessaires à la génération de problèmes dans le cadre du projet AMBRE.

Ce travail est décrit dans le chapitre A5 du mémoire.

Comment permettre à un enseignant de concevoir un EIAH AMBRE ?

Le coût de conception et de développement d'un EIAH AMBRE est important. Pour faire diminuer ce coût et permettre à l'enseignant de concevoir un EIAH répondant à ses attentes, j'ai souhaité travailler sur la conception d'un outil auteur pour le projet AMBRE. La principale question de recherche dans le cadre de ce travail, qui constitue la thèse d'Awa Diattara, est celle de l'acquisition auprès de l'auteur des connaissances nécessaires à un EIAH AMBRE. Il s'agit donc de définir des méta-modèles de connaissance et de fournir à l'auteur des outils fondés sur ces méta-modèles lui permettant d'éliciter les connaissances nécessaires à l'architecture SYRCLAD pour résoudre les problèmes, et les connaissances nécessaires à l'architecture CHAMADE pour diagnostiquer les réponses de l'apprenant et lui fournir aide et explications sur ses erreurs.

Cette question de recherche est étudiée dans le chapitre A6 du mémoire.

2.2 Personnalisation des EIAH

L'un des principaux atouts des EIAH est leur capacité à s'adapter à l'apprenant, de manière à ce que chaque élève puisse apprendre à son rythme. La personnalisation est donc une des problématiques fondamentales de ce domaine de recherche, et l'un des principaux verrous de leur développement (Marty & Mille, 2009). On peut entendre par personnalisation le fait d'adapter l'EIAH au niveau scolaire visé et aux pratiques pédagogiques des enseignants (comme dans AMBRE-enseignant par exemple). Je considèrerai ici la personnalisation comme une individualisation de l'enseignement, de manière à ce qu'un EIAH s'adapte à chaque élève. L'objectif de mes travaux sur cette problématique est que l'EIAH s'adapte automatiquement à chaque apprenant, mais selon une stratégie définie par l'enseignant. De ceci découlent trois questions de recherche.

S'adapter à chaque apprenant nécessite de disposer d'un profil des connaissances et compétences de l'apprenant. C'est pourquoi la première question de recherche que

j'étudierai ci-dessous est celle de l'élaboration d'un profil de l'apprenant à partir de l'analyse des traces de ses interactions avec l'EIAH.

Puisqu'il s'agit d'adapter l'EIAH, et que je travaille sur des EIAH proposant aux apprenants de résoudre des problèmes, je considérerai surtout la question de l'adaptation des *activités* proposées à l'apprenant. Comme il ne s'agit pas ici de se restreindre à l'enseignement de méthodes, je parlerai d'exercices pour englober l'ensemble des activités concernées. Adapter les exercices peut consister à les choisir en fonction d'un ensemble de critères dans une base prédéfinie d'exercices. Je discuterai plutôt ici de la question de générer des exercices en fonction de critères fixés par l'enseignant ou par le module pédagogique d'un EIAH, c'est-à-dire de génération semi-automatique d'exercices.

Si l'on dispose d'un profil des connaissances et compétences de l'apprenant, et que l'on peut générer des exercices répondant à certains critères, la personnalisation de l'EIAH va consister à définir comment fixer ces critères en fonction du contenu du profil de l'apprenant. L'objectif est alors d'alléger le travail de l'enseignant en lui proposant de mettre en œuvre automatiquement une individualisation fondée sur les consignes qu'il aura données de manière globale. Ce sera la troisième question de recherche discutée ci-dessous.

Comment élaborer des profils d'apprenants en analysant les traces d'interactions ?

Afin de pouvoir individualiser l'apprentissage, il est nécessaire de disposer d'informations sur l'apprenant. Supposer que l'enseignant dispose de ces informations, issues de sources diverses (Jean-Daubias, 2011) est une hypothèse forte, et requiert un travail important de l'enseignant. On peut aussi considérer que les traces d'interactions de l'apprenant avec l'EIAH que l'on souhaite personnaliser est une source d'informations qui peut être exploitée de manière automatique. Ainsi, en me fondant sur les travaux de l'équipe SILEX du LIRIS sur la notion de trace modélisée (Champin, Mille, & Prié, 2013), je me suis intéressée à la question de l'interprétation des traces d'interactions pour calculer les informations attendues dans un profil d'apprenant.

Si l'élaboration des profils à partir des traces peut être effectuée de manière ad-hoc pour un EIAH donné, l'objectif de mes travaux sur cette problématique est de fournir des modèles et des outils génériques, permettant à un utilisateur de mettre en œuvre cette élaboration, dans un domaine donné, sans nécessiter de compétences en programmation. Cet utilisateur pourra être un concepteur d'EIAH, un analyste, ou un enseignant.

Une première contribution à cette question de recherche est la proposition d'un langage de haut niveau permettant à l'utilisateur d'exprimer des requêtes sur les traces stockées dans un système à base de traces, afin de calculer des indicateurs. La proposition d'un méta-modèle de traces d'EIAH qui permettra de définir le modèle des traces issues d'un EIAH donné permet également d'envisager l'adaptation de patrons de requêtes et de patrons d'indicateurs proposés à l'utilisateur.

Lorsque l'EIAH repose sur un modèle explicite des connaissances enjeu de l'apprentissage et que les activités sont liées à ces connaissances, il est possible de définir un mécanisme calculant le taux de maîtrise de l'apprenant pour chacune des connaissances ou compétences enjeu de l'apprentissage. C'est ce que nous avons fait dans le cadre du projet « Cartographie des savoirs », en nous appuyant sur un référentiel de compétences (Chaachoua, 2010) que l'équipe METAH du LIG représente à l'aide d'une ontologie.

Un tel processus de diagnostic des connaissances de l'apprenant est indépendant du domaine d'application. D'autres techniques génériques de diagnostic existent dans la littérature (par exemple *Knowledge Tracing* (A. Corbett & Anderson, 1994) ou *Constraint-based Modelling* (Ohlsson, 1994)). Dans le cadre de la thèse de Sébastien Lallé, nous avons défini un méta-modèle permettant de décrire ces techniques génériques de diagnostic, ainsi qu'un processus permettant d'instancier ces techniques génériques de diagnostic des connaissances pour un domaine donné. Un utilisateur souhaitant disposer d'une technique de diagnostic sur un domaine donné peut utiliser ce processus en fournissant au système des traces d'interactions ainsi qu'une ontologie des concepts du domaine, et ainsi obtenir plusieurs techniques de diagnostic appliquées à son domaine. Cet utilisateur peut ensuite comparer les résultats des techniques instanciées afin de choisir celle répondant le mieux à ses attentes.

L'ensemble des contributions liées à cette question de l'élaboration des profils à partir des traces est décrit dans le chapitre B1 de ce mémoire.

Comment générer des exercices ?

Pour adapter les activités proposées aux apprenants, une approche est de choisir les activités les plus pertinentes parmi un ensemble prédéfini de ressources ou d'exercices. Afin d'alléger le travail de l'auteur de ces exercices qui doivent alors être produits en grand nombre, nous avons choisi quand cela est possible de favoriser la génération semi-automatique d'exercices, reprenant ainsi les idées développées avec GenAMBRE dans le module enseignant de AMBRE. Afin de généraliser les principes de GenAMBRE, nous avons conçu dans le cadre de la thèse de Marie Lefevre des générateurs d'exercices pour des types d'exercices que l'on retrouve dans des domaines variés (appariements, textes à trous…). Ces générateurs sont donc indépendants du domaine d'application. L'approche GEPPETO-P développée dans le logiciel Adapte permet ainsi à un enseignant de définir des modèles d'exercices qui sont ensuite utilisés par les générateurs pour construire des exercices constituant une feuille d'exercices. Les modèles de l'approche GEPPETO-P ont été repris dans le projet CLAIRE afin de créer des générateurs d'exercices interactifs.

Ces types d'exercices (QCM, textes à trous, appariements, etc.) peuvent être utilisés dans tous les domaines, mais il serait dommage de se limiter à ce type d'activités pédagogiques alors que des EIAH existants peuvent proposer des activités de résolution de problèmes prédéfinies ou inclure des générateurs d'exercices de types plus spécifiques et permettant un apprentissage différent. Dans le but d'offrir à l'auteur une interface unique permettant d'utiliser ces bases d'exercices ou ces générateurs existants, nous avons (à nouveau dans le cadre de la thèse de Marie Lefevre) défini l'approche GEPPETO-S, qui permet de piloter le choix ou la génération d'exercices dans des EIAH existants. Cette approche repose sur la définition par un expert d'un modèle des connaissances pédagogiques et techniques pour chacun de ces EIAH.

L'ensemble des contributions liées à cette question de la génération d'exercices est décrit dans le chapitre B2 de ce mémoire.

Comment personnaliser les exercices en fonction des profils d'apprenants ?

Une fois que l'on dispose de profils d'apprenants, et d'un moyen pour choisir ou générer des exercices, il faut encore définir comment piloter la création des exercices à partir des données contenues dans le profil. En effet, il n'est pas envisageable (en termes de charge de

travail) que l'enseignant étudie le profil de chacun de ses élèves pour ensuite décider comment créer des activités intéressantes pour lui. C'est pourquoi nous avons (toujours dans le cadre de la thèse de Marie Lefevre) défini le méta-modèle PERSUA2, qui permet à l'enseignant, grâce à sa mise en œuvre dans le logiciel Adapte, de modéliser sa stratégie de personnalisation. Le système applique ensuite la stratégie définie par l'enseignant, qui consiste en un ensemble de règles spécifiant comment affecter des activités pédagogiques en fonction des éléments présents dans le profil des apprenants.

Cette approche a été étendue d'une part pour le contexte spécifique des MOOCs et d'autre part dans le cadre du projet « Cartographie des savoirs ».

Ces travaux sont décrits dans le chapitre B3 de ce mémoire.

2.3 Lien entre les deux thématiques de recherche

Après avoir conçu l'EIAH AMBRE-add, nous avons développé un module permettant d'analyser les traces d'interactions de l'élève avec l'EIAH, afin de pouvoir disposer d'un profil de l'élève rassemblant des informations sur son comportement et ses compétences. Ce profil était destiné d'une part à être présenté à l'enseignant et d'autre part à être utilisé afin de personnaliser l'EIAH. Nous souhaitions également utiliser les traces afin de pouvoir étudier l'usage de l'EIAH par les élèves dans le but d'apporter des améliorations à l'outil. Ces travaux m'ont ensuite amenée à étudier de manière plus générique la question de l'analyse des traces d'interactions pour l'élaboration de profils d'apprenant ainsi que la personnalisation des EIAH à partir de ces profils.

Dans le cadre de AMBRE-enseignant, nous avons travaillé sur le générateur de problèmes GenAMBRE, qui a également initié les recherches conduites ensuite sur la génération d'activités et la manière d'utiliser des générateurs de problèmes pour personnaliser les EIAH.

Ainsi mes recherches sur le projet AMBRE m'ont conduite à aborder des questions de recherche que j'ai ensuite traitées de manière plus générique. Les contributions issues de ces recherches génériques sur la personnalisation des EIAH ont ensuite été tout naturellement évaluées en les appliquant entre autres aux EIAH du projet AMBRE.

La **Figure 1** présente un synopsis des questions de recherche présentées dans cette section. La section suivante montrera que le point commun à l'ensemble de ces questions est la problématique de l'élicitation des connaissances.

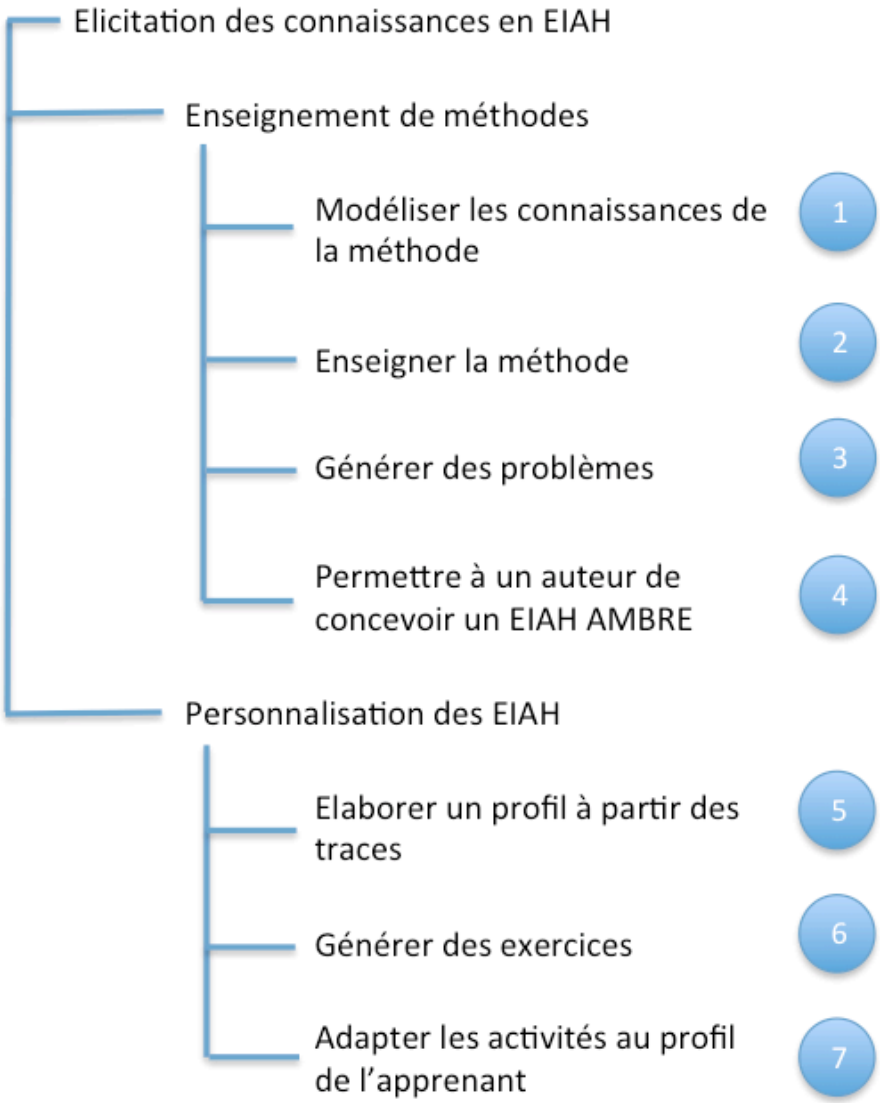

Figure 1 : Synopsis des questions de recherche

3 Problématique générale : l'élicitation des connaissances

La problématique générale de mes travaux, et qui est présente dans l'ensemble des questions de recherche présentées ci-dessus, est celle de l'élicitation des connaissances. Il s'agit de permettre à un utilisateur d'expliciter les connaissances qui seront nécessaires au système informatique pour accomplir la tâche que l'on en attend.

En effet, pour ce qui relève de l'enseignement de méthodes, il s'agit de (cf. **Figure 1**) :

1. Permettre à un expert de modéliser les connaissances définissant la méthode de résolution de problèmes qu'il souhaite enseigner : connaissances de classification, de reformulation, et techniques de résolution. Ces connaissances sont nécessaires à un système devant appliquer la méthode pour résoudre des problèmes.
2. Permettre à un expert de définir les connaissances nécessaires à un EIAH AMBRE pour accompagner l'élève dans son apprentissage : connaissances de diagnostic des

réponses, connaissances de construction de messages d'aide et d'explications sur les erreurs.

3. Permettre à un enseignant de définir des contraintes sur les problèmes qu'il souhaite faire résoudre à ses élèves, contraintes qui seront utilisées par le système pour générer les problèmes. Permettre à un expert de définir les connaissances nécessaires au système pour générer ces problèmes.
4. Permettre à un auteur de définir l'ensemble des connaissances nécessaires à un EIAH AMBRE. Il s'agit des connaissances mentionnées dans les points précédents et qu'un expert peut modéliser. Ici l'on cherche à acquérir interactivement ces connaissances auprès d'un utilisateur non informaticien, à travers une interface.

Pour ce qui relève de la personnalisation des EIAH, il s'agit de (cf. **Figure 1**) :

5. Permettre à un utilisateur de définir les connaissances d'élaboration du profil à partir des traces, c'est-à-dire les requêtes sur les traces ou la manière de calculer des indicateurs. Permettre à un utilisateur d'instancier une technique générique de diagnostic à un domaine donné, c'est-à-dire expliciter les connaissances du domaine nécessaires à la technique de diagnostic.
6. Permettre à un auteur de définir des contraintes sur les exercices qu'il souhaite poser à ses élèves, contraintes qui seront utilisées par les générateurs d'exercices. Permettre à un auteur de définir des connaissances du domaine afin de créer plus facilement des modèles d'exercices. Permettre à un expert de définir un modèle de son EIAH, c'est-à-dire les connaissances pédagogiques et techniques qui permettront de personnaliser cet EIAH.
7. Permettre à un enseignant de définir des connaissances de personnalisation, sous la forme d'une stratégie pédagogique.

Pour ce qui est de mes premiers travaux, la réponse à cette question de l'élicitation des connaissances était de proposer des méta-modèles de connaissances indépendants du domaine, ainsi que des moteurs de raisonnement capables d'exploiter les modèles conformes à ces méta-modèles. Ainsi il était laissé à la charge de l'expert le soin de définir (souvent en Prolog) des modèles de connaissances conformes aux méta-modèles, sans lui proposer d'outil permettant de lui faciliter cette tâche. Au fur et à mesure de l'avancement de mes recherches, j'ai adopté un point de vue plus proche de l'acquisition de connaissances, en cherchant à concevoir des outils permettant - *via* une interface - à des utilisateurs non informaticiens de définir les connaissances nécessaires au système. C'est le cas pour l'ensemble des travaux liés à la personnalisation des EIAH, ainsi que pour le projet d'un outil auteur pour les EIAH AMBRE. Cette évolution a permis d'élargir le périmètre des personnes auxquelles ces recherches permettent d'éliciter leurs connaissances.

De quelles connaissances parle-t-on ?

Comme nous l'avons vu ci-dessus, en EIAH nous avons certes besoin d'éliciter les connaissances de résolution de problèmes qui sont enjeu de l'apprentissage, mais aussi les connaissances qui permettent au système d'accompagner l'apprenant dans cet apprentissage : les connaissances de diagnostic des réponses de l'élève, les connaissances permettant de lui fournir une rétroaction (construction de messages d'aide et d'explications sur ses erreurs), les connaissances permettant de générer de nouvelles activités à lui

proposer, les connaissances d'élaboration d'un profil de l'apprenant à partir de ses traces d'interaction avec le système, et les connaissances permettant de personnaliser les activités proposées à l'apprenant en fonction de son profil.

De Jong et Ferguson (Ton De Jong & Ferguson-Hessler, 1996) distinguent quatre types de connaissances permettant au sujet humain d'accomplir une tâche (de type résolution de problèmes) : des connaissances de représentation du problème, des connaissances conceptuelles sur le domaine (souvent nommées connaissances déclaratives), des connaissances procédurales de résolution, et des connaissances stratégiques permettant de piloter la résolution. Ils définissent également un certain nombre de *propriétés* des connaissances. J'utilise ici certaines d'entre elles pour caractériser les connaissances listées au paragraphe précédent et qu'il faut éliciter pour permettre à un EIAH d'accompagner l'apprenant dans son apprentissage.

L'une des propriétés les plus souvent utilisées pour caractériser les connaissances est la distinction entre connaissances déclaratives et connaissances procédurales. Les connaissances déclaratives (savoirs) donnent des informations sur le domaine tandis que les connaissances procédurales (savoir-faire) donnent des indications sur les procédures de résolution et les conditions d'utilisation de ces procédures (Weil-Barais & Dubois, 1994, p. 433). Selon Anderson (Anderson, 1983), l'apprentissage dans le cadre de la résolution de problèmes consiste à construire des connaissances procédurales à partir de connaissances déclaratives. Cependant, les procédures peuvent rester implicites (Luengo, 2009), ce qui nécessite d'accompagner le processus de procéduralisation des connaissances (Weil-Barais & Dubois, 1994, p. 434).

J'adopterai la définition suivante proposée par P. Lecocq (Weil-Barais & Dubois, 1994, p. 380) : « Une connaissance a le statut de connaissance déclarative lorsqu'elle est manipulée par une autre connaissance, la même connaissance a le statut de connaissance procédurale lorsqu'elle manipule d'autres connaissances ». En effet, lorsqu'il s'agit des connaissances du système, c'est plus le caractère *explicite* qui est déterminant, que celui de *déclaratif*. C'est bien le caractère *explicite* des connaissances qui permet de les modifier facilement, et de les manipuler (par exemple *via* d'autres connaissances, les méta-connaissances). Je rejoins ainsi De Jong et Ferguson (Ton De Jong & Ferguson-Hessler, 1996), pour qui le pendant du caractère *procédural* de la connaissance est son caractère *conceptuel*, le caractère *déclaratif* étant lui le pendant du caractère *compilé* ou *implicite* de la connaissance. Dans un contexte d'accompagnement de l'apprentissage humain, le caractère explicite des connaissances est particulièrement important, puisqu'il semble illusoire d'envisager la transmission d'une connaissance qui serait automatisée et tacite.

L'exemple de l'enseignement de méthodes permet d'illustrer l'importance du caractère explicite des connaissances. En effet, l'enseignement de méthodes est particulièrement pertinent dans des domaines où les élèves rencontrent des difficultés à appliquer dans une situation de résolution de problèmes (qui nécessite des connaissances procédurales ou concrètes) les connaissances (déclaratives ou conceptuelles) apprises en cours. Le fait d'expliciter la méthode (i.e. la rendre déclarative) permet de faciliter l'apprentissage.

Une autre propriété des connaissances proposée par De Jong et Ferguson (Ton De Jong & Ferguson-Hessler, 1996) est leur caractère *général* ou *spécifique au domaine*. Ces auteurs soulignent que des stratégies générales de résolution de problèmes peuvent exister, mais que la plupart du temps ce sont des connaissances spécifiques au domaine que l'on cherche

à enseigner. Dans mes travaux de recherche, je cherche à définir des modèles indépendants du domaine pour représenter les connaissances nécessaires au système, ainsi que des moteurs de raisonnement permettant d'exploiter ces modèles, qui sont ainsi également indépendants du domaine. Les modèles permettent de guider l'élicitation de connaissances qui sont, elles, spécifiques au domaine.

Enfin, les connaissances explicitées doivent pouvoir évoluer, du fait des interactions du système avec l'utilisateur qui les définit ou celui qui les utilise. Ainsi, l'acquisition de connaissances peut être facilitée par l'utilisation des expériences (Mille, 1998). En effet, les expériences utilisant les connaissances peuvent amener à les remettre en cause et donc à les modifier.

Je me situe ainsi clairement dans une démarche d'ingénierie des connaissances, telle que la définit N. Aussenac-Gilles (Aussenac-Gilles, 2005) : « l'Ingénierie des Connaissances s'intéresse à la mise en place de systèmes informatiques s'intégrant dans des tâches humaines faisant appel à des connaissances spécialisées, systèmes qui assistent leur utilisateurs dans la réalisation de ces tâches. L'objectif est que le couple utilisateur-système réalise conjointement la tâche mieux, plus rapidement ou plus efficacement que l'utilisateur seul. Le système doit donc aussi maîtriser ces connaissances ».

Les inscriptions de connaissances (Bachimont, 2004) issues de l'ingénierie des connaissances peuvent être soit documentaires, soit formelles. Je me place résolument dans la même posture que D. Lenne (Lenne, 2009), en cherchant à exprimer les connaissances dans des langages permettant d'en contrôler la sémantique, afin de pouvoir les exploiter pour résoudre des problèmes, mais aussi accompagner l'apprenant.

4 Approche scientifique

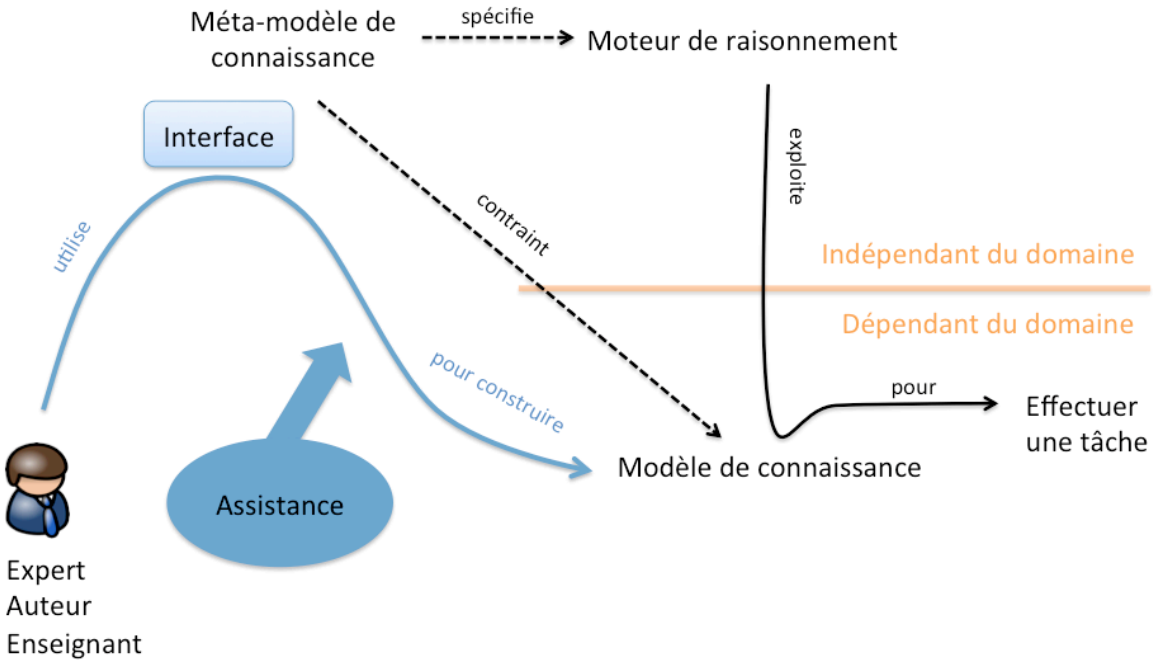

Figure 2 : Approche scientifique *(en bleu le processus d'élicitation des connaissances, en noir les modèles et mécanismes de raisonnement utilisés par le système informatique)*

4.1 S'appuyer sur des méta-modèles pour guider l'élicitation des connaissances

Pour répondre à cette problématique de l'élicitation de connaissances, mon approche (cf. **Figure 2**) consiste à définir des méta-modèles décrivant les connaissances que l'utilisateur doit définir pour construire un modèle de connaissances spécifique à un domaine donné. Selon les contextes ou les questions de recherche, cet utilisateur peut être soit un expert de la connaissance, soit un enseignant, soit un auteur de contenu pédagogique, soit un concepteur d'EIAH.

J'adopterai la définition de (Kleppe, Warmer, & Bast, 2003), selon laquelle un modèle est une description de tout ou partie d'un système à partir d'un langage clairement défini. Un méta-modèle est alors un modèle définissant un langage de description d'un modèle (OMG, 2008).

Pour une question de recherche donnée, le chercheur définit un méta-modèle dont l'objectif est de guider l'utilisateur dans sa définition des connaissances pour un domaine donné. Ce méta-modèle est indépendant du domaine et décrit quelles connaissances définir et sous quelle forme. Dans certains travaux, une interface fondée sur le méta-modèle aide l'utilisateur à éliciter les connaissances. Le résultat de ce processus d'élicitation est un modèle de connaissance dépendant du domaine et conforme au méta-modèle.

Le chercheur s'appuie sur ce méta-modèle pour définir un moteur de raisonnement capable d'exploiter les modèles de connaissances conformes au méta-modèle. Comme le méta-modèle, ce moteur de raisonnement est indépendant du domaine. Dans un domaine donné, le moteur exploite le modèle de connaissances pour effectuer la tâche attendue du

système : résoudre un problème, diagnostiquer une réponse, construire une rétroaction, élaborer un profil de l'apprenant, générer des exercices, personnaliser des activités…

J'utiliserai souvent le terme *d'architecture* pour désigner le couple formé par le méta-modèle de connaissances et le moteur de raisonnement, dans le sens qu'ils forment l'architecture d'un système à base de connaissances, qu'il faut "remplir" dans un domaine donné en définissant les modèles de connaissances requis par le méta-modèle, pour obtenir un système à base de connaissances fonctionnel pour le domaine concerné.

4.2 Quelle assistance à ce processus d'élicitation des connaissances ?

Dans certains travaux, nous proposons des outils complémentaires pour assister l'utilisateur dans son élicitation des connaissances. Il ne s'agit pas d'un processus générique d'assistance à la tâche, tel qu'il est par exemple proposé par (Ginon, 2014), mais seulement de moyens divers pour faciliter l'élicitation, selon les questions de recherche, et que je regroupe sur la **Figure 2** sous le terme d'assistance. Il s'agit de procédés qui sont indépendants du domaine et dont je donne ci-dessous deux exemples.

Un premier moyen de faciliter la tâche d'élicitation des connaissances conformément à un méta-modèle est de proposer à l'utilisateur des versions plus spécialisées du méta-modèle, donc plus contraintes, pour certains contextes. Un deuxième moyen est de lui proposer des patrons de modèles de connaissances, conformes au méta-modèle, et qu'il sera plus aisé d'instancier et de compléter, que de définir un modèle de connaissances uniquement en fonction du méta-modèle.

Prenons le méta-modèle PERSUA2, destiné à permettre l'élicitation de stratégies de personnalisation (cf. Chapitre B3). Une telle stratégie utilise un modèle des profils d'apprenants dont on dispose et un modèle des activités que l'on souhaite personnaliser, et précise comment personnaliser les activités en fonction du contenu du profil. Pour faciliter la tâche d'un auteur de MOOC qui souhaite définir une stratégie de personnalisation, nous avons conçu une version spécialisée du méta-modèle PERSUA2 pour le contexte des MOOCS. Nous proposons alors à l'auteur un patron décrivant les éléments qui sont en général présents dans un profil d'apprenant de MOOC, et un patron des activités souvent présentes dans un MOOC et personnalisables. Ces patrons de modèles pourront être instanciés par l'auteur du MOOC pour définir une stratégie de personnalisation pour son MOOC, ce qui représente moins de travail que de construire ces modèles à partir des méta-modèles.

Sur une autre question de recherche, nous avons défini un langage permettant de définir des requêtes sur les traces d'interaction, langage qui peut être considéré comme un méta-modèle d'indicateur. Afin d'alléger le travail de l'utilisateur qui utilise ce langage pour définir des requêtes sur les traces d'un EIAH donné, nous avons défini des patrons de requêtes correspondant à des indicateurs souvent utilisés dans le domaine des EIAH. L'utilisateur pourra instancier ces patrons à son EIAH, cette instanciation des patrons de requêtes étant même effectuée automatiquement par le système, pour peu que le modèle de traces de l'EIAH en question soit conforme au méta-modèle de traces d'EIAH que nous avons défini.

4.3 Positionnement

Mon approche est cohérente avec celle souhaitée par T. Nodenot (Nodenot, 2005, p. 40) : « Pour soulager les chercheurs en EIAH non informaticiens du poids des développements, il est urgent de leur demander de développer et réutiliser des modèles plutôt que du code.

Pour faciliter le dialogue entre ces modèles et les modèles que savent produire les informaticiens, pour rendre les modèles exploitables par les informaticiens, il est nécessaire d'expliciter les méta-modèles qui permettent d'interpréter les divers modèles proposés ». En tant que chercheur en informatique dans le domaine des EIAH, j'explicite ainsi des méta-modèles qui permettent ensuite aux non informaticiens (chercheurs mais aussi praticiens) de définir des modèles qui, du fait de leur conformité avec ces méta-modèles, seront exploitables. Le fait que ces modèles soient conformes à un méta-modèle permet également de les faire évoluer plus facilement, puisque le mécanisme d'exploitation du modèle mis en œuvre par le moteur de raisonnement (cf. **Figure 2**) repose sur le méta-modèle et non sur le modèle.

Pour N. Aussenac-Gilles (Aussenac-Gilles, 2005), un modèle est un médiateur, une représentation dont l'opérationnalisation va donner à l'utilisateur et au système les moyens de raisonner, d'agir et de produire des connaissances. Elle formule le problème de l'Ingénierie des Connaissances (IC) comme un problème de modélisation. Le *modèle conceptuel* est ainsi une représentation des connaissances propre au processus d'acquisition, qui favorise l'abstraction et la structuration progressive des connaissances, leur interprétation, leur évaluation et leur révision. Ces modèles conceptuels de l'IC sont des représentations intermédiaires définies pour guider l'explicitation de connaissances d'une part et la mise au point de représentations informatiques d'autre part (Aussenac-Gilles, 2005). Le modèle conceptuel est donc tout d'abord un instrument du processus d'Ingénierie des Connaissances lui-même avant d'en être le résultat, c'est-à-dire une composante du système à base de connaissances en usage auprès d'utilisateurs. On parle alors *d'acquisition guidée par les modèles.* Je m'inscris entièrement dans cette démarche, même si pour moi c'est le méta-modèle qui guide le processus d'Ingénierie des Connaissances, car il spécifie les connaissances qui doivent figurer dans le modèle pour chaque domaine. N. Aussenac-Gilles utilise plutôt de terme de *langage de modélisation* plutôt que celui de méta-modèle. Il y a pour elle deux parties dans un modèle conceptuel : le modèle du raisonnement et le modèle du domaine. Pour moi le modèle de raisonnement est associé au méta-modèle et exploite le modèle des connaissances spécifique au domaine.

L'acquisition des connaissances guidée par les modèles est la démarche adoptée par la méthode CommonKads (Breuker & Velde, 1994), conçue pour guider la définition d'un modèle conceptuel qui explicite une expertise. Cette méthode propose de construire le modèle conceptuel autour de cinq modèles : le modèle d'organisation, le modèle des tâches, le modèle des agents, le modèle des connaissances (qui précise les types, les structures et les rôles de la connaissance) et le modèle de communication. Des composants génériques sont proposés et peuvent être réutilisés. Ils décrivent comment résoudre des problèmes très généraux tels que diagnostic et prédiction du comportement, évaluation, conception, configuration, planification, affectation et ordonnancement.

Dans le domaine des EIAH, ces types de tâches très génériques de résolution de problèmes ne sont pas forcément adaptés à ce que l'on cherche à enseigner. En effet, ce qui particularise la démarche d'élicitation des connaissances dans le domaine des EIAH, c'est d'une part qu'il ne s'agit pas de modéliser les connaissances de l'expert, mais les connaissances telles qu'elles devraient fonctionner chez l'élève après l'apprentissage, pour ce qui est des connaissances enjeu de l'apprentissage : les connaissances du domaine. D'autre part, il faut en EIAH modéliser d'autres types de connaissances, comme les connaissances de diagnostic des compétences de l'apprenant, ou les connaissances

permettant de fournir des rétro-actions ou de l'assistance. Or ces connaissances ne sont pas des connaissances de résolution de problèmes à proprement dit : lorsque l'on parle par exemple de diagnostic des connaissances de l'apprenant en EIAH, il ne s'agit pas de la même tâche qu'un diagnostic de panne que l'on pourrait modéliser avec la méthode CommonKads. C'est pourquoi même si ma démarche relève de la même approche que CommonKads, elle s'applique à des problématiques spécifiques au champ de recherche des EIAH.

La méthode MISA (Paquette, 2007) propose un processus de conception d'EIAH fondé sur le langage MOT, un langage graphique de modélisation de connaissances permettant de décrire de manière déclarative les concepts, procédures et principes constituant des éléments d'apprentissage, pour ensuite scénariser des ressources pédagogiques liées à des compétences . Ce langage est conçu de manière à pouvoir décrire n'importe quel EIAH, et cette généricité a deux inconvénients : d'une part elle permet difficilement d'associer aux modèles de connaissances des mécanismes de raisonnement capables de les exploiter pour accompagner l'apprenant dans des tâches complexes de résolution de problème, et d'autre part elle peut être source de difficulté pour celui ou celle qui doit utiliser ces méta-modèles très génériques pour définir des modèles de connaissances. Je préfère proposer des méta-modèles de connaissances qui, tout en étant génériques, sont plus spécifiques à certaines questions de recherches du domaine des EIAH, comme la génération d'activités, leur personnalisation, la modélisation de l'apprenant, ou l'enseignement de méthodes. Dans le même ordre d'idées, pour faciliter la définition de modèles de connaissances conformes à des méta-modèles, je tente également dans certains cas de définir des patrons de modèles qui pourront être adaptés par l'utilisateur, peut-être plus rapidement que s'il devait construire le modèle seulement à partir du méta-modèle.

T. Winograd (Winograd, 2006) considère que cette approche de modélisation des connaissances à partir de modèles est une approche « rationaliste » issue de la « Good Old-Fashioned AI », et plaide pour une approche plus proche du « design », prenant en compte les interactions entre l'utilisateur et l'environnement. L'évolution de mes travaux va bien dans cette direction, en considérant que l'utilisateur définit des modèles de connaissances qui sont le résultat des interactions entre l'utilisateur et le système, l'utilisateur pouvant tester puis raffiner ses modèles. Une manière d'être encore plus proche de cette approche « design » serait, comme dans nos projets en cours et à venir, d'exploiter les traces d'interaction de l'utilisateur avec le système pour l'assister dans la construction des modèles de connaissances. Par exemple, l'utilisateur peut montrer au système le fonctionnement attendu sur un exemple, et le système généralise ce fonctionnement pour construire une connaissance qu'il propose à l'utilisateur pour validation.

4.4 Démarche de recherche

Dans mes projets de recherche, la démarche qui conduit à l'élaboration d'un méta-modèle commence souvent par la **constitution d'un corpus** (par exemple corpus de traces, ou corpus d'activités) ou le **choix de deux ou trois domaines d'application** (par exemple pour l'enseignement de méthodes).

La **conception du méta-modèle** est ensuite effectuée à partir des points communs aux éléments du corpus ou aux domaines considérés, afin de s'abstraire du domaine d'application dans la mesure du possible.

Le **moteur de raisonnement** associé au méta-modèle est également défini, afin de spécifier le processus d'exploitation des modèles conformes au méta-modèle.

Le méta-modèle est ensuite **évalué** en définissant des modèles conformes à ce méta-modèle, pour le **corpus** ou les **domaines considérés**, ce qui peut amener à revoir le méta-modèle.

Le méta-modèle est ensuite utilisé pour définir des modèles pour un **autre corpus** ou d'autres domaines d'application, ce qui peut encore amener à affiner le méta-modèle (et le moteur de raisonnement associé).

Dans un deuxième temps, un **outil** permettant à un utilisateur de définir les modèles de connaissances conformes au méta-modèle est conçu et **développé**, puis **testé** auprès du public visé afin de vérifier qu'il lui permet de définir les modèles de connaissances nécessaires à la tâche visée, ce qui peut bien sûr également entrainer des boucles de réingénierie.

Dans le même temps, le **moteur de raisonnement** permettant d'exploiter les modèles est **évalué** afin de vérifier qu'il permet bien d'effectuer la tâche visée par les modèles de connaissances.

Selon les projets, toutes ces étapes n'ont pas systématiquement été mises en œuvre (ce que je préciserai dans les chapitres correspondant).

Cette démarche de recherche est menée en collaboration avec des chercheurs issus d'autres disciplines que l'informatique au sein de la communauté de recherche en EIAH. J'ai par exemple sur l'enseignement de méthodes collaboré avec des chercheurs en psychologie cognitive et des chercheurs en didactiques des disciplines. Une constante de l'ensemble de mes projets de recherche est la collaboration avec des praticiens (enseignants, auteurs, concepteurs), puisqu'ils sont les utilisateurs primaires ou secondaires des outils développés dans le cadre de ces projets de recherche.

Pour élaborer un modèle de connaissance, on distingue souvent deux méthodes : une méthode descendante, dirigée par les (méta-)modèles, et une méthode ascendante, par abstraction à partir des données, par exemple des textes (Studer, Benjamins, & Fensel, 1998).

La démarche de recherche que je mets en œuvre utilise une méthode ascendante pour l'élaboration des méta-modèles, puisqu'il s'agit de définir les points communs à un corpus ou à un ensemble de domaines, mais ce n'est pas un système informatique qui effectue cette abstraction à partir des données. L'élaboration des modèles à partir des méta-modèles utilise alors une méthode descendante, en privilégiant l'interaction entre le système informatique et l'utilisateur qui élicite les connaissances. Cependant, une méthode ascendante peut être pertinente pour fournir à l'utilisateur une assistance dans sa tâche d'élicitation des connaissances. C'est que qu'a fait Sébastien Lallé pendant sa thèse, en utilisant des techniques d'apprentissage automatique pour extraire des traces les connaissances nécessaires à la définition de techniques de diagnostic. Le raisonnement à partir de l'expérience tracée (RàPET) tel que défini par l'équipe SILEX (Cordier, Lefevre, Champin, Georgeon, & Mille, 2013) relève de la même approche : exploiter les traces d'interaction de l'utilisateur avec le système pour assister l'utilisateur dans sa tâche, qui est ici l'élicitation des connaissances. Le RàPET est ainsi une perspective que je souhaite développer dans plusieurs de mes projets de recherche.

Synthèse

Dans ce premier chapitre, j'ai tout d'abord caractérisé les EIAH issus de mes travaux. On peut synthétiser ces caractéristiques de la manière suivante :

- L'apprenant est dans une **situation d'apprentissage individuelle**, en classe accompagné par l'enseignant, ou en autonomie.
- Les activités proposées à l'apprenant relèvent de la **résolution de problèmes**.
- Ces activités sont préparées par **l'enseignant** qui **participe à la conception de l'EIAH**, *via* le paramétrage de l'application, la spécification d'activités pédagogiques ou la création d'activités à l'aide d'un outil auteur.
- L'ensemble des fonctionnalités de l'EIAH s'appuie sur une **représentation explicite des connaissances** enjeu de l'apprentissage et des connaissances pédagogiques.

J'ai ensuite présenté les questions de recherche abordées dans mes travaux sur **l'enseignement de méthodes** de résolution de problèmes :

- Comment modéliser les connaissances de la méthode à enseigner ?
- Comment permettre à l'apprenant d'acquérir une telle méthode ?
- Comment permettre au système d'accompagner l'apprenant dans cet apprentissage ?
- Comment permettre à l'enseignant d'agir sur l'EIAH et de générer des problèmes ?
- Comment permettre à un auteur de concevoir un EIAH AMBRE ?

Les questions de recherche relatives à mes travaux sur la **personnalisation des EIAH** sont les suivantes :

- Comment élaborer des profils d'apprenant à partir de l'analyse des traces ?
- Comment générer des exercices ?
- Comment adapter les exercices au contenu du profil de l'apprenant ?

Que ce soit dans mes travaux sur l'enseignement de méthodes ou sur la personnalisation des activités proposées à l'apprenant, la problématique commune à l'ensemble des questions de recherches sur lesquelles j'ai travaillé est celle de **l'élicitation des connaissances**. Mon approche pour répondre à cette problématique est la suivante :

- Définir un méta-modèle de connaissance, indépendant du domaine, destiné à guider l'utilisateur (expert, auteur ou enseignant) dans son élicitation des connaissances.
- Définir un moteur de raisonnement, également indépendant du domaine, capable d'exploiter les modèles de connaissances conformes au méta-modèle.

En s'appuyant sur le méta-modèle de connaissances, l'utilisateur définit, pour un domaine donné, un modèle de connaissances que le moteur de raisonnement exploite pour accomplir une tâche destinée à accompagner l'apprenant dans son apprentissage.

Si dans mes premiers travaux l'élicitation des connaissances était guidée uniquement par les méta-modèles, mes travaux récents et en cours portent davantage sur une acquisition interactive des connaissances auprès de l'utilisateur. Ce processus d'acquisition des connaissances s'appuie d'une part sur des interfaces liées aux méta-modèles et d'autre part sur une assistance à l'utilisateur qui peut prendre des formes variées.

L'approche scientifique décrite dans ce chapitre et présentée sur la **Figure 2** sera illustrée dans chacun des chapitres constituant la suite de ce mémoire, qui décrivent les contributions apportées aux questions de recherche présentées ici. C'est en effet une approche qui a permis de proposer des contributions aux travaux de recherche que j'ai menés et qui a également été efficace dans les travaux des étudiants que j'ai encadrés.

PARTIE A
ENSEIGNEMENT DE MÉTHODES

INTRODUCTION 35

CHAPITRE A1 : ÉLICITER LES CONNAISSANCES D'UNE MÉTHODE DE RÉSOLUTION DE PROBLÈMES 39

CHAPITRE A2 : MODÉLISER UN PROCESSUS DE RÉSOLUTION DE PROBLÈMES MOBILISANT CONJOINTEMENT DES CLASSES ET DES CAS 53

CHAPITRE A3 : AMBRE : ENSEIGNER UNE MÉTHODE À PARTIR D'EXEMPLES 65

CHAPITRE A4 : ÉLICITER LES CONNAISSANCES PERMETTANT D'ACCOMPAGNER L'APPRENANT 81

CHAPITRE A5 : PERMETTRE À L'ENSEIGNANT D'ADAPTER L'EIAH À SES BESOINS 95

CHAPITRE A6 : VERS UN OUTIL AUTEUR POUR AMBRE 107

CONCLUSION ET PERSPECTIVES 115

INTRODUCTION

Pour Piaget (Piaget, 1975), l'apprentissage consiste à construire en mémoire des structures mentales (schèmes), que l'on complète et que l'on adapte en fonction des situations rencontrées. Ces schèmes sont utilisés en situation de résolution de problèmes et éventuellement restructurés. Weil-Barais (Weil-Barais & Dubois, 1994, p. 433) définit les connaissances déclaratives (savoirs) comme des informations sur le domaine et les connaissances procédurales (savoir-faire) comme des indications sur les procédures de résolution et les conditions d'utilisation de ces procédures. Selon Anderson (Anderson, 1983), l'apprentissage consiste à construire des connaissances procédurales à partir de connaissances déclaratives. Cependant dans l'enseignement, si les connaissances déclaratives sont souvent explicitées, les procédures peuvent rester implicites (Luengo, 2009), ce qui nécessite d'accompagner le processus de procéduralisation des connaissances (Weil-Barais & Dubois, 1994, p. 434).

Certaines recherches en didactiques des disciplines proposent d'enseigner explicitement des méthodes (Rogalski, 1994 ; Schoenfeld, 1985), c'est-à-dire des connaissances procédurales de résolution de problèmes, qui se fondent sur la notion de classe de problème. Résoudre un problème dans un domaine donné consiste alors à convoquer le bon schème au sens de Piaget. Il s'agit en effet pour l'élève de raisonner sur l'énoncé du problème avant d'entamer sa résolution (Pitrat, 1990, chap. 6), en identifiant la classe à laquelle il appartient, afin de pouvoir choisir une technique de résolution adaptée aux problèmes de cette classe. De manière plus générale, il semble important, dans le monde actuel où les connaissances sont facilement accessibles, d'enseigner aux enfants à prendre du recul et à utiliser des méta-connaissances pour exploiter au mieux les connaissances factuelles à leur disposition.

Mes recherches sur l'enseignement de méthodes ont commencé pendant ma thèse de doctorat et se poursuivent actuellement. Le fondement de l'enseignement de méthodes, qui est de rendre explicite les procédures de résolution, rejoint parfaitement mon parti pris de représenter explicitement dans les EIAH les connaissances enjeu de l'apprentissage. Ces recherches sont pour la plupart menées au sein du projet AMBRE, dont les fondements seront présentés au chapitre A3.

La **Figure 3** présente l'enchaînement temporel des questions de recherche traitées sur cette thématique de l'enseignement de méthodes.

Les premiers travaux menés pendant ma thèse ont consisté à définir une architecture permettant de représenter les connaissances d'une méthode, afin de permettre à un système (à base de connaissances) d'appliquer cette méthode pour résoudre des problèmes. Le chapitre A1 décrit ces travaux et présente l'architecture SYRCLAD permettant de construire des résolveurs de problèmes.

La littérature en psychologie cognitive montre que les experts en résolution de problèmes classent les problèmes en fonction des techniques qu'ils vont pouvoir appliquer pour les résoudre (Chi et al., 1981). Les experts utilisent donc des méthodes, mais mobilisent également des connaissances plus contextuelles (S. Blessing & Ross, 1996). Le chapitre 2 décrit les travaux menés avec André Didierjean sur l'utilisation conjointe en résolution de problèmes de ces deux types de connaissances : les connaissances abstraites, liées aux classes de problèmes, et les connaissances contextualisées, liées aux cas. L'architecture

CASCADE issue de ces travaux s'appuie sur l'architecture SYRCLAD et permet d'obtenir pour chaque domaine un résolveur de problèmes capable de mobiliser conjointement les deux types de connaissances.

Ces travaux ainsi que la littérature en psychologie cognitive sur le raisonnement par analogie m'ont conduite à proposer un processus inspiré du raisonnement à partir de cas pour enseigner des méthodes : le cycle AMBRE. Ce cycle s'appuie en effet sur la construction de connaissances abstraites, de type classes, à partir de connaissances contextualisées, liées au raisonnement à partir de cas. Le chapitre A3 décrit plus précisément le cycle AMBRE et la manière dont nous l'avons mis en œuvre pour réaliser l'EIAH AMBRE-add appliqué au domaine des problèmes additifs à l'école primaire. Ce chapitre décrit également brièvement les expérimentations menées dans le cadre de la thèse de Sandra Nogry afin d'évaluer l'impact du cycle AMBRE sur l'apprentissage de la méthode.

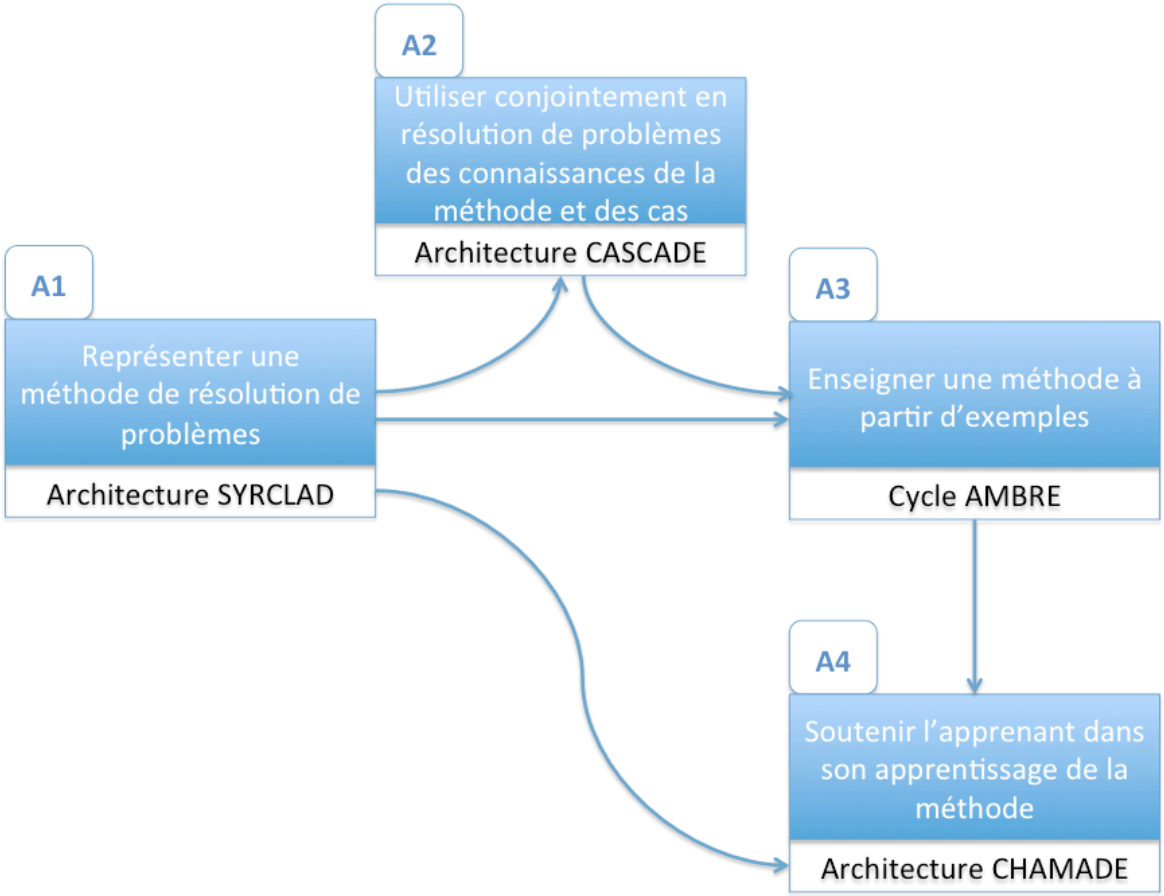

Figure 3. Premières questions de recherche sur l'enseignement de méthodes

Un système qui résout les problèmes d'un domaine en appliquant une méthode ne dispose pas pour autant des connaissances lui permettant de diagnostiquer les réponses de l'apprenant, ni de lui fournir des explications sur ses erreurs. Le chapitre A4 décrit l'architecture CHAMADE, destinée à définir des modèles de connaissance qui complètent les bases de connaissance de l'architecture SYRCLAD, en permettant la modélisation de connaissances de diagnostic, de connaissances permettant de fournir de l'aide à l'apprenant,

et de connaissances permettant de générer des explications adaptées aux réponses de l'élève. Elle permet ainsi à l'EIAH d'accompagner l'apprenant dans son processus d'apprentissage de la méthode.

Un EIAH AMBRE s'appuie donc sur un système à base de connaissances issu de l'architecture CHAMADE, intégrant un résolveur de problèmes issu de l'architecture SYRCLAD (cf. **Figure 4**). Le chapitre A3 présente l'exemple de l'EIAH AMBRE-add issu de l'application de ce principe au domaine des problèmes additifs.

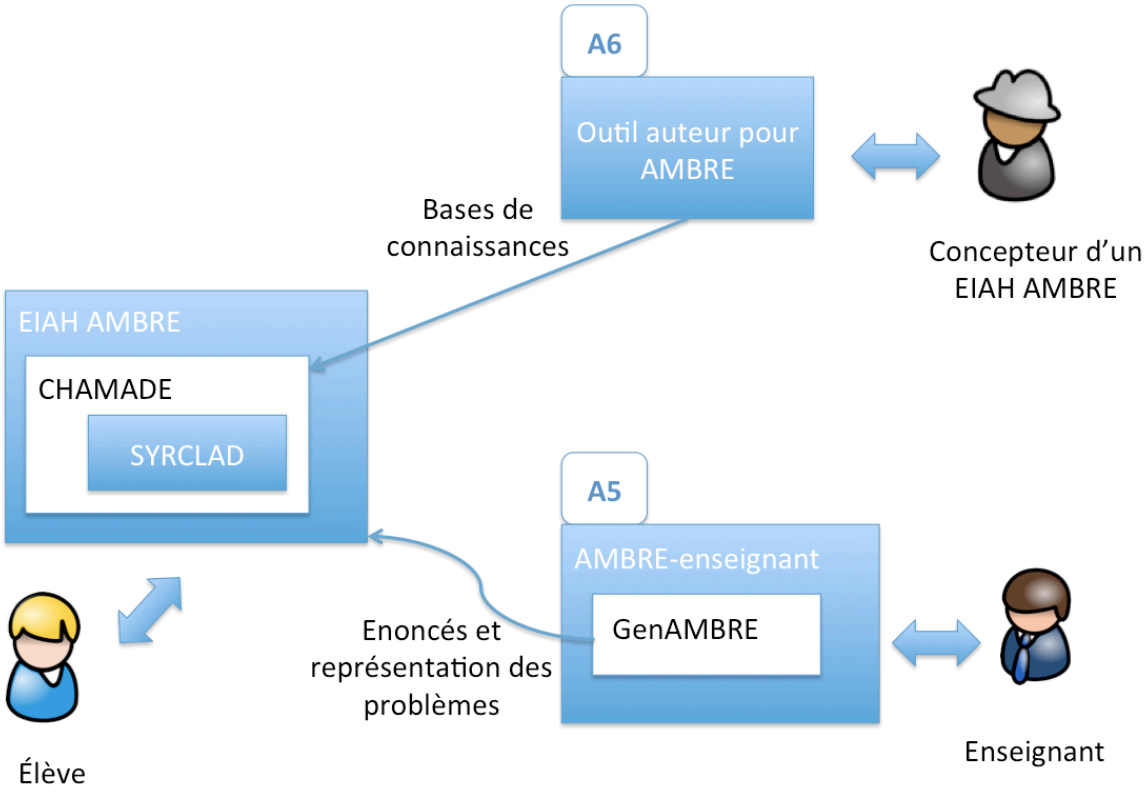

Figure 4. Outils pour la conception d'un EIAH AMBRE

Pour favoriser l'utilisation de tels EIAH en classe, nous avons étudié comment permettre à l'enseignant d'agir sur l'EIAH destiné à ses élèves, afin qu'il puisse l'adapter à ses besoins. C'est l'objectif du logiciel AMBRE-enseignant, qui est décrit dans le chapitre A5. Nous avons en particulier étudié la question de la génération semi-automatique de problèmes, pour que le système génère automatiquement des problèmes à poser aux apprenants à partir de contraintes sur ces problèmes fixées par l'enseignant. Cette étude a débouché sur l'architecture GenAMBRE, qui s'appuie également sur le résolveur SYRCLAD. GenAMBRE fournit à l'EIAH AMBRE des problèmes à poser aux élèves et répondant aux prescriptions de l'enseignant (cf. **Figure 4**).

Le coût de conception et de développement d'un EIAH AMBRE est important. Pour faire diminuer ce coût et permettre à un enseignant ou à un concepteur d'EIAH non informaticien de concevoir un EIAH répondant à ses attentes, j'ai souhaité travailler sur la conception d'un outil auteur pour le projet AMBRE. La principale question de recherche associée, qui est abordée dans le cadre de la thèse d'Awa Diattara, est celle de l'acquisition auprès de

l'auteur des connaissances nécessaires à un EIAH AMBRE. Le chapitre A6 discute des outils à fournir à l'auteur pour lui permettre d'éliciter les connaissances nécessaires à l'architecture SYRCLAD pour résoudre les problèmes, et les connaissances nécessaires à l'architecture CHAMADE pour diagnostiquer les réponses de l'apprenant et lui fournir aide et explications sur ses erreurs (cf. **Figure 4**).

CHAPITRE A1
ÉLICITER LES CONNAISSANCES D'UNE MÉTHODE DE RÉSOLUTION DE PROBLÈMES

Résumé

Ce chapitre présente l'architecture SYRCLAD, qui permet à un expert d'éliciter les connaissances permettant, dans un domaine donné, d'appliquer une méthode pour résoudre les problèmes. Cette architecture est constituée, d'une part, de méta-modèles permettant d'éliciter une classification de problèmes ainsi que des connaissances de reformulation et de résolution, et d'autre part, d'un moteur de résolution de problèmes capable d'exploiter ces connaissances. Dans un domaine donné, l'expert définit des modèles de connaissances de classification, de reformulation et de résolution conformes aux méta-modèles. Le moteur de raisonnement, indépendant du domaine, utilise ces modèles de connaissance pour obtenir un résolveur de problèmes du domaine qui fonctionne selon la méthode modélisée.

INTRODUCTION 40

1 CADRE DES RECHERCHES 40

2 QUESTION DE RECHERCHE : ÉLICITER LES CONNAISSANCES D'UNE MÉTHODE DE RÉSOLUTION DE PROBLÈMES 41

3 CONTRIBUTION : L'ARCHITECTURE SYRCLAD 42

3.1 TROIS MÉTA-MODÈLES DE CONNAISSANCE 43
3.2 PROCESSUS DE RÉSOLUTION DE PROBLÈMES 46
3.3 ÉVALUATION DE LA PROPOSITION 47

CONCLUSION 48

SYNTHÈSE DES CONTRIBUTIONS 50

POUR EN SAVOIR PLUS 50

PUBLICATIONS LIÉES 50

Introduction

Chi a montré dans (Chi et al., 1981) que les experts se distinguent des novices par la classification qu'ils donnent d'un problème. Lorsqu'on leur demande de classer des problèmes de mécanique, les novices parlent de "plan incliné" ou de "ressort", c'est-à-dire des traits de surface des problèmes. Les experts parlent de "conservation de l'énergie" ou de "troisième loi de Newton". Ils ont directement une idée de la technique de résolution qu'il faudra appliquer. Pour Chi, la classification est l'activation d'un schéma qui correspond à des connaissances. Cela débouche d'une part sur une représentation du problème (il s'agit de l'aspect déclaratif des connaissances) et d'autre part sur une solution (il s'agit de l'aspect procédural des connaissances).

Une méthode permet d'organiser une résolution de problèmes en se fondant sur un classement des problèmes et des outils de résolution. Elle permet ainsi d'expliciter les méta-connaissances associées à certaines connaissances du domaine et à certaines classes de problèmes. Dans un domaine donné, ce sont souvent des recherches en didactique de la discipline concernée qui aboutissent à la proposition d'une telle méthode.

Dans le domaine de la didactique des mathématiques, des études sur l'enseignement de méthodes ont été menées par Marc Rogalski (Rogalski, 1990) qui propose « d'enseigner explicitement les méthodes qui, dans chaque domaine relativement limité, permettent d'amorcer la résolution de problèmes de ce domaine. Ne serait-ce pas un moyen de combler un manque évident de l'enseignement […] ? ».

Selon Rogalski, « l'enseignement de méthodes de résolution de problèmes dans un domaine mathématique bien défini est une utilisation privilégiée d'un tuteur intelligent » et « le résolveur d'un tuteur donneur de leçons de méthodes doit fonctionner selon les méthodes qu'il veut enseigner, et non pas selon les méthodes expertes les plus générales du domaine concerné ». Il rejoint ainsi Nicaud (Nicaud, 1994), qui insiste également sur le fait que le module résolveur d'un EIAH doit être adapté à un niveau d'apprentissage donné et ainsi servir de résolveur de référence ayant une plus forte plausibilité cognitive qu'un système expert.

C'est la raison pour laquelle j'ai souhaité définir un méta-modèle permettant de modéliser les connaissances de méthodes de résolution de problèmes, pour pouvoir ensuite produire dans chaque domaine concerné un résolveur de problèmes utilisant une méthode. L'objectif de ce travail est que ces résolveurs puissent servir de résolveurs de référence pour des EIAH.

1 Cadre des recherches

Ce travail a été conduit pendant mon DEA et ma thèse de doctorat, entre 1994 et 1997, à l'Université Pierre et Marie Curie – Paris 6 et au sein du laboratoire LAFORIA. J'ai mené ces recherches sous la direction de Jacques Pitrat, Hélène Giroire et Gérard Tisseau, en interaction avec le groupe *Combien?* qui travaillait à la conception d'un EIAH pour enseigner la résolution de problèmes de dénombrement en terminale scientifique.

2 Question de recherche : éliciter les connaissances d'une méthode de résolution de problèmes

L'objectif étant d'utiliser des résolveurs appliquant des méthodes comme résolveurs de référence au sein d'EIAH, il est important que la démarche de résolution utilisée par le système soit explicite, de manière à faciliter (pour un futur EIAH) la production d'explications. Par rapport à l'approche générale décrite sur la **Figure 2** du chapitre 1, la question est ici (cf. **Figure 5**) de permettre à un expert disposant, dans un domaine donné, de connaissances sur une méthode de résolution de problème, de les expliciter, afin que ces connaissances puissent être utilisées par un système informatique. Pour faciliter cette élicitation, j'ai défini un méta-modèle décrivant la nature des connaissances constituant une méthode. Un moteur de raisonnement conçu conjointement permet d'exploiter les modèles de connaissances conformes au méta-modèle, ce qui permet de disposer de résolveurs de problèmes fonctionnant selon les méthodes explicitées.

Le méta-modèle de connaissance ainsi que le moteur de raisonnement associé constituent l'architecture SYRCLAD (SYstème de Résolution de problèmes basé sur une CLAssification du Domaine) et sont décrits dans la section suivante.

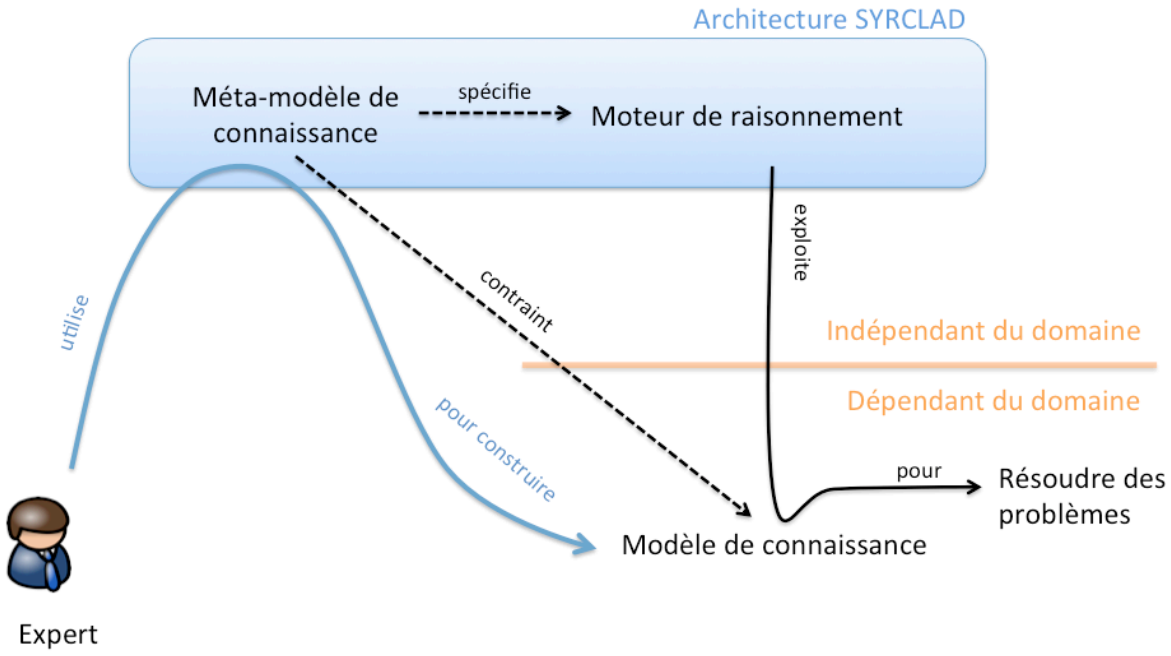

Figure 5. Instanciation de l'approche générale présentée dans le chapitre 1 pour la question de l'élicitation des connaissances d'une méthode de résolution de problèmes.

Cette architecture a été conçue par généralisation du concept de méthode appliqué au domaine des problèmes de dénombrement abordés en terminale scientifique, puis a été consolidée en l'appliquant aux problèmes additifs abordés à l'école élémentaire, aux problèmes de choix d'un plan de jeu au tarot, et aux problèmes de thermodynamique. Plus tard, l'architecture SYRCLAD a également été appliquée à certains problèmes du jeu d'échecs ainsi qu'à la conjugaison française. Elle peut potentiellement être utilisée pour tous

les domaines pour lesquels il est possible de définir une classification des problèmes permettant de choisir des techniques de résolution pertinentes.

3 Contribution : l'architecture SYRCLAD

L'architecture SYRCLAD représente les connaissances d'une méthode de résolution de problèmes selon trois méta-modèles : celui des connaissances de classification, celui des connaissances de reformulation et celui des connaissances de résolution.

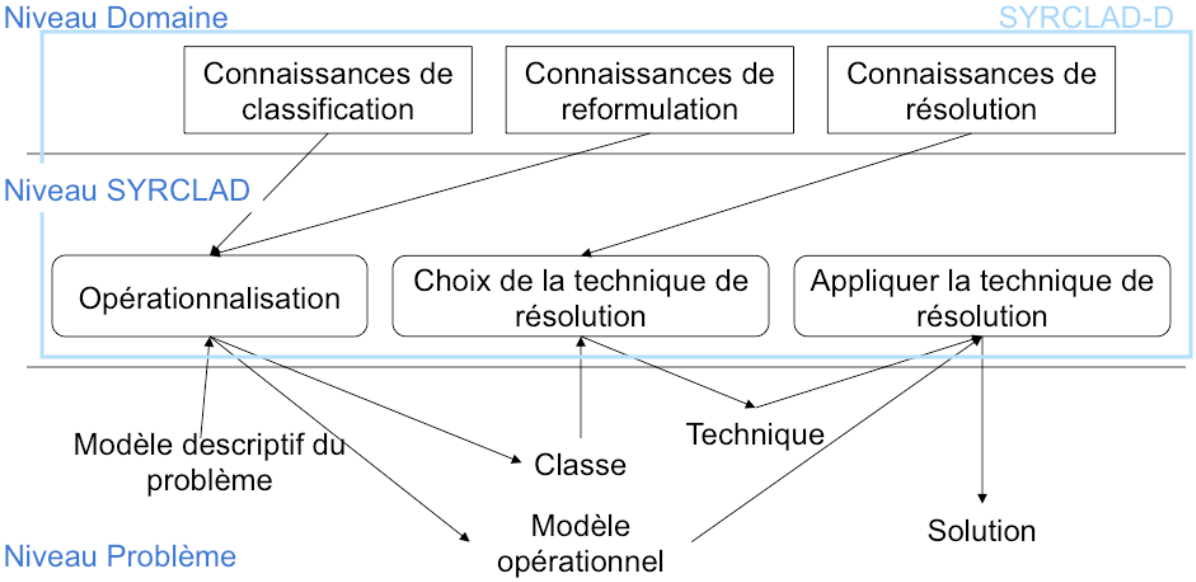

Figure 6. L'architecture SYRCLAD

Dans un domaine donné D, l'expert définit trois modèles de connaissances (cf. **Figure 6**) à partir de ces trois méta-modèles : les connaissances de classification, celles de reformulation et celles de résolution.

En associant ces trois modèles de connaissances au moteur de raisonnement de l'architecture SYRCLAD, on obtient SYRCLAD-D, un résolveur de problèmes pour le domaine D.

Les problèmes à résoudre ne sont pas donnés au résolveur SYRCLAD-D en langue naturelle, mais sous la forme d'un modèle du problème le plus proche possible de l'énoncé en langue naturelle, que j'appelle *modèle descriptif du problème*. Ce modèle décrit une situation concrète, qui est celle présentée dans l'énoncé. Il contient des traits de surface du problème considéré : ce sont des éléments concrets tels que les objets et les personnages.

Dans l'architecture SYRCLAD, qui est implémentée en Prolog, le modèle descriptif d'un problème est exprimé dans un langage de la logique du premier ordre. Ce modèle descriptif est composé d'un ensemble de propositions représentant les données fournies par l'énoncé du problème.

Pour résoudre un problème, un résolveur SYRCLAD-D effectue une opérationnalisation du problème, en utilisant les connaissances de classification et celles de reformulation. Il obtient ainsi d'une part la classe dont relève le problème, et d'autre part un nouveau modèle du problème, appelé *modèle opérationnel*. Les connaissances de résolution associent à certaines classes du graphe de classification dites *opérationnelles* une technique

de résolution adaptée. Pour obtenir une solution au problème, SYRCLAD-D applique la technique de résolution associée à la classe du problème sur le modèle opérationnel.

Je vais maintenant décrire les trois méta-modèles de connaissance de SYRCLAD, avant de présenter plus en détail le processus de résolution mis en œuvre par le moteur.

3.1 Trois méta-modèles de connaissance

Pour représenter les connaissances d'une méthode de résolution de problèmes avec SYRCLAD, il faut définir trois types de modèles de connaissances conformes à des méta-modèles. Ces trois méta-modèles étant liés entre eux, on pourrait aussi considérer qu'il s'agit d'un seul méta-modèle de connaissance dans lequel on distingue trois parties.

Connaissances de classification

Dans chaque domaine d'application, l'expert doit définir un graphe de classification des problèmes. La classification représentée par ce graphe est évidemment dépendante du domaine. Mais la structure de représentation du graphe et son utilisation sont les mêmes quel que soit le domaine.

Le méta-modèle définit ce graphe comme une hiérarchie de classes au sens de la relation de généralisation / spécialisation, où une classe C2 est une sous-classe d'une classe C1 si tout problème de C2 est aussi un problème de C1. Dans un tel graphe, on définit une classe racine comme étant la classe la plus générale, les feuilles étant les classes les plus spécifiques.

Pour chaque classe, on définit l'*attribut discriminant* comme étant l'attribut du problème qui discrimine les sous-classes. Les valeurs de cet attribut permettent de répartir les instances d'une classe suivant les différentes sous-classes.

Certaines classes suffisamment spécifiques pour que l'on puisse leur associer une technique de résolution sont appelées *classes opérationnelles*.

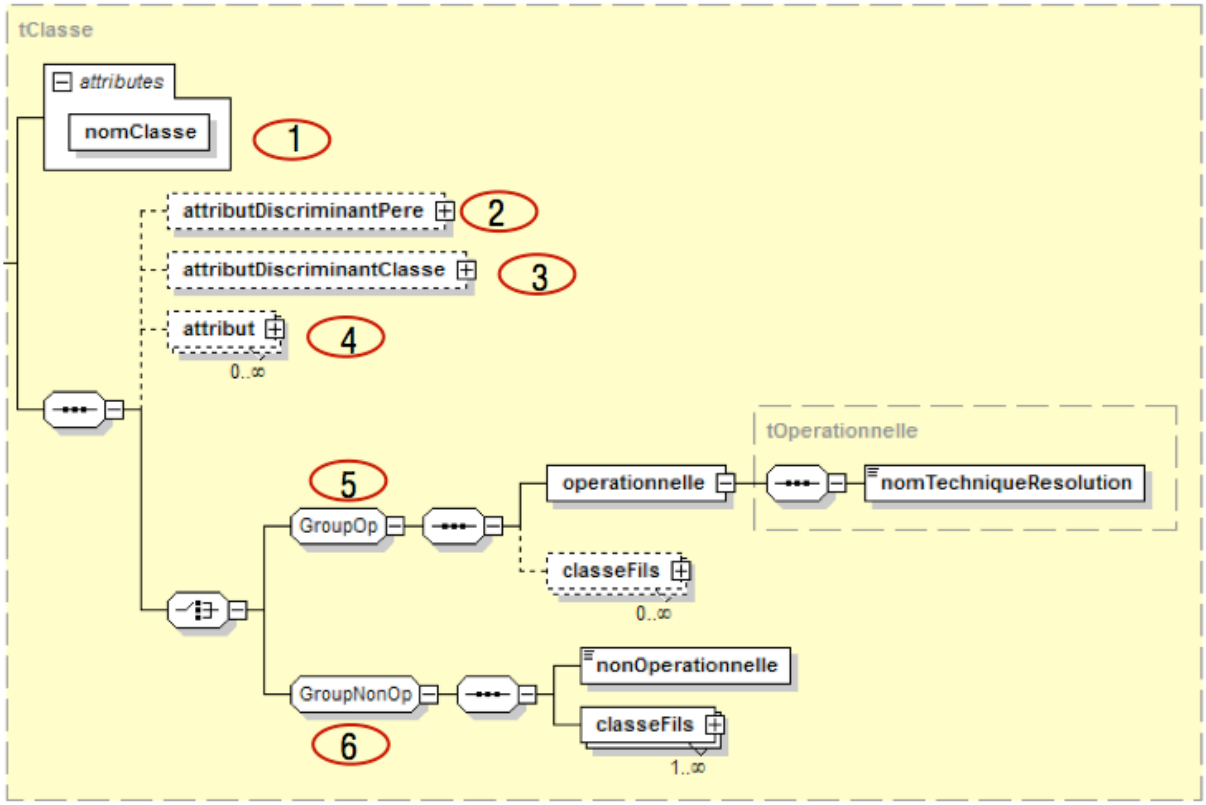

Figure 7. Schéma XML décrivant une classe de problèmes

Pour chaque classe C du graphe, l'expert doit définir (cf. **Figure 7**) :

1. le nom de la classe C ;
2. l'attribut discriminant du père de cette classe ainsi que sa valeur pour C (ce qui spécialise C par rapport à son père) ;
3. l'attribut qui permet de discriminer les classes Fils de la classe C ;
4. l'ensemble des attributs (discriminants ou non) qui seront utiles pour la résolution des problèmes relevant de cette classe ;
5. si la classe est opérationnelle, la technique de résolution associée. Les classes opérationnelles peuvent également avoir des classes Fils auxquelles on associera des techniques de résolution plus spécifiques ;
6. si la classe est non opérationnelle, elle doit avoir au moins une classe Fils.

La **Figure 8** présente un extrait de l'instanciation du méta-modèle des connaissances de classification pour SYRCLAD-add, le résolveur issu de SYRCLAD pour les problèmes additifs posés à l'école élémentaire. Le nom de la classe racine est c_problème. Cette classe n'a pas d'attribut discriminant pour le père puisqu'elle est la classe racine. L'attribut discriminant de cette classe est type_problème. L'attribut à déterminer est type_problème et les règles permettant de déterminer sa valeur sont respectivement type_pro_reunion, type_pro_changement, type_pro_comparaison1, type_pro_comparaion2 (cf. section suivante). Cette classe est non opérationnelle. Elle possède plusieurs classes Fils dont la première nommée c_changement. La classe c_changement a pour attribut discriminant opérateur. Cette classe est non opérationnelle.

```
<classification nomDomaine="problèmes_additifs">
    <classeRacine nomClasse="c_problème">
        <attributDiscriminantClasse nomAttribut="type_problème"/>
        <attribut nomAttribut="type_problème">
            <nomRegle>type_pro_réunion</nomRegle>
            <nomRegle>type_pro_changement</nomRegle>
            <nomRegle>type_pro_comparaison1</nomRegle>
            <nomRegle>type_pro_comparaison2</nomRegle>
        </attribut>
        <nonOperationnelle></nonOperationnelle>
        <classeFils nomClasse="c_changement">
            <attributDiscriminantPere nomAttribut="type_problème">
                <valeurAttributDiscrPere>changement</valeurAttributDiscrPere>
            </attributDiscriminantPere>
            <attributDiscriminantClasse nomAttribut="opérateur"/>
            <attribut nomAttribut="opérateur">
                    <nomRegle>op_ajouter</nomRegle>
                    <nomRegle>op_retrancher</nomRegle>
                    <nomRegle>op_inconnu</nomRegle>
            </attribut>
            <nonOperationnelle></nonOperationnelle>
```

Figure 8. Extrait des connaissances de classification pour les problèmes additifs

Pour que le système utilise le graphe de classification afin d'identifier la classe d'un problème, l'expert doit définir des règles de reformulation. Celles-ci permettent de calculer, à partir de l'énoncé du problème, les valeurs des attributs discriminants, afin de situer ce problème dans le graphe de classification. Ces règles constituent les connaissances de reformulation.

Connaissances de reformulation

Pour utiliser le graphe de classification, il faut posséder des connaissances permettant de déterminer les valeurs des attributs discriminants. Le moteur de résolution de SYRCLAD déterminera les valeurs d'attributs discriminants du problème, ainsi que les valeurs d'autres attributs du problème, qui ne sont pas discriminants pour le classement mais qui sont néanmoins utiles. À chaque attribut du problème est attaché un ensemble de règles si-alors qui permettent de conclure sur la valeur de cet attribut. C'est l'expert qui donne au système ces connaissances de reformulation du problème.

Dans une classe donnée, l'expert définit un attribut du problème si cet attribut a un sens pour les problèmes de cette classe. Il précise les règles qui permettent de trouver la valeur de cet attribut. La plupart du temps, ces attributs ne sont pas présents dans les modèles descriptifs des problèmes. Ces connaissances permettent alors de passer du modèle descriptif au modèle opérationnel, plus adapté à l'utilisation d'une technique de résolution.

Chaque règle est définie par :

- son nom ;
- un ensemble de prémisses portant sur des éléments du modèle descriptif du problème ou sur des attributs du problème qui pourront eux-mêmes nécessiter l'application de règles pour calculer leur valeur. Les prémisses peuvent également faire appel à des fonctions de calcul en Prolog ;

- un ensemble de conclusions permettant de calculer ou de modifier les valeurs d'attributs du problème. On y retrouve également des appels à des fonctions de calcul.

Par exemple, dans le domaine des dénombrements, le premier attribut discriminant à déterminer dans la classification est le type du problème (jointure, complémentaire, disjonction, répartition, spectre, liste avec position). Il existe dans SYRCLAD-dénombrements 16 règles qui permettent de trouver la valeur de cet attribut. Les prémisses de ces règles portent par exemple sur l'ensemble dans lequel on choisit (jeu de cartes, lettres, jetons), les catégories des objets de l'énoncé (couleur des jetons, hauteur des cartes, …), l'objet formé (une main de cartes, un mot…), les contraintes sur les éléments à dénombrer, etc. Il s'agit d'attributs du problème qui doivent eux-mêmes être déterminés par des règles analysant le modèle descriptif du problème.

Connaissances de résolution

Pour choisir une technique de résolution à employer, il faut savoir quelle technique est adaptée pour une classe de problèmes. Il s'agit d'une expertise difficile à acquérir : Chi a montré [Chi et al 81] que les novices possèdent parfois de bons critères de classification, mais qu'ils ne savent pas les relier aux différentes techniques de résolution possibles.

Dans SYRCLAD, l'expert attache donc à chaque classe opérationnelle du graphe de classification une technique de résolution.

Une technique de résolution est définie par :

- son nom ;
- un ensemble de données correspondant à des attributs du problème ;
- un ensemble d'éléments de solution ;
- une fonction de calcul permettant de déterminer les éléments de solution à partir des données.

Ces techniques de résolution sont spécifiques aux domaines. En dénombrement ou pour les problèmes additifs, il s'agit un plan de résolution type pour les exercices de cette classe, ainsi qu'une formule permettant de calculer la solution numérique. Pour la conjugaison française, il s'agit d'une part du processus permettant d'obtenir la partie du verbe fixe pour l'ensemble des personnes et d'autre part de la liste des terminaisons pour les six personnes.

3.2 Processus de résolution de problèmes

Pour réaliser un résolveur SYRCLAD-D, l'expert du domaine D fournit à SYRCLAD trois bases de connaissances issues des méta-modèles présentés en 3.1. Un utilisateur d'un résolveur SYRCLAD-D lui présente alors un modèle descriptif du problème à résoudre (cf. **Figure 6**). Afin de choisir une technique de résolution pertinente, SYRCLAD-D doit classer ce problème et en construire un nouveau modèle. Ces processus de classement et de construction du modèle opérationnel ont lieu en même temps, le graphe de classification permettant le classement, et les règles permettant la construction du modèle opérationnel. Mais les règles sont nécessaires au classement et le graphe est nécessaire pour guider la reformulation du problème. Il y a donc un seul processus, indépendant du domaine, que l'on appellera *opérationnalisation* du problème.

Le processus de classement prend en entrée un modèle descriptif D d'un problème P. En sortie, on obtient une classe C telle qu'il existe une instance M de C représentant un nouveau modèle de P. Pour classer le problème, SYRCLAD utilise le graphe de classification comme un guide. Il part de la racine du graphe, et affine le classement du problème petit à petit, en passant à une classe plus spécifique. Il essaie d'aller le plus loin possible vers les feuilles, au moins jusqu'à une classe opérationnelle. Il obtient ainsi la classe la plus spécifique selon laquelle les connaissances permettent de modéliser le problème.

Descendre dans une sous-classe revient à déterminer la valeur de l'attribut discriminant. Les règles permettant de déterminer sa valeur se déclenchent en chainage arrière, par nécessité. En effet, les règles permettant de conclure sur la valeur de cet attribut peuvent faire intervenir plusieurs autres attributs du problème, dont il faut alors déterminer la valeur. Lorsque l'on arrive dans une classe opérationnelle, on essaie quand même de descendre vers une sous-classe pour avoir un classement le plus précis possible. Il se peut que l'on ne puisse plus descendre, soit parce que l'on est arrivé à une feuille, soit parce que la valeur de l'attribut discriminant n'est pas de celles qui permettent de descendre vers les cas particuliers.

La construction du modèle opérationnel du problème a lieu pendant le classement de celui-ci. En effet, on a vu que pour classer un problème, on détermine la valeur d'attributs du problème ; ces nouveaux attributs sont ceux qui définissent le modèle opérationnel. Au début du classement, le modèle opérationnel est vide ; il est enrichi à chaque fois que l'on descend dans une classe plus spécifique. Un modèle est une description d'un problème suivant un certain point de vue, ici les points de vue sont des classes du graphe de classification.

Le résultat de l'opérationnalisation est un couple (C, M) : une classe à laquelle appartient le problème posé, et un nouveau modèle du problème. Le modèle opérationnel M est un problème qui se suffit à lui-même, et qui contient assez d'informations pour que l'on puisse le résoudre. On n'a plus besoin du modèle descriptif (c'est-à-dire de l'énoncé) pour terminer la résolution. Le modèle opérationnel instancie le problème par rapport à la classe. Il est constitué d'un ensemble d'attributs avec leurs valeurs, d'une part les attributs discriminants déterminés pendant la classification, et d'autre part des attributs dont les valeurs seront utilisées par la technique de résolution associée à la classe (par exemple des valeurs numériques qui seront utilisées pour des calculs, ou le radical d'un verbe pour la conjugaison).

Pour trouver une solution au problème, le résolveur SYRCLAD-D applique sur le modèle opérationnel M la technique de résolution associée à la classe C du problème.

3.3 Évaluation de la proposition

Les méta-modèles de connaissances de l'architecture SYRCLAD ont été évalués en vérifiant qu'ils permettent d'exprimer dans des domaines variés les connaissances d'une méthode de résolution de problème. Le moteur de raisonnement associé à ces méta-modèles a été évalué en vérifiant qu'il permet d'exploiter les modèles de connaissances de ces domaines et ainsi résoudre des problèmes.

Le premier domaine d'application de SYRCLAD a été celui des exercices de dénombrement au niveau de la classe de terminale scientifique. Par rapport au système expert réalisé pendant mon DEA [Guin et al 95], l'utilisation de SYRCLAD a permis une représentation plus

déclarative des connaissances nécessaires à la résolution des exercices de dénombrement. Le système SYRCLAD-dénombrements a été testé en résolvant 76 problèmes variés.

Par la suite, j'ai testé SYRCLAD sur le domaine des problèmes additifs proposés à l'école élémentaire, en me fondant sur une classification sur papier établie par des didacticiens. Le système SYRCLAD-add a été testé en résolvant 50 problèmes additifs, en utilisant une modélisation fondée sur la notion d'opérateur, adaptée aux connaissances des élèves. Plus tard, au sein du projet AMBRE, la robustesse du résolveur SRYCLAD-add a été démontrée puisqu'il permet de résoudre un très grand nombre de problèmes proposés aux élèves dans l'EIAH AMBRE-add (cf. chapitre A3).

J'ai ensuite étudié comment le système Bateleur (Nigro, 1995) choisit un plan de jeu au tarot, pour expliciter une classification cachée dans des règles. Le système SYRCLAD-tarot choisit un plan de jeu pour une trentaine de situations, et obtient les mêmes résultats que le système Bateleur, de manière plus explicite.

J'ai également étudié la résolution d'exercices de thermodynamique par le système Modélis (Tisseau, 1990), dans le but d'élaborer une classification. Le système SYRCLAD-thermodynamique résout une soixantaine d'exercices de thermodynamique en utilisant le graphe de classification ainsi construit. Le résolveur de Modélis n'avait pas de vision globale du problème et n'utilisait pas de connaissances cachées concernant une classification de problèmes. Le fait d'utiliser une classification de problèmes permet à SYRCLAD-thermodynamique de reformuler le problème en interprétant les données du problème afin de le présenter de manière structurée, en faisant ressortir les points qui seront importants pour la résolution.

Après ma thèse, l'architecture SYRCLAD a été appliquée à la résolution de certains problèmes de jeu d'échecs (cf. chapitre A2).

Nous l'avons également plus récemment appliquée à la conjugaison française. La difficulté dans ce domaine a été la construction du graphe de classification. En effet, les nombreux cas particuliers ont nécessité de construire un grand nombre de classes.

Ainsi, l'application de SYRCLAD à six domaines variés montre sa généricité. Elle pourrait ainsi être appliquée à tout domaine de résolution de problèmes dans lequel on peut expliciter une classification des problèmes qui permet de choisir une technique de résolution pertinente.

Conclusion

J'ai présenté dans ce chapitre l'architecture SYRCLAD, issue de mon travail de thèse. Il s'agit d'une architecture destinée à la résolution de problèmes dans laquelle les phases de modélisation et de résolution sont séparées et où le processus de modélisation est guidé par une classification des problèmes. J'ai ainsi proposé, d'une part, des méta-modèles permettant d'éliciter une classification de problèmes ainsi que des connaissances de reformulation et de résolution, et d'autre part, un moteur d'exploitation de ces connaissances. L'architecture SYRCLAD a été testée sur six domaines d'application.

Si un expert du domaine D souhaite utiliser SYRCLAD pour réaliser un résolveur de problèmes du domaine D (SYRCLAD-D), cet expert doit définir un langage de la logique du premier ordre qui permette de décrire les problèmes du domaine D, un graphe de

classification des problèmes du domaine D qui permette d'associer une technique de résolution à certaines classes de problèmes, des règles de reformulation permettant de modéliser un problème concret afin d'obtenir un modèle plus proche de la théorie du domaine D, et des technique de résolution efficaces pour certaines classes de problèmes.

Dans SYRCLAD, les connaissances sont exprimées de manière déclarative. Cela permet d'expliciter une démarche de résolution à un niveau connaissance. De plus, le fait de séparer la résolution d'un problème en différents processus (modélisation, classement, application d'une technique) permet de mieux distinguer les démarches et les types de difficultés propres à chaque processus. L'abstraction possède également l'avantage de permettre une réutilisabilité plus aisée des connaissances contenues dans la classification, et en particulier de faire évoluer l'expertise plus facilement et avec moins de risques d'erreurs. J'ai montré dans plusieurs domaines que l'on peut facilement ajouter des classes au graphe de classification.

L'architecture SYRCLAD permet de concevoir des systèmes SYRCLAD-D destinés à servir de résolveurs de référence dans des EIAH sur chacun des domaines D (cf. chapitres A3 et A4). Un système destiné à enseigner une stratégie de résolution fondée sur une classification des problèmes doit avoir accès à une représentation explicite de la classification et des processus mis en jeu. L'objectif est, dans chaque domaine et dans un cadre d'enseignement à un niveau donné, de pouvoir expliquer à un élève la démarche de résolution d'un élève-expert, et les méta-connaissances qu'elle utilise. Les explications seront d'autant plus faciles à donner que les connaissances seront explicitées. Chacun des systèmes SYRCLAD-D est adapté à un niveau d'apprentissage donné, et son comportement peut être fourni comme exemple à un apprenant du domaine. J'ai conçu l'architecture SYRCLAD pour qu'elle soit adaptée à des domaines où les énonces des problèmes correspondent à des situations concrètes, alors que le cours est donné sous forme de connaissances théoriques. Il faut pour résoudre ces problèmes effectuer une reformulation du problème pour obtenir un modèle adapté aux connaissances du cours. C'est cette phase de reformulation qui est difficile pour les élèves. SYRCLAD facilite l'expression des connaissances dans de tels domaines en permettant d'exprimer de manière déclarative une classification des problèmes et de distinguer connaissances de reformulation et connaissances de résolution.

Même si l'architecture SYRCLAD est issue de mon travail de thèse, sa présentation était nécessaire dans ce mémoire. Nous verrons en effet au chapitre A2 comment nous l'avons complétée afin de modéliser une démarche de résolution de problèmes associant classes de problèmes et cas. Nous verrons ensuite au chapitre A4 comment elle est utilisée pour soutenir l'enseignement de méthodes au sein des EIAH AMBRE, et au chapitre A5 comment elle est également utilisée pour la génération de problèmes.

Pour appliquer SYRCLAD à un domaine, l'expert du domaine doit fournir au système un langage de description des problèmes, un graphe de classification, des règles de reformulation et des techniques de résolution. L'élicitation des connaissances est un travail intrinsèquement difficile. De plus, le fait que ces connaissances doivent être exprimées en Prolog limite fortement l'utilisabilité du système et nécessite de fait la collaboration entre un expert et un informaticien. C'est pourquoi nous travaillons actuellement (cf. chapitre A6) à la définition d'un outil auteur qui permettra à l'expert d'éliciter *via* une interface les modèles de connaissances conformes aux méta-modèles de SYRCLAD.

Synthèse des contributions

Question de recherche : Modéliser les connaissances d'une méthode de résolution de problèmes.

- **Contributions théoriques** :
  - Méta-modèles de connaissances de classification, de reformulation et de résolution.
  - Processus de résolution associé.
- **Système développé** :
  - Architecture SYRCLAD, intégrant un moteur de résolution capable d'exploiter les modèles de connaissances conformes aux méta-modèles.
  - Définition des modèles de connaissances pour six domaines, permettant d'obtenir avec le moteur de résolution six résolveurs de problèmes.
- **Déploiement** : SYRCLAD-add (le résolveur pour les problèmes additifs) est intégré à l'EIAH AMBRE-add qui sera décrit au chapitre A3.

Pour en savoir plus

Pour avoir davantage de précisions sur l'architecture SYRCLAD, le lecteur intéressé pourra consulter l'article suivant :

Guin-Duclosson, N. (1999). SYRCLAD: une architecture de résolveurs de problèmes permettant d'expliciter des connaissances de classification, reformulation et résolution. *Revue d'Intelligence Artificielle*, *13*(2), 225–282.

Le mémoire de ma thèse est également disponible en ligne :

Guin, N. (1997). Reformuler et classer un problème pour le résoudre. L'architecture SYRCLAD et son application à quatre domaines. Université Paris 6, Paris. http://liris.cnrs.fr/nathalie.guin/Articles/thesengd.pdf

Publications liées

Conférences internationales

Guin, N. (1996a). Problem solving using classification of the domain problems. In *EXPERTSYS-96, Expert Systems Applications & Artificial Intelligence* (pp. 15–20). Marne la Vallée.

Guin, N. (1997a). Building a new model in order to solve a problem : use of classification. In *Fifteenth International Joint Conference on Artificial Intelligence* (p. 42). Nagoya (Japon).

Guin, N. (1997c). Changing the representation of a problem in order to solve it : use of classification. In B. Du Boulay & R. Mizoguchi (Eds.), *Artificial Intelligence in Education* (pp. 583–585). IOS Press.

Revues nationales

Guin-Duclosson, N. (1999). SYRCLAD : une architecture de résolveurs de problèmes permettant d'expliciter des connaissances de classification, reformulation et résolution. *Revue d'Intelligence Artificielle*, *13-2*.

Conférences nationales

Guin, N. (1995). Le classement de problèmes : une méthode de résolution de problèmes pour le module expert d'un EIAO. Application aux problèmes de dénombrement. In D. Guin, J.-F. Nicaud, & D. Py (Eds.), *Quatrièmes journées EIAO de Cachan* (pp. 113–124). Eyrolles.

Guin, N. (1996b). Résolution de problèmes à partir de la classification du domaine. In *Troisièmes rencontres nationales des jeunes chercheurs en intelligence artificielle (RJCIA)* (p. 264). Nantes.

Guin, N. (1997b). Changer de modèle pour résoudre un problème : utilisation de la classification. In M. Baron, P. Mendelshon, & J.-F. Nicaud (Eds.), *Cinquièmes journées EIAO de Cachan* (pp. 49–60). Hermès.

Guin, N. (1998). Utiliser une classification pour résoudre et décomposer des problèmes. In *Reconnaissance des Formes et Intelligence Artificielle* (pp. 135–144). Clermont-Ferrand.

CHAPITRE A2
MODÉLISER UN PROCESSUS DE RÉSOLUTION DE PROBLÈMES MOBILISANT CONJOINTEMENT DES CLASSES ET DES CAS

Résumé

Ce chapitre présente un travail pluridisciplinaire, mené en collaboration avec un chercheur en psychologie cognitive, dont l'objectif était de proposer un modèle psychologiquement valide d'un processus de résolution de problèmes mobilisant conjointement des connaissances abstraites de type classes et des connaissances contextualisée de type cas.

J'y décris l'architecture CASCADE, qui donne à un expert la possibilité d'éliciter les connaissances permettant, dans un domaine donné, d'obtenir un résolveur de problèmes mobilisant conjointement ces deux types de connaissances. Réutilisant l'architecture SYRCLAD, cette nouvelle architecture est constituée, d'une part, de méta-modèles de connaissances et, d'autre part, d'un moteur d'exploitation de modèles de connaissances conformes à ces méta-modèles.

INTRODUCTION 54

1 CADRE DES RECHERCHES : LE PROJET CASES 54

2 QUESTION DE RECHERCHE : MODÉLISER UN PROCESSUS DE RÉSOLUTION DE PROBLÈMES MOBILISANT CONJOINTEMENT DES CLASSES ET DES CAS 54

3 CONTRIBUTION : L'ARCHITECTURE CASCADE 56

3.1 EXPÉRIMENTATIONS PRÉLIMINAIRES 56
3.2 MÉTA-MODÈLES DE CONNAISSANCE 58
3.3 PROCESSUS DE RAISONNEMENT 61
3.4 ÉVALUATION DE L'ARCHITECTURE 62

CONCLUSION 63

SYNTHÈSE DES CONTRIBUTIONS 64

POUR EN SAVOIR PLUS 64

PUBLICATIONS LIÉES 64

Introduction

Dans l'enseignement, l'apprentissage passe souvent par la présentation aux élèves de problèmes corrigés. De nombreuses expérimentations de psychologie cognitive consistent à proposer aux élèves des problèmes corrigés qu'ils doivent analyser et expliquer à haute voix, leurs verbalisations faisant l'objet d'une étude. Les travaux sur l'analogie issus de (Gick & Holyoak, 1980, 1983) montrent qu'il faut présenter plusieurs exemples pour que l'élève puisse construire une structure abstraite de connaissances, appelée *schéma*, qui est analogue à la notion de classe de problèmes utilisée dans le chapitre A1. Didierjean et Cauzinille-Marmèche montrent l'existence de différences individuelles parmi les sujets analysant des corrigés-types (Didierjean & Cauzinille-Marmèche, 1997).

Certains élèves cherchent, pendant l'analyse de ces corrigés, à anticiper les étapes de la résolution et créent ainsi des attentes déçues lorsque le corrigé ne correspond pas à leurs prévisions. Ils détectent ainsi comment résoudre certaines étapes spécifiques et construisent des *cas*, en mémorisant la résolution d'un problème spécifique. Ces élèves résolvent bien les problèmes proches des exemples, mais n'arrivent pas à résoudre des problèmes dont l'aspect contextuel est éloigné. D'autres élèves ont des verbalisations qui font apparaitre qu'ils comparent les exemples entre eux. Ces élèves construisent des schémas et résolvent bien les problèmes éloignés des exemples. C'est cet apprentissage que l'on souhaite déclencher pour enseigner des méthodes, et nous verrons au chapitre suivant comment nous l'avons mis en œuvre dans le projet AMBRE. Didierjean et Cauzinille-Marmèche observent également que certains élèves utilisent à la fois des schémas et des cas. Ces élèves obtiennent les meilleurs résultats au moment de résoudre, mais la charge mentale requise semble plus importante.

Le travail décrit dans ce chapitre avait pour objectif une modélisation psychologiquement valide d'une utilisation conjointe en résolution de problèmes de connaissances abstraites (de type classes de problèmes) et de connaissances contextualisées (de type cas).

1 Cadre des recherches : le projet CASES

Ce travail a été mené dans le cadre du projet CASES (Connaissances Abstraites et/ou Spatialement Évoquées en Situation), financé par une ACI Cognitique entre 1999 et 2002. Il est le fruit d'une collaboration avec André Didierjean, qui était alors maître de conférences en psychologie cognitive à l'Université de Provence, et qui est maintenant professeur en psychologie cognitive à l'Université de Franche-Comté.

2 Question de recherche : modéliser un processus de résolution de problèmes mobilisant conjointement des classes et des cas

L'architecture SYRCLAD décrite dans le chapitre A1 permet de modéliser une méthode de résolution de problèmes, c'est-à-dire un processus de résolution fondé sur des classes de problèmes. Nous souhaitions dans le projet CASES définir des modèles de connaissances et de raisonnement permettant l'utilisation en résolution de problèmes de ces connaissances abstraites, mais aussi de connaissances plus contextualisées issues de l'étude d'exemples.

Par rapport à l'approche générale décrite sur la **Figure 2** du chapitre 1, la question est ici la même que dans le chapitre A1 (cf. **Figure 9**). Il s'agit de permettre à un expert disposant, dans un domaine donné, de connaissances de résolution de problèmes, de les expliciter, afin que ces connaissances puissent être utilisées par un système informatique. Cette élicitation est guidée par un méta-modèle décrivant la nature des connaissances utilisées pour résoudre les problèmes. Par rapport à SYRCLAD, on veut ici modéliser à la fois les connaissances de la méthode et les connaissances issues de l'expérience des sujets. Un moteur de raisonnement conçu conjointement permet d'exploiter les modèles de connaissances conformes au méta-modèle, ce qui permet de disposer de résolveurs de problèmes fonctionnant en utilisant conjointement les deux types de connaissance.

Le méta-modèle de connaissance ainsi que le moteur de raisonnement associé constituent l'architecture CASCADE (CAS et Connaissances Abstraites : une Double Expertise) et sont décrits dans la section suivante.

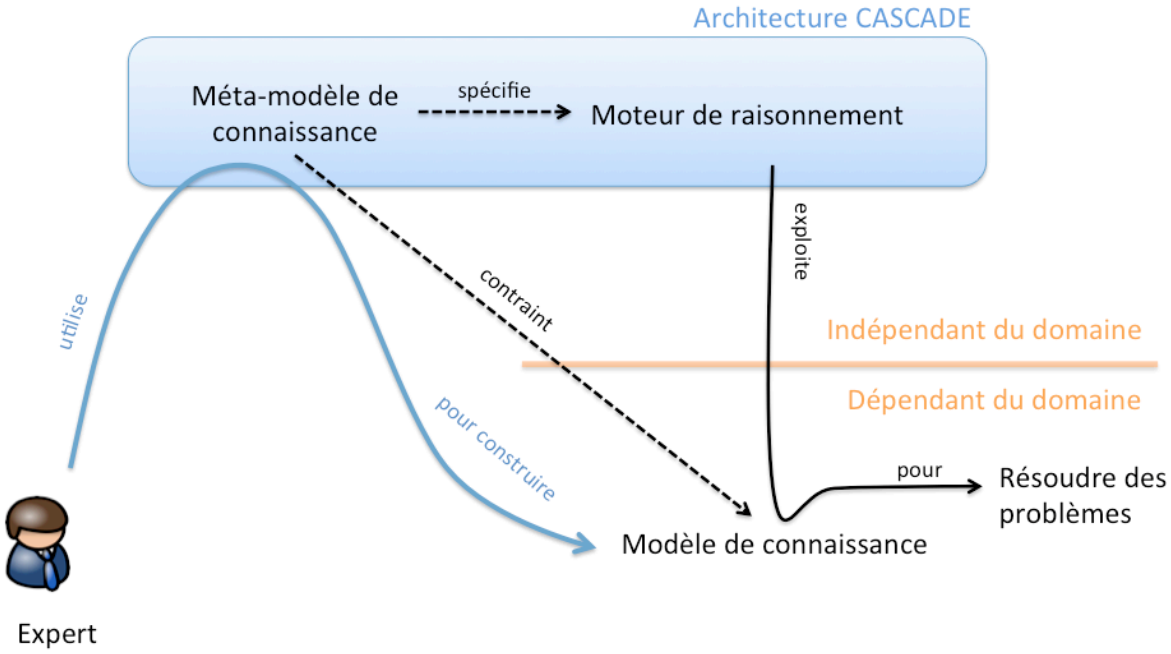

Figure 9. Instanciation de l'approche générale présentée dans le chapitre 1 pour l'élicitation de connaissances de résolution de problèmes combinant des classes et des cas.

Pour concevoir cette architecture, nous nous sommes appuyés sur des expérimentations destinées à mieux appréhender l'utilisation de ces deux types de connaissances par le sujet humain en situation de résolution de problèmes. À partir des résultats de ces expérimentations, nous avons conçu un modèle de cette double expertise, que j'ai mis en œuvre dans l'architecture de résolveurs de problèmes CASCADE fondée sur deux types de connaissances : les classes et les cas.

3 Contribution : l'architecture CASCADE

Lorsque des humains étudient des exemples de problèmes résolus, quelle est la nature des connaissances qu'ils construisent ?

Des travaux historiques en psychologie cognitive avancent que les sujets pourraient élaborer une représentation abstraite de la structure de résolution des exemples sources, débarrassée des éléments de contexte spécifiques à chacun des problèmes. Ils utiliseraient ensuite cette connaissance abstraite pour résoudre de nouveaux problèmes (Cummins, 1992 ; Gick & Holyoak, 1983).

Selon une deuxième hypothèse, les sujets élaboreraient une connaissance conservant la plupart des éléments de contexte des exemples étudiés, y compris des éléments contextuels non pertinents pour la résolution. Face au problème cible à résoudre, le sujet tenterait d'adapter cette connaissance spécifique. Cette forme de raisonnement par analogie, nommée Raisonnement à Partir de Cas (Kolodner, 1993), est couramment utilisée par les novices (Didierjean, 2003 ; Ross, 1987, 1989), mais aussi par les experts, ce mode de raisonnement s'avérant très économique (S. Blessing & Ross, 1996).

Une troisième hypothèse consiste à envisager des modèles mixtes d'apprentissage et de résolution. Les sujets pourraient, face à un exemple particulier, élaborer plusieurs représentations de la situation source : une connaissance abstraite détachée des éléments de contexte et une connaissance contextualisée. Deux versions de cette hypothèse peuvent être envisagées :

- Version 1 : les sujets élaborent une connaissance abstraite (classe) et stockent la situation source (cas) dans sa spécificité, mais seuls les éléments de structure sont adaptés sur la cible. Les éléments de contexte servent uniquement d'indice de récupération lors de la phase d'évocation.
- Version 2 : les sujets stockent les deux types de connaissances et peuvent adapter sur la cible l'une ou l'autre connaissance selon la distance entre le problème source et le problème cible. Lorsque le problème à résoudre est proche de la source, les sujets pourraient adapter la connaissance spécifique en mettant en œuvre un raisonnement à partir de cas. Lorsque le problème à résoudre en est plus éloigné, ils auraient alors recours à la connaissance abstraite (classe).

3.1 Expérimentations préliminaires

La distinction entre ces deux versions de la troisième hypothèse nécessite de considérer séparément l'analogie inter-domaines de l'analogie intra-domaine, qui est celle qui nous intéresse. Si pour l'analogie inter-domaines, c'est la première version qui semble être privilégiée (Gentner, Rattermann, & Forbus, 1993; Holyoak & Koh, 1987), la question est posée pour l'analogie intra-domaine. Pour y répondre, André Didierjean a créé une situation expérimentale où les sujets étudiaient des exemples pour lesquels l'extraction des principes généraux en œuvre est très simple. L'objectif était de mesurer dans un cadre intra-domaine si les effets de contexte sur la résolution perdurent après la généralisation. Dans une première phase, les sujets étudiaient des exemples illustrant des coups classiques au jeu d'échecs dont la compréhension est élémentaire (problèmes d'alignement et de fourchette). Ils passaient dans une deuxième phase une série de problèmes qui nécessitaient la mise en œuvre de ces coups, mais qui étaient plus ou moins similaires à la source.

Pour chaque exemple étudié, le sujet voyait trois problèmes différents :

- Un problème très *proche* de l'exemple de départ.
- Un problème *moins proche*. Celui-ci était construit à partir du problème *proche* par une symétrie axiale.
- Enfin les sujets passaient un problème *éloigné*. Celui-ci était construit à partir du problème *moins proche* par une nouvelle symétrie axiale.

Tous les problèmes étaient très simples et la plupart des sujets après avoir étudié un exemple n'ont aucune difficulté à les résoudre. Les trois hypothèses énoncées préalablement amènent cependant à des prédictions différentes en ce qui concerne les temps de résolution des différents problèmes :

- Si les sujets élaborent à partir de l'exemple source une connaissance abstraite et n'utilisent pas les aspects spécifiques de la source (comme le coin de l'échiquier où doit se jouer la pièce), on peut s'attendre à ce qu'ils aient un temps de résolution identique quelle que soit la partie de l'échiquier où se situe le problème à résoudre.
- Si toutefois les sujets élaborent uniquement une connaissance contenant des informations de contexte, et qu'ils tentent de l'adapter sur l'ensemble des problèmes, le temps de réponse devrait augmenter avec l'augmentation de la distance entre problème source et problème cible : le problème *proche* devrait alors être résolu plus vite que le problème *moins proche*, qui devrait lui-même être résolu plus vite que le problème *éloigné*.
- Si les sujets élaborent deux types de connaissances différentes, à savoir élaborent un schéma mais conservent aussi des aspects spécifiques de la source, le temps de réponse au problème *proche* devrait être plus rapide que le temps de réponse aux deux autres problèmes, et les temps de réponse aux problèmes *moins proche* et *éloigné* devraient être équivalents. Ceci dans le cadre des deux versions de notre troisième hypothèse.

Les résultats obtenus montrent que les sujets réussissent plus vite les problèmes *proches* que les autres problèmes construits par symétrie à partir du problème *proche* (qu'il s'agisse des problèmes *moins proches* (une symétrie) ou *éloignés* (deux symétries), aucune différence n'étant observée dans les temps de résolution de ces deux dernières catégories.

Ces résultats sont donc en faveur de l'hypothèse selon laquelle les sujets élaborent deux types de connaissances : un schéma abstrait relatif au principe du coup à jouer et une connaissance contextualisée intégrant des traits perceptifs spécifiques. Face aux problèmes les plus proches des exemples, la connaissance perceptive interviendrait, soit en étant adaptée sur le problème à résoudre (cf. notre troisième hypothèse, deuxième version), soit en accélérant le recours à la connaissance abstraite (cf. notre troisième hypothèse, première version). Face aux problèmes plus éloignés, seule la connaissance abstraite serait mise en œuvre.

Afin de choisir entre les deux versions de la troisième hypothèse, une expérimentation complémentaire a été réalisée, en complément des travaux de Sander et Richard (Sander & Richard, 1997) et de Blessing et Ross (S. Blessing & Ross, 1996), qui penchent pour la deuxième version. Dans cette expérience, nous avons présenté à des sujets des séries d'exemples illustrant les coups utilisés dans la première expérience, mais nous avons également introduit dans les exemples un élément non pertinent pour la résolution. Les résultats obtenus montrent un effet de l'élément contextuel introduit dans l'exemple. Les

sujets réutilisent de manière systématique cet élément (non nécessaire pour la résolution) dans les problèmes à résoudre. De plus, les sujets utilisent cette connaissance non seulement sur le problème *proche*, mais également sur les problèmes *moins proches* et *éloignés*. Ceci suggère que la connaissance contextuelle influence la résolution des problèmes *moins proches* et *éloignés*, conjointement au recours à la connaissance abstraite.

Suite aux résultats de ces expérimentations, nous proposons un méta-modèle des connaissances de résolution d'un expert. Ce modèle permet d'envisager le recours conjoint à des connaissances abstraites et à des connaissances contextualisées en résolution de problèmes. Nous avons défini l'architecture CASCADE (CAS et Connaissances Abstraites : une Double Expertise), qui permet de représenter de telles connaissances et de les mettre en œuvre pour la résolution de problèmes. Cette architecture a été utilisée pour implémenter un système qui résout les problèmes simples d'échecs utilisés dans les expérimentations précédemment décrites : pour chaque problème, le résolveur propose une solution fondée soit sur ses connaissances contextualisées soit sur ses connaissances abstraites, selon la similarité entre le problème rencontré et les problèmes de sa base de connaissances.

Je décris dans un premier temps comment sont modélisées les connaissances puis je présente le processus de raisonnement du moteur utilisant les deux types de connaissances pour la résolution de problèmes.

3.2 Méta-modèles de connaissance

Dans l'architecture CASCADE, les connaissances abstraites de type classes sont représentées grâce à l'architecture SYRCLAD (cf. chapitre A1). Nous avons défini une hiérarchie de classes de problèmes pour les problèmes d'alignement aux échecs étudiés dans le projet CASES (cf. partie supérieure de la **Figure 10**). Lorsque la technique de résolution associée à la classe du problème conduit à plusieurs solutions équivalentes (par exemple lorsque la pièce peut être placée sur plusieurs cases), celle qui est choisie est la plus proche de celles figurant dans les cas déjà résolus qui sont attachés à la classe du problème. On illustre ainsi l'influence de la connaissance contextuelle sur une résolution utilisant la connaissance abstraite, comme montré par les expérimentations décrites ci-dessus.

Pour les connaissances contextualisées, nous utilisons une représentation des connaissances adaptée au Raisonnement à Partir de Cas (RàPC). Ce paradigme permet d'une part de modéliser le raisonnement (Kolodner, 1993) et d'autre part de réaliser des systèmes d'Intelligence Artificielle permettant une interprétation et/ou une résolution de problèmes (Aamodt & Plaza, 1994 ; Leake, 1996).

Le RàPC, qui modélise un raisonnement par analogie intra-domaine, peut être décrit par un ensemble d'étapes séquentielles que l'on représente souvent par un cycle. La première étape de ce cycle consiste à décrire le problème à résoudre selon des *traits* (ou *indices*) pour *élaborer* le *cas cible*. L'étape suivante consiste à retrouver un cas similaire au cas cible, qu'on appelle *cas source*. Cette étape de *remémoration* utilise les indices des deux cas pour les comparer et réaliser un *appariement* du cas cible au cas source. Il s'agit ensuite *d'adapter* la solution (ou le processus ayant mené à cette solution) au cas cible. Après cette étape d'adaptation, la solution proposée pour le cas cible est testée et éventuellement *révisée*. Enfin, le cas cible (avec sa solution) est *mémorisé* dans la base de cas en fonction de ses indices ; il devient ainsi un cas source potentiel pour un problème futur.

Dans le méta-modèle que nous proposons, les connaissances contextualisées sont représentées par des cas. Un cas est défini par :

- son identifiant
- le groupement de cas auquel il appartient
- la liste de ses attributs
- sa solution

Figure 10. Modèle des connaissances de l'expert pour les problèmes d'alignement aux échecs

L'étape d'élaboration du cycle de RàPC revient pour nous à déterminer les valeurs d'indices de surface. En ce qui concerne le domaine des échecs, ces indices sont liés à la représentation spatiale du problème. L'étape de remémoration consiste à trouver un cas source semblable en termes d'indices de surface.

Pour une remémoration efficace, on utilise une hiérarchie de groupements de cas fondée sur ces indices de surface. Cette hiérarchie est définie *a priori* par l'expert, et seules sont actives (et donc utilisables pour la remémoration) les branches qui conduisent à des cas (cf. partie inférieure de la **Figure 10**). On définit alors un *cas proche* (au sens de l'expérimentation décrite en 3.1) comme un cas qui appartient au même groupement de cas que le cas cible. Un cas *moins proche* est un cas appartenant à un sous-arbre de la hiérarchie dont la racine est définie par l'expert du domaine. Pour le domaine des problèmes d'échecs que nous traitons, c'est le niveau des directions d'alignement qui définit les cas *moins proches* : un cas du groupement *vertical gauche* est un cas *moins proche* d'un cas du groupement *vertical droite*. Les cas appartenant aux sous-arbres issus du groupement *horizontal* ou du groupement *diagonale* sont des cas *éloignés*.

L'étape de remémoration consiste alors à classer le cas cible dans la hiérarchie de groupements de cas, puis à récupérer les cas proches et moins proches grâce à la hiérarchie de groupements de cas, comme défini ci-dessus.

Pour préparer l'étape d'adaptation, la fin de l'étape de remémoration consiste à classer les cas sources potentiels selon leur degré de similarité avec le cas cible (il s'agit de l'étape d'appariement). La mesure de similarité donne ici priorité aux cas qui pourront être adaptés facilement. Ainsi, on préfère sélectionner un cas *moins proche* si sa solution peut être adaptée de manière certaine pour le cas cible. Par exemple, on pourra préférer un cas source du groupement *horizontale basse* à un cas source du groupement *horizontale haute* même si le cas cible appartient au groupement *horizontale haute*, quand la solution du premier peut s'adapter de manière certaine au cas cible (par exemple s'il y a par analogie "plus de place" pour la solution éventuelle du cas cible que dans le cas source).

La solution du cas source sélectionné est adaptée au cas cible. Pour le domaine des échecs, on applique au roi du cas cible le vecteur (roi noir, pièce blanche) du cas source. Si cette solution n'est pas issue d'une adaptation "certaine", elle est testée puis corrigée dans des cas simples (par exemple par une rectification de la norme du vecteur). Si cette adaptation n'aboutit pas, on essaie d'adapter le prochain cas source dans la liste ordonnée des cas candidats.

La **Figure 11** décrit la partie de l'architecture CASCADE dédiée aux connaissances contextualisées (l'architecture SYRCLAD présentée au chapitre A1 permettant de représenter les connaissances abstraites). Le niveau *domaine* contient des modèles de connaissance spécifiques au domaine, qui doivent être définis par l'expert du domaine conformément aux méta-modèles évoqués ci-dessus. Il s'agit des règles permettant de trouver la valeur des attributs de surface du problème servant à la remémoration, de la hiérarchie de groupements de cas, des connaissances d'appariement permettant d'évaluer la distance entre les cas, des connaissances destinées à adapter la solution d'un cas et à évaluer la solution ainsi produite, ainsi que les attributs du problème à utiliser pour mémoriser un nouveau cas.

Le niveau *CASCADE* est le moteur du système, indépendant du domaine ; il contient des processus qui utilisent les modèles de connaissances du domaine. Le niveau *problème* décrit comment le système résout par RàPC un problème posé par l'utilisateur.

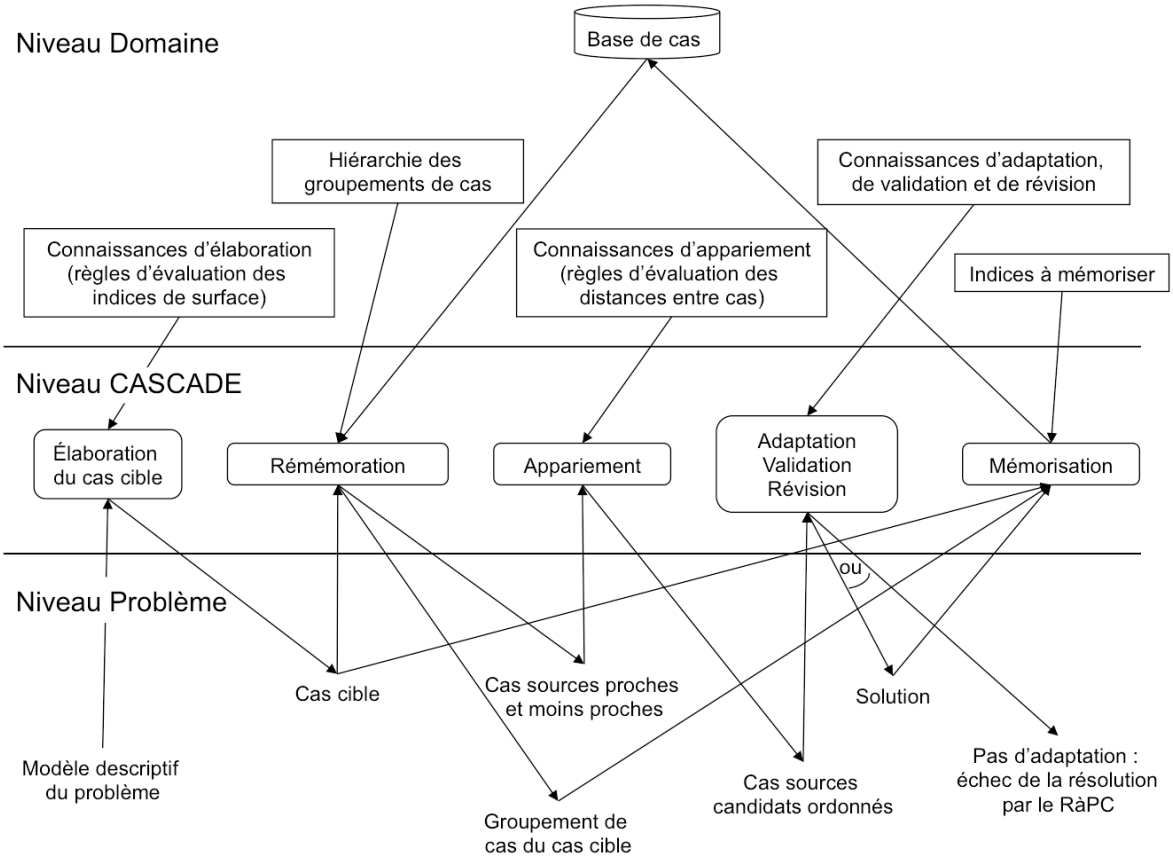

Figure 11. Méta-modèles des connaissances contextualisées et leur utilisation dans CASCADE pour résoudre des problèmes.

3.3 Processus de raisonnement

Le processus de raisonnement permettant d'exploiter les connaissances abstraites est celui de SYRCLAD, présenté au chapitre A1. Le processus de raisonnement permettant d'exploiter les connaissances contextualisées est celui du RàPC, pour lequel j'ai décrit en 3.2 l'implémentation que j'en ai effectuée. Voyons à présent la manière dont ces deux mécanismes sont utilisés conjointement par le moteur de l'architecture CASCADE.

Parce qu'il est montré que les experts tendent à utiliser préférentiellement lorsqu'ils le peuvent des connaissances spécifiques pour la résolution de problèmes familiers (S. Blessing & Ross, 1996), nous avons choisi de privilégier le RàPC par rapport au raisonnement par classement. Ainsi, dans le modèle que nous proposons, résoudre un problème consiste d'abord à utiliser un raisonnement à partir de cas ; si cette démarche n'aboutit pas, on utilise alors le raisonnement par classement.

Nous avons vu que dans la démarche de RàPC, nous distinguons des adaptations de cas "certaines". Lorsque le RàPC débouche sur une telle adaptation, on distingue deux possibilités :

- Si le cas source est un cas *proche* du cas cible (i.e. les deux cas appartiennent au même groupement de cas), le cas cible n'est pas mémorisé car il est jugé trop semblable au cas source.
- Si le cas source est un cas *moins proche* (i.e. il appartient à un groupement de cas "voisin" de celui du cas cible), le cas cible est mémorisé.

Lorsque l'adaptation n'est pas certaine, la solution issue de cette adaptation est vérifiée. Si la solution est validée, le cas est mémorisé.

Lorsque le RàPC n'a pas débouché sur une solution satisfaisante, on utilise le raisonnement par classement pour résoudre le problème, raisonnement qui aboutit toujours à une solution valide. Les raisons d'un échec du RàPC peuvent être d'une part que l'on n'a pas trouvé de cas source (proche ou moins proche), et d'autre part que pour les cas sources retenus, aucun n'a permis une adaptation certaine, et les adaptations peu sûres ont toutes abouti à des solutions invalidées ou à des solutions négatives (i.e. « il n'y a pas de solution »), donc non validables. Les problèmes résolus par classement sont systématiquement mémorisés ; ils illustrent en effet une situation qui est nouvelle puisqu'elle n'a pas pu être résolue par analogie.

Mémorisation d'un problème résolu

Lorsque l'on a trouvé une solution au problème, il faut le mémoriser en l'attachant à la hiérarchie de classes et à la hiérarchie de groupements de cas. Cette étape de mémorisation est indispensable puisqu'elle construit la base de cas qui sera par la suite utilisée pour résoudre d'autres problèmes par RàPC. De plus, c'est la mémorisation des problèmes qui permet d'obtenir une représentation conjointe des connaissances abstraites et des connaissances contextualisées, en fusionnant les deux parties du modèle.

Pour attacher le problème résolu à la hiérarchie de classes, lorsque le problème a été résolu par classement, on l'attache à la classe opérationnelle déterminée pendant le classement. Lorsque le problème a été résolu par RàPC, la solution est "certaine" ou a été validée et l'on peut donc attacher le cas cible à la classe opérationnelle dont relève le cas source.

Pour attacher le problème résolu à la hiérarchie de groupements de cas, on parcourt la hiérarchie à partir de sa racine, et on compare les indices de surface du problème à ceux de la hiérarchie. Lorsque le problème a été résolu par RàPC, la plupart de ces indices ont été déterminés dans l'étape d'élaboration ; lorsqu'il a été résolu par classification, ces indices doivent être déterminés pendant cette étape de mémorisation. Le parcours de la hiérarchie de groupements de cas peut amener à "activer" certaines branches prédéfinies qui n'étaient pas actives (par exemple « verticale gauche » sur la **Figure 10**).

3.4 Évaluation de l'architecture

Le fonctionnement de l'architecture CASCADE est, par construction, cohérent avec les résultats expérimentaux : dans ces résultats, les temps sont plus courts pour les problèmes proches que pour les autres problèmes. Le processus de résolution d'un problème par un résolveur CASCADE-D comporte deux étapes : une première étape de recherche de cas, puis éventuellement une deuxième étape d'application de la connaissance abstraite. Pour les

problèmes proches, la première étape suffit ; pour les autres problèmes, la première étape est complétée par la deuxième.

Conclusion

Ce chapitre décrit l'architecture CASCADE, qui permet à un expert de définir des modèles de connaissances permettant d'obtenir dans un domaine donné un résolveur de problèmes capable d'utiliser un raisonnement à partir de cas où un raisonnement fondé sur des classes de problèmes.

Ce travail a d'une part donné lieu à une proposition visant à fournir plusieurs types d'aide à l'apprenant dans un environnement d'aide à la résolution de problèmes (Guin-Duclosson & Nova, 2002). En effet, on peut choisir de présenter à l'apprenant soit le schéma à appliquer pour résoudre le problème, soit un problème analogue qu'il a déjà rencontré. Ce choix peut être effectué en tenant compte des aides précédemment fournies à l'apprenant et qui ont été efficaces, mais aussi des recommandations de l'enseignant, qui voudra peut-être amener un apprenant qui utilise dans un premier temps de manière préférentielle le raisonnement par analogie à utiliser davantage des schémas, afin qu'il acquière les connaissances abstraites du domaine. Il s'agit alors de personnaliser l'aide à l'apprenant, en fonction d'une stratégie définie par l'enseignant, thématique qui sera développée dans la partie B de ce mémoire.

Ce travail sur l'architecture CASCADE a d'autre part servi d'inspiration à la conception du cycle AMBRE, qui sera présenté au chapitre suivant. En effet, le principe du cycle AMBRE est de déclencher chez l'apprenant un raisonnement à partir de cas, dans le but de lui faire acquérir des schémas. Nous verrons également dans le chapitre A4 que pour soutenir l'activité de l'élève, le système doit disposer de modèles de connaissance que l'on trouve déjà dans CASCADE, comme les connaissances sur l'analogie entre problèmes.

Enfin ce travail sur l'architecture CASCADE est également une source d'inspiration pour nos travaux actuels sur un outil auteur pour les EIAH AMBRE. En effet, pour acquérir auprès de l'expert le graphe de classification, nous envisageons de lui permettre de décrire des cas, et que le système lui propose des classes généralisant ces cas.

Synthèse des contributions

Question de recherche : Modéliser un processus de résolution de problème combinant l'utilisation de classes et de cas.

- **Contribution théorique** :
  - Méta-modèles de connaissances contextualisées
  - Processus de résolution de problèmes combinant les connaissances de classification et les connaissances contextualisées.
- **Système développé** :
  - Architecture CASCADE, incluant un moteur de résolution de problèmes.
  - Élicitation des connaissances pour les problèmes d'alignement et de fourchette aux échecs, permettant d'obtenir avec le moteur de résolution un résolveur pour ces types de problèmes.
- **Co-auteur** : André Didierjean

Pour en savoir plus

Didierjean, A., Duclosson, N., & Marmèche, E. (2005). Utilisation conjointe de connaissances abstraites et de connaissances contextualisées en résolution de problèmes d'échecs. In C. Thinus-Blanc & J. Bullier (Eds.), *Agir dans l'espace*. Editions de la MSH.

Publications liées

Chapitre d'ouvrage

Didierjean, A., Duclosson, N., & Marmèche, E. (2005). Utilisation conjointe de connaissances abstraites et de connaissances contextualisées en résolution de problèmes d'échecs. In C. Thinus-Blanc & J. Bullier (Eds.), *Agir dans l'espace*. Editions de la MSH.

Conférence nationale

Didierjean, A., & Guin-Duclosson, N. (2001). Utilisation conjointe de connaissances abstraites et de connaissances contextualisées en résolution de problèmes. In *ARCo'2001 - Colloque Interdisciplinaire en Sciences Cognitives* (pp. 333–344).

CHAPITRE A3
AMBRE : ENSEIGNER UNE MÉTHODE À PARTIR D'EXEMPLES

Résumé

Ce chapitre présente le principe des EIAH AMBRE, dont l'objectif est d'enseigner des méthodes de résolution de problèmes. J'y décris le cycle AMBRE, un processus d'apprentissage qui propose à l'apprenant d'étudier des problèmes-types résolus puis de résoudre de nouveaux problèmes par analogie avec ces problèmes-types.

Ce chapitre décrit également comment nous avons conçu des EIAH AMBRE mettant en œuvre ce principe, en collaboration avec des chercheurs en psychologie cognitive et en didactique des mathématiques. Il présente en particulier l'EIAH AMBRE-add, appliqué au domaine des problèmes additifs, et comment nous avons évalué son impact sur l'apprentissage de la méthode.

INTRODUCTION 66

1 CADRE DES RECHERCHES : LE PROJET AMBRE 66

2 QUESTION DE RECHERCHE : COMMENT ENSEIGNER UNE MÉTHODE ? 67

3 CONTRIBUTION : LE CYCLE AMBRE 67

4 APPLICATIONS : LES EIAH AMBRE 69

4.1 PREMIER PROTOTYPE ET PREMIÈRE ÉVALUATION 69
4.2 ÉTUDES EN PSYCHOLOGIE COGNITIVE ET RECOMMANDATIONS POUR LE PROJET AMBRE 70
4.3 L'EIAH AMBRE-ADD 71

5 ÉVALUATION DU CYCLE AMBRE 75

CONCLUSION ET PERSPECTIVES 77

SYNTHÈSE DES CONTRIBUTIONS 78

POUR EN SAVOIR PLUS 78

PUBLICATIONS LIÉES 79

Introduction

Pour concevoir un EIAH, Rogalski propose de partir des connaissances telles qu'elles devraient fonctionner chez l'apprenant après l'apprentissage (Rogalski, 1994). Il précise que le résolveur d'un logiciel "donneur de leçons de méthodes" doit fonctionner selon les méthodes qu'il veut enseigner, et non pas selon des méthodes expertes du domaine concerné. Grâce à l'architecture SYRCLAD, il est possible de disposer d'un résolveur mettant en œuvre la méthode que l'on souhaite enseigner.

Toutefois, si l'objectif est d'enseigner une méthode, même s'il existe un système informatique capable de l'appliquer, il n'est pas forcément souhaitable de présenter explicitement la méthode à l'apprenant. En effet, dans certains domaines, les termes définissant les classes de problèmes et les techniques de résolution ne sont pas utilisés institutionnellement et ne sont pas connus des apprenants. De plus, il semble préférable que l'apprenant soit actif et se construise sa propre méthode.

C'est pourquoi, suite au projet CASES, j'ai proposé un processus inspiré du raisonnement à partir de cas (RàPC) et de l'apprentissage à partir d'exemples pour permettre à l'apprenant d'acquérir une méthode de résolution de problèmes dans un domaine donné. Ce processus, le cycle AMBRE, est présenté dans la section 3 de ce chapitre. La section 4 décrit les EIAH mis en œuvre en suivant ce principe, et en particulier l'EIAH AMBRE-add appliqué au domaine des problèmes additifs à l'école élémentaire. La section 5 présente l'évaluation du cycle AMBRE et en particulier son impact sur l'apprentissage.

1 Cadre des recherches : le projet AMBRE

J'ai commencé à travailler sur le projet AMBRE (Apprentissage de Méthodes Basé sur le Raisonnement à partir de l'Expérience) en 2000 et les travaux se poursuivent actuellement.

En 2000-2001, suite au projet CASES, j'ai posé les bases du projet AMBRE, et en particulier défini le cycle AMBRE qui est présenté dans ce chapitre. Sandra Nogry, dans le cadre de son DEA en 2001, a réalisé et évalué un premier prototype d'un EIAH AMBRE appliqué au domaine des dénombrements au niveau de la terminale scientifique : AMBRE-dénombrement.

Suite à ce travail, j'ai obtenu une bourse Cognitique sur le projet AMBRE, pour financer la thèse en Sciences Cognitives de Sandra Nogry, avec un co-encadrement en psychologie cognitive assuré par André Didierjean. Dans le même temps, j'ai obtenu un financement du projet AMBRE par le Programme interdisciplinaire STIC-SHS « Société de l'Information » du CNRS, pour les années 2001-2005.

Ce double financement a tout d'abord permis à Sandra Nogry de réaliser des expérimentations complémentaires en psychologie cognitive afin de mieux comprendre comment induire certains processus d'apprentissage. Suite à la quasi-disparition des dénombrements du programme de terminale scientifique, j'ai choisi un nouveau domaine d'application : celui des problèmes additifs à l'école élémentaire. Le financement du programme interdisciplinaire nous a permis de concevoir et développer l'EIAH AMBRE-add, avec les contributions de Sandra Nogry, Stéphanie Jean-Daubias, François Fallet-Kahn et

Philippe Daubias. Sandra Nogry a évalué pendant sa thèse l'impact de cet EIAH sur l'apprentissage.

Une version web de l'EIAH AMBRE-add a récemment été développée par Michaël Souvy, grâce au soutien du LIRIS.

2 Question de recherche : comment enseigner une méthode ?

La question posée ici est celle des activités à proposer à l'élève pour l'amener à acquérir la méthode de résolution de problèmes que l'on souhaite lui enseigner. On ne retrouve donc pas l'approche générale décrite dans le chapitre 1, où il s'agit de définir des méta-modèles qui permettront l'élicitation de connaissances *via* la définition de modèles conformes à ces méta-modèles. Cette approche sera reprise dans le chapitre suivant qui décrira les systèmes à base de connaissances sur lesquels s'appuient les EIAH AMBRE. En effet, ici ce n'est pas le système qui doit acquérir des connaissances, mais l'apprenant.

Pour aider l'apprenant à acquérir la méthode de résolution de problèmes, je propose de l'inciter à mettre en œuvre un raisonnement par analogie, c'est-à-dire à se remémorer une expérience précédemment vécue (ici, un problème proche déjà rencontré) et à l'adapter pour résoudre un nouveau problème. L'idée est d'utiliser des problèmes prototypiques qui serviront de référence, en représentant une classe de problèmes de la méthode. En effet, de nombreuses études en psychologie cognitive montrent que le raisonnement par analogie permet d'acquérir des connaissances contextuelles et des connaissances abstraites (voir par exemple (Cummins, 1992; Didierjean, 2001; Gick & Holyoak, 1983)). Cependant, il est à noter qu'en situation d'apprentissage, lorsque les apprenants sont novices dans un domaine, ils ont peu ou pas d'expérience dans le domaine étudié. Aussi, dans cette situation, des cas représentatifs de la situation (par exemple des problèmes résolus) peuvent leur être présentés, pour leur permettre une première expérience dans le domaine. De nombreuses études conduites en psychologie (Carrol, 1994; Cooper & Sweller, 1987; Ward & Sweller, 1990) ont montré l'intérêt pour l'apprentissage de présenter des problèmes résolus aux apprenants. Par ailleurs, comme le montrent Gick et Holyoak (Gick & Holyoak, 1983), le raisonnement par analogie n'est pas toujours spontané et l'acquisition de connaissances lors la résolution de problèmes par analogie n'est pas toujours automatique. Aussi, pour inciter l'apprenant à raisonner par analogie, le cycle AMBRE consiste à lui présenter des problèmes résolus puis à le guider à travers plusieurs étapes dans lesquelles on lui demande explicitement de se remémorer un problème qu'il a déjà rencontré pour résoudre un nouveau problème.

3 Contribution : le cycle AMBRE

Dans le projet AMBRE, le RàPC n'est pas utilisé par le système mais proposé à l'apprenant comme une stratégie d'apprentissage. Pour chaque problème, l'apprenant doit passer par toutes les étapes du RàPC, de l'élaboration du cas à sa mémorisation, pour retrouver lui-même un cas approprié au problème et l'adapter au problème à résoudre. Par ailleurs, comme l'apprenant n'a pas *a priori* d'expérience dans le domaine étudié, des problèmes résolus lui sont proposés.

Ainsi, une séance de travail avec un EIAH AMBRE se divise en deux phases. La première phase est destinée à présenter à l'apprenant des problèmes résolus (problèmes-types) qui serviront de référence par la suite. Après plusieurs séances, l'apprenant aura vu un problème-type correspondant à chaque classe de problèmes définie au sein de la méthode.

La deuxième phase de la séance est consacrée à la résolution par l'apprenant de plusieurs problèmes. L'EIAH guide l'apprenant dans sa résolution à travers cinq étapes inspirées du cycle du RÀPC (cf. **Figure 12**).

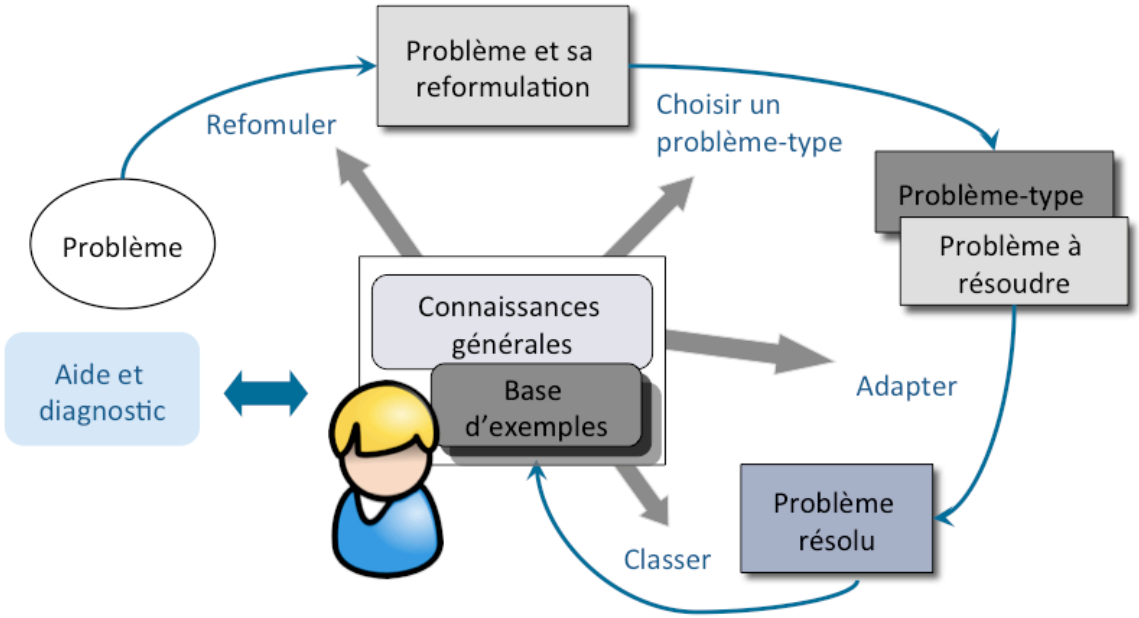

Figure 12. Le cycle AMBRE : cycle du RàPC adapté au projet AMBRE. Ce cycle décrit les activités de l'apprenant lors de l'utilisation de l'EIAH.

Après une première étape qui consiste à lire l'énoncé du problème, l'apprenant doit reformuler le problème en identifiant les éléments de l'énoncé pertinents pour la résolution (étape analogue à l'élaboration du RàPC). Dans une troisième étape, l'apprenant choisit parmi les problèmes-types qui lui ont été présentés celui qui lui semble le plus proche du problème à résoudre (étape analogue à la remémoration). Il identifie ainsi de manière implicite la classe du problème à résoudre. Ensuite, l'apprenant doit rédiger la solution du problème en s'inspirant de la solution du problème-type (étape analogue à l'adaptation). Nous supposons que cette adaptation va l'amener par généralisation à identifier une technique de résolution adaptée aux problèmes de cette classe. Enfin, la dernière étape consiste pour l'apprenant à associer le problème qu'il vient de résoudre à un problème-type, de manière à constituer des groupes de problèmes correspondant aux classes, un groupe étant représenté par un problème-type (étape analogue à la mémorisation).

Ces étapes sont donc guidées par le système mais effectuées par l'apprenant. L'étape de révision du RàPC est remplacée dans le cycle AMBRE par un diagnostic réalisé par le système (cf. **Figure 12**). Celui-ci évalue la production de l'apprenant à la fin de chaque étape et lui propose éventuellement des explications pour l'amener à comprendre ses erreurs et à corriger ses réponses.

Ce cycle de résolution d'un problème, indépendant du domaine d'application, est constitué de plusieurs étapes, ce qui peut sembler un peu fastidieux alors que les problèmes pourraient être résolus directement. Cependant, dans des domaines où la résolution de problèmes occasionne beaucoup d'erreurs liées à l'identification des caractéristiques du problème et au choix d'une technique de résolution correspondante, ces étapes devraient aider l'apprenant à acquérir la méthode.

Par ces étapes imposées et la manière de guider l'apprenant dans sa résolution du problème, les EIAH AMBRE s'apparentent aux "tuteurs intelligents". C'est néanmoins une approche constructiviste qu'ils mettent en œuvre. En effet, selon Piaget (Piaget, 1975), l'apprentissage consiste à construire en mémoire des structures mentales (schèmes) qui sont convoqués lors de la réalisation d'une activité de type résolution de problèmes. Ces schèmes sont complétés et adaptés en fonction des situations rencontrées. L'objectif du cycle AMBRE est la construction par l'apprenant d'un schème par problème type, schème qui sera adapté par l'apprenant à chaque fois qu'il résout un problème par analogie, et on l'espère généralisé afin de s'approcher de la classe de problèmes définie par la méthode.

4 Applications : les EIAH AMBRE

Pour concevoir et réaliser un environnement d'apprentissage suivant le principe décrit précédemment, nous avons adopté une démarche de conception incrémentale et itérative, en spécifiant et implémentant des prototypes ne comprenant d'abord qu'un nombre limité de fonctionnalités. Ces prototypes ont ensuite été évalués, ce qui nous a permis de produire des spécifications afin d'implémenter une nouvelle version du logiciel modifiée et complétée par de nouvelles fonctionnalités. Cette démarche a été choisie afin de valider les choix de conception et de détecter les problèmes précocement.

L'objectif du projet AMBRE étant la conception d'environnements informatiques pour l'apprentissage de méthodes, plusieurs partenaires sont intervenus dans chacune de ces étapes : des chercheurs en informatique, mais aussi en psychologie cognitive et en didactique des mathématiques, une conseillère pédagogique, des enseignants et des apprenants.

La section suivante décrit le développement d'un premier prototype appliqué aux problèmes de dénombrement, après quoi seront présentées les études complémentaires réalisées par Sandra Nogry afin de mieux comprendre comment déclencher les processus d'apprentissage recherchés dans le cycle AMBRE. Je décris ensuite l'EIAH AMBRE-add appliqué aux problèmes additifs.

4.1 Premier prototype et première évaluation

En appliquant le principe du projet AMBRE présenté ci-dessus, un premier prototype d'EIAH dédié au domaine des dénombrements au niveau de la terminale scientifique a été développé : AMBRE-dénombrement (Guin-Duclosson, Jean-Daubias, & Nogry, 2002). Ce prototype implémentait le principe de AMBRE avec un nombre de problèmes limités et des fonctionnalités limitées (il n'y avait pas de résolveur de problèmes intégré au logiciel). Nous l'avons ensuite, dans le cadre du DEA de Sandra Nogry, évalué en classe de terminale scientifique afin de mesurer son impact sur l'apprentissage en utilisant une méthode comparative. Dans le cadre de cette première évaluation, conduite selon des techniques de psychologie cognitive, nous avons comparé cette maquette à une maquette contrôle

n'intégrant pas le cycle AMBRE. Les résultats de cette évaluation n'ont pas permis de mettre en évidence un effet du logiciel sur l'apprentissage. Ils nous ont cependant conduits à identifier les difficultés d'utilisation rencontrées par les apprenants et à spécifier de nouvelles recommandations.

Cette première expérimentation a également montré la nécessité de mieux comprendre comment et dans quelles conditions la résolution de problèmes par analogie conduit à élaborer des connaissances générales, ceci afin de favoriser l'apprentissage de méthodes en effectuant les bons choix de conception pour chaque étape du cycle AMBRE. J'ai donc mis en place une collaboration avec le Laboratoire de Psychologie Cognitive de l'Université de Provence, en la personne d'André Didierjean, afin de réaliser cet objectif dans le cadre de la thèse en Sciences Cognitives de Sandra Nogry.

4.2 Études en psychologie cognitive et recommandations pour le projet AMBRE

En psychologie cognitive, différentes études ont d'une part montré que la résolution de problèmes par analogie peut permettre d'acquérir des connaissances générales et ont d'autre part identifié différents processus cognitifs de généralisation de connaissances (détection de similitudes, processus explicatifs, généralisation par adaptation) que les apprenants peuvent mettre en œuvre face aux exemples et aux problèmes (voir (Cauzinille-Marmèche & Didierjean, 1999) pour une synthèse). Cependant, la résolution de problèmes par analogie n'est pas toujours associée à l'élaboration de connaissances générales. Certaines conditions semblent plus que d'autres associées à un transfert analogique positif et à l'élaboration de connaissances.

D'autres études ont identifié différentes conditions qui favorisent l'acquisition de connaissances générales (Sandra Nogry & Didierjean, 2006). Ces études ont montré que certaines activités telles que la comparaison (Cummins, 1992), l'auto-explication de problèmes résolus (Bielaczyc, Pirolli, & Brown, 1995; Chi, Bassok, Lewis, Reimann, & Glaser, 1989; Renkl, 1997) ou l'adaptation d'un problème déjà vu pour résoudre un nouveau problème (Ross & Kennedy, 1990) favorisent la généralisation de connaissances. Elles ont également mis en évidence le rôle sur l'apprentissage de la similarité entre les problèmes présentés successivement (Quilici & Mayer, 1996, 2002). Ainsi, il semble que l'élaboration de connaissances générales soit favorisée si des problèmes isomorphes ayant des traits de surface dissimilaires sont comparés ou si des problèmes de classes différentes ayant des traits de surface identiques sont comparés (Gick & Paterson, 1992), ou encore lorsque l'on adapte un problème ayant des traits de surface proches du problème à résoudre (Ross & Kennedy, 1990).

Nous nous sommes appuyés sur ces études théoriques pour préciser les processus cognitifs que nous souhaitions voir mis en œuvre par l'apprenant dans le cycle AMBRE, puis pour proposer différentes recommandations destinées à faciliter la mise en œuvre de ces processus (Nogry, 2003). Ces recommandations portaient sur le contenu des consignes, la manière de présenter les problèmes-types, les traits de surface des problèmes-types ou des problèmes à résoudre et le type de diagnostic.

Ces études théoriques nous ayant permis d'améliorer la mise en œuvre concrète du cycle AMBRE, nous avons souhaité réaliser un nouveau prototype d'EIAH. Suite à un changement des programmes scolaires, le domaine des dénombrements n'étant pratiquement plus

étudié en classe de terminale scientifique, nous nous sommes intéressés au domaine des problèmes arithmétiques posés en classe de CE1-CE2, souvent nommés problèmes additifs.

4.3 L'EIAH AMBRE-add

Le domaine des problèmes additifs

Le domaine des problèmes additifs est l'un des domaines les plus largement étudiés en didactique des mathématiques, psychologie cognitive et linguistique. Les problèmes additifs décrivent une situation concrète, par exemple un jeu de billes : « Alex avait 32 billes. À la fin de la récréation, il en a 45. Combien a-t-il gagné de billes pendant la récréation ? ».

Pour un problème de ce type, on attend une résolution en plusieurs étapes :

- décrire le problème à l'aide d'une "opération à trou" : 32 + ? = 45
- écrire comment on effectue le calcul : 45 - 32 = ?
- effectuer le calcul : 13
- écrire la réponse à la question : Alex a gagné 13 billes.

Riley, Greeno et Heller ont proposé une classification des problèmes additifs qui distingue trois catégories de problème : réunion, changement et comparaison (Riley, Greeno, & Heller, 1983). Greeno et Riley montrent que les difficultés que rencontrent les jeunes enfants dans la résolution de problèmes additifs viennent essentiellement du fait qu'ils n'arrivent pas à se représenter correctement la situation décrite dans l'énoncé. Ce qui permet à des enfants plus âgés de bien résoudre les problèmes additifs est leur capacité à modéliser les problèmes (Greeno & Riley, 1987).

Ces travaux en didactique des mathématiques m'ont conduit à penser que le domaine des problèmes additifs est un bon domaine pour un EIAH AMBRE. En effet, il s'agit de problèmes difficiles pour les élèves de l'école élémentaire, dans lesquels la modélisation joue un rôle important et pour lesquels une classification a été établie.

Ces travaux nous ont également apporté des connaissances très utiles pour adapter le principe de AMBRE à ce domaine. Certains travaux ont apporté les informations nécessaires à la conception du système d'aide et de diagnostic, tandis que d'autres travaux ont été utiles pour concevoir la seconde étape du cycle AMBRE, l'étape de reformulation. Ainsi, pour concevoir cette étape, dont l'objectif est l'identification par l'apprenant des éléments de l'énoncé pertinents pour la résolution, nous nous sommes appuyés sur des travaux de Vergnaud (Vergnaud, 1982) et Willis et Fuson (Willis & Fuson, 1988). Ceux-ci proposent de représenter les problèmes sous forme de schémas et montrent que l'utilisation de ces schémas au cours de l'apprentissage peut améliorer les performances de résolution et de catégorisation de problèmes. Aussi, nous avons adapté les schémas proposés dans ces travaux pour représenter les catégories de problèmes « réunion » et « changement », et nous avons modifié le schéma correspondant à la catégorie « comparaison » au sein d'une équipe pluridisciplinaire pour améliorer sa compréhension (Jean-Daubias, 2004).

Je décris maintenant AMBRE-add, l'EIAH conçu suivant le principe de AMBRE pour le domaine des problèmes additifs. Cet EIAH est destiné à être utilisé régulièrement par des élèves de CE1-CE2, de préférence en situation scolaire, en présence d'un enseignant ou d'un tuteur. Le vocabulaire utilisé dans le logiciel a été adapté pour des enfants de ce niveau, en collaboration avec des enseignants et une conseillère pédagogique. Cet EIAH est ici présenté du point de vue de l'apprenant en soulignant les différents choix de conception. Le chapitre

suivant précisera les fonctionnalités d'aide, de diagnostic des réponses de l'apprenant et d'explications proposées par ce système.

Présentation des problèmes résolus

Comme précisé précédemment, étant donné que les apprenants sont novices, une séance de travail avec AMBRE-add commence généralement par une présentation de problèmes résolus représentatifs des classes de problèmes à acquérir au cours de la séance. Conformément aux recommandations issues des études en psychologie cognitive, ces problèmes-types représentent chacun une classe de problèmes et ont tous des traits de surface identiques, afin de favoriser la comparaison entre exemples et l'identification des caractéristiques qui les différencient. Pour chaque problème-type, l'énoncé est d'abord présenté, suivi par sa reformulation puis sa résolution (cf. **Figure 13**). Cette présentation séquentielle devrait favoriser la mise en œuvre par l'apprenant d'un raisonnement anticipatif et de processus explicatifs pour s'expliquer le passage d'une étape à l'autre.

Figure 13. Présentation d'un problème-type résolu (version web de AMBRE-add)

Une fois que les différents problèmes-types ont été étudiés, un bilan présente tous les problèmes-types côte à côte, afin de favoriser la comparaison entre problèmes (ceux étudiés pendant la séance courante, ainsi que ceux étudiés pendants les séances précédentes).

La suite de la séance est consacrée à la résolution de problèmes. Les problèmes sont choisis afin qu'un problème de chaque classe ayant des traits de surfaces proches des problèmes-types étudiés en début de séance soit d'abord présenté, avant de présenter des problèmes ayant des traits de surfaces plus éloignés. Pour chaque problème, AMBRE-add guide la résolution de l'apprenant en suivant le cycle AMBRE.

Reformuler le problème

Après l'étape de lecture de l'énoncé, l'apprenant doit reformuler le problème (cf. **Figure 14**) en le représentant grâce au schéma adéquat. Dans ce but, l'apprenant doit choisir parmi les trois schémas représentatifs des catégories de problème, celui qui correspond au problème, puis doit indiquer la place de l'inconnue par un point d'interrogation avant de compléter cette reformulation par les valeurs déjà connues (et l'opérateur pour les problèmes de changement). L'objectif de cette étape de reformulation est que l'apprenant soit progressivement capable d'identifier dans l'énoncé les éléments pertinents pour la résolution. Cette reformulation devient une référence pour la suite de la résolution.

Figure 14. Étape de reformulation du problème

Choisir un problème-type

L'apprenant doit ensuite comparer le problème aux différents problèmes-types afin de choisir celui qui est le plus proche du problème à résoudre (cf. **Figure 15**). Dans cette étape, la consigne incite les élèves à comparer le problème à résoudre aux différents problèmes-types que l'apprenant a déjà vus (en début de séance et pendant les séances précédentes). Ceux-ci sont représentés par leur énoncé et leur reformulation. Le fait de reformuler le problème en utilisant un schéma facilite la tâche de comparaison de problèmes nécessaire à cette étape. Si l'apprenant éprouve des difficultés à faire cette comparaison, il peut faire appel à l'aide ; le système facilite alors la comparaison en colorant en vert les éléments identiques entre les problèmes et en rouge les éléments différents. À travers cette activité de comparaison, nous souhaitons accentuer la généralisation de connaissances tout en préparant l'étape d'adaptation.

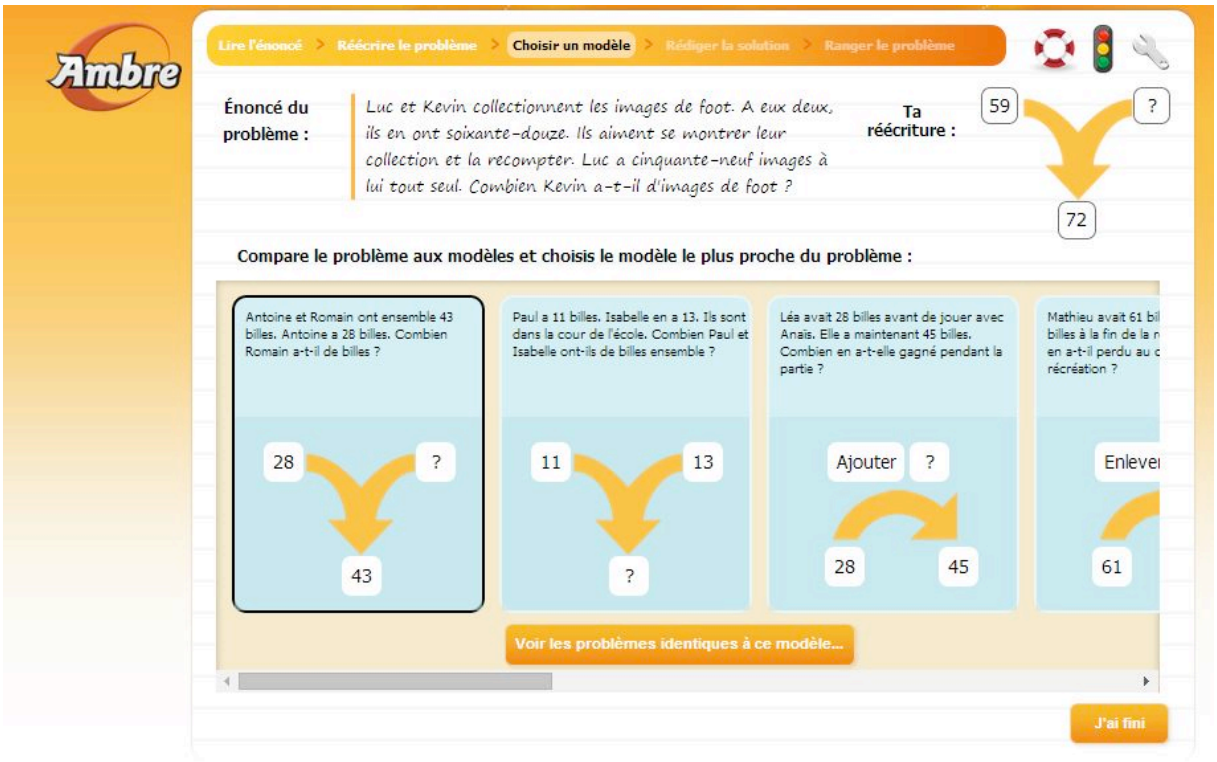

Figure 15. Étape de choix d'un problème-type

Adapter la solution du problème-type

L'étape suivante consiste, pour l'apprenant, à rédiger la solution du problème en s'aidant de la solution du problème-type qu'il a choisi à l'étape précédente (cf. **Figure 16** : problème-type à gauche, problème à résoudre à droite). Comme nous l'avons vu précédemment, la résolution se décompose en plusieurs sous-étapes. L'apprenant doit d'abord écrire l'équation qui correspond au problème. Ensuite, il doit écrire l'opération à réaliser pour aboutir à la solution, puis la calculer. Enfin il doit écrire la phrase de réponse à l'aide de listes déroulantes. Selon le niveau (CE1 ou CE2), la deuxième partie de la résolution peut ne pas être demandée.

Figure 16. Résolution du problème par analogie

Pour faciliter la mise en correspondance du problème-type avec le problème à résoudre, une fonctionnalité du système permet de colorier les éléments de l'énoncé, de la reformulation et de la solution qui ont une fonction identique ou qui sont liés (cf. Chapitre A4). De plus, la sous-étape du problème-type correspondant à la sous-étape traitée par l'apprenant est accentuée par une police en gras. Les résultats des études en psychologie cognitive nous permettent de penser que l'adaptation de la résolution du problème-type choisi au problème à résoudre est source de généralisation.

Classer le problème

Enfin, dans la dernière étape, le système présente d'abord à l'apprenant un bilan de la résolution du problème : l'énoncé, sa reformulation et la solution. Ensuite l'apprenant est invité à ranger ce problème avec l'un des problèmes-types, afin de constituer des groupes de problèmes de même classe. Ce faisant, l'apprenant identifie la classe à laquelle appartient le problème.

5 Évaluation du cycle AMBRE

Dans le cadre de la thèse de Sandra Nogry, nous avons conduit trois expérimentations afin d'évaluer l'utilisabilité de AMBRE-add et son impact sur l'apprentissage de la méthode. La première expérimentations réalisée en laboratoire avec cinq apprenants a permis d'identifier les plus gros problèmes d'utilisation et de modifier le logiciel avant l'expérimentation en classe.

La seconde expérimentation s'est déroulée en classe avec 78 élèves de CE1, dans des conditions réelles d'utilisation. Cette expérience consistait à comparer AMBRE-add avec un logiciel contrôle qui proposait une résolution de problème simple. Cette expérience n'a pas permis de montrer que le cycle AMBRE a un impact plus important qu'une résolution simple sur l'apprentissage de la méthode. Nous avons identifié trois facteurs qui peuvent expliquer ces résultats. On peut d'abord souligner des difficultés à pouvoir comparer objectivement les performances obtenues après utilisation de AMBRE-add et du logiciel contrôle. En effet, le nombre de problèmes résolus et le type d'interventions des encadrants ont varié suivant le logiciel utilisé. Par ailleurs, les élèves ont rencontré de nombreuses difficultés tout au long de l'utilisation du logiciel. Ces difficultés étaient dues à un manque de maîtrise de la lecture (difficultés à lire l'énoncé et les messages d'aide et de diagnostic) et des techniques calculatoires, mais aussi à la complexité du système (difficultés pour situer leurs erreurs ou pour comprendre le lien entre les différentes étapes) pour des élèves de cet âge là. Enfin, l'utilisation du système par les élèves de CE1 était assez différente de l'utilisation prescrite. En effet, ils n'adaptaient que rarement le problème-type pour résoudre le problème.

Cette expérimentation nous a conduit à modifier les messages d'aide et de diagnostic proposés par le système.

Par ailleurs, il est apparu qu'AMBRE-add était trop complexe pour des élèves de CE1 mais qu'il pourrait être plus adapté à des élèves de CE2. En effet, les classes de problèmes traitées par AMBRE-add sont également abordées en classe de CE2 et posent encore des problèmes de modélisation aux élèves ; de plus ces élèves ont plus de facilité à lire et maîtrisent davantage les techniques calculatoires. Nous avons par conséquent conduit une expérimentation auprès d'élèves en classe de CE2 consistant à observer l'utilisation effective du logiciel. Les observations ont montré que les élèves en classe de CE2 utilisent le logiciel de manière autonome (à partir de la seconde séance, ils posent très peu de questions), ils comprennent rapidement comment utiliser le logiciel et n'ont pas de difficulté à lire les messages d'aide et d'explication. En outre, une proportion importante d'élèves a une utilisation du logiciel proche de l'utilisation attendue. Ainsi, 16 enfants sur 21 ont utilisé le problème-type pour résoudre le problème. Les analyses quantitatives montrent que les élèves ayant des performances faibles lors du pré-test ont obtenu de meilleures performances après avoir utilisé AMBRE-add. Les résultats de cette observation sont donc encourageants : les élèves en classe de CE2 n'ont pas de difficulté particulière à utiliser le logiciel AMBRE-add.

Ainsi, à travers ces différentes expérimentations, nous avons identifié les pratiques et les difficultés des apprenants face au logiciel. Nous avons également produit des recommandations qui ont conduit à améliorer le logiciel et identifier un public pour lequel AMBRE-add pourrait être bien adapté.

Les difficultés rencontrées par les élèves de CE1 ont conduit l'équipe de conception à imaginer des activités très simples, préparatoires à AMBRE-add pour des apprenants jeunes. De plus, nous avons intégré un synthétiseur vocal au système pour limiter les difficultés dues à la lecture et réduire la charge cognitive pour des apprenants faibles en lecture.

Ces premières expérimentations n'ont cependant pas montré que le cycle AMBRE facilite davantage l'apprentissage de la méthode qu'une résolution simple des mêmes problèmes. Comme les élèves de CE2 semblent avoir moins de difficulté à utiliser le logiciel que ceux de CE1, une nouvelle expérimentation avec plusieurs classes de CE2 a été menée. Si elle

confirme la progression des élèves ayant utilisé le logiciel, l'impact de AMBRE-add sur l'apprentissage n'est pas plus important qu'avec le logiciel contrôle lorsque l'on étudie les résultats de l'ensemble des élèves. En effet, cet impact semble concerner seulement une partie des élèves. Ainsi, la principale conclusion de cette dernière expérimentation est l'existence d'une grande variation interindividuelle, d'une part du fait des grandes facilités ou au contraire des difficultés majeures des apprenants en mathématique ou par rapport à l'utilisation du logiciel, et d'autre part au niveau de l'attitude de l'élève face au logiciel, plus ou moins active. Au vu des résultats obtenus dans cette expérimentation, une personnalisation du logiciel en fonction du niveau et des difficultés spécifiques de chaque enfant paraît indispensable.

Conclusion et perspectives

Ce chapitre a présenté les principes du projet AMBRE, qui propose de concevoir des EIAH destinés à l'enseignement de méthode fondés sur un processus d'apprentissage inspiré du raisonnement à partir de cas. Le *cycle AMBRE* propose d'aider l'apprenant à exploiter sa propre expérience pour résoudre de nouveaux problèmes en se remémorant un problème proche et en l'adaptant. Les résultats d'études en psychologie cognitive nous permettent de penser que guider la résolution de problèmes à travers les étapes du raisonnement à partir de cas devrait permettre à l'apprenant d'acquérir des connaissances généralisées.

Après avoir décrit le cycle AMBRE, j'ai présenté comment nous avons conçu l'EIAH AMBRE-add au sein d'une équipe pluridisciplinaire, en nous appuyant sur des études en didactique des mathématiques sur le domaine des problèmes additifs et sur des études en psychologie cognitive sur l'apprentissage à partir d'exemples. J'ai enfin décrit comment nous avons procédé à plusieurs expérimentations destinées à évaluer cet environnement.

Après une application du cycle AMBRE au domaine des dénombrements et au domaine des problèmes additifs, des travaux récents ont abouti à la réalisation d'un prototype d'EIAH AMBRE appliqué à la conjugaison française. En effet, la conjugaison française est souvent enseignée à l'aide de verbes prototypiques. Dans ce nouvel EIAH AMBRE, il est probable que nous nous dirigions vers une utilisation du cycle AMBRE seulement en cas de difficulté. Nous avons en effet constaté dans les expérimentations menées en CE2 avec AMBRE-add que les élèves utilisent le problème type et l'adaptent uniquement lorsqu'ils n'arrivent pas à résoudre directement leur problème.

Afin de faciliter l'utilisation de l'EIAH AMBRE-add par les élèves de CE1, et pour remédier à des difficultés spécifiques de certains élèves, nous avons conçu un ensemble d'activités autour de AMBRE-add. Ces activités se focalisent sur certaines compétences du cycle comme par exemple la reformulation. Elles pourront pour certaines préparer les élèves à l'utilisation de AMBRE-add ou pour d'autres leur permettre d'en prolonger l'apprentissage. Le développement de ces activités ne soulevant pas de questions de recherche spécifique, et faute de forces de développement et d'expérimentation, cette piste de travail est en attente depuis quelques années.

Une autre piste de travail issue des expérimentations menées sur AMBRE-add visait à étudier comment transformer AMBRE-add pour qu'il puisse s'adapter non seulement aux spécificités de l'apprenant, mais également à celles du contexte d'apprentissage. Pour ce faire, nous avons d'une part conçu un outil permettant aux enseignants de personnaliser

l'environnement, de créer des séquences d'apprentissage et de générer les problèmes qu'ils souhaitent faire résoudre à leurs élèves. Ce travail est décrit dans le chapitre A5. Nous avons d'autre part étudié comment construire un profil de l'apprenant en analysant l'activité de cet apprenant lorsqu'il utilise l'EIAH AMBRE-add. Ce profil rassemble un ensemble d'informations sur l'apprenant caractérisant ses connaissances, compétences et/ou conceptions. Nous avons ensuite mené des recherches sur la manière de permettre une adaptation automatique de l'EIAH à partir de tels profils. Ces recherches sont décrites dans la partie B du mémoire. J'y présenterai comment nous avons appliqué les modèles et outils développés à l'EIAH AMBRE-add, pour permettre au système d'adapter ses interventions à l'apprenant, et en particulier de générer les problèmes ou les activités les plus adaptées à ses difficultés.

Synthèse des contributions

Question de recherche : Comment enseigner une méthode de résolution de problèmes ?

- **Contribution théorique** : le cycle AMBRE
- **Systèmes développés** : AMBRE-dénombrement et AMBRE-add
- **Déploiement** : AMBRE-add a été expérimenté dans une dizaine de classes. Sa version web sera prochainement librement accessible en ligne.
- **Co-auteurs** : Sandra Nogry, Stéphanie Jean-Daubias

Pour en savoir plus

• Sur AMBRE-dénombrement :

Guin-Duclosson, N., Jean-Daubias, S., & Nogry, S. (2002). The AMBRE ILE: How to Use Case-Based Reasoning to Teach Methods. In G. G. S. Cerri F. Paraguaçu (Ed.), *Intelligent Tutoring Systems (ITS'2002)* (Vol. 2363, pp. 782–791). San Sebastian (Espagne) & Biarritz (France): Springer, Lecture Notes in Computer Science.

• Sur AMBRE-add :

Nogry, S., Guin, N., & Jean-Daubias, S. (2008). AMBRE-add : An ITS to Teach Solving Arithmetic Word Problems. Technology, Instruction, Cognition and Learning, 6(1), 53–61.

Guin, N., Nogry, S., & Jean-Daubias, S. (2009). Le projet AMBRE : inviter un apprenant à adapter une expérience passée afin de l'amener à généraliser ses connaissances. In E. Egyed-Zsigmond, N. Guin, & A. Mille (Eds.), Réutilisation de l'expérience : modèles et applications. Hermès Sciences Publications.

• Sur les expérimentations effectuées pour évaluer AMBRE-add :

Nogry, S., Jean-Daubias, S., & Duclosson, N. (2004). ITS Evaluation in Classroom: The Case of AMBRE-AWP. In J. C. L. & al. (Ed.), *Intelligent Tutoring Systems (ITS'2004)* (pp. 511–520). Laceio, Brasil.

Nogry, S. (2005). *Faciliter l'apprentissage à partir d'exemples en situation de résolution de problèmes - Application au projet AMBRE*. Université Lumière Lyon 2, Lyon, France.

Publications liées

Revues internationales

Nogry, S., Guin, N., & Jean-Daubias, S. (2008). AMBRE-add : An ITS to Teach Solving Arithmetic Word Problems. *Technology, Instruction, Cognition and Learning*, *6*(1), 53–61.

Nogry, S., Jean-Daubias, S., & Guin, N. (2012). How to combine objectives and methods of evaluation in iterative ILE design: lessons learned from designing AMBRE-add. *Interactive Learning Environments*, *20*(2), 155–175.

Conférences internationales

Guin-Duclosson, N., Jean-Daubias, S., & Nogry, S. (2002). The AMBRE ILE: How to Use Case-Based Reasoning to Teach Methods. In G. G. S. Cerri F. Paraguaçu (Ed.), *Intelligent Tutoring Systems (ITS'2002)* (Vol. 2363, pp. 782–791). San Sebastian (Espagne) & Biarritz (France): Springer, Lecture Notes in Computer Science.

Nogry, S., Jean-Daubias, S., & Duclosson, N. (2004). ITS Evaluation in Classroom: The Case of AMBRE-AWP. In J. C. L. & al. (Ed.), *Intelligent Tutoring Systems (ITS'2004)* (pp. 511–520). Laceio, Brasil.

Chapitres d'ouvrages

Guin, N., Nogry, S., & Jean-Daubias, S. (2009). Le projet AMBRE : inviter un apprenant à adapter une expérience passée afin de l'amener à généraliser ses connaissances. In E. Egyed-Zsigmond, N. Guin, & A. Mille (Eds.), *Réutilisation de l'expérience : modèles et applications*. Hermès Sciences Publications.

Guin-Duclosson, N., Jean-Daubias, S., & Nogry, S. (2007). Le projet AMBRE : utiliser un paradigme d'apprentissage fondé sur le RÀPC pour faire acquérir à des élèves une méthode de résolution de problèmes. In J. Renaud, B. Chebel Morello, B. Fuchs, & J. Lieber (Eds.), *Raisonnement à partir de cas 2 - surveillance, diagnostic et maintenance* (pp. 203–226). Hermès-Lavoisier.

Revues nationales

Nogry, S., Jean-Daubias, S., & Guin-Duclosson, N. (2006). Comment combiner les objectifs et méthodes d'évaluation pour la conception itérative des EIAH ? Leçons tirées de la conception de AMBRE-add. *Sciences et Technologies de l´Information et de La Communication Pour l´Éducation et La Formation (STICEF)*, *13*, 147–186.

Conférences nationales

Guin-Duclosson, N., Jean-Daubias, S., & Nogry, S. (2001). Le projet AMBRE : utiliser le RàPC pour enseigner des méthodes. *RàPC 2001 - Atelier Raisonnement À Partir de Cas, Plate-Forme AFIA*. Grenoble, France.

Nogry, S., Guin-Duclosson, N., & Jean-Daubias, S. (2001). Évaluation cognitive d'un Environnement Informatique d'Apprentissage Humain pour l'apprentissage de méthode. *ARCo'2001 - Colloque Interdisciplinaire En Sciences Cognitives*. Lyon, France.

Nogry, S., Jean-Daubias, S., & Guin-Duclosson, N. (2002). La psychologie cognitive au service de la conception de l'environnement d'apprentissage AMBRE. In *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie (TICE'2002)*. Villeurbanne, France.

CHAPITRE A4
ÉLICITER LES CONNAISSANCES PERMETTANT D'ACCOMPAGNER L'APPRENANT

Résumé

Ce chapitre aborde la question des connaissances nécessaires pour accompagner l'apprenant dans son utilisation d'un EIAH AMBRE et dans son apprentissage. L'objectif est de pouvoir apporter de l'aide à l'apprenant dans la tâche de résolution de problèmes, de pouvoir diagnostiquer ses réponses et de lui proposer des explications pertinentes par rapport à ses erreurs.

J'y présente l'architecture CHAMADE, qui permet à un expert d'éliciter dans un domaine donné les connaissances nécessaires à un EIAH AMBRE pour accompagner l'apprenant. Ces modèles de connaissances sont définis à partir de méta-modèles de connaissances de communication avec l'apprenant, de connaissances d'aide et de diagnostic. Un moteur associé à ces méta-modèles permettent d'exploiter les modèles de connaissances définis et d'obtenir ainsi dans un domaine donné les systèmes à base de connaissances sous-jacents à un EIAH AMBRE.

Ce chapitre s'appuie sur des notions présentées dans les chapitres A1 et A3.

INTRODUCTION 82

1 CADRE DES RECHERCHES : LE PROJET AMBRE 82

2 QUESTION DE RECHERCHE : ÉLICITER LES CONNAISSANCES PERMETTANT DE DIAGNOSTIQUER LES RÉPONSES DE L'APPRENANT, ET DE LUI FOURNIR AIDE ET EXPLICATIONS SUR SES ERREURS 82

2.1 ACCOMPAGNER L'APPRENANT DANS SA RÉSOLUTION DE PROBLÈMES 82
2.2 ÉLICITER LES CONNAISSANCES PERMETTANT D'ACCOMPAGNER L'APPRENANT 85

3 CONTRIBUTION : L'ARCHITECTURE CHAMADE 86

3.1 MÉTA-MODÈLES DE CONNAISSANCES 87
3.2 PROCESSUS DE RAISONNEMENT 90

CONCLUSION ET PERSPECTIVES 92

SYNTHÈSE DES CONTRIBUTIONS 93

PUBLICATIONS LIÉES 93

Introduction

Le chapitre précédent a présenté le cycle AMBRE, ainsi que sa mise en œuvre dans l'EIAH AMBRE-add. Dans le cycle AMBRE, l'apprenant résout un problème en suivant des étapes inspirées par le Raisonnement à Partir de Cas. Le système guide l'apprenant dans ces étapes en lui apportant une aide sur ce qu'il doit faire, et en évaluant les productions de l'apprenant à la fin de chaque étape ou même sous-étape. Il lui propose éventuellement des explications pour l'amener à comprendre ses erreurs et à corriger ses réponses.

La question étudiée ici est celle des modèles de connaissances permettant au système d'accompagner ainsi l'apprenant dans sa résolution de problèmes et dans son apprentissage. Nous avons vu que les EIAH AMBRE s'appuient sur l'architecture SYRCLAD, qui permet d'éliciter les connaissances de la méthode de résolution de problèmes. Un système qui résout les problèmes d'un domaine en appliquant une méthode ne dispose pas pour autant des connaissances lui permettant de diagnostiquer les réponses de l'apprenant, ni de lui fournir des explications sur ses erreurs. Ce chapitre discute donc des modèles de connaissances qu'il faut adjoindre à SYRCLAD afin de pouvoir accompagner l'apprenant. La section 2 présente le type d'aide, de diagnostic et d'explications que nous souhaitons pouvoir apporter à l'apprenant, et la section 3 présente l'architecture CHAMADE qui permet de mettre en œuvre ces fonctionnalités.

1 Cadre des recherches : le projet AMBRE

Ce travail a été mené dans le cadre du projet AMBRE, grâce au financement du programme interdisciplinaire STIC-SHS du CNRS entre 2001 et 2005. La première version des systèmes à base de connaissances de CHAMADE a été réalisée grâce la contribution de François Fallet-Kahn. Récemment, Bryan Kong Win Chang et Aurélie Kong Win Chang ont contribué en 2013, en collaboration avec Marie Lefevre, à formaliser davantage les modèles de connaissance de l'architecture, ce travail étant actuellement repris par Awa Diattara dans le cadre de sa thèse dont les travaux seront présentés au chapitre A6.

2 Question de recherche : éliciter les connaissances permettant de diagnostiquer les réponses de l'apprenant, et de lui fournir aide et explications sur ses erreurs

Avant d'aborder la question de l'élicitation des connaissances permettant au système d'accompagner l'apprenant, je vais - afin d'illustrer le besoin - préciser dans une première sous-section quel type d'aide, de diagnostic et d'explications sont attendues dans le cadre d'un EIAH AMBRE. Des exemples issus des fonctionnalités d'aide et d'explications de l'EIAH AMBRE-add seront utilisés.

2.1 Accompagner l'apprenant dans sa résolution de problèmes

Afin d'assister l'apprenant dans sa résolution de problème, et dans le but de favoriser l'apprentissage, nous souhaitons que le système puisse d'une part répondre à une demande d'aide de l'apprenant, et d'autre part diagnostiquer ses réponses pour chaque étape du cycle et l'aider à réviser une réponse erronée en lui proposant des explications pertinentes.

L'EIAH doit proposer des aides et des explications de différentes natures selon l'étape en cours de réalisation et les erreurs commises. Pour l'EIAH AMBRE-add, la conception de ces messages a été réalisée à partir d'une analyse *a priori* des difficultés rencontrées et des erreurs possibles dans chacune des étapes et sous-étapes proposées par le logiciel.

Dans AMBRE-add, le diagnostic peut être réalisé automatiquement par le système à chaque sous-étape ou à la fin d'une étape suivant un paramétrage défini (le plus souvent par l'enseignant), ou être demandé par l'apprenant en cliquant sur l'icône représentant un feu tricolore (cf. **Figure 17** en haut à droite). L'apprenant peut également demander de l'aide à tout moment en utilisant l'icône représentant une bouée de sauvetage. Les messages d'aide ou de diagnostic sont souvent à deux niveaux : si le premier message ne suffit pas à l'élève pour comprendre, il peut utiliser le deuxième niveau (cf. **Figure 17**).

Les messages d'aide consistent souvent à rappeler à l'apprenant ce qu'il doit faire. Cette aide peut également renvoyer l'élève vers des pages de rappel (par exemple les différents schémas utilisés pour reformuler un problème arithmétique) ou vers des outils favorisant la comparaison entre problème-type et problème à résoudre à travers la coloration des éléments de l'énoncé et de la reformulation ayant la même fonction dans les deux problèmes. (cf. **Figure 18**).

Figure 17. Exemple de fenêtre d'aide à deux niveaux pour l'étape d'adaptation. Le deuxième niveau active le coloriage facilitant la mise en correspondance du problème-type et du problème à résoudre (cf. **Figure 18**).

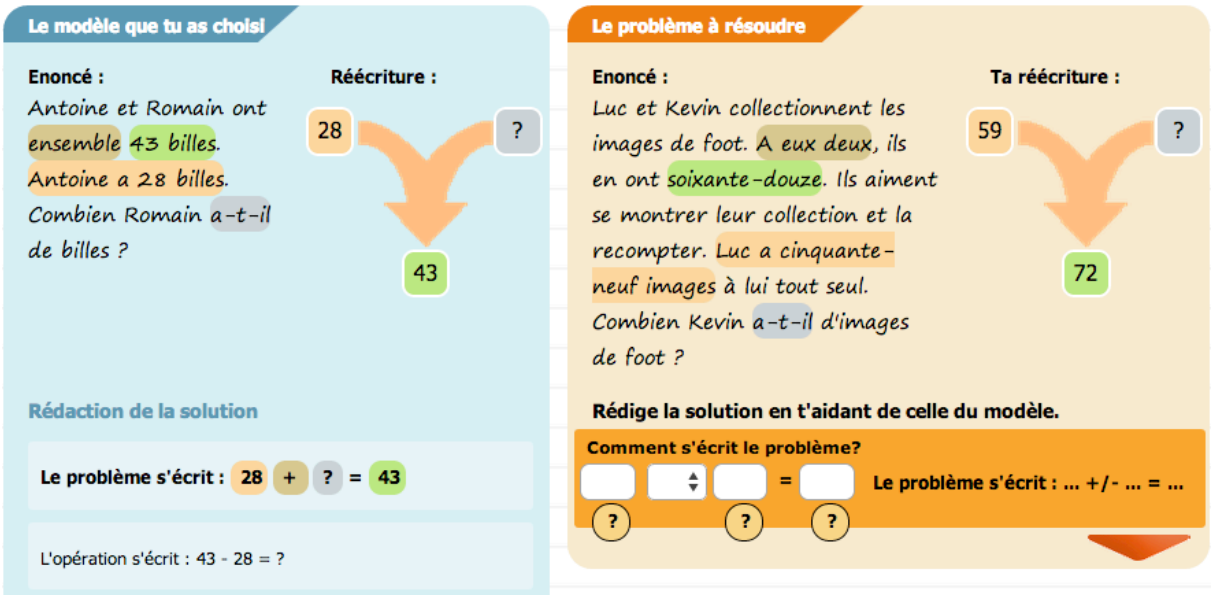

Figure 18. Aide à l'analogie dans l'étape d'adaptation

Les messages d'explication faisant suite au diagnostic peuvent prendre plusieurs formes. Ils peuvent présenter une phrase explicative de l'erreur. Par exemple, dans AMBRE-add, lors de la rédaction de la phrase de réponse, il est fréquent que les élèves écrivent une phrase qui est dans l'énoncé. Dans ce cas, le système explique « Ce que tu as écrit est vrai, mais ce n'est pas la réponse à la question ».

Ces messages d'explication peuvent aussi contenir un énoncé généré par le système correspondant à la réponse de l'élève. Cette génération d'énoncé peut souvent lui permettre de comprendre en quoi sa réponse est erronée. Cette fonctionnalité de génération d'énoncé est par exemple utilisée dans AMBRE-add pour expliquer à l'apprenant pourquoi sa reformulation à l'aide d'un schéma n'est pas correcte : le système génère un énoncé de problème correspondant au schéma donné par l'élève (cf. **Figure 19**).

Des éléments graphiques peuvent aussi être utilisés lorsque le système ne sait pas interpréter la réponse de l'apprenant, ou parce que ce format est plus facile à comprendre que des explications textuelles complexes. On peut ainsi mettre en vert les éléments de la réponse qui sont corrects et en rouge ceux qui sont erronés.

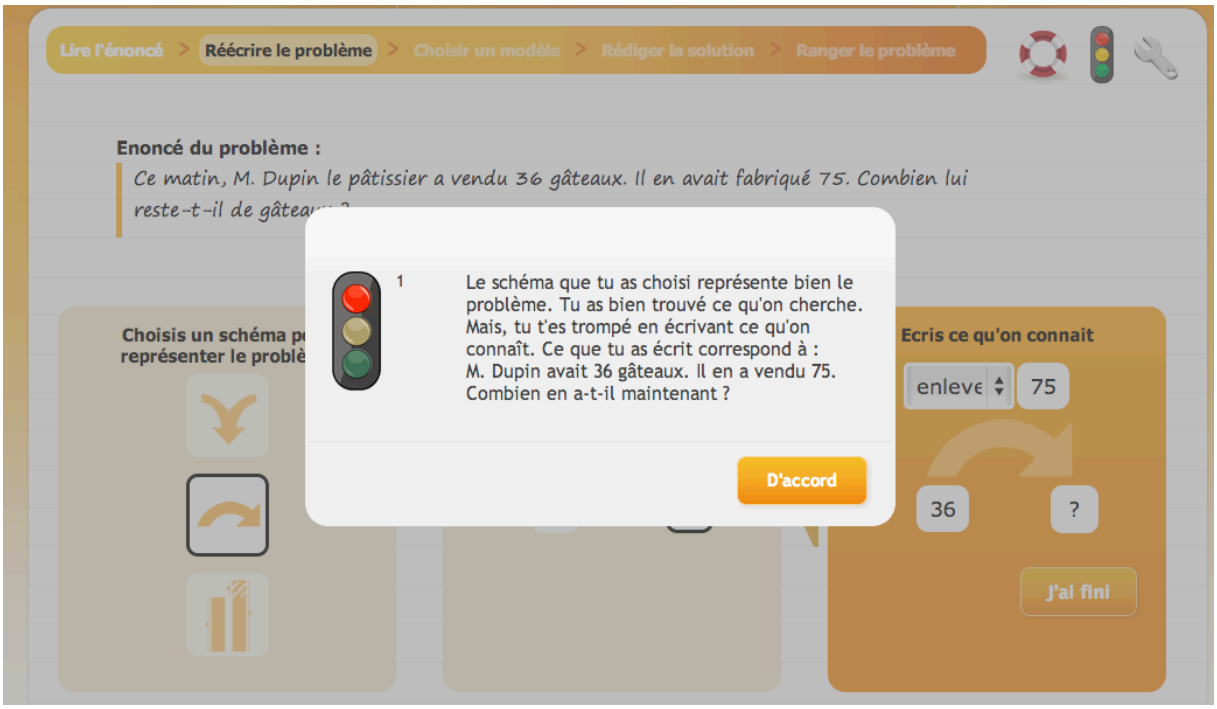

Figure 19. Exemple de message d'explication généré à partir de la réponse de l'élève

2.2 Éliciter les connaissances permettant d'accompagner l'apprenant

La question de recherche abordée dans ce chapitre est de permettre à un expert de définir les connaissances nécessaires à un EIAH AMBRE pour accompagner l'élève dans son apprentissage : connaissances de diagnostic des réponses, connaissances de construction de messages d'aide et d'explications sur les erreurs.

On retrouve bien ainsi l'approche présentée au chapitre 1 : il s'agit de définir des méta-modèles qui permettront dans un domaine donné de définir les modèles de connaissance nécessaires à l'EIAH pour assurer ces fonctionnalités (cf. **Figure 20**). Ces modèles de connaissance spécifiques au domaine seront exploités par un moteur associé aux méta-modèles, afin de permettre les fonctionnalités recherchées, c'est-à-dire le diagnostic des réponses de l'apprenant et la construction de messages d'aide et d'explications.

Je présente dans la section suivante l'architecture CHAMADE, qui regroupe les méta-modèles et le moteur de raisonnement associé. Cette architecture s'appuie sur l'architecture SYRCLAD décrite au chapitre A1. Les EIAH AMBRE proposent à l'apprenant une démarche de raisonnement à partir de cas, alors que les résolveurs SYRCLAD se fondent sur une classification des problèmes du domaine. Nous sommes donc confrontés à la difficulté d'assister l'apprenant dans sa résolution de problèmes, en nous fondant sur des connaissances de type classification, alors que lui utilise l'analogie entre problèmes. C'est la notion de problème prototypique représentant une classe qui permet de faire le lien entre les connaissances du système et le raisonnement de l'apprenant. Comme nous le verrons, il est en effet nécessaire que le système dispose de connaissances pour soutenir le RàPC, comme par exemple des connaissances d'analogie qui permettent d'accompagner l'apprenant, comme illustré sur la **Figure 18**. Il ne s'agit pas pour autant d'un système de raisonnement à partir de cas, puisque le résolveur n'utilise pas le cycle de RàPC pour résoudre les problèmes. En effet, il est important que le système accompagnant l'apprenant utilise les connaissances de la méthode que celui-ci doit acquérir.

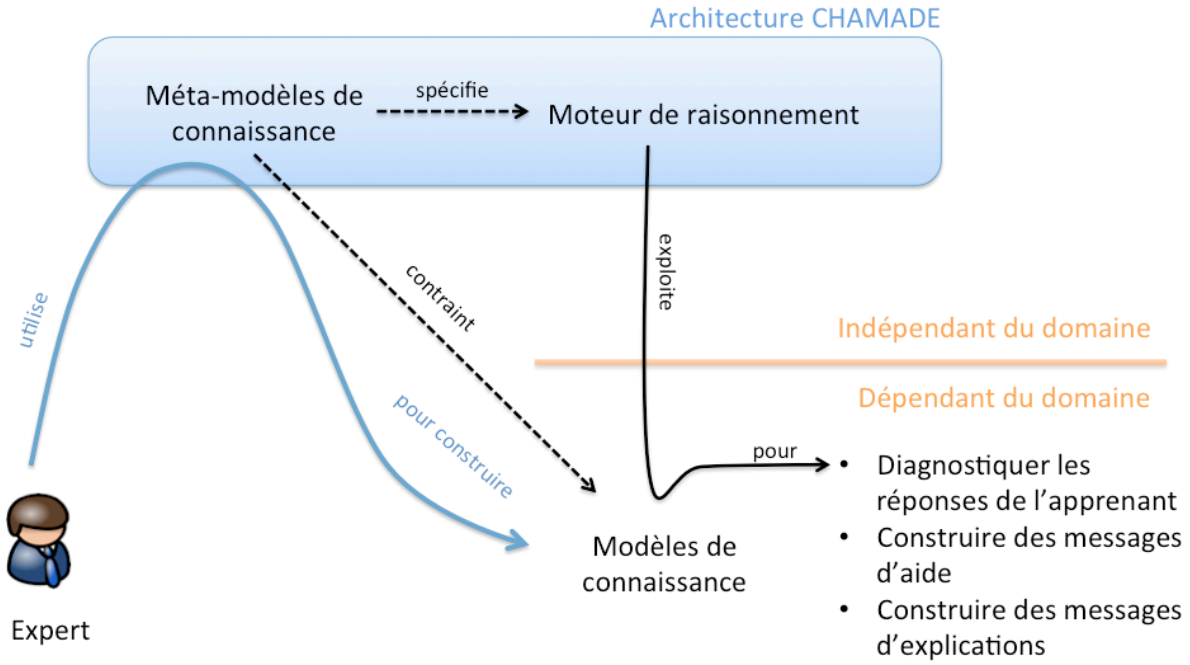

Figure 20. Instanciation de l'approche générale présentée dans le chapitre 1 à la question de l'élicitation des connaissances permettant d'accompagner l'apprenant dans son apprentissage avec un EIAH AMBRE.

3 Contribution : l'architecture CHAMADE

L'architecture CHAMADE (arCHitecture pour l'Apprentissage de Méthodes permettant Aide, Diagnostic et Explications) a été élaborée de manière à pouvoir définir des modèles de connaissances pertinents pour les EIAH AMBRE, indépendamment du domaine, mais en prenant comme exemple celui des problèmes additifs, de manière à concevoir les systèmes à base de connaissances sous-jacents à l'EIAH AMBRE-add. Comme l'architecture SYRCLAD, CHAMADE est codée en langage Prolog. Les structures de représentation des connaissances ainsi que les algorithmes exploitant ces connaissances sont définis indépendamment du domaine, et des bases de connaissances spécifiques sont définies pour le domaine des problèmes additifs.

L'architecture CHAMADE contient tout d'abord un résolveur SYRCLAD ainsi que les connaissances sur la méthode associée. Elle contient de plus deux grands groupes de connaissances (cf. **Figure 21**) : celles destinées à la communication avec l'apprenant, et celles destinées à diagnostiquer les réponses de l'apprenant et à construire les messages d'aide et d'explication des erreurs. De plus, une base de problèmes contient les problèmes-types déjà présentés à l'apprenant ainsi que les problèmes qu'il a déjà résolus.

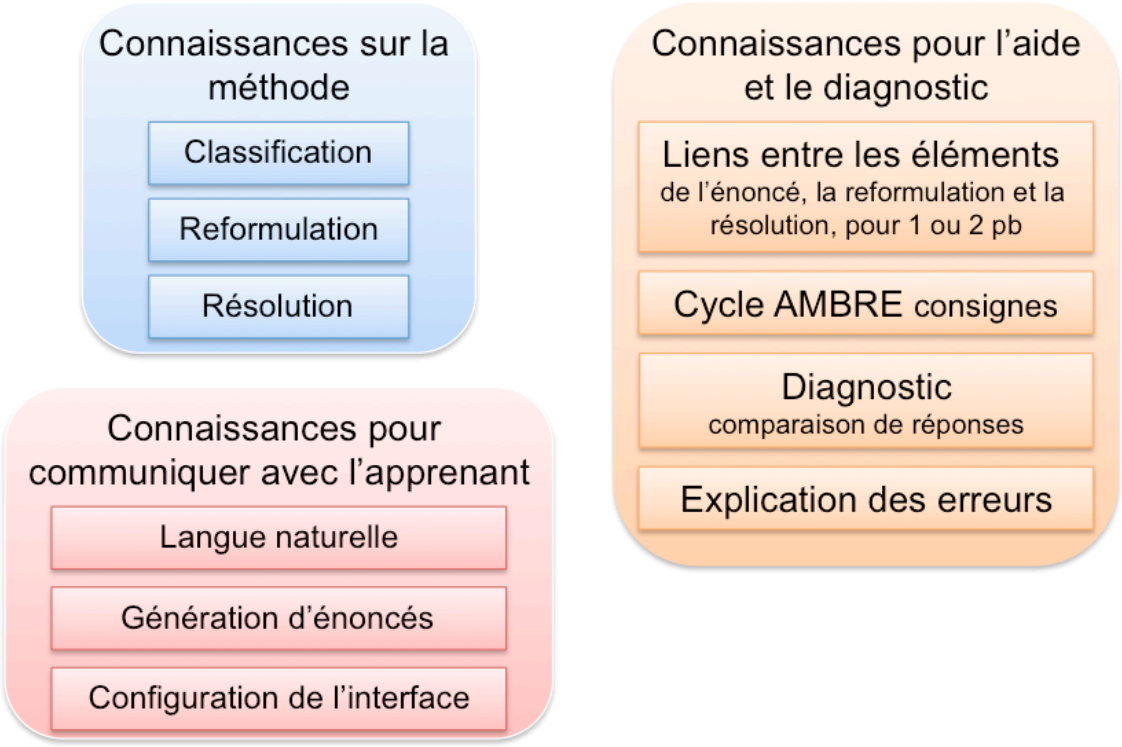

Figure 21. Les différentes bases de connaissances de l'architecture CHAMADE

Le moteur exploitant ces connaissances diagnostique les différentes réponses de l'apprenant en les comparant aux réponses proposées par le résolveur SYRCLAD, et compose des messages d'aide et d'explication adaptés à la situation.

3.1 Méta-modèles de connaissances

Les connaissances destinées à la communication avec l'apprenant sont de trois types :

- Les connaissances liées à la langue naturelle, qui permettent de construire des messages grammaticalement corrects.
- Les connaissances de génération, qui permettent de générer un énoncé de problème, ou une formulation en langue naturelle d'une connaissance du domaine. Ces connaissances permettent par exemple de générer le message présenté sur la **Figure 19**, d'un énoncé correspondant à la reformulation de l'élève.
- Les connaissances de configuration de l'interface, qui sont destinées à fournir à l'interface des éléments qu'elle interprète, afin de par exemple mettre en couleur des éléments du problème ou de la réponse de l'apprenant, ou de construire des listes déroulantes. Par exemple, pour rédiger la phrase de réponse, l'apprenant utilise des listes déroulantes contenant des éléments choisis par CHAMADE-add, grâce à sa connaissance du problème.

Les connaissances destinées à l'aide et au diagnostic sont de quatre types :

- Les connaissances permettant de connaître le rôle des différentes parties d'un problème (énoncé, reformulation et solution), ainsi que les liens entre ces parties. Elles permettent de mettre en évidence (grâce à des couleurs) les éléments analogues de l'énoncé, de la reformulation et de la solution, et de les mettre en relation avec les éléments

analogues d'un autre problème (cf. **Figure 18**). Ces connaissances sont similaires aux connaissances d'adaptation du RàPC.

- Les connaissances sur le cycle AMBRE tel qu'il est mis en œuvre dans l'EIAH (étapes et sous-étapes), afin de pouvoir rappeler à l'apprenant ce qu'il doit faire en cas de demande d'aide (cf. **Figure 17**).

- Les connaissances définissant comment comparer les réponses de l'apprenant à la réponse correcte fournie par le résolveur SYRCLAD, afin de pouvoir déterminer si une réponse est correcte ou non. En effet, le résolveur donne une réponse correcte, mais il est possible que d'autres réponses équivalentes à la réponse du résolveur soient également correctes. C'est pour définir cette équivalence que des connaissances sont nécessaires.

- Les connaissances permettant de donner à l'élève des explications pertinentes sur ses erreurs : il s'agit de définir un ensemble de réponses erronées attendues, et d'y associer des messages explicatifs. Il faut également savoir comment aider l'élève à remédier à une erreur que le système n'a pas su interpréter. Enfin, il peut être nécessaire de re-contextualiser un message d'erreur.

Des méta-modèles de ces différents types de connaissances ont été définis en XML Schema, et les modèles de connaissance pour un domaine donné peuvent alors être définis par un fichier XML. La **Figure 22** présente un de ces méta-modèles, qui décrit comment représenter les connaissances permettant de diagnostiquer la réponse de l'apprenant et de proposer une rétroaction appropriée.

Pour chaque point de diagnostic, l'expert doit spécifier :

- L'étape ou sous-étape du cycle AMBRE concernée par la réponse de l'apprenant. Par exemple la sous-étape 4 de l'étape 4 pour AMBRE-add, où l'élève doit rédiger la phrase de réponse (e.g. "Julie a gagné 3 billes").
- L'attribut du problème correspondant pour le résolveur. Dans notre exemple la phrase de réponse du problème. Lorsque c'est nécessaire, le résolveur doit fournir l'ensemble des réponses correctes.
- La rétroaction à proposer à l'apprenant lorsque sa réponse est conforme à celle du résolveur. Dans notre exemple, si le diagnostic est demandé par l'apprenant, on lui affiche un message indiquant que jusqu'ici ses réponses sont justes (si les réponses aux sous-étapes précédentes sont également justes). Si le diagnostic a été effectué automatiquement par le système avant de passer à l'étape suivante, il n'y a pas de retour à l'apprenant.
- Un ensemble de réponses fausses attendues. Pour chacune d'elles, l'expert doit spécifier les rétroactions associées. Dans notre exemple, cela peut être que la réponse reprend une phrase vraie de l'énoncé mais qui ne répond pas à la question, qu'il y a une erreur sur le temps (*a* au lieu de *avait*), que la réponse numérique est correcte mais pas la phrase, ou au contraire que la phrase est correcte mais que l'apprenant n'a pas reporté la valeur qu'il vient de calculer, qu'il s'agit de la réponse du modèle, etc. Dans chaque cas, un message spécifique est affiché à l'apprenant.
- La rétroaction à proposer à l'apprenant lorsque la réponse fausse ne correspond à aucun des cas prévus. Dans notre exemple, il s'agira d'envoyer à l'interface les

éléments lui permettant d'afficher en rouge les éléments erronés de la phrase de réponse et en vert ceux qui sont corrects.

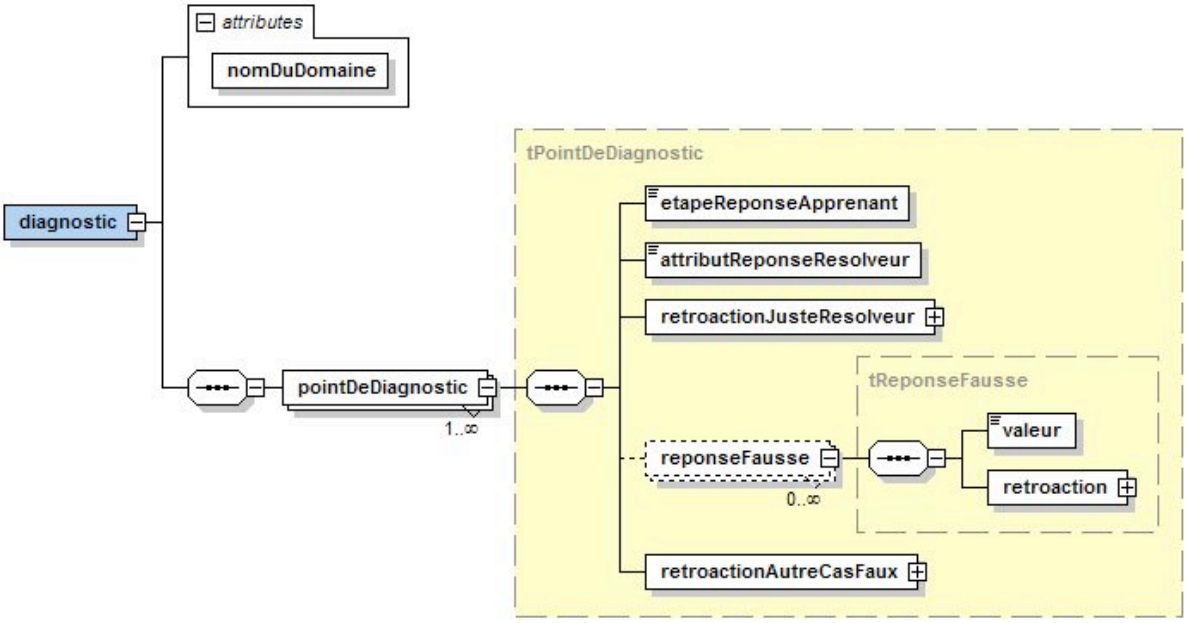

Figure 22. Méta-modèle de connaissances permettant le diagnostic de la réponse de l'apprenant et la construction d'une rétroaction appropriée.

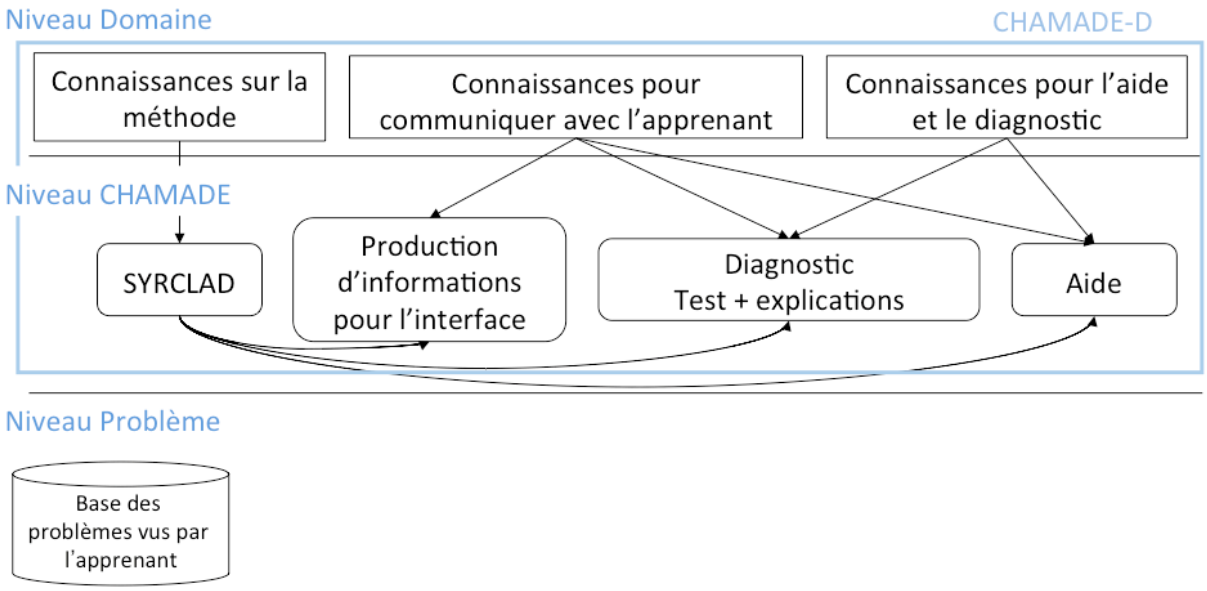

Figure 23. Architecture CHAMADE

La **Figure 23** présente l'architecture CHAMADE, où pour un domaine D donné il faut définir les connaissances correspondant aux trois méta-modèles présentés ci-dessus. En associant ces modèles de connaissances aux processus décrits dans la section suivante, on obtient un

système CHAMADE-D capable d'accompagner l'apprenant dans son utilisation d'un EIAH AMBRE-D.

3.2 Processus de raisonnement

L'ensemble des connaissances que nous venons de décrire peut être utilisé par le moteur de CHAMADE pour répondre à trois types de demande : une demande de l'interface de l'EIAH, une demande d'aide de l'apprenant ou une demande de diagnostic (de l'apprenant ou de l'interface). Je décris ci-dessous les processus permettant de répondre à ces demandes.

Production d'informations pour l'interface

L'interface de l'EIAH AMBRE fait appel à CHAMADE, et plus précisément au résolveur SYRCLAD, pour présenter à l'apprenant en début de séance des problèmes-types résolus. L'interface a également besoin de connaître les problèmes-types qu'a déjà vus l'apprenant pour lui demander de choisir un problème-type ; elle doit aussi savoir quels problèmes a déjà résolu l'apprenant et à quel problème-type il les a associés afin de lui permettre de consulter les problèmes rangés avec un problème-type. L'interface peut également faire appel à CHAMADE pour constituer des éléments d'interface. Par exemple pour les problèmes additifs, CHAMADE construit les listes de déroulantes utilisées par l'apprenant pour construire la phrase de réponse. Ces listes sont construites en fonction du problème, pour proposer à l'apprenant les éléments corrects ainsi que des éléments incorrects, par exemple sur le sujet de la phrase (les prénoms de l'énoncé et d'autres prénoms), les objets (ceux de l'énoncé - e.g. les autocollants - et ceux figurant dans d'autres problèmes tels que le problème-type), le verbe de la phrase (a, avait, a gagné, a perdu, a mangé, a fabriqué). Pour produire ces informations destinées à l'interface, ce processus (cf. **Figure 24**) utilise donc SYRCLAD et sa connaissance des problèmes, mais aussi des connaissances pour configurer l'interface figurant dans les connaissances pour communiquer avec l'apprenant.

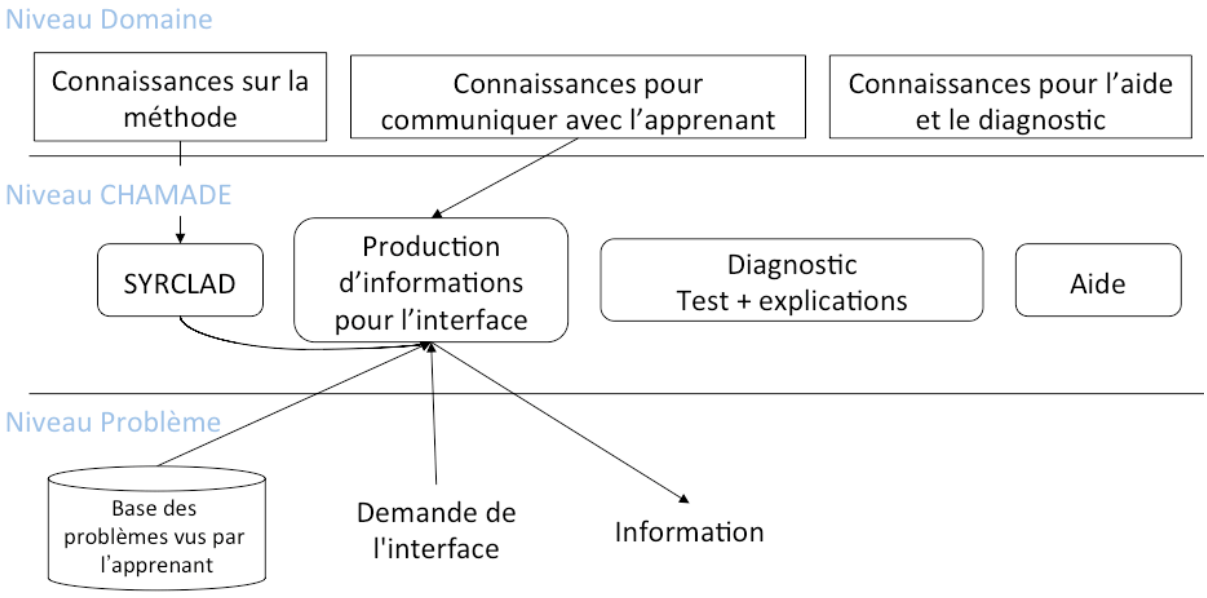

Figure 24. Production d'informations pour l'interface

Production d'aide

Lorsque l'apprenant demande de l'aide (cf. **Figure 25**), l'interface de l'EIAH indique à CHAMADE quel est exactement l'avancement de l'élève dans le cycle AMBRE. CHAMADE construit alors un message d'aide en utilisant ses connaissances sur le cycle AMBRE et ses connaissances liées à la langue naturelle. Afin de pouvoir mettre en évidence les éléments analogues du problème-type et du problème à résoudre, CHAMADE utilise également les connaissances sur les liens entre ces éléments ainsi que les connaissances du résolveur SYRCLAD sur les deux problèmes.

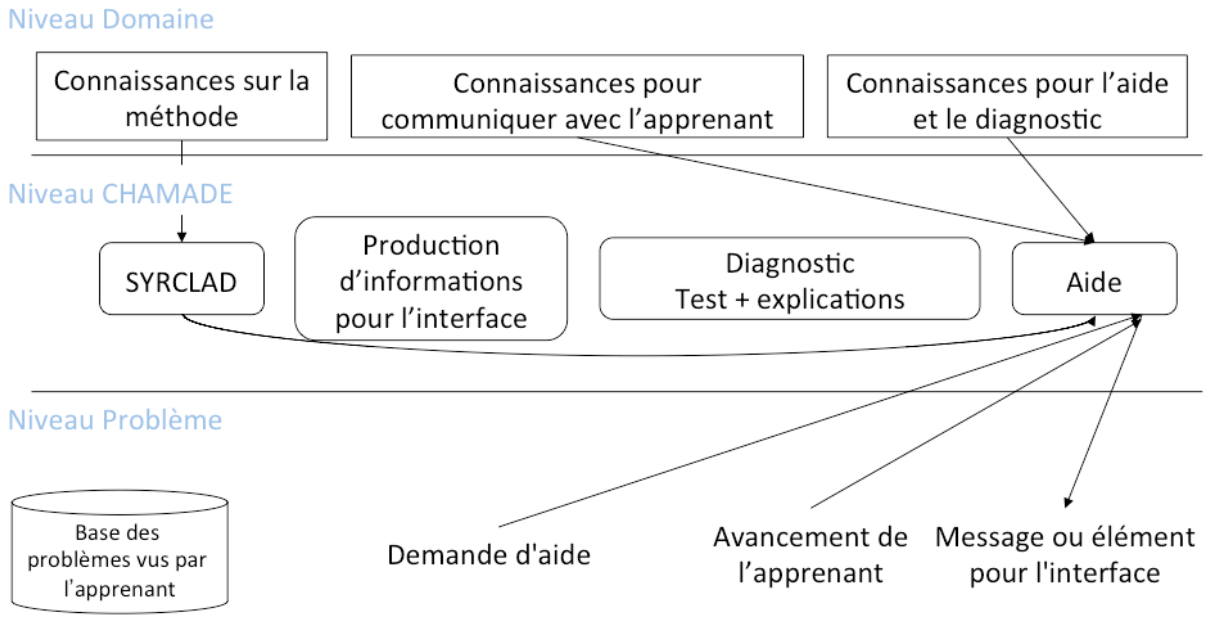

Figure 25. Production d'aide

Diagnostic des réponses et production d'explications

Lorsque l'apprenant demande un diagnostic de ses réponses (ou que l'interface demande ce diagnostic avant de laisser passer l'élève à l'étape suivante), CHAMADE procède en deux temps (cf. **Figure 26**). Dans un premier temps, le système utilise ses connaissances de diagnostic pour comparer les réponses de l'élève à celles du résolveur. Dans un deuxième temps, il construit en cas d'erreur un message d'explication ou des informations que l'interface mettra en œuvre visuellement. Pour cela, il utilise les connaissances d'explication des erreurs, en s'appuyant sur le résolveur SYRCLAD, ainsi que sur l'ensemble des connaissances destinées à la communication avec l'apprenant.

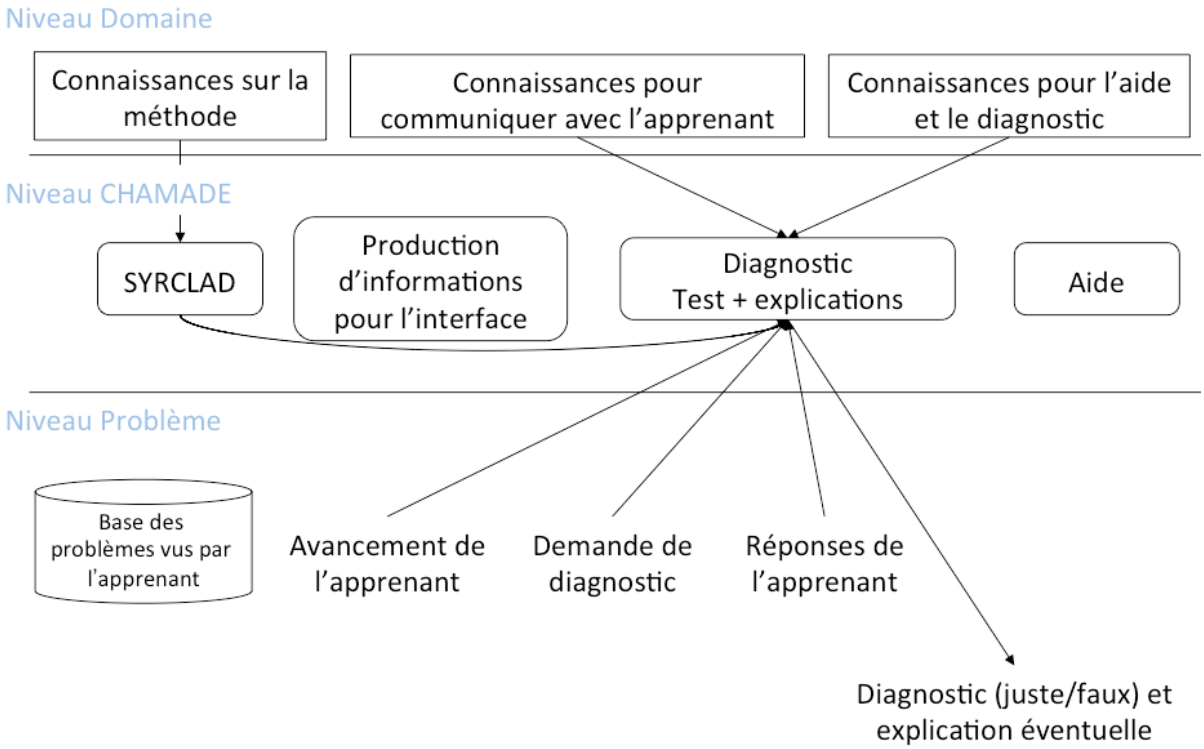

Figure 26. Diagnostic des réponses et production d'explications

Conclusion et perspectives

Ce chapitre a présenté l'architecture CHAMADE, qui permet d'éliciter les connaissances nécessaires pour obtenir un système capable d'accompagner l'apprenant dans son utilisation d'un EIAH AMBRE, en lui proposant de l'aide, diagnostiquant ses réponses, et lui fournissant des explications sur ses erreurs. Si l'architecture SYRCLAD a été évaluée sur plusieurs domaines, ce n'est pas le cas de CHAMADE, qui a été testée uniquement sur le domaine des problèmes additifs, même si elle a été conçue selon la même démarche, de manière indépendante du domaine. Pour le domaine des problèmes additifs, le système CHAMADE-add remplit entièrement les objectifs fixés en terme d'accompagnement de l'apprenant. Une application de CHAMADE au domaine de la conjugaison française est en cours, ce qui nous permettra de prouver la généricité de cette architecture.

Si les méta-modèles de l'architecture CHAMADE (comme ceux de SYRCLAD) sont définis afin de guider l'élicitation par l'expert des connaissances nécessaires au système, ces connaissances doivent pour l'instant être exprimées en langage Prolog, ce qui limite fortement l'utilisation de ces architectures pour concevoir des EIAH AMBRE. Notre objectif est maintenant de concevoir un outil permettant à un auteur de définir ces connaissances. Cet auteur sera le concepteur de l'EIAH mais ne sera pas nécessairement un informaticien. Il est donc nécessaire de proposer un formalisme s'appuyant sur les méta-modèles de CHAMADE ainsi qu'une interface qui permettrait à un utilisateur non informaticien d'éliciter les connaissances nécessaires aux systèmes à base de connaissances des EIAH AMBRE. Les connaissances explicitées par l'utilisateur pourront ensuite être traduites *via* cette interface en Prolog afin d'enrichir les bases de connaissances des EIAH AMBRE. Ce travail est présenté dans le chapitre A6 de ce mémoire.

Synthèse des contributions

Question de recherche : Éliciter les connaissances permettant à un EIAH AMBRE d'accompagner l'apprenant.

- **Contribution théorique** :
  - Méta-modèles de connaissances de diagnostic, d'aide, de production d'explications
  - Processus d'utilisation de ces connaissances pour répondre aux demandes de l'apprenant.
- **Système développé** :
  - Architecture CHAMADE intégrant un moteur de raisonnement
  - Définition des bases de connaissances pour les problèmes additifs pour obtenir le système CHAMADE-add.
- **Déploiement** : CHAMADE-add est intégré à l'EIAH AMBRE-add
- **Co-auteurs** : Marie Lefevre, Awa Diattara, Bryan Kong Win Chang

Publications liées

Chapitres d'ouvrages

Guin, N., Nogry, S., & Jean-Daubias, S. (2009). Le projet AMBRE : inviter un apprenant à adapter une expérience passée afin de l'amener à généraliser ses connaissances. In E. Egyed-Zsigmond, N. Guin, & A. Mille (Eds.), *Réutilisation de l'expérience : modèles et applications*. Hermès Sciences Publications.

Guin-Duclosson, N., Jean-Daubias, S., & Nogry, S. (2007). Le projet AMBRE : utiliser un paradigme d'apprentissage fondé sur le RÀPC pour faire acquérir à des élèves une méthode de résolution de problèmes. In J. Renaud, B. Chebel Morello, B. Fuchs, & J. Lieber (Eds.), *Raisonnement à partir de cas 2 - surveillance, diagnostic et maintenance* (pp. 203–226). Hermès-Lavoisier.

Conférences nationales

Duclosson, N. (2004). Représentation des connaissances dans l'EIAH AMBRE-add. *TICE 2004 - Colloque Technologies de L'information et de La Connaissance Dans L'enseignement Supérieur et L'industrie*. Compiègne, France.

Guin, N., & Lefevre, M. (2013). Systèmes à base de connaissances pour des EIAH fondés sur le RàPC Le projet AMBRE : s'inspirer du RàPC pour enseigner des méthodes. In *Journée EIAH&IA 2013*. Toulouse, France.

CHAPITRE A5
PERMETTRE À L'ENSEIGNANT D'ADAPTER L'EIAH À SES BESOINS

Résumé

Ce chapitre présente AMBRE-enseignant, un outil permettant à un enseignant d'adapter un EIAH AMBRE à ses besoins, en configurant le logiciel pour ses élèves, en générant des exercices et en construisant des séquences d'exercices destinées à ses élèves.

Ce module destiné à l'enseignant s'appuie sur l'architecture GenAMBRE, constituée par des méta-modèles de connaissance associés à un moteur de génération de problèmes. Cette architecture a permis de mettre en œuvre un générateur semi-automatique de problèmes additifs. Ce générateur crée des problèmes variés en fonction de contraintes fixées par l'enseignant.

INTRODUCTION 96

1 CADRE DES RECHERCHES : LE PROJET AMBRE 96

2 QUESTIONS DE RECHERCHE : COMMENT PERMETTRE À L'ENSEIGNANT D'AGIR SUR L'EIAH ? COMMENT GÉNÉRER DES PROBLÈMES ? 96

3 CONTRIBUTION : AMBRE-ENSEIGNANT 98

4 CONTRIBUTION : GENAMBRE, VERS LA GÉNÉRATION DE PROBLÈMES 99

4.1 QUEL TYPE DE GÉNÉRATEUR POUR AMBRE ? 99
4.2 L'ENVIRONNEMENT DE GÉNÉRATION DE PROBLÈMES VU PAR L'ENSEIGNANT 100
4.3 L'ARCHITECTURE GENAMBRE 102

5 ÉVALUATION DES CONTRIBUTIONS 105

CONCLUSION ET PERSPECTIVES 105

SYNTHÈSE DES CONTRIBUTIONS 106

POUR EN SAVOIR PLUS 106

PUBLICATIONS LIÉES 106

Introduction

On constate toujours que les enseignants utilisent peu d'EIAH avec leurs élèves. Une des raisons à cette faible intégration des EIAH dans les pratiques est que l'on ne donne pas aux enseignants la possibilité d'agir sur les environnements qu'on leur propose. Or, pour s'approprier un EIAH, il faut que l'enseignant puisse l'adapter au contexte d'apprentissage et à sa démarche pédagogique (Grandbastien, 1999; Major, 1995). Afin de favoriser l'utilisation des EIAH AMBRE en classe, nous avons étudié comment permettre à l'enseignant d'agir sur l'EIAH destiné à ses élèves, afin qu'il puisse l'adapter à ses besoins.

Ce travail a donné lieu à la conception d'un module destiné à l'enseignant pour l'EIAH AMBRE-add. Ce module, AMBRE-enseignant, permet à l'utilisateur de configurer l'EIAH destiné à être utilisé en classe par ses élèves. Nous avons, en collaboration avec des enseignants, identifié les fonctionnalités de AMBRE-enseignant nécessaires à l'adaptabilité de l'EIAH. Nous avons en particulier étudié comment permettre à l'enseignant de créer les problèmes de son choix pour les proposer à ses élèves dans l'EIAH.

1 Cadre des recherches : le projet AMBRE

Ce travail a été mené dans le cadre du projet AMBRE, et en particulier pendant le stage de DEA en informatique de Stéphanie Riot, au printemps 2004, avec la collaboration de Stéphanie Jean-Daubias.

2 Questions de recherche : Comment permettre à l'enseignant d'agir sur l'EIAH ? Comment générer des problèmes ?

Notre objectif était dans ce travail de proposer à l'enseignant un environnement lui permettant de paramétrer le logiciel destiné à ses élèves, de créer des séquences destinées à l'ensemble de sa classe ou à certains profils d'apprenants, et plus particulièrement de créer des problèmes à faire résoudre à ses élèves. Cet environnement, exclusivement destiné aux enseignants, devait donc répondre à leurs besoins, et leur permettre d'intégrer et d'adapter un EIAH AMBRE à leur démarche, à leurs stratégies pédagogiques, mais aussi aux contextes dans lesquels ils évoluent.

Cet objectif nécessitait donc d'identifier les fonctionnalités nécessaires aux enseignants et de les mettre en œuvre. Après avoir identifié la nécessité de permettre aux enseignants de créer de nouveaux problèmes, une deuxième question de recherche s'est posée : celle d'acquérir les connaissances nécessaires, dans un domaine donné, à un système générant des problèmes de ce domaine à partir de contraintes fixées par l'utilisateur. En effet, il ne s'agit pas uniquement de créer l'énoncé en langue naturelle du problème, mais également son modèle descriptif (cf. Chapitre A1), c'est-à-dire une représentation du problème compréhensible par les systèmes à bases de connaissances sous-jacents à un EIAH AMBRE. Il est en particulier nécessaire que le résolveur SYRCLAD intégré à l'EIAH puisse raisonner sur le problème créé afin de le résoudre. Il s'agit donc bien d'une problématique d'ingénierie des connaissances.

Ainsi, la **Figure 27** instancie l'approche générale présentée au chapitre 1 : il s'agit de permettre à un expert de définir des modèles de connaissances permettant la génération de problèmes. Ces modèles de connaissances sont conformes à des méta-modèles indépendants du domaine et sont exploités par un moteur associé aux méta-modèles.

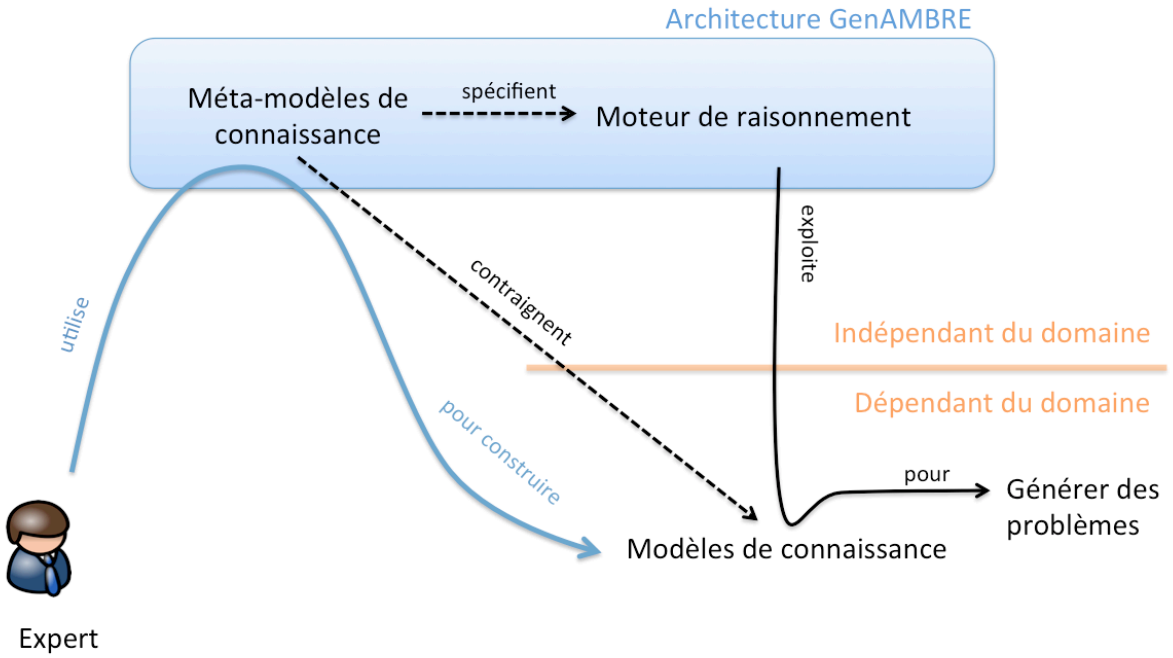

Figure 27. Instanciation de l'approche générale présentée dans le chapitre 1 à la question de l'élicitation de connaissances de génération de problèmes.

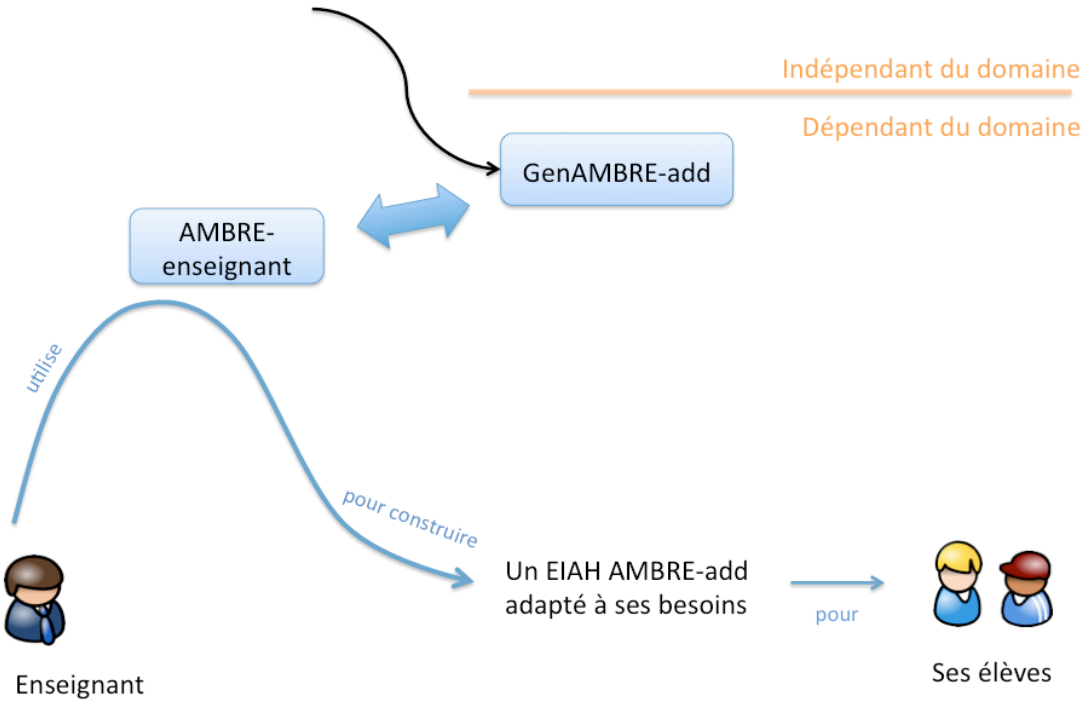

Figure 28. Utilisation de AMBRE-enseignant

Dans le domaine des problèmes additifs, ces modèles de connaissances ont été définis. En les associant au moteur qui les exploite, on obtient GenAMBRE-add, un système capable de générer des problèmes additifs conformes à un ensemble de contraintes.

La **Figure 28** présente alors comment, de manière analogue à l'approche générale, l'enseignant utilise les modèles de connaissance de GenAMBRE-add *via* l'interface de AMBRE-enseignant, pour construire un EIAH AMBRE-add adapté à ses besoins. Ce faisant, il définit en particulier des modèles de problèmes représentés par l'ensemble des contraintes que les problèmes générés devront respecter.

La suite de ce chapitre présente tout d'abord le logiciel AMBRE-enseignant, avant de décrire plus précisément l'architecture GenAMBRE.

3 Contribution : AMBRE-enseignant

Grâce à des entretiens avec des enseignants, nous avons identifié les fonctionnalités dont ils souhaitaient pouvoir disposer dans un logiciel qui leur serait destiné. Il s'agissait de générer des problèmes, de créer des séquences d'apprentissage (ensemble d'activités destinées aux apprenants), mais aussi de paramétrer l'environnement élève, de créer les listes d'élèves des classes et de distribuer le travail (séquences et exercices) aux classes ou aux élèves. Ces fonctionnalités ont ensuite été mises en œuvre dans le logiciel AMBRE-enseignant.

Le paramétrage de l'environnement élève consiste en la personnalisation par l'enseignant de l'interface du logiciel élève : choix des couleurs et de la langue principalement. AMBRE-enseignant doit également permettre aux enseignants d'établir la liste des élèves de leur(s) classe(s).

Un certain nombre d'exercices, créés par les concepteurs, sont fournis avec le logiciel élève. Mais afin de permettre aux enseignants qui le désirent de faire travailler leurs élèves sur des problèmes dont ils souhaitent préciser les caractéristiques, AMBRE-enseignant comporte un module de **génération de problèmes**. Cette génération doit être un plus pour les enseignants et non une obligation. Ce passage semble toutefois nécessaire pour augmenter et diversifier la batterie d'exercices de l'EIAH, et adapter les énoncés au public à former. Générer un problème à résoudre dans un EIAH AMBRE consiste à proposer un énoncé en langue naturelle ainsi qu'une description du problème compatible avec le résolveur utilisé par le logiciel élève. Du point de vue de l'enseignant, il s'agit de préciser un certain nombre de caractéristiques que doit comporter le problème. La génération du problème peut être plus ou moins automatisée selon le choix de l'enseignant : il peut préciser toutes les caractéristiques du problème, certaines seulement, ou au contraire aucune.

AMBRE-enseignant offre également la possibilité à l'enseignant de **créer des séquences** d'apprentissage (ensemble de problèmes à résoudre dans une ou plusieurs séances d'utilisation du logiciel) en utilisant le matériel pédagogique (les problèmes) qu'il a créé. Cette fonctionnalité lui permet de mettre en place sa stratégie, sa démarche pédagogique, en intégrant comme il le souhaite les problèmes de son choix dans une séquence. On propose pour cela à l'utilisateur deux manières de créer une séquence : manuelle ou automatisée. Pour la création manuelle, l'enseignant sélectionne de façon chronologique ou de façon aléatoire les exercices qu'il souhaite intégrer dans la séquence qu'il crée. Il peut par exemple choisir un ordre de présentation des problèmes en fonction de leur niveau de difficulté. L'enseignant peut ensuite définir le comportement du logiciel élève pour les

exercices de la séquence : nombre d'essais autorisés par étape de l'exercice (et que faire si ce nombre est atteint ?), nombre d'accès à la vérification ou à l'aide autorisés pendant l'exercice, fonctionnement du diagnostic (toujours diagnostiquer ou laisser l'apprenant se tromper à certaines étapes de la résolution), etc. On définit par défaut ce comportement pour tous les exercices de la séquence. Toutefois, si l'enseignant souhaite intégrer une progression dans la séquence, il a la possibilité de définir un comportement différent en fonction des exercices (il peut ainsi par exemple mettre un diagnostic maximum en début de séquence, et le diminuer au fur et à mesure). Pour la création automatisée d'une séquence, l'enseignant ne choisit que le(s) dossier(s) d'exercices dans lequel le système prendra les exercices, ainsi que le nombre d'exercices qu'il souhaite voir figurer dans la séquence. Charge ensuite au système de choisir aléatoirement les exercices. L'enseignant peut ensuite s'il le souhaite modifier les propositions du système (supprimer, ajouter ou remplacer un problème), ou ordonner les exercices différemment, définir un comportement particulier, etc.

AMBRE-enseignant permet aussi de **distribuer le travail** aux élèves, c'est-à-dire associer à l'ensemble de la classe, à un groupe d'élèves ou à chaque élève une ou plusieurs séquences. L'enseignant peut en effet donner les mêmes séquences à tous les élèves ou choisir d'individualiser l'enseignement en proposant des séquences spécifiques à certains élèves.

Nous allons à présent étudier de plus près la partie de AMBRE-enseignant dédiée à la génération de problèmes.

4 Contribution : GenAMBRE, vers la génération de problèmes

Dans cette section, j'explique tout d'abord les raisons de notre choix d'un procédé semi-automatique de génération de problèmes pour AMBRE, après quoi je décris le fonctionnement du générateur de problèmes de AMBRE-add, du point de vue de l'enseignant qui l'utilise, et je présente enfin l'architecture GenAMBRE qui a permis de créer ce générateur.

4.1 Quel type de générateur pour AMBRE ?

Dans la plupart des EIAH fondés sur la résolution de problèmes, les exercices proposés aux apprenants sont issus d'une bibliothèque prédéfinie. Leur diversité et leur nombre en sont alors limités. Une solution répondant à ce problème consiste en l'utilisation de générateurs de problèmes. Si l'on considère la place de l'auteur dans le processus de création des exercices et la capacité du système à générer automatiquement des variantes d'un exercice, on peut répartir les générateurs de la littérature en trois grandes familles.

La première regroupe les **générateurs automatiques** (Bouhineau & Nicaud, 2006; Mostow et al., 2004; Selva, 2002). Avec ce genre de générateurs, un grand nombre d'exercices est créé automatiquement sans que l'auteur puisse influencer les choix du système. Il peut simplement choisir la catégorie de l'exercice (forme, thème, connaissances abordées) mais ne peut pas agir sur le contenu ni sur des contraintes précises. Ce type de systèmes offrant peu de liberté à l'enseignant, ce n'est pas l'approche que nous avons retenue pour AMBRE-enseignant.

La deuxième grande famille de générateurs d'exercices est celle des **générateurs manuels** (Cogne, David, & Lacombe, 1998). Souvent utilisés dans le cadre des outils auteurs et dans

les plateformes comme Moodle ou Claroline, ces générateurs laissent une grande liberté à l'auteur. À l'inverse des générateurs automatiques, les générateurs manuels ne construisent pas eux-mêmes les énoncés des problèmes, ni leur solution. C'est à l'enseignant que revient cette tâche. Il construit chaque énoncé, en précisant la ou les réponses attendues. Le système ne possède aucune connaissance de l'exercice qu'il propose, les réponses étant préenregistrées. Le système n'est donc pas apte à résoudre les problèmes qu'il permet de créer et n'offre pas de fonctionnalité d'aide ou de diagnostic des réponses de l'apprenant. Le seul diagnostic possible consiste à indiquer à l'apprenant s'il a répondu juste ou faux aux questions posées. Quant à l'aide, elle doit être préalablement définie par l'enseignant. Cette approche, qui laisse une totale liberté à l'enseignant, ne nous convenait toutefois pas dans le cadre du projet AMBRE. En effet, dans un EIAH AMBRE, les problèmes posés à l'apprenant doivent donc être compréhensibles par le résolveur SYRCLAD, afin que l'EIAH puisse fournir aide, diagnostic et explications à l'apprenant. Par conséquent, il n'est pas possible de laisser l'enseignant saisir un énoncé en langue naturelle sur lequel le résolveur ne pourrait pas raisonner.

Nous avons donc choisi une approche relevant de la troisième famille, celle des **générateurs semi-automatiques,** qui construisent eux-mêmes les énoncés des problèmes, mais en laissant intervenir l'utilisateur dans le processus de création. L'enseignant a donc la possibilité, avec de tels systèmes, d'influer sur les problèmes qui seront générés, en spécifiant un ensemble de contraintes sur l'exercice à générer.

Pour AMBRE-enseignant, le générateur de problèmes doit permettre à l'enseignant d'adapter les problèmes produits à son contexte d'apprentissage et à sa démarche pédagogique. Il est donc important que celui-ci puisse caractériser les exercices à générer, c'est-à-dire saisir un ensemble de contraintes qui spécifieront ces problèmes. Les problèmes étant construits par le système à partir de ces contraintes, le résultat de la génération sera non seulement un énoncé en langue naturelle, mais aussi un modèle du problème qui sera compréhensible par le résolveur.

Le projet AMBRE a pour objectif la conception d'EIAH pour apprendre des méthodes dans des domaines variés. Il est donc important que l'approche choisie pour la génération de problèmes soit indépendante du domaine. Nous avons donc conçu une architecture indépendante du domaine, GenAMBRE, à laquelle il faut fournir des bases de connaissances spécifiques à un domaine donné pour obtenir un générateur de problèmes pour ce domaine.

La section suivante présente l'outil de génération de problèmes tel que nous l'avons élaboré pour le domaine des problèmes additifs, puis nous décrivons l'architecture GenAMBRE qui permet de réaliser un tel générateur de problèmes.

4.2 L'environnement de génération de problèmes vu par l'enseignant

Notre objectif concernant le module de génération de problèmes était de permettre à un enseignant de créer ses propres problèmes et de les adapter au public à qui il les destine. De plus, il nous semblait important de pouvoir lui permettre de construire avec exactitude un énoncé souhaité, ou au contraire qu'il puisse demander au système d'en construire un automatiquement. Pour arriver à une telle souplesse, nous avons décidé d'utiliser l'approche de la génération à partir de contraintes : l'enseignant définit les contraintes qu'il souhaite voir vérifiées par le problème que le système va construire.

Afin que la tâche de création de problèmes par l'enseignant soit simplifiée, nous voulions lui éviter d'avoir à créer un à un les nouveaux énoncés. Nous avons donc décidé qu'un même jeu de contraintes pourrait permettre la création de plusieurs exercices. Moins il y aura de contraintes, plus les problèmes générés seront variés.

Pour le domaine des problèmes additifs, les contraintes que l'enseignant peut définir relèvent de quatre catégories différentes : traits de structure, traits de surface, valeurs et complication.

La **structure** d'un problème à générer correspond à la classe du problème. Cette classe est définie par plusieurs attributs qui peuvent ou non être fixés.

Les **traits de surface** sont les éléments qui permettent d'habiller les énoncés produits. L'enseignant peut préciser certains éléments de cette catégorie, par exemple les thèmes, les objets et les personnages pour les problèmes additifs.

Pour les problèmes additifs, l'enseignant peut choisir les **valeurs numériques** des données qui seront utilisées dans les problèmes. Il peut définir un intervalle global de valeurs pour les éléments et l'écart souhaité entre les deux valeurs (également par un intervalle). Une autre possibilité consiste à fixer une valeur ou un intervalle pour chaque élément connu de l'énoncé. L'enseignant peut également, s'il veut simplifier les opérations, interdire l'utilisation de la retenue dans les calculs.

La **complication** concerne toutes les options qui permettront de compliquer l'énoncé du problème pour l'adapter au niveau des élèves. Cette partie a nécessité une collaboration étroite avec les enseignants, afin de connaître leurs besoins. Pour les problèmes additifs, l'environnement propose différents types de difficultés à intégrer dans les exercices : des complications de la langue (difficulté du vocabulaire employé et des tournures des phrases) et de l'énoncé lui-même (écriture des nombres en lettres, modification de l'ordre des propositions de l'énoncé, ajout de phrases inutiles à la résolution, introduisant éventuellement dans le problème des données non pertinentes).

Précisons qu'aucune des contraintes citées précédemment n'est obligatoire pour la création d'exercices. Les contraintes non spécifiées par l'enseignant seront définies aléatoirement par le système. L'enseignant peut de plus attribuer aux problèmes créés un nom, un niveau (par exemple CE1 1er trimestre) et une description, afin de les retrouver plus facilement pour la création de séquences.

Ces quatre catégories de contraintes définies pour les problèmes additifs ne seront pas réutilisables telles quelles pour un autre domaine d'application de AMBRE. Néanmoins, il est certain que les catégories « traits de structure » et « traits de surface » seront présentes dans tous les domaines. La catégorie « valeurs » sera uniquement présente pour les domaines numériques, alors que la catégorie « complication » sera probablement utile dans tous les domaines. Les contraintes figurant dans chacune de ces catégories dépendront bien évidemment du domaine.

La **Figure 29** présente l'écran de bilan du module de génération de AMBRE-add. Cet écran résume les contraintes définies par l'enseignant pour les quatre catégories (structure, traits de surface, valeurs et complication) et présente un exemple de problème qui pourra être généré à partir de ces contraintes. Le schéma de la partie "structure" sert à représenter la classe du problème.

La section suivante présente l'architecture GenAMBRE qui a permis de mettre en œuvre le générateur de problèmes de AMBRE-add.

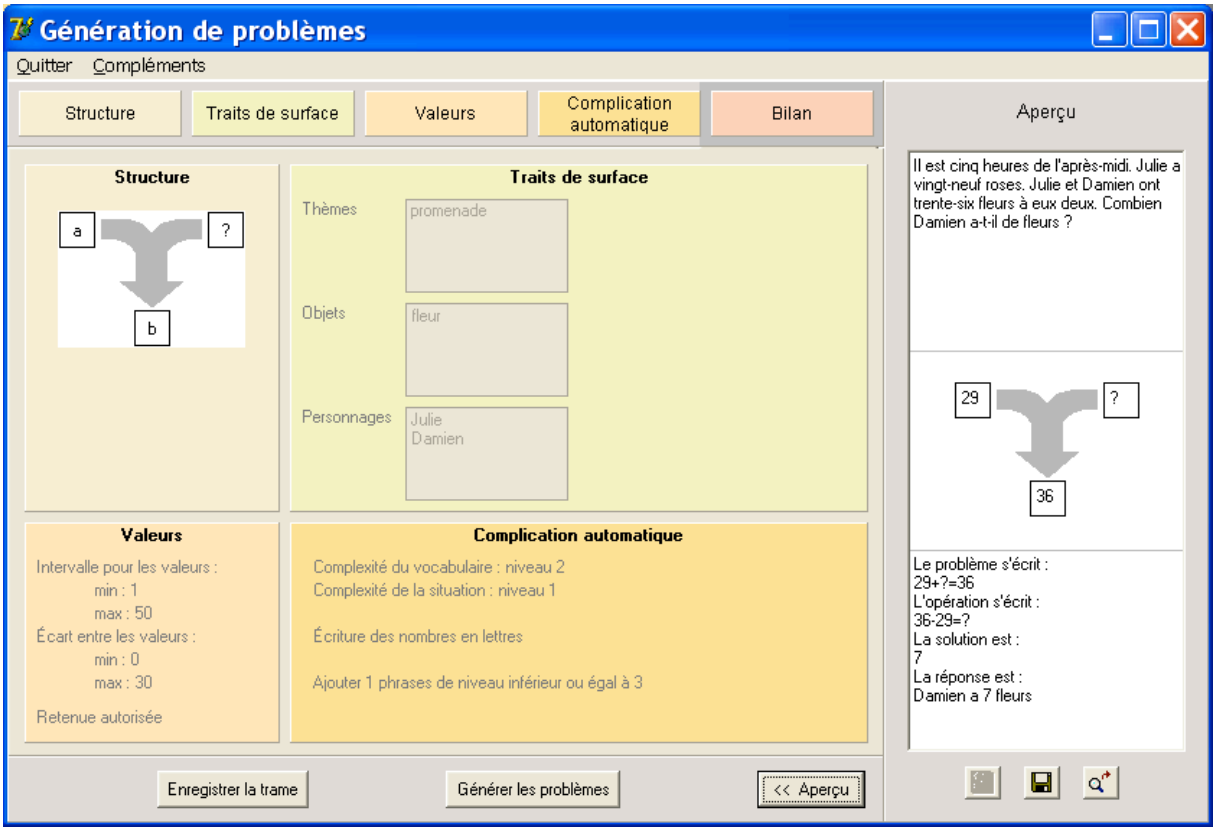

Figure 29. L'écran de bilan du module de génération de problèmes de AMBRE-enseignant.

4.3 L'architecture GenAMBRE

Le moteur de génération de problèmes que nous avons mis en œuvre dans l'architecture GenAMBRE prend en entrée le jeu de contraintes spécifié par l'enseignant et fournit en sortie deux éléments : l'énoncé du problème en langue naturelle, destiné aux apprenants, et le modèle descriptif du problème, destiné au résolveur SYRCLAD (cf. Chapitre A1).

L'architecture GenAMBRE est présentée en **Figure 30** et chacun de ses constituants est décrit dans la suite du texte. Pour un domaine D (par exemple celui des problèmes additifs), il faut définir les cinq bases de connaissances du "niveau domaine" à partir des méta-modèles de connaissance. Le processus de génération du problème s'effectue en deux temps : le moteur construit un modèle de génération du problème, puis construit l'énoncé en langue naturelle ainsi que le modèle descriptif du problème. Ces deux processus sont indépendants du domaine. Associés aux bases de connaissances du domaine D, ils forment un générateur de problèmes pour le domaine D : GenAMBRE-D.

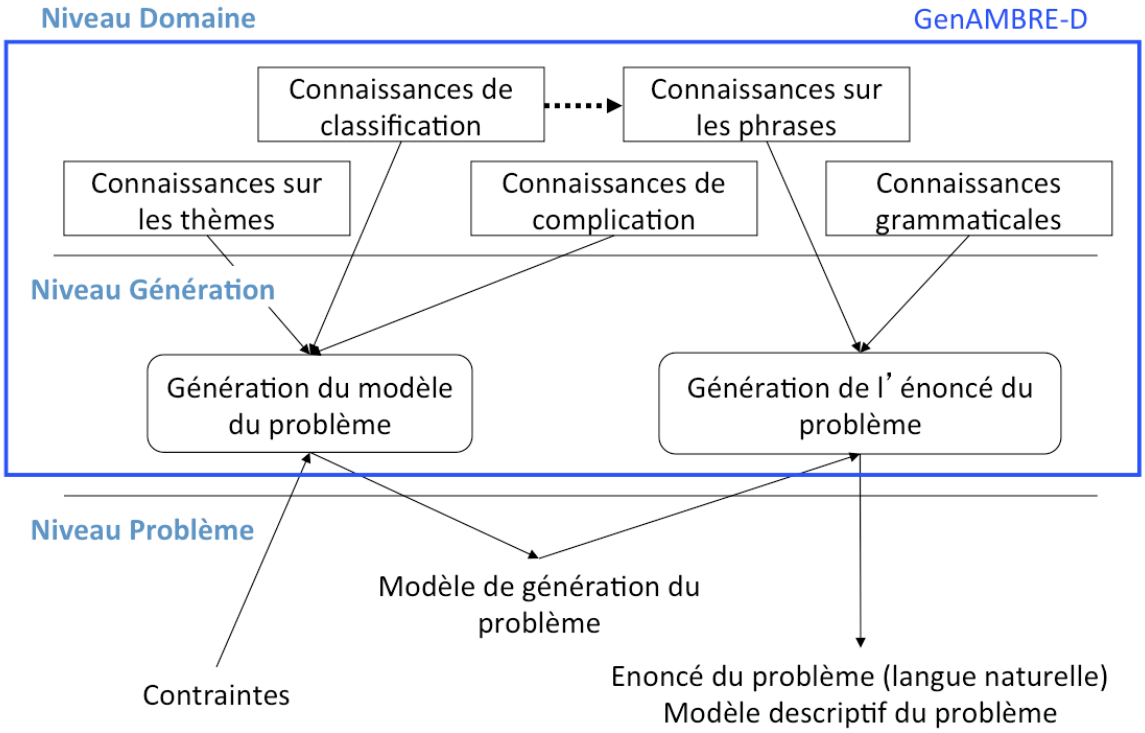

Figure 30. L'architecture GenAMBRE

Méta-modèles de connaissances

Connaissances de classification

Dans chaque domaine d'application, un expert fournit au résolveur SYRCLAD, et donc par extension au générateur GenAMBRE, un graphe de classification de problèmes. Ce graphe est utilisé dans le résolveur pour classer le problème. C'est une hiérarchie de classes dépendante du domaine, mais dont la représentation est la même quel que soit le domaine.

Connaissances sur les thèmes

Pour pouvoir générer un problème, nous avons besoin de connaître le thème concerné et les traits de surface qui lui sont associés (par exemple les objets, personnages et actions pour le domaine des problèmes additifs). De telles connaissances, fournies par l'expert, sont dépendantes du domaine pour lequel elles sont utilisées, mais leur représentation est indépendante du domaine.

Connaissances de complication

Pour pouvoir compliquer un exercice et donc son énoncé, il est nécessaire de fournir des connaissances de complication au système de génération. Pour les problèmes additifs, compliquer un énoncé revient principalement à changer l'ordre des phrases et à ajouter des phrases inutiles. Les connaissances de complication permettent alors de répondre aux questions suivantes : comment et dans quel cas est-il possible de changer l'ordre des phrases utiles du problème ? Quelles phrases inutiles peut-on ajouter aux problèmes et où les placer ?

Connaissances grammaticales

L'expert du domaine doit fournir une grammaire au système de génération, c'est-à-dire un ensemble de structures de phrases qui pourront être utilisées dans le domaine. Dans le cadre des problèmes additifs, cette grammaire permet de générer des phrases affirmatives et interrogatives. Les unités syntaxiques de base sont les groupes sujets, les verbes et les compléments. Cette grammaire est dépendante du domaine dans le sens où, pour un domaine donné, on ne prévoit pas de pouvoir créer toutes les phrases de la langue française. Par exemple, pour le domaine des problèmes additifs, nous n'utilisons pas de propositions relatives ni d'adjectifs. Lorsque nous appliquerons GenAMBRE à d'autres domaines, nous pourrons probablement avoir une seule grammaire dont nous spécifierons pour chaque domaine les parties à utiliser.

Connaissances sur les phrases

Les énoncés générés, notamment la structure de leurs phrases, dépendent de la classe à laquelle le problème appartient. Il est donc nécessaire pour pouvoir générer l'énoncé en langue naturelle, de savoir quelles structures de phrases (de la grammaire) pourront être utilisées pour le problème à générer. Ce sont ces informations que contiennent les connaissances sur les phrases. Elles permettent d'associer à la classe du problème les structures de phrases utilisables, et les éléments du problème qui seront associés à ces structures.

Processus de raisonnement

Processus de génération du modèle du problème

À partir des connaissances de classification, des connaissances sur les thèmes, des connaissances de complication et des contraintes définies par l'enseignant, le moteur doit générer ce que nous appelons le modèle de génération du problème. Ce modèle est en fait un modèle descriptif étendu, puisqu'il contient le modèle descriptif du problème mais précise également quelques informations absentes du modèle descriptif, tels que la classe du problème ou encore son thème. Pour construire ce modèle, le moteur complète les contraintes définies par l'enseignant, notamment en choisissant aléatoirement des valeurs pour celles qui n'ont pas été précisées.

Processus de génération de l'énoncé du problème

Pour générer un énoncé en langue naturelle, le moteur utilise les connaissances sur les phrases et la grammaire du domaine, ainsi que le modèle de génération du problème créé auparavant. Les connaissances sur les phrases permettent au moteur de prendre les décisions d'ordre conceptuel (décider quoi dire) : on spécifie ici la structure des phrases qui seront générées en précisant pour chaque phrase les éléments qui la composeront (issus du modèle du problème : objets, personnages…). Une fois les décisions d'ordre conceptuel prises, le moteur passe à l'étape de génération de texte, et met donc en place différents mécanismes linguistiques : traitements syntaxiques (décider comment le dire), lexicaux et morphologiques (décider comment l'écrire).

5 Évaluation des contributions

Même si nous avons intégré des enseignants à la conception de AMBRE-enseignant, il faudrait évaluer ce logiciel en situation réelle auprès d'un nombre significatif d'enseignants.

En ce qui concerne l'architecture GenAMBRE, elle n'a, comme CHAMADE, été appliquée qu'au seul domaine des problèmes additifs. Il faudrait l'utiliser pour créer un générateur de problèmes pour un autre domaine afin de montrer sa généricité.

Pour ce qui est de GenAMBRE-add, ce générateur permet de créer des problèmes variés et satisfaisant les contraintes définies par l'utilisateur.

Pour les contraintes définies par l'enseignant dans l'exemple de la **Figure 29**, le générateur de problèmes lui propose l'énoncé suivant : « Il est cinq heures de l'après-midi. Julie a vingt-neuf roses. Julie et Damien ont trente-six fleurs à eux deux. Combien Damien a-t-il de fleurs ? ». Si l'on demande au système de générer un autre problème à partir des mêmes contraintes, il pourra proposer : « Damien a dix billes. Il a dix-sept fleurs. Si Damien et Julie mettent leurs fleurs ensemble, ils en ont trente-cinq. Combien Julie a-t-elle de fleurs ? ». Ces deux énoncés sont proches parce que l'enseignant a dans cet exemple défini des contraintes très précises (Julie et Damien qui ont des fleurs).

Si l'enseignant choisit la même classe de problème dans l'onglet "structure" que dans l'exemple précédent, mais ne définit aucune contrainte supplémentaire dans les onglets "traits de surface", "valeurs" et "complication", le générateur produira des énoncés plus variés, tels que ceux ci-dessous :

• Damien est dans la cour de récréation avec Paul. Il a 32 billes vertes. Si Damien et Paul mettent leurs billes vertes ensemble, ils en ont 46. Cherche le nombre de billes vertes de Paul.

• Il est 11h. Sophie a 25 billes. Sophie et Romain en ont 38 à eux deux. Calcule le nombre de billes de Romain.

• Trouve le nombre de roses rouges de Sabrina, sachant qu'en ajoutant les roses rouges de Sabrina, Julie et Sabrina en ont 40 à elles deux, qu'elles ont 10 ans et qu'elles sont amies, et que Julie a 7 roses rouges.

• La semaine dernière, c'était l'anniversaire de Jean, maintenant il a 11 ans. Il a 9 billes jaunes. Quand Jean et Julie mettent leurs billes jaunes ensemble ils en ont 37. Trouve le nombre de billes jaunes de Julie.

Conclusion et perspectives

Ce chapitre a permis de présenter un module destiné à l'enseignant pour l'EIAH AMBRE-add, pour donner la posibilité à l'enseignant d'adapter l'EIAH au contexte d'apprentissage et à sa démarche pédagogique. En adoptant une démarche générique, nous avons identifié en concertation avec des enseignants les fonctionnalités que doit posséder un module enseignant pour un EIAH AMBRE (AMBRE-enseignant). Nous avons ainsi prévu que l'enseignant puisse paramétrer l'environnement, générer des problèmes de son choix, créer des séquences d'apprentissage adaptées à ses élèves en choisissant les problèmes et le comportement de l'EIAH, et distribuer ces séquences à ses élèves.

Nous avons pour cela conçu un système de génération de problèmes dont l'architecture est indépendante du domaine. Il s'agit de l'architecture GenAMBRE qui s'appuie sur des méta-modèles de connaissances qu'il conviendrait maintenant de tester sur un autre domaine que celui des problèmes additifs.

Ce travail sur l'architecture GenAMBRE a préfiguré d'autres travaux sur la génération semi-automatique d'exercices menés ensuite dans le cadre de la thèse de Marie Lefevre et du projet CLAIRE. Ces travaux feront l'objet du chapitre B2 de ce mémoire.

Synthèse des contributions

Question de recherche : Permettre à l'enseignant d'agir sur un EIAH AMBRE et de générer des problèmes.

- **Contributions théoriques** :
  - Méta-modèles de connaissance pour la génération de problèmes
  - Processus de génération associé aux méta-modèles
- **Systèmes développés** :
  - Architecture GenAMBRE, incluant un moteur de génération, et son application au domaine des problèmes additifs
  - AMBRE-enseignant
- **Co-auteurs** : Stéphanie Jean-Daubias, Stéphanie Riot

Pour en savoir plus

Duclosson, N., Jean-Daubias, S., & Riot, S. (2005b). *AMBRE-enseignant : un module partenaire de l'enseignant pour faciliter l'intégration de AMBRE en classe*. Rapport de recherche LIRIS RR-LIRIS-2005-007.

Publications liées

Conférence internationale

Jean-Daubias, S., & Guin, N. (2009). AMBRE-teacher: a module helping teachers to generate problems. In *2nd Workshop on Question Generation, AIED'09 - Artificial Intelligence in Education*. Brighton, Great Britain.

Conférence nationale

Duclosson, N., Jean-Daubias, S., & Riot, S. (2005a). AMBRE-enseignant : un module partenaire de l'enseignant pour créer des problèmes. In *Environnements Informatiques pour l'Apprentissage Humain (EIAH'2005)* (pp. 353–358). Montpellier, France.

CHAPITRE A6
VERS UN OUTIL AUTEUR POUR AMBRE

Résumé

Ce chapitre décrit les travaux en cours pour la conception d'un outil auteur permettant à un utilisateur non informaticien d'éliciter les connaissances nécessaires à un EIAH AMBRE, c'est-à-dire les modèles de connaissances de l'architecture CHAMADE. Je présente un état de l'art par rapport à cette problématique ainsi que les questions de recherche qui en découlent et la démarche envisagée pour y répondre.

Ce chapitre s'appuie sur des notions présentées dans les chapitres A1, A3 et A4.

INTRODUCTION 108

1. CADRE DES RECHERCHES 108

2. QUESTION DE RECHERCHE : ACQUÉRIR LES CONNAISSANCES NÉCESSAIRES À UN EIAH AMBRE 108

3. ÉTAT DE L'ART 109

4. DÉMARCHE ENVISAGÉE 111

CONCLUSION ET PERSPECTIVES 113

PUBLICATIONS LIÉES 114

Introduction

Les résultats des recherches menées dans le cadre du projet AMBRE et présentées dans les chapitres précédents de ce mémoire permettent la conception d'EIAH destinés à l'enseignement de méthodes. Ces EIAH proposent à l'apprenant un processus d'apprentissage (le cycle AMBRE), s'appuient sur des systèmes à base de connaissance permettant d'accompagner l'apprenant (grâce à l'architecture CHAMADE), et peuvent proposer à l'enseignant un module lui permettant d'adapter l'EIAH à ses besoins et de générer des problèmes à poser à ses élèves.

La conception de tels EIAH est cependant très coûteuse, en particulier du fait que les connaissances nécessaires doivent être élicitées en Prolog. Pour remédier à ce problème, et pour poursuivre mes recherches sur l'acquisition de connaissances, j'ai souhaité entreprendre des recherches visant à concevoir un outil auteur pour les EIAH AMBRE. L'objectif d'un tel outil auteur est de permettre à un utilisateur non informaticien, souhaitant concevoir un EIAH AMBRE, d'éliciter les connaissances nécessaires à l'architecture CHAMADE.

1 Cadre des recherches

Ces travaux sont menés dans le cadre du projet AMBRE. Après avoir effectué un état de l'art et conduit des travaux exploratoires dans le cadre d'un TER, j'ai obtenu pour ce projet un financement de l'ARC 6 de la Région Rhône-Alpes incluant une bourse de thèse. Il s'agit de la thèse d'Awa Diattara, commencée en novembre 2013, dont la directrice est Vanda Luengo (laboratoire LIG à Grenoble), avec la contribution également d'Amélie Cordier (laboratoire LIRIS).

2 Question de recherche : acquérir les connaissances nécessaires à un EIAH AMBRE

L'objectif de ce travail est de concevoir un outil auteur générique permettant à des enseignants ou à des pédagogues plus experts mais non informaticiens de construire eux-mêmes l'EIAH AMBRE correspondant à leurs besoins dans le domaine qui les intéresse.

Les outils auteurs ont souvent été présentés comme une réponse au goulot d'étranglement qu'est l'acquisition des connaissances pour la conception de tuteurs intelligents. Les connaissances à acquérir ici sont celles nécessaires aux EIAH AMBRE, c'est-à-dire les connaissances correspondant aux méta-modèles de l'architecture CHAMADE (qui inclut rappelons-le l'architecture SYRCLAD).

On retrouve bien alors l'approche décrite dans le chapitre 1 de ce mémoire. La **Figure 31** illustre cette approche dans le cadre de ce chapitre. Par rapport aux figures analogues des chapitres A1 (SYRCLAD) ou A4 (CHAMADE), la différence vient du fait que l'élicitation des connaissances s'effectue *via* une interface, l'outil auteur, et non plus directement en Prolog. Cette différence traduit l'évolution de mes recherches initialement centrées sur la représentation des connaissances de l'expert au sein du système informatique, vers l'acquisition par le système des connaissances auprès de l'utilisateur. L'utilisateur n'a plus

alors à être un expert (au sens expert des connaissances *et* capable de définir les connaissances en Prolog), mais peut être ce que j'appellerai le concepteur de l'EIAH : enseignant, pédagogue, didacticien, donc expert de la méthode qu'il souhaite enseigner, sans être nécessairement informaticien. Un des enjeux est également l'assistance que le système pourra offrir à l'utilisateur de manière à lui faciliter la tâche dans l'élicitation des connaissances *via* l'outil auteur.

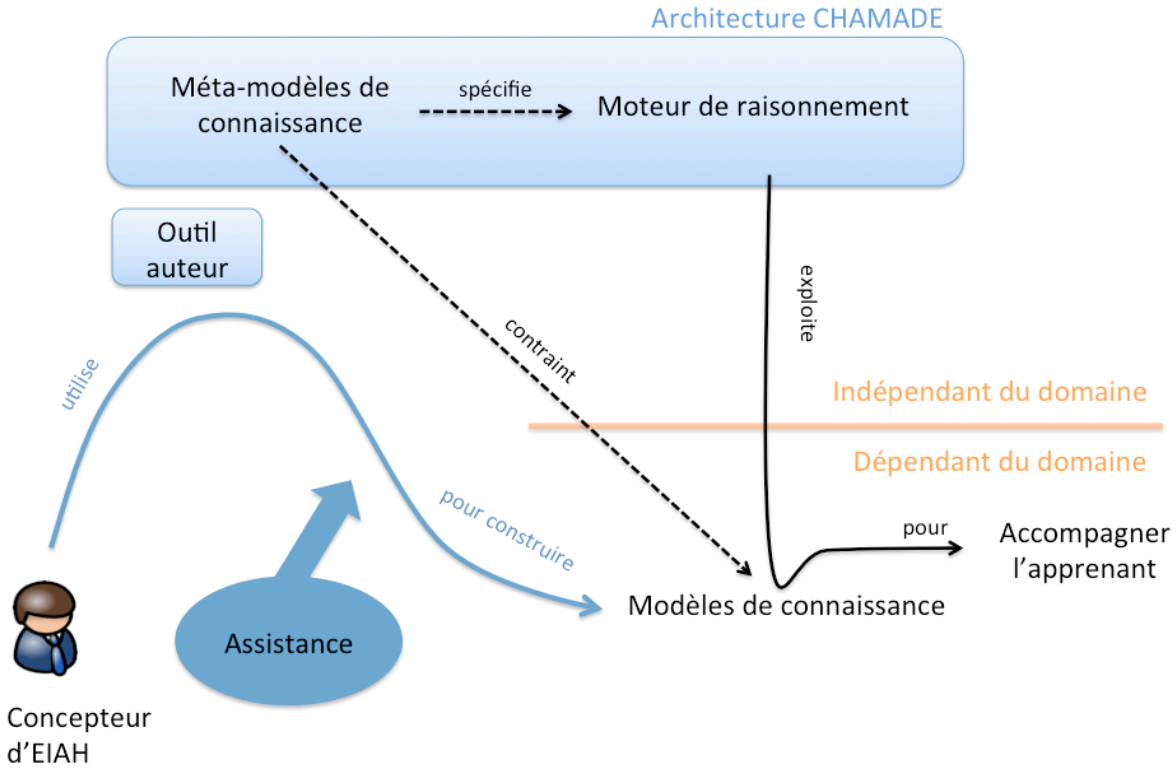

Figure 31. Instanciation de l'approche générale présentée dans le chapitre 1 à la question de l'acquisition des connaissances nécessaires à un EIAH AMBRE.

3 État de l'art

Les recherches sur les outils auteurs dans le domaine des EIAH ont commencé il y a une quinzaine d'années. Ce champ de recherche permet d'espérer résoudre le problème de la difficulté et du coût de conception des EIAH. Une plus grande offre d'outils auteurs permettrait de supporter le développement plus rapide d'EIAH plus nombreux et plus variés (Woolf, 2009, Chap. 10)

Les outils auteurs sont souvent divisés en deux catégories, ceux s'attachant essentiellement à construire des parcours pédagogiques à partir de ressources existantes, et ceux visant la construction d'EIAH de type "tuteurs intelligents" fondés sur une représentation des connaissances permettant la résolution de problèmes et la construction de rétroactions pertinentes pour l'apprenant (Murray, 2003).

Les travaux français autour des scénarios d'apprentissage (Pernin, Emin, & Guéraud, 2009) et de la modélisation pédagogique (Laforcade, Nodenot, & Sallaberry, 2005), en particulier ceux menés dans le cadre du projet Ariadne (David, Guilloux, & Flament, 2002; Nanard &

Nanard, 2002) font partie de la première catégorie d'outils auteurs. On y trouve aussi les travaux sur les outils auteurs pour créer des hypermédias adaptatifs (Brusilovsky, Schwarz, & Weber, 1996) et ceux permettant d'expliciter des stratégies d'enseignement (Major, Ainsworth, & Wood, 1997; Murray, 1998). La faiblesse des EIAH produits par ces outils est le caractère relativement passif de l'enseignement, du fait du manque de connaissances sur le domaine et sur l'apprenant (Murray, 2003).

Les outils auteurs destinés à créer des environnements d'apprentissage fondés sur la simulation modélisent le processus simulé et sont bien adaptés à l'apprentissage de compétences procédurales. On trouve dans ce champ de recherche des systèmes opérationnels proposés au niveau national (Guéraud et Pernin 1999, Guéraud et al 1999) et international (Hsieh, Halff, & Redfield, 1999; De Jong & Joolingen, 1998; Munro, Johnson, & Pizzini, 1997). Les rétroactions ne sont pas construites à partir de connaissances du domaine mais à partir de comportements prévus pour chaque étape du processus simulé.

Dans la catégorie des outils auteurs visant la construction d'EIAH de type "tuteurs intelligents" fondés sur une représentation des connaissances, on peut citer ASPIRE (Mitrovic et al., 2006), outil auteur destiné à créer des modèles du domaine pour des tuteurs à base de contraintes. La limitation de ces tuteurs à base de contraintes est le manque de fiabilité du diagnostic lorsque l'apprenant s'écarte de la solution proposée par le modèle (Woolf, 2009, Chap. 3). Les outils auteurs ayant eu le plus de succès et les plus proches de nos besoins sont ceux qui ont été développés à l'Université Carnegie Mellon à Pittsburgh, pour créer des *model tracing tutors*. Les *model tracing tutors* sont fondés sur un modèle cognitif à base de règles de production, et portent en général sur des tâches de résolution de problèmes. La résolution est conduite pas à pas, et le comportement de l'élève est à chaque étape analysé et corrigé s'il s'écarte de la procédure prévue. Des outils auteurs destinés à créer de tels tuteurs ont été conçus depuis une vingtaine d'année à Pittsburgh, il s'agit des *Cognitive Tutor Authoring Tools* (CTAT) (Aleven, McLaren, Sewall, & Koedinger, 2006 ; Anderson & Pelletier, 1991 ; Blessing, 1997; Koedinger et al , 2004). Les enseignants auteurs doivent définir des règles de production explicitant les stratégies de résolution des apprenants et leurs misconceptions. Une version de l'outil incluant le système SimStudent permet à l'enseignant auteur de définir sur des exemples de réponses de l'élève le comportement de l'EIAH, que le système tente de généraliser (Matsuda et al., 2010).

Les *model tracing tutors* se limitent aux domaines où la tâche de résolution de problèmes s'effectue pas à pas et où l'ensemble des connaissances du domaine peut s'exprimer sous forme de règles de production. Ils ne permettent donc pas l'apprentissage de connaissances déclaratives comme les concepts, ni l'apprentissage à partir d'exemples (Woolf, 2009, Chap. 4). Quant aux *example tracing tutors* (Aleven, Mclaren, Sewall, & Koedinger, 2009), ils ont l'avantage d'être développés rapidement, mais ils n'assurent pas l'ensemble des fonctionnalités qu'on attend d'un tuteur intelligent, par manque d'une solide représentation des connaissances.

Par rapport à notre problématique de l'acquisition de connaissances pour les EIAH centrés connaissances que sont les EIAH AMBRE, nous voyons donc que la plupart des outils auteurs développés jusqu'à présent se sont essentiellement intéressés à la construction d'un parcours pédagogique par l'enseignant, mais peu aux connaissances nécessaires pour assurer les fonctionnalités d'assistance à l'apprenant attendues d'un tuteur intelligent. Cette problématique a été prise en compte dans les outils auteurs développés à Pittsburgh, mais les model tracing tutors se limitent à des domaines où la résolution de problèmes peut

consister à appliquer séquentiellement des règles de production. Ces outils ne répondent donc pas à notre problématique d'outils auteurs pour l'enseignement de méthodes, ce mode résolution de problèmes demandant à l'élève de prendre du recul et de raisonner globalement sur le problème à résoudre.

4 Démarche envisagée

Nous avons vu que les Cognitive Tutor Authoring Tools (CTAT) développés à l'université Carnegie Mellon à Pittsburgh sont les outils les plus aboutis actuellement en terme de création d'EIAH centrés connaissances. Cependant, leur représentation de la résolution d'un problème comme une suite d'applications de règles de production ne permet pas de représenter les connaissances nécessaires aux EIAH AMBRE. En effet, les systèmes à base de connaissance de AMBRE utilisent des connaissances de classification de problème, des connaissances de reformulation du problème, des connaissances sur l'analogie entre les problèmes, des connaissances de génération de problème et de communication avec l'apprenant.

L'approche que j'ai adoptée pour ces systèmes à base de connaissances est de concevoir des architectures indépendantes du domaine utilisant des méta-modèles de connaissance communs à l'ensemble des domaines abordés. J'estime donc possible la définition de langages de description des connaissances permettant à un non-informaticien d'expliciter *via* une interface ces différents types de connaissances nécessaires à la création d'un EIAH AMBRE. La définition des schémas XML correspondant aux différents méta-modèles des architectures SYRCLAD et CHAMADE est une première étape vers ces langages de description des connaissances, puisqu'ils explicitent les connaissances à acquérir.

L'approche *de SimStudent* me paraît également intéressante dans le cadre du projet AMBRE, qui repose justement sur l'apprentissage de connaissances abstraites à partir d'exemples. On pourrait ainsi envisager une acquisition interactive des connaissances, l'utilisateur définissant les connaissances sur un exemple, et le système lui proposant une généralisation que l'utilisateur pourra valider ou modifier. Nous espérons pouvoir compléter l'offre existante de CTAT en définissant de nouveaux modules permettant d'expliciter des connaissances plus riches que les bases de règles utilisées actuellement.

La conception d'un outil auteur pour créer des EIAH AMBRE doit s'effectuer au sein d'une équipe pluridisciplinaire intégrant des enseignants. Durant toute la phase de conception de l'outil auteur, nous aurons le souci de l'évolution possible des EIAH créés, et donc de prendre en compte la tâche de réingénierie au sein de l'outil auteur.

Un des premiers verrous à lever concernant ce travail est la question de l'articulation entre la conception de l'interface de l'EIAH et l'élicitation des connaissances nécessaires au système. En effet, nous ne prétendons pas développer un outil auteur permettant de définir l'interface d'un EIAH AMBRE mais uniquement un outil auteur destiné à acquérir les connaissances nécessaires à l'architecture CHAMADE. Cependant, les connaissances destinées à accompagner l'apprenant (connaissances d'aide, de diagnostic et d'explication) sont fortement dépendantes des tâches que l'apprenant doit accomplir lors de sa résolution de problèmes, et donc de l'interface de l'EIAH AMBRE. Cette articulation entre la conception de l'interface et l'élicitation des connaissances pourrait s'effectuer de la manière suivante :

1. Le concepteur de l'EIAH élicite les connaissances de la méthode grâce à l'outil auteur. Il définit les connaissances de classification, ainsi que les connaissances de reformulation et de résolution qui y sont liées.
2. Le concepteur conçoit l'interface de l'EIAH, et en particulier les différentes tâches que l'apprenant doit effectuer pour résoudre des problèmes, en s'appuyant sur le cycle AMBRE. Le développement de cette interface doit être effectué par un informaticien.
3. Le concepteur définit grâce à l'outil auteur les connaissances d'aide, de diagnostic et d'explication, en lien avec les connaissances de la méthode préalablement définies en 1, et avec les étapes de résolution de l'EIAH définies en 2.

Pour choisir sur quel formalisme de représentation des connaissances construire l'outil auteur, il est possible qu'une piste intéressante à explorer soit celle des modèles et outils issus du Web Sémantique. En effet, le Web Sémantique apporte des formalismes de représentation des connaissances, des mécanismes de raisonnement, et un environnement (le web) facilitant le travail collaboratif, le partage de connaissances et la dynamicité des interactions.

Une question à trancher dans le cadre de ce travail sera donc de choisir si les connaissances élicitées par le concepteur dans l'outil auteur seront traduites en Prolog par le système, afin de pouvoir alimenter les architectures actuelles du projet AMBRE, ou si nous devons passer complètement aux technologies du Web Sémantique et redévelopper par conséquent les architectures SYRCLAD, CHAMADE et GenAMBRE.

Un premier travail exploratoire sur **l'acquisition des connaissances de classification** a été effectué par Damien Fourure dans le cadre d'un TER de M1. Le principe est de permettre au concepteur de définir un arbre de classification à l'aide d'une interface graphique (cf. **Figure 32**). L'utilisateur peut choisir de construire son arbre à partir de la racine ou à partir des feuilles. La définition des classes de problèmes est fondée sur les attributs (discriminants ou non) des problèmes, les classes pouvant être créées à partir d'autres classes : la classe père, la classe fils ou la classe frère. Le système contrôle que les relations de parenté définies par l'utilisateur sont correctes par rapport aux attributs. Il peut également faciliter une partie du travail de l'utilisateur en construisant la hiérarchie à partir d'un ensemble de classes opérationnelles. L'auteur peut ainsi définir uniquement des cas (des problèmes avec leur technique de résolution), préciser quels sont les attributs discriminants et ceux qui instancient le problème par rapport la classe, et le système construit une hiérarchie de classes. Ce procédé est un premier exemple du principe que nous souhaitons mettre en œuvre pour l'outil auteur, c'est-à-dire la construction de connaissances par la généralisation d'exemples.

L'utilisateur peut associer une technique de résolution à une classe, afin d'en faire une classe opérationnelle. L'élicitation des connaissances de reformulation pour les attributs définis dans les classes devrait également être possible pendant la construction de l'arbre.

L'outil peut exporter en XML l'arbre de classification construit par l'utilisateur, ce fichier XML étant conforme au schéma XML défini pour les connaissances de classification de l'architecture SYRCLAD (cf. **Figure 6**).

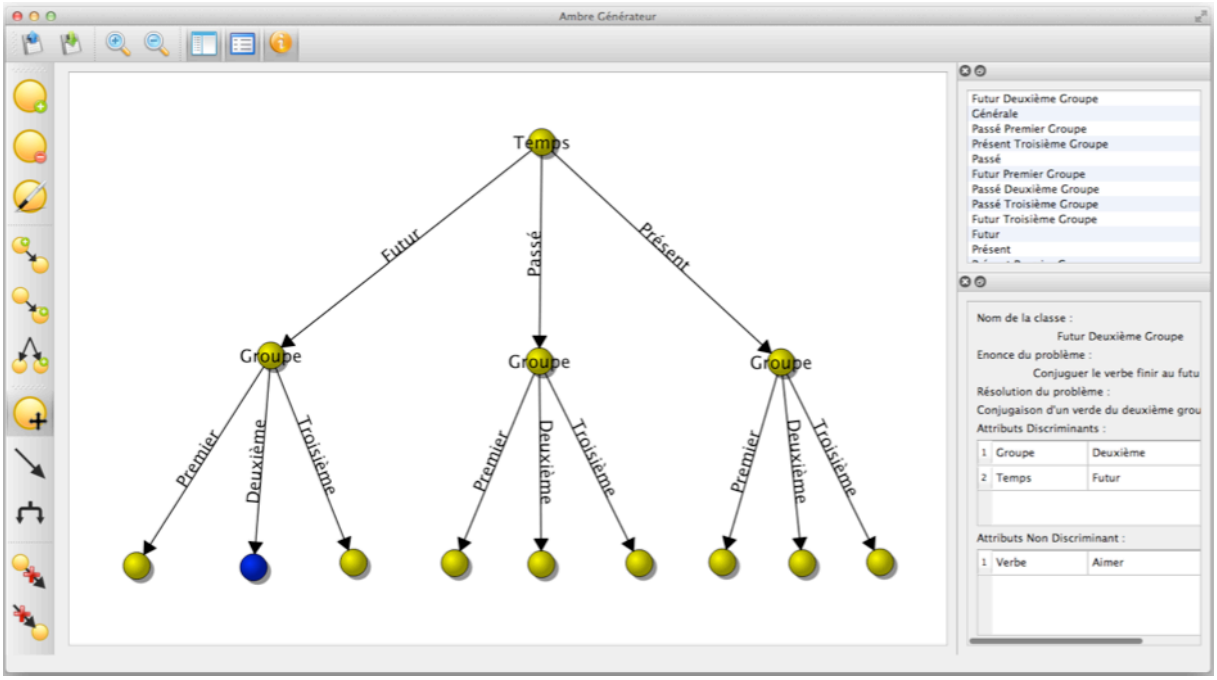

Figure 32. Interface de construction du graphe de classification

Conclusion et perspectives

Ce chapitre décrit des travaux en cours dont l'objectif est la conception d'un outil auteur permettant l'acquisition, auprès du concepteur d'un EIAH AMBRE, des connaissances nécessaires à l'architecture CHAMADE. La motivation de ce travail est d'une part de diminuer le coût de conception d'un EIAH AMBRE et d'autre part de permettre à un concepteur d'EIAH non informaticien, comme un enseignant, de mettre en œuvre un EIAH AMBRE dans le domaine qui l'intéresse.

Nous travaillons actuellement à la définition de langages de description des connaissances accessibles à des non-informaticiens (éventuellement *via* une interface graphique) et fondés sur les méta-modèles de l'architecture CHAMADE et les schémas XML les décrivant. Nous concevrons ensuite, en collaboration avec des praticiens, l'outil auteur s'appuyant sur ces langages de description des connaissances. L'objectif est d'obtenir un outil permettant une acquisition interactive des connaissances, et qui fournisse une assistance à l'utilisateur dans sa tâche. Permettre à l'utilisateur de décrire un exemple plutôt qu'une connaissance abstraite est l'un des moyens d'assistance envisagés. Ce principe a été mis en œuvre pour les connaissances de classification. Il pourrait probablement être très pertinent également pour les connaissances de diagnostic et d'explications : si l'utilisateur décrit un ensemble de réponses justes et fausses et les rétroactions correspondantes, un mécanisme de généralisation permettrait au système d'acquérir les connaissances lui permettant de diagnostiquer une réponse et de construire un rétroaction adaptée.

Notre objectif est également de monter une collaboration avec le laboratoire HCII à Pittsburgh pour étudier la possibilité d'ajouter à CTAT un module permettant la création d'EIAH AMBRE, mais aussi pour voir s'il serait pertinent de réutiliser une partie des modules de CTAT pour notre outil auteur, en particulier le moteur d'inférences de SimStudent, pour généraliser un comportement défini sur des exemples.

Le travail sur des outils indépendants du domaine nécessite d'étudier conjointement plusieurs domaines. Ainsi, pour la conception de l'outil auteur, nous veillerons à ce qu'il puisse nous permettre de construire à nouveau CHAMADE-add, mais aussi qu'il permette de construire un système CHAMADE appliqué à la conjugaison française, et un EIAH AMBRE-conjugaison. Pour valider les contributions liées à l'outil auteur, son test pour la conception d'un EIAH AMBRE appliqué à un troisième domaine semble également nécessaire. Il pourrait s'agir de l'apprentissage de la programmation récursive et fonctionnelle en Scheme.

Publications liées

Conférence nationale

Diattara, A., Guin, N., Lefevre, M., Kong Win Chang, B., & Luengo, V. (2014). Vers un outil auteur pour des EIAH destinés à l'enseignement de méthodes : formalisation des connaissances à acquérir. In *Technologies de l'Information et de la Communication pour l'Enseignement*. (Article court)

CONCLUSION ET PERSPECTIVES SUR LA PARTIE A

Cette conclusion à la partie A du mémoire présente un bilan des recherches menées sur l'enseignement de méthodes, et dresse quelques perspectives pour le projet AMBRE. Une dernière section présente les travaux engagés pour permettre la personnalisation des EIAH AMBRE, et comment ils ont motivé les travaux sur la personnalisation des EIAH décrits dans la partie B.

1 Bilan

La partie A de ce mémoire a permis d'explorer l'ensemble de mes travaux de recherche sur la question de l'enseignement de méthodes, dont l'objectif est la conception d'EIAH destinés à enseigner des méthodes de résolution de problèmes. Ces travaux sur l'élicitation de connaissances, menés le cadre du projet AMBRE, ont conduit à la définition de plusieurs architectures.

Ces architectures sont constituées d'une part de méta-modèles permettant la définition de modèles de connaissances dans un domaine donné, et d'autre part de moteurs de raisonnement capables d'exploiter ces modèles de connaissances.

Ainsi, si l'on souhaite concevoir un EIAH AMBRE-D (cf. **Figure 33**), destiné à enseigner une méthode dans le domaine D, on peut dans un premier temps, grâce à l'architecture SYRCLAD (cf. Chapitre A1), représenter les connaissances sur la méthode que l'on souhaite enseigner, et obtenir un résolveur de problèmes du domaine D, SYRCLAD-D. En appliquant le cycle AMBRE (cf. Chapitre A3) au domaine D, on peut ensuite concevoir un EIAH AMBRE-D dans lequel l'apprenant est invité à résoudre des problèmes du domaine D par analogie avec des problèmes-types de ce domaine, qu'on lui aura auparavant présentés avec leur résolution. L'architecture CHAMADE (cf. Chapitre A4) permet d'expliciter les connaissances de communication avec l'apprenant, d'aide et de diagnostic, afin d'obtenir un système à base de connaissances CHAMADE-D, capable d'accompagner l'apprenant dans son utilisation de l'EIAH AMBRE-D.

Nous travaillons actuellement sur un outil auteur (cf. Chapitre A6) qui permettrait à un utilisateur non informaticien d'éliciter plus facilement l'ensemble de ces connaissances, *via* un processus d'acquisition interactive des connaissances.

De plus, l'architecture GenAMBRE (cf. Chapitre A5) permet de définir des connaissances de génération, de manière à obtenir un générateur semi-automatique de problèmes du domaine D, GenAMBRE-D, pilotable par l'enseignant dans le cadre d'un logiciel AMBRE-enseignant-D. Ceci permet à l'enseignant de construire les problèmes qu'il souhaite faire résoudre à ses élèves avec AMBRE-D et de paramétrer le logiciel.

Dans le cadre de l'enseignement de méthodes, l'élicitation des connaissances intervient à double titre : d'une part comme je viens de le décrire pour représenter les connaissances nécessaires au système informatique pour accompagner l'utilisateur humain dans sa tâche, mais également parce que le principe de l'enseignement de méthodes est justement de

rendre explicites des méta-connaissances de résolution que les experts dans un domaine donné utilisent mais qui sont rarement enseignées aux novices.

Cette partie sur l'enseignement de méthodes permet également d'illustrer un glissement dans mes travaux de recherche concernant l'élicitation de connaissances. En effet, on passe de recherches initialement centrées sur la représentation des connaissances de l'expert au sein du système informatique (SYRCLAD, CASES, CHAMADE, GenAMBRE), vers l'acquisition par le système des connaissances auprès de l'utilisateur, comme dans l'outil auteur pour AMBRE. C'est également ce deuxième point de vue qui sera adopté pour les recherches décrites dans la partie B de ce mémoire.

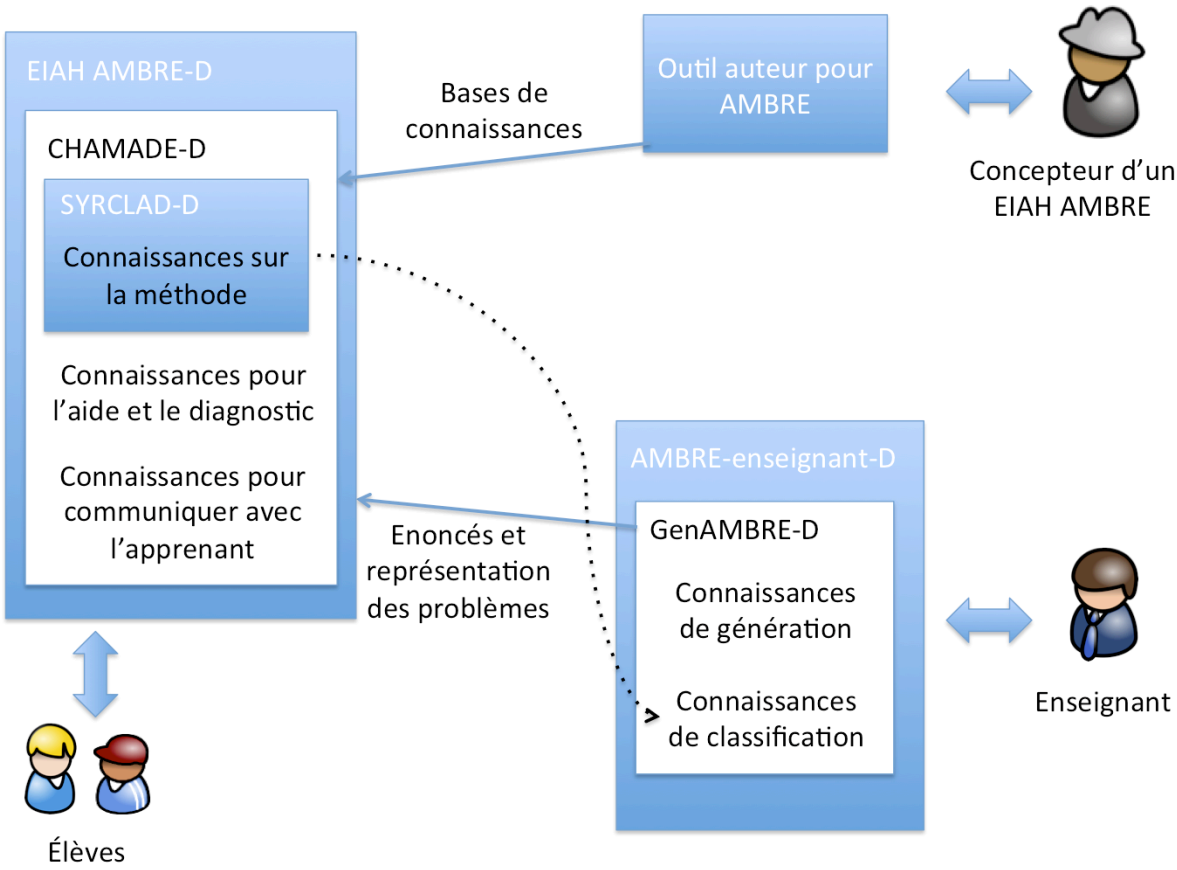

Figure 33. Les systèmes à base de connaissances de AMBRE

2 Perspectives pour le projet AMBRE

L'ensemble des travaux décrits dans cette partie va se poursuivre, à l'exception des travaux autour de l'architecture CASCADE menés dans le cadre du projet CASES. En effet, ce travail a été utile au projet AMBRE, car étudier le lien entre l'utilisation de connaissances abstraites et l'utilisation de cas pour résoudre des problèmes nous a permis de proposer le cycle AMBRE et nous sert encore aujourd'hui dans nos travaux sur l'outil auteur pour AMBRE, pour lesquels nous souhaitons acquérir des connaissances abstraites à partir d'exemples. Cependant l'architecture CASCADE n'est pas réutilisée telle quelle.

La première version de l'interface de l'EIAH AMBRE-add était développée en Delphi, et l'utilisation de l'EIAH souffrait de cette technologie ancienne, difficile à maintenir. Une

version web a été développée cette année (sans que cela nécessite de modification des architectures SYRCLAD et CHAMADE). Cela va nous permettre de reprendre des expérimentations dans de meilleures conditions, quand nous aurons effectué le développement d'une version web de AMBRE-enseignant-add, que nous souhaitons également expérimenter en conditions réelles. Les dernières expérimentations effectuées avec AMBRE-add nous ont montré qu'il faudrait pouvoir expérimenter une version du logiciel qui s'adapte à l'élève. Le travail qui a mené à une version adaptative de AMBRE-add est décrit dans la section suivante puis dans la partie B de ce mémoire.

La principale perspective actuellement est le travail en cours sur la conception d'un outil auteur pour AMBRE et l'acquisition interactive des connaissances nécessaires à un EIAH AMBRE. Comme je l'ai dit, l'évaluation de l'outil auteur s'effectuera en l'utilisant pour concevoir des EIAH AMBRE appliqués à de nouveaux domaines, ce qui nous permettra également de montrer la généricité des architectures CHAMADE et GenAMBRE.

3 Vers la personnalisation des EIAH

Dans le but de faire évoluer l'EIAH AMBRE-add afin qu'il s'adapte à chaque apprenant, nous avons développé un module destiné à élaborer un profil de l'apprenant, à partir de l'analyse de ses traces d'interactions avec le logiciel. Ce travail a été effectué par Kevin Sanlaville en 2011 et co-encadré par Marie Lefevre.

L'activité des élèves utilisant AMBRE-add est intégralement tracée : les traces contiennent l'ensemble des actions de l'apprenant à chaque étape et sous-étape de la résolution d'un problème. L'ensemble des réponses, ou même des projets de réponse, des demandes d'aide ou de diagnostic, des utilisations d'outils de calcul spécifiques, et toutes les rétroactions de l'EIAH (messages d'aide et de diagnostic) se trouvent dans les traces, l'ensemble de ces observés étant temporellement situés.

À partir de l'ensemble des données présentes dans ces traces, le logiciel calcule un certain nombre d'indicateurs sur les temps de lecture et de résolution selon les différentes étapes et sous étapes du cycle AMBRE, sur le nombre de réponses fournies par l'apprenant (pour savoir s'il répond ou non correctement du premier coup), sur la consultation des messages d'aide et de diagnostic, sur les erreurs-types identifiées par le système de diagnostic, et sur la manière dont l'apprenant se sert des outils mis à sa disposition dans les différentes étapes. Ces indicateurs sont disponibles pour chaque problème, mais sont aussi établis pour chaque classe de problème et pour chaque étape de résolution.

Le profil se décompose en deux parties : d'une part les données personnelles de l'apprenant, renseignées par l'enseignant, et d'autre part ses compétences et son comportement, déterminés par le logiciel.

Les **données personnelles** de l'apprenant sont fournies par l'enseignant lors de l'initialisation du profil. Elles contiennent son identité, les informations concernant la classe à laquelle il appartient et enfin ses aptitudes particulières.

Les compétences et le comportement de l'apprenant forment la majeure partie du profil. Elles sont déduites des indicateurs calculés à partir des traces. Pour faire ces déductions, il est nécessaire d'utiliser des valeurs seuils. Certains seuils sont définis par le concepteur mettant en œuvre l'élaboration des profils, comme par exemple le nombre minimum

d'exercices que l'apprenant aura dû faire avant que l'on estime pertinentes les informations qu'on en déduit. D'autres seuils sont définis par l'enseignant, comme ceux qui permettent de passer des taux de réussite à une étape aux valeurs que l'on donne à la maîtrise d'une compétence (acquis, en cours d'acquisition, non acquis). Ces valeurs sont tirées du consensus utilisé à l'école primaire mais passer d'un taux de réussite à ces valeurs est dépendant de la façon d'agir de chaque enseignant.

Pour ce qui est des **compétences**, on se demande si l'apprenant sait résoudre un problème additif, de manière générale, mais surtout en fonction de la classe du problème. On étudie aussi l'impact de certains paramètres (utilisation de grands nombres, de la retenue, l'écriture des chiffres en toutes lettres, l'ajout de valeurs ou phrases inutiles) sur la réussite de l'apprenant dans la résolution des problèmes. On étudie aussi toutes les compétences relatives au cycle AMBRE : les compétences de reformulation, de choix du problème-type, d'adaptation de ce problème et du classement du problème résolu dans le bon groupe de problèmes. Pour chacune de ces compétences, on note les résultats selon les classes de problèmes, ainsi que selon le détail de chaque étape. Par exemple, pour l'étape d'adaptation, on regarde si l'élève sait reformuler l'énoncé, écrire l'opération correspondante, faire le calcul et enfin écrire la solution, et ce de manière générale et pour chaque classe de problème. On étudie de même si l'apprenant commet certaines erreurs-type, comme celle de résoudre le problème avec les valeurs du problème-type.

Pour ce qui est du **comportement** de l'élève qui utilise AMBRE-add, on observe si l'apprenant lit effectivement les informations qui lui sont présentées à plusieurs moments de l'utilisation du logiciel. On essaie de détecter l'utilisation de stratégies d'essai-erreur pour certaines étapes-clé de la résolution, et en particulier selon les classes de problèmes. On essaie de repérer également les pertes de motivation. On étudie aussi quelle utilisation fait l'apprenant des outils proposés dans le menu, des demandes d'aide et de diagnostic. Enfin, on analyse l'usage fait des outils de calculs mis à la disposition de l'apprenant. Si l'apprenant se sert des outils de calculs, on regardera s'il sait les utiliser et dans quels buts (grand nombre, calcul avec retenue, addition, soustraction).

L'objectif de ce profil rassemblant toutes ces informations, est d'utiliser AMBRE-enseignant et GenAMBRE pour adapter l'EIAH AMBRE-add à l'apprenant.

Dans la partie du profil contenant ses données personnelles, ses aptitudes indiquent si l'apprenant a du mal à lire, à calculer, à entendre. En fonction de celles-ci, on lui proposera des exercices ou des fonctionnalités adaptées. Par exemple, à un apprenant qui a de gros problèmes de lecture, on proposera la synthèse vocale ; pour des problèmes de lecture moins importants, on pourra jouer sur la longueur de l'énoncé et la difficulté du vocabulaire employé. Pour ce qui est de difficultés de calcul, on pourra jouer sur les grandeurs des nombres ainsi que sur les opérations avec ou sans retenue.

Les informations contenues dans le profil à propos des compétences de l'élève peuvent être utilisées par l'enseignant pour personnaliser avec AMBRE-enseignant la nature des problèmes posés à l'élève. En effet, les classes de problèmes n'ont pas toutes la même difficulté, l'enseignant pourra donc vérifier que l'apprenant sait résoudre les classes de problèmes simples avant de lui proposer des classes de problèmes plus difficiles. En ce qui concerne les compétences de calcul, il pourra aussi, au fur et à mesure de l'avancement de l'élève, introduire des calculs avec retenue, choisir de plus grands nombres. Pour les élèves

qui ont de bonnes performances, il pourra choisir d'écrire les nombres en lettres, et d'introduire des valeurs et des phrases inutiles. Si l'enseignant constate dans le profil de l'élève que celui-ci a une difficulté avec une étape particulière du cycle AMBRE, il pourra lui proposer un exercice spécifique pour lui faire travailler cette compétence.

Les informations contenues dans le profil de l'élève relatives à son comportement peuvent être utilisées par l'enseignant pour personnaliser les fonctionnalités de AMBRE-add. Si l'élève ne prend pas le temps de lire des informations importantes, il pourra allonger le temps avant lequel l'élève peut passer à l'étape suivante. Si dans le profil il est indiqué que l'apprenant utilise une stratégie d'essai-erreur, par exemple pour choisir le problème-type, il pourra limiter le nombre d'essais ou l'accès au diagnostic. Si l'élève utilise systématiquement des outils de calcul, il pourra de la même manière en limiter l'usage.

Pour personnaliser l'EIAH AMBRE-add avec les outils décrit dans cette partie A, l'enseignant disposait donc des profils de ses élèves, qui lui fournissent des informations sur leurs spécificités, ainsi que du logiciel AMBRE-enseignant, qui lui permet de configurer les problèmes qu'il souhaite faire résoudre à ses élèves et les fonctionnalités qu'ils pourront utiliser. Cependant, la tâche de personnalisation restait très lourde (Guin, Lefevre, & Jean-Daubias, 2011). En effet, pour mettre en œuvre une stratégie de personnalisation proche des exemples mentionnés ci-dessus, le fait de devoir traiter individuellement chaque apprenant rendait la tâche difficilement réalisable. Au mieux, l'enseignant pouvait définir des groupes d'apprenants à partir de son observation des profils, et créer avec AMBRE-enseignant des séquences pour chacun de ces groupes.

Ce travail sur la mise en œuvre de la personnalisation dans AMBRE-add a donc mené aux travaux décrits dans la partie B, sur des modèles et outils génériques permettant une personnalisation semi-automatique de l'apprentissage, c'est-à-dire effectuée par le système mais pilotée par l'enseignant (cf. Chapitre B3). Les travaux décrits dans la partie B ont également été l'occasion de généraliser certaines approches déjà étudiées dans le projet AMBRE, comme l'élaboration de profils à partir de l'analyse des traces d'interactions (cf. Chapitre B1) ou la génération semi-automatique d'exercices (cf. Chapitre B2).

PARTIE B
PERSONNALISATION DES EIAH

INTRODUCTION 122

CHAPITRE B1 : ÉLABORER UN PROFIL DE L'APPRENANT À PARTIR DE L'ANALYSE DE SES TRACES 125

CHAPITRE B2 : GÉNÉRER DES EXERCICES 153

CHAPITRE B3 : PILOTER LA GÉNÉRATION D'EXERCICES À PARTIR DES PROFILS 171

CONCLUSION ET PERSPECTIVES 187

INTRODUCTION

Le réseau d'excellence européen STELLAR[2] a identifié la personnalisation comme l'un des grands challenges de la recherche en EIAH. En effet, d'après (Marty & Mille, 2009), la personnalisation des EIAH est l'un des principaux verrous de leur développement. Selon eux, les objectifs de la personnalisation sont la production de rétroactions pertinentes, le choix des activités à proposer, et un retour à l'enseignant sur ses élèves. Ils précisent qu'un prérequis à la personnalisation est la capacité à produire et analyser les traces d'interaction de l'élève avec l'EIAH.

Dans les recherches décrites dans cette partie du mémoire, l'analyse des traces est en effet utilisée pour choisir les activités qui seront proposées à l'élève et pour effectuer un retour à l'enseignant sur les activités de l'élève et son apprentissage. Nous avons choisi pour cela d'utiliser un objet intermédiaire : le profil de l'apprenant (Jean-Daubias, 2011). Ce profil, élaboré grâce à l'analyse des traces, contient des informations sur le comportement et les compétences de l'apprenant. Il peut ainsi être présenté à l'enseignant, au tuteur ou même à l'apprenant (Jean-Daubias, 2011). Il peut également être utilisé pour personnaliser les activités à proposer à l'apprenant.

On peut entendre par personnalisation le fait d'adapter l'EIAH au niveau scolaire visé et aux pratiques pédagogiques des enseignants (comme dans AMBRE-enseignant par exemple), sans pour autant individualiser l'apprentissage. Je considèrerai ici la personnalisation comme une individualisation de l'enseignement, de manière à ce qu'un EIAH s'adapte à *chaque* élève. L'objectif de nos travaux sur cette problématique est que l'EIAH s'adapte automatiquement à chaque apprenant, mais selon une stratégie définie par l'enseignant. Lorsqu'il est question dans cette partie *d'adaptation* de l'EIAH, il s'agit essentiellement d'adapter les activités à proposer à l'élève ainsi que les modalités de ces activités.

La **Figure 34** présente ainsi les trois principales questions de recherche qui seront traitées dans les chapitres de la partie B.

La première concerne la modélisation de l'apprenant. Il s'agit d'élaborer un profil de l'apprenant à partir de l'analyse de ses traces d'interaction avec un EIAH. Les travaux menés autour de cette problématique sont présentés dans le chapitre B1. Ils visent à proposer des modèles et outils génériques qui permettent de mettre en œuvre le processus d'élaboration du profil pour un EIAH dans un domaine donné. Ce processus d'élaboration consiste à calculer les valeurs d'éléments d'un profil dont la structure a été préalablement établie.

La deuxième question de recherche concerne la construction d'exercices répondant à certains critères fixés en fonction du profil de l'apprenant. Adapter un exercice peut consister à choisir en fonction de ces critères dans un ensemble d'exercices existants, ou à fournir à un EIAH des éléments lui permettant de construire ces exercices, ou encore à utiliser des générateurs semi-automatiques d'exercices, comme nous l'avons fait pour le projet AMBRE, mais en nous intéressant à des types d'exercices plus génériques. Ces travaux sont décrits dans le chapitre B2.

[2] http://www.stellarnet.eu/

La troisième question de recherche concerne la manière dont on va piloter la génération d'exercices en fonction des profils. L'adaptation ne dépendant pas seulement du profil de l'apprenant, mais aussi des objectifs de l'enseignant et du contexte d'apprentissage (Luengo, 2009, p. 12), il est important de confier la définition de la stratégie de personnalisation à l'enseignant. Le chapitre B3 décrit nos propositions pour une mise en œuvre par le système d'un processus d'individualisation défini par l'enseignant.

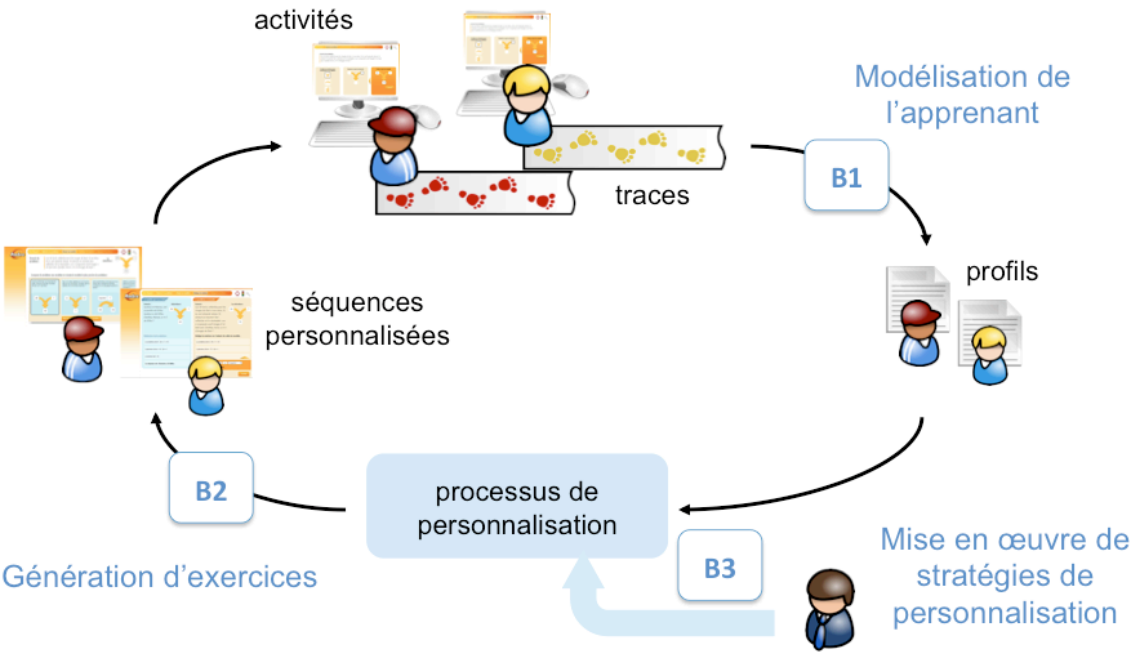

Figure 34. Cycle de personnalisation des EIAH

Une fois que l'enseignant a défini une stratégie de personnalisation, le processus de mise en œuvre de la personnalisation par le système peut être décrit par un cycle (cf. **Figure 34**). L'apprenant utilise l'EIAH. À partir des traces de son activité avec l'EIAH, le système élabore un profil de son comportement et de ses compétences. À partir des données contenues dans ce profil, le système, en appliquant la stratégie de personnalisation définie par l'enseignant, fixe des critères pour générer des exercices. La génération d'exercices respectant ces critères permet de construire une séquence de travail avec l'EIAH, personnalisée pour chaque élève. L'apprenant effectue cette séquence, ce qui permet de mettre à jour son profil et de poursuivre le cycle de personnalisation pour préparer la séquence suivante. Comme le dit Shute (Shute & Zapata-Rivera, 2007), au fur et à mesure des cycles, les informations du profil de l'apprenant deviennent de plus en plus précises et fiables, car elles se fondent sur l'analyse d'un plus grand nombre d'activités, ce qui permet de proposer à l'apprenant des activités qui seront de plus en plus pertinentes pour lui.

Plusieurs travaux présentés dans cette partie B ont été menés dans le cadre du projet "Personnalisation des EIAH" du Cluster ISLE (Informatique, Signal et Logiciels Embarqués) de la Région Rhône-Alpes. Ce projet était divisé en cinq sous-projets. Trois d'entre eux ont été le cadre de certaines contributions présentées ici : le SP1 qui étudiait la production des traces et leur représentation, le SP3 qui s'intéressait à l'interprétation des traces et à la représentation des connaissances, et le SP5 qui étudiait la question de la constitution de corpus de traces permettant d'effectuer des benchmarks.

CHAPITRE B1
ÉLABORER UN PROFIL DE L'APPRENANT À PARTIR DE L'ANALYSE DE SES TRACES

Résumé

Ce chapitre présente plusieurs travaux de recherche traitant de l'élaboration d'un profil de l'apprenant à partir de l'analyse de ses traces d'interaction avec un EIAH.

Les premiers travaux sont destinés à permettre à un utilisateur non informaticien d'exploiter les fonctionnalités de la plateforme de gestion de traces kTBS, développée au sein de l'équipe SILEX. L'accent est mis sur la manière de permettre l'importation de traces à kTBS puis l'interrogation de ces traces dans le but de définir des méthodes de calcul d'indicateurs destinées à alimenter un profil de l'apprenant.

Un modèle de diagnostic est également présenté, qui permet d'estimer le taux de maîtrise par l'apprenant de types de tâches représentées au sein d'une organisation ontologique.

Enfin un méta-modèle de techniques génériques de diagnostic est présenté. Il a permis de définir une méthode d'instanciation de techniques génériques de diagnostic issues de la littérature et rend possible la comparaison de plusieurs techniques de diagnostic pour un même domaine.

INTRODUCTION 127

1 INTERROGER DES TRACES POUR CALCULER DES INDICATEURS 128

1.1 CADRE DES RECHERCHES 128
1.2 QUESTIONS DE RECHERCHE 129
1.3 CONTRIBUTIONS 129
IMPORTER DES TRACES EXISTANTES AU SEIN DE KTBS 129
ANALYSER LES TRACES POUR ÉLABORER DES PROFILS 132
PERMETTRE À UN UTILISATEUR NON INFORMATICIEN D'INTERROGER DES TRACES 134
1.4 BILAN 138

2 ÉLABORER UN PROFIL DE L'APPRENANT EN SE FONDANT SUR UN RÉFÉRENTIEL DES CONNAISSANCES DU DOMAINE 139

2.1 CADRE DES RECHERCHES 139
2.2 QUESTION DE RECHERCHE 139
2.3 CONTRIBUTIONS 142
2.4 ÉVALUATION ET PERSPECTIVES 143

3 INSTANCIER ET COMPARER DES TECHNIQUES GÉNÉRIQUES DE DIAGNOSTIC 144

3.1 CADRE DES RECHERCHES 144
3.2 QUESTIONS DE RECHERCHE 144
3.3 CONTRIBUTIONS 145
3.4 ÉVALUATIONS 147

CONCLUSION ET PERSPECTIVES 148

SYNTHÈSE DES CONTRIBUTIONS 149

POUR EN SAVOIR PLUS 150

PUBLICATIONS LIÉES 151

Introduction

L'objectif des travaux présentés dans ce chapitre est de permettre d'obtenir des informations sur le comportement et les compétences d'un apprenant utilisant un EIAH, en analysant les traces de ses interactions avec le système. La conclusion de la partie A présentait un travail de ce type effectué pour l'EIAH AMBRE-add. Ce travail était dépendant à la fois de l'EIAH, du domaine, et des traces collectées. Le but des travaux décrits ici est de proposer des modèles et outils génériques permettant de faciliter la tâche d'un utilisateur qui voudrait définir, dans le contexte d'un EIAH donné et pour une activité donnée, un processus d'élaboration d'un profil de l'apprenant.

Les traces d'interactions de l'apprenant permettent de suivre les étapes de son travail avec le système et incluent ses productions. Historiquement, seules les productions de l'apprenant étaient exploitées en EIAH, mais depuis une dizaine d'années on s'intéresse davantage au processus de construction de ces productions. D'où un besoin plus important d'analyse des traces. Cette évolution s'est traduite par la constitution de communautés de recherche autour de ces questions, telles que *Educational Data Mining* ou *Learning Analytics*. L'intérêt pour ces recherches s'est encore renforcé depuis deux ans avec l'avènement des MOOCs.

Dans les recherches en EIAH, et en particulier au sein de la communauté internationale, le terme de modèle d'apprenant est celui qui est le plus souvent utilisé pour décrire un ensemble d'informations sur les compétences de l'apprenant. Cependant, ce terme de modèle est, selon les auteurs, utilisé à la fois pour décrire le processus de modélisation (comment calculer les informations), la théorie d'apprentissage sous-jacente, la structure de données permettant de représenter les informations, et les informations calculées pour un apprenant donné. Par conséquent, au sein de l'équipe SILEX, nous avons choisi d'utiliser le terme de *profil de l'apprenant* pour désigner l'ensemble des informations calculées pour un apprenant donné, et le terme de *modèle de profils* pour désigner la structure des profils, c'est-à-dire l'ensemble des informations attendues, dont il faut calculer la valeur pour obtenir le profil d'un apprenant donné. Pour ce qui est du processus de calcul de ces valeurs, c'est le terme d'*élaboration du profil*, ou de *diagnostic* qui sera utilisé. Il s'agit effectivement de mettre en œuvre un diagnostic des compétences de l'apprenant. Ce terme se retrouve souvent dans la littérature, mais ce diagnostic des connaissances de l'apprenant ne doit pas être confondu avec le diagnostic des réponses de l'élève, évoqué dans la partie A, qui est beaucoup plus local. Les travaux décrits dans ce chapitre portent sur l'élaboration du profil de l'apprenant, c'est-à-dire la manière d'alimenter un profil, dont la structure a été définie préalablement. Ils ne traitent pas de la définition du modèle de profil, qui a été étudiée par ailleurs (Jean-Daubias, 2011).

Selon Marty et Mille (Marty & Mille, 2009), l'enjeu de l'analyse des traces d'un EIAH est d'une part de comprendre le comportement de l'apprenant qui utilise l'EIAH. En effet, l'analyse du comportement permet l'analyse des usages de l'environnement par leurs utilisateurs, particulièrement utile dans des dispositifs massifs comme les MOOCs. L'utilisateur visé par ce que nous pouvons proposer comme outils est alors l'auteur du MOOC. Marty et Mille estiment que l'analyse des traces a d'autre part comme objectif de fournir à l'enseignant une information adaptée à ses besoins sur l'évolution des apprentissages de ses élèves. Dans un cas comme dans l'autre, il semble indispensable que

l'enseignant, qu'il soit auteur ou tuteur, puisse agir directement sur les processus d'analyse. Toujours selon les mêmes auteurs, la spécification des traces et celle des processus de diagnostic sont interdépendantes. Effectivement, pour analyser des traces, il faut connaître leur contenu et leur format. L'ambition des travaux que je mène est de proposer des outils d'analyse qui pourront s'adapter à la fois au format et au contenu des traces de celui qui souhaite les analyser.

La première section de ce chapitre présente des propositions pour permettre à un utilisateur non informaticien d'interroger des traces afin de calculer des éléments du profil de l'apprenant. Nous avons défini un langage de haut niveau qui peut également être utilisé pour effectuer une analyse des usages.

La deuxième section décrit une technique de diagnostic des compétences de l'apprenant fondée sur une représentation ontologique des savoirs et savoir-faire du domaine. Cette technique de diagnostic est indépendante du domaine.

La troisième section présente un méta-modèle de techniques génériques de diagnostic, ainsi qu'une méthode d'instanciation de telles techniques à un domaine donné. Le fait de pouvoir instancier plusieurs techniques génériques de diagnostic grâce à un processus unique permet également de les comparer afin de choisir la plus pertinente pour ce domaine.

1 Interroger des traces pour calculer des indicateurs

La question est ici de permettre à un utilisateur non informaticien (auteur, enseignant) mais ayant une bonne connaissance des traces qu'il souhaite analyser, d'interroger ces traces pour élaborer un profil de l'apprenant, ou pour effectuer une analyse des usages. Pour regrouper ces deux objectifs en une problématique unique, considérons qu'il s'agit de définir des indicateurs sur l'activité des apprenants avec l'EIAH. Un indicateur est une variable généralement calculée ou établie à partir de données issues des traces (Dimitrakopoulou, 2004).

Dans le cadre de l'analyse des usages, on veut en effet calculer des indicateurs comme le taux de participation à un forum ou la quantité de vues effectuées pour une ressource. Dans le cas de l'élaboration d'un profil de l'apprenant, chaque élément du modèle de profil, par un exemple le taux de réussite à un type d'exercice, peut être considéré comme un indicateur. Il faut définir une méthode permettant de calculer la valeur de cet élément pour un apprenant donné.

La littérature sur l'ingénierie des indicateurs ne distingue pas en général le résultat du calcul (l'indicateur) de la méthode qui permet d'obtenir ce résultat (la formule). Nous parlerons donc dans la suite de *méthode de calcul d'indicateurs*. Notre objectif est donc de proposer des outils permettant de définir de telles méthodes de calcul d'indicateurs.

1.1 Cadre des recherches

Ces travaux ont commencé dans le cadre du projet « Personnalisation des EIAH » du cluster ISLE de la Région Rhône-Alpes, et en particulier au sein du sous-projet 3 (Interprétation des traces et représentation des connaissances), avec les travaux de la thèse (non soutenue, 2008-2011) de Lemya Settouti, co-encadrée par Vanda Luengo.

Ils ont été poursuivis avec la définition d'un langage d'interrogation des traces dans le cadre du stage de M2R de Bryan Kong Win Chang en 2014 au sein du projet COAT (Connaissances Ouvertes à Tous), avec la participation de Pierre-Antoine Champin et Marie Lefevre.

Nous travaillons également depuis 2013 sur la question de la collecte des traces avec Mohamed Besnaci et Pierre-Antoine Champin.

1.2 Questions de recherche

L'équipe SILEX a défini la notion de Système à Base de Traces (SBT), qui permet de stocker, transformer et interroger les traces d'interaction. Une trace est définie comme un ensemble d'observés situés dans le temps, nommés *obsels* (pour "observed element"). Au sein d'un SBT, une trace est systématiquement accompagnée de son modèle, qui définit les types d'obsels pouvant être présents dans la trace, la manière de représenter le temps, les types d'attributs des obsels et les types de relations entre obsels (Champin et al., 2013). On parle alors de *trace modélisée* (M-Trace). Une transformation de traces consistera à réécrire une trace selon un nouveau modèle de traces.

kTBS[3] (kernel for Trace-Based Systems) est une plateforme qui a été développée dans l'équipe SILEX pour mettre en œuvre cette notion de système à base de traces. Cette plateforme fondée sur RDF permet d'interroger les traces d'interaction en effectuant des requêtes SPARQL qui permettent d'obtenir des informations sur l'activité de l'utilisateur.

Les systèmes à base de traces ainsi que kTBS ne sont pas spécifiques au domaine des EIAH, mais permettent d'analyser l'activité des utilisateurs d'un environnement numérique quelconque. Les fonctionnalités de stockage et d'interrogation des traces sont cependant particulièrement intéressantes pour un concepteur d'EIAH ou un enseignant qui souhaiterait exploiter les traces d'interaction de ses élèves. De plus, la communauté EIAH dispose de bases de traces existantes. Cependant, l'utilisation de kTBS, même par des chercheurs en EIAH, est très limitée par le fait qu'il faut définir un modèle de traces en RDF puis implémenter un programme informatique permettant d'importer les bases de traces existantes au sein de kTBS, et il devient alors possible d'interroger ces traces, pour peu qu'on maîtrise le langage SPARQL. La conception d'interfaces à kTBS, accessibles à des non programmeurs, est donc un préalable indispensable à son utilisation dans le cadre des EIAH.

Trois questions de recherche permettant d'atteindre cet objectif sont étudiées dans la section suivante. La première consiste à permettre à un utilisateur ne possédant pas nécessairement de compétences en informatique d'importer au sein de kTBS des traces existantes. La deuxième consiste à lui permettre de définir comment alimenter un profil d'apprenant à partir de ces traces. Ce qui pose une troisième question de recherche sur la conception d'un langage lui permettant d'exprimer des requêtes sur les traces.

1.3 Contributions

Importer des traces existantes au sein de kTBS

Nous nous plaçons ici dans la situation où un utilisateur dispose d'une base de traces d'apprentissage qu'il souhaite importer à kTBS. Cet utilisateur ne possède pas de compétences en programmation mais a une bonne connaissance des traces qu'il souhaite exploiter.

[3] http://kernel-for-trace-based-systems.readthedocs.org/

Les traces au sein de kTBS sont toujours accompagnées de leur modèle. Avant de pouvoir importer des traces à kTBS, il faut donc définir un modèle de traces qui correspond aux informations présentes dans les traces existantes (que j'appellerai traces externes) et que l'utilisateur souhaite retrouver dans les nouvelles traces créées au sein de kTBS. L'équipe SILEX a défini un méta-modèle RDF décrivant comment est constitué un modèle de traces. Nous avons défini une version plus spécialisée de ce méta-modèle décrivant les types d'obsels, d'attributs et de relations qui sont souvent présents dans des traces d'EIAH (Settouti, Guin, Luengo, & Mille, 2010). Ce méta-modèle, dont une partie est présentée sur la **Figure 35,** peut être utilisé afin de définir un modèle de traces pour un EIAH donné. L'objectif est donc avec ce méta-modèle de guider l'utilisateur dans le choix et la définition des types d'obsels présents dans son modèle de traces.

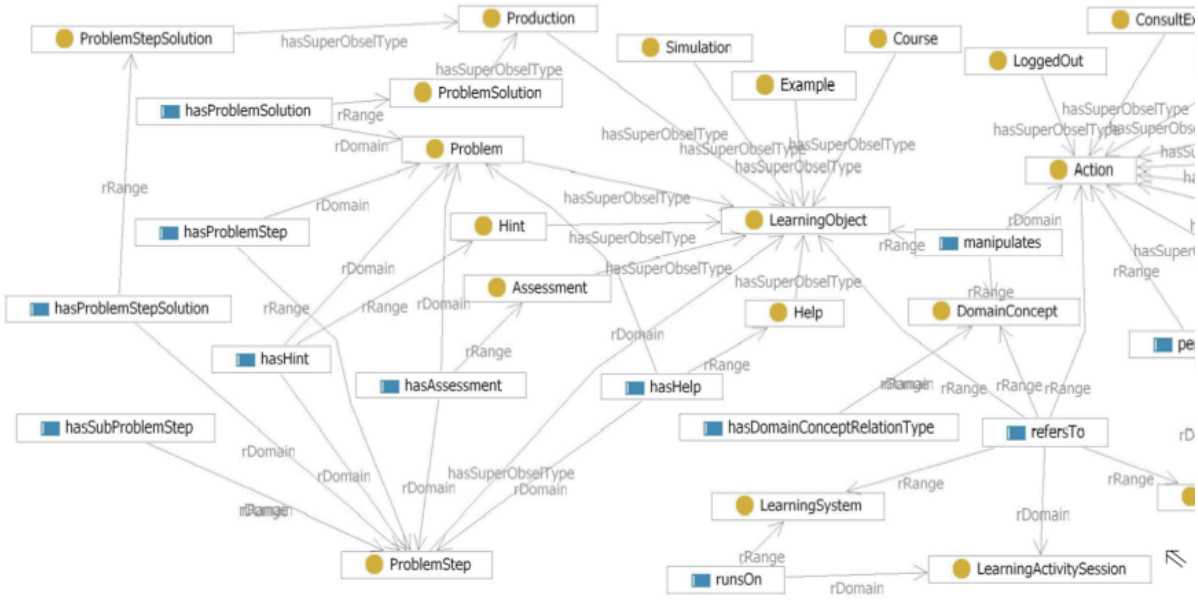

Figure 35. Un méta-modèle des traces d'EIAH

La **Figure 36** présente le processus proposé pour importer des traces externes dans kTBS. Si l'utilisateur veut importer les traces d'un EIAH X, il doit donc définir un modèle de ses traces au sein de kTBS, conforme au méta-modèle de traces. Pour importer les traces externes, c'est-à-dire construire au sein de kTBS des traces modélisées (m-traces) conformes au modèle défini, l'utilisateur doit définir un *collecteur*. Ce collecteur est un programme informatique qui va convertir les données présentes dans les traces externes pour construire des m-traces. Pour qu'un utilisateur non programmeur puisse créer un tel collecteur, nous nous sommes inspirés de l'approche utilisée par Marie Lefevre pour créer Tornade (Lefevre & Jean-Daubias, 2012), logiciel permettant d'importer des profils externes à l'environnement Eprofilea (Jean-Daubias, 2011).

Cette démarche consiste à acquérir auprès de l'utilisateur les connaissances permettant de convertir les traces. Le principe est de demander à l'utilisateur de mettre en correspondance chaque élément du modèle de traces avec un ou des éléments des traces externes. De cette mise en correspondance, le système va déduire une règle de construction d'un élément de la m-trace à partir d'un élément de la trace externe à importer. Cette tâche est demandée une seule fois à l'utilisateur, afin de construire le collecteur. Une fois le collecteur construit, il permettra d'importer une grande quantité de traces externes analogues à celles utilisées pour construire le collecteur.

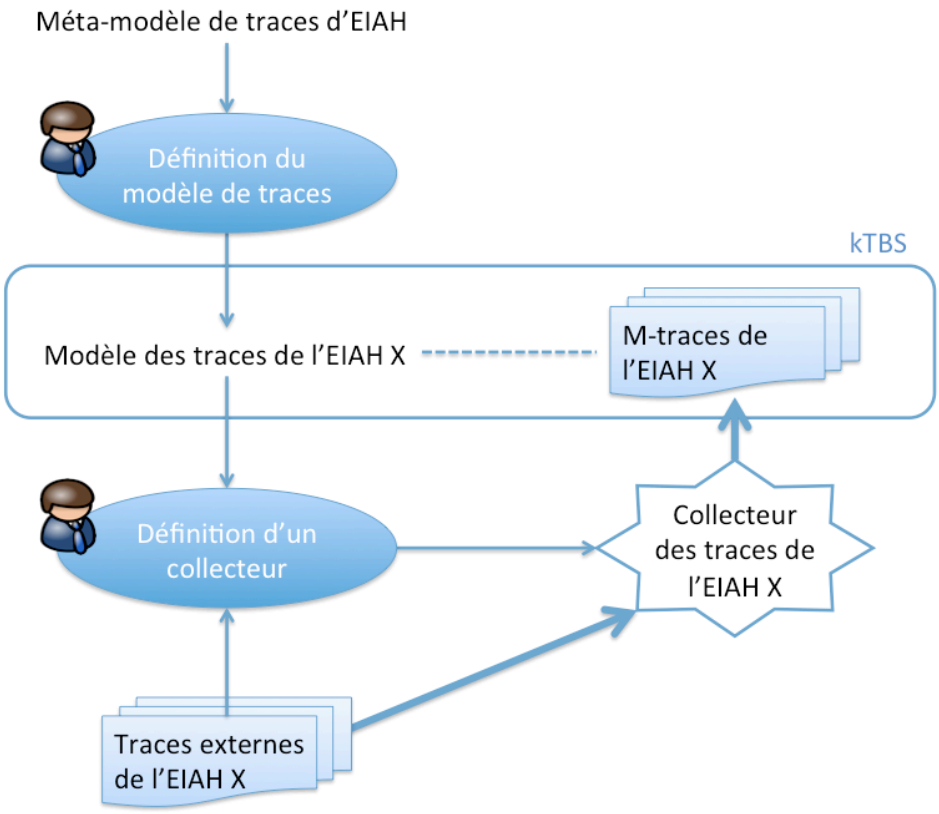

Figure 36. Processus d'importation de traces externes dans kTBS. *Les flèches fines représentent des actions qui ne seront effectuées qu'une fois et qui permettent de créer le collecteur. Les flèches épaisses représentent le processus automatique d'importation de traces mis en œuvre par le collecteur.*

xCollector (cf. **Figure 37**) est un prototype mettant en œuvre ce principe. Cet outil permet d'importer des traces aux formats XML, base de données, et texte structuré. Il utilise une première phase convertissant les bases de données et textes structurés en fichiers XML.

Nous avons décidé de proposer à l'utilisateur de xCollector de définir dans le même temps le modèle de traces et les mises en correspondance, dans le cas où le modèle de traces n'a pas été défini au préalable. L'utilisateur peut en effet observer ses traces externes et créer au fur et à mesure les éléments du modèle nécessaires pour conserver les éléments qui l'intéressent. À partir d'une mise en correspondance exprimée par l'utilisateur, le système effectue plusieurs propositions à l'utilisateur sur la manière de repérer l'élément sélectionné dans les traces externes. Ces propositions sont des expressions xPath libellées en français. Ces expressions xPath sont construites en utilisant la valeur de l'élément sélectionné (e.g. tous les nœuds dont le texte est égal à "exercice"), les valeurs de ses éventuels attributs (e.g. tous les nœuds dont l'attribut "action" vaut "Envoyer Msg"), de ses fils, sa position, et ce relativement à un certain contexte (cf. (Besnaci, Guin, & Champin, 2014) pour en savoir plus).

Ce logiciel en cours de finalisation devra être expérimenté auprès d'utilisateurs souhaitant importer leurs traces dans kTBS afin de savoir s'il répond effectivement au besoin pour lequel il a été conçu.

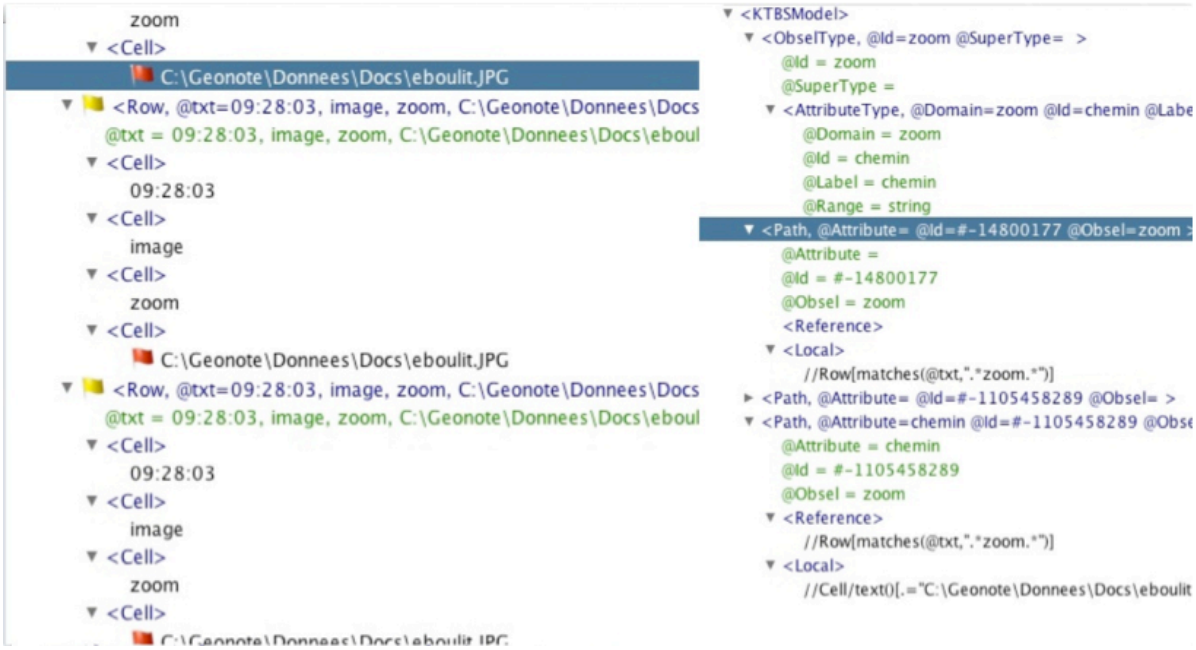

Figure 37. Mise en correspondance avec xCollector entre un élément de la trace externe et un élément du modèle de traces.

Analyser les traces pour élaborer des profils

Une fois que les traces d'un EIAH X sont stockées ou importées dans kTBS, la question qui se pose est d'analyser ces traces pour alimenter un profil d'apprenant. La **Figure 38** présente le processus proposé pour répondre à cette question.

Nous disposons dans kTBS de m-traces d'apprenants qui utilisent l'EIAH X, ces m-traces étant conformes au modèle des traces de l'EIAH X. L'utilisateur a d'autre part défini un modèle de profil décrivant les informations qu'il souhaite déduire de l'analyse des traces. Nous ne traitons pas ici de cette question, en considérant le modèle de profil comme un ensemble d'éléments dont il faut calculer la valeur pour élaborer un profil de l'apprenant. L'utilisateur pourrait pour définir ce modèle de profil utiliser par exemple Batisseur (Jean-Daubias, Eyssautier-Bavay, & Lefevre, 2009), un module de l'environnement Eprofilea, fondé sur le langage de modélisation de profils évolutifs PMDLe (Ginon, Jean-Daubias, & Lefevre, 2011).

La question est alors pour chaque élément du modèle de profil, de définir une méthode de calcul d'indicateurs (MCI), dépendant des éléments du modèle de traces. Cette MCI s'appliquera ensuite sur les m-traces de l'apprenant satisfaisant ce modèle de traces, afin de calculer la valeur permettant d'alimenter le profil de cet apprenant. Un langage permettant à l'utilisateur de formuler la MCI sera présenté dans la sous-section suivante.

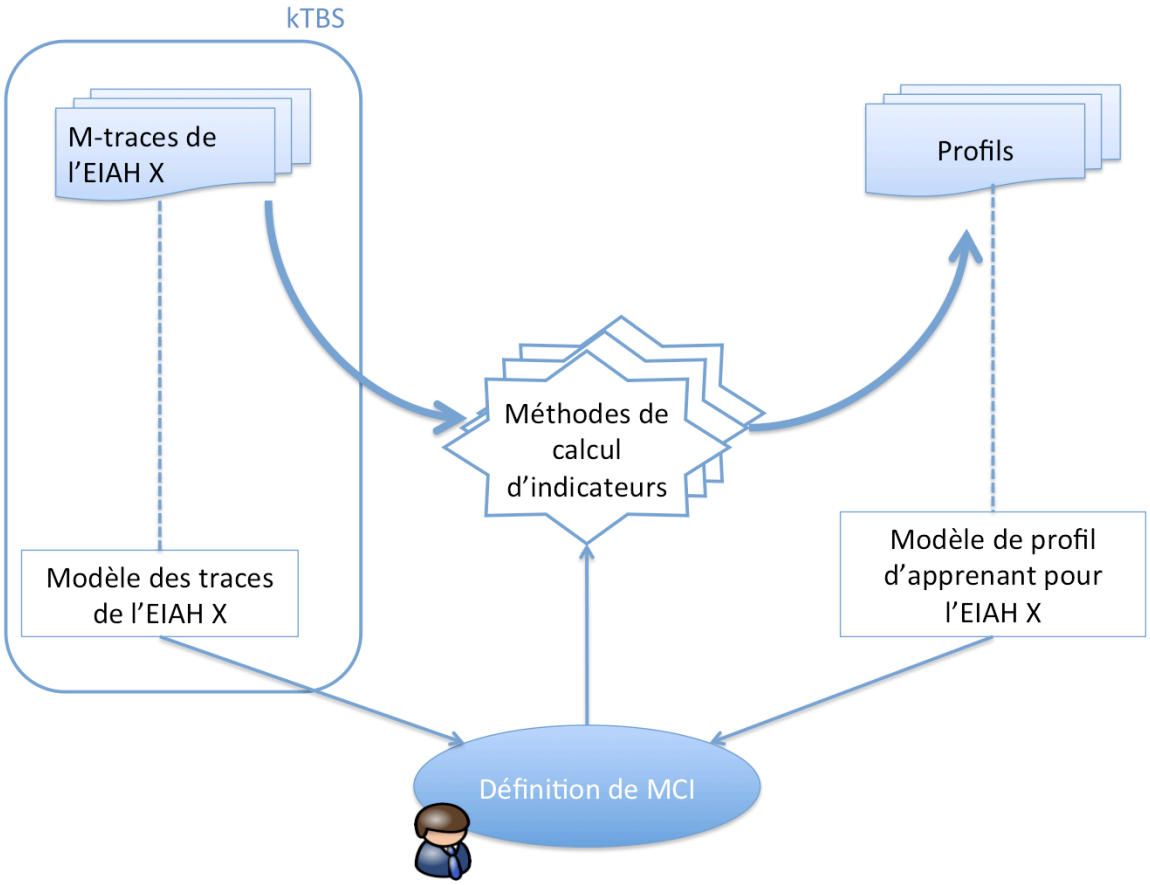

Figure 38. Processus de définition de méthodes de calcul d'indicateurs permettant d'alimenter le profil. *Les flèches fines représentent des actions qui ne seront effectuées qu'une fois et qui permettent de créer les MCI. Les flèches épaisses représentent le processus automatique d'alimentation des profils à partir des traces, mis en œuvre par chaque MCI.*

Pour faciliter la tâche de l'utilisateur qui souhaite formuler des MCI, nous proposons de définir des patrons de MCI. La **Figure 39** présente comment de tels patrons de MCI, indépendants du domaine et de l'EIAH, peuvent être automatiquement adaptés aux traces d'un EIAH donné, pour peu que le modèle des traces de l'EIAH en question satisfasse bien le méta-modèle des traces d'EIAH. En effet, les types d'obsels du modèle de traces sont définis à partir des types d'obsels du méta-modèle de trace. Si un patron de MCI est défini en utilisant les types d'obsels du méta-modèle de traces (cf. **Figure 35**), on peut déduire de la relation entre le modèle de traces et le méta-modèle une instanciation de la MCI par rapport au patron de MCI (cf. **Figure 39**). Pour une description plus précise et plus formelle de ce processus, le lecteur pourra consulter (Settouti, Guin, & Luengo, 2011).

Nous avons par exemple défini un patron de MCI permettant d'évaluer si l'apprenant joue avec le système (Baker et al., 2006), c'est-à-dire essaie sans réfléchir toutes les réponses possibles les unes après les autres jusqu'à ce que l'EIAH lui renvoie un diagnostic positif. (Settouti et al., 2011) montre comment ce patron de MCI peut être adapté à la fois à l'EIAH AMBRE et à l'EIAH Copex Chimie (d'Ham & Girault, 2009). Nous avons également défini des patrons de MCI sur le comportement de l'apprenant avec l'EIAH (est-ce qu'il consulte le cours avant de faire les exercices, est-ce qu'il demande de l'aide avant de répondre, est-ce qu'il demande un diagnostic après avoir répondu…), et des patrons de MCI sur les

performances de l'apprenant (nombre d'essais sur les problèmes, taux de bonnes réponses, temps moyen de réponse…).

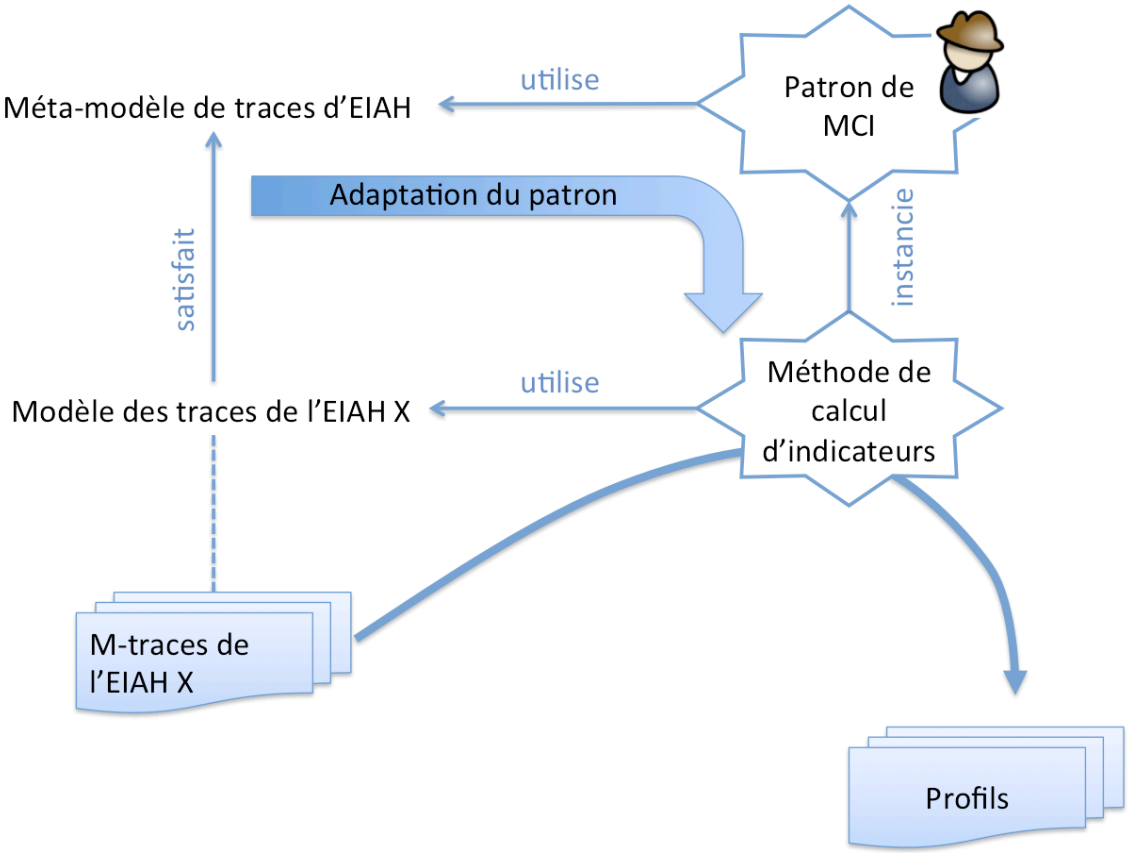

Figure 39. Processus d'adaptation d'un patron de MCI à un EIAH X

D'autres projets menés au sein du réseau d'excellence Kaléidoscope (DPULS), au laboratoire LIUM (Choquet & Iksal, 2007 ; Choquet, 2007; Pham-Thi, Iksal, Choquet, & Klinger, 2009), au laboratoire LIG (Diagne, 2009), ainsi qu'à Chambéry (Gendron, 2010), ont également proposé des définitions de patrons d'indicateurs, dans le but de permettre le partage et la réutilisation d'indicateurs. Notre approche, fondée sur le modèle de traces d'un EIAH et la manière dont il satisfait le méta-modèle de traces d'EIAH, devrait permettre une adaptation automatique de la suite de requêtes permettant de calculer la valeur de l'indicateur. Contrairement aux travaux cités ci-dessus, il ne s'agit pas ici de réutiliser et adapter à la main un indicateur défini par un autre enseignant.

Permettre à un utilisateur non informaticien d'interroger des traces

Nous avons vu qu'au sein de kTBS, l'interrogation des traces doit s'effectuer *via* des requêtes SPARQL. L'objectif du travail décrit ici était de définir un langage d'interrogation des traces sur-couche de SPARQL, permettant la définition de méthodes de calcul d'indicateurs. Ce langage se devait d'être suffisamment simple pour être utilisé par un non informaticien mais suffisamment riche pour permettre des requêtes sur les traces prenant en particulier en compte la dimension temporelle (par exemple les différentes relations de précédence temporelle d'un événement par rapport à un autre).

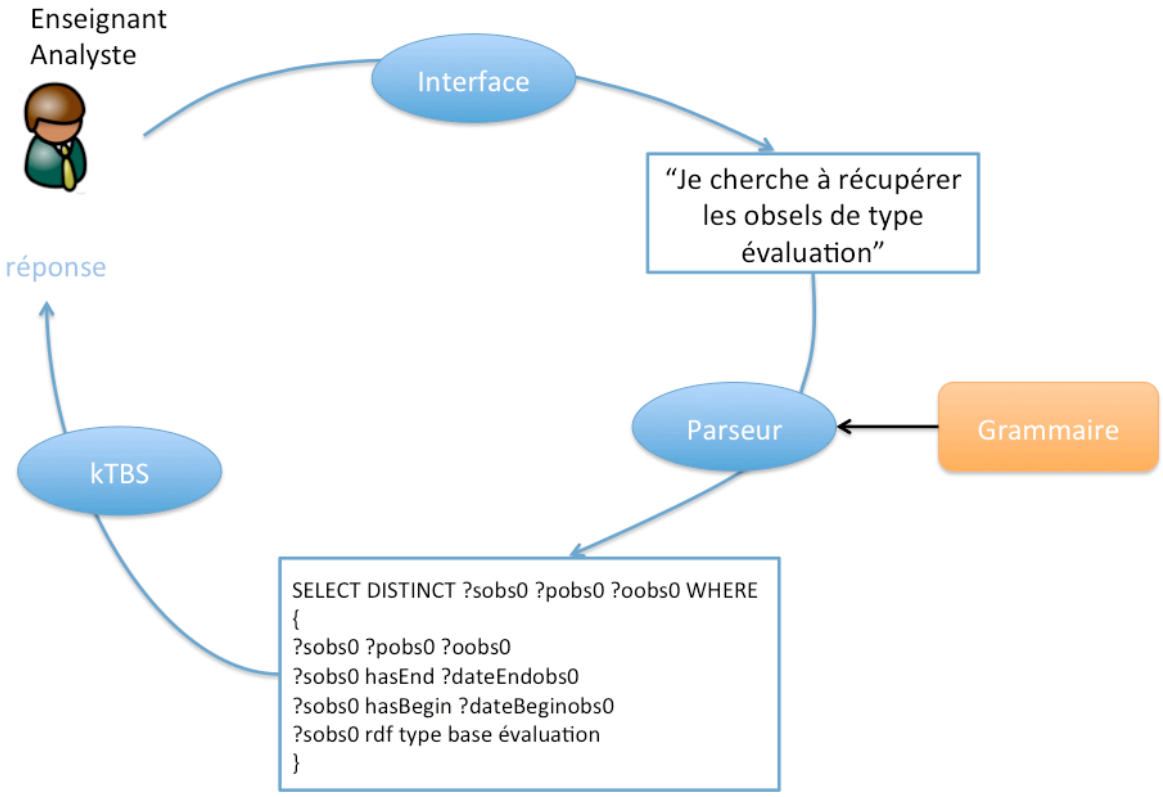

Figure 40. Processus d'interrogation des traces *via* SPARE-LNC

Nous avons choisi de définir un langage naturel contrôlé, fondé sur une grammaire décrivant un ensemble restreint de formulations possibles en français. La **Figure 40** présente le processus d'interrogation des traces par l'utilisateur avec l'outil développé : SPARE-LNC (SPARql REquêtes en Langage Naturel Contrôlé). L'utilisateur formule une phrase en langage naturel contrôlé, *via* une interface qui peut être soit textuelle soit graphique (cf. **Figure 41**). Un parseur fondé sur la grammaire analyse cette phrase et la traduit en une requête SPARQL qui est envoyée à kTBS, et le résultat de cette requête est enfin affiché à l'utilisateur. Celui-ci peut alors modifier ou compléter la requête.

Le choix du français plutôt que de l'anglais comme langue a été dicté par le fait que dans le cadre du projet COAT, nous souhaitions pouvoir proposer ce nouvel outil d'interrogation des traces à des auteurs de MOOCs lyonnais.

Un langage naturel contrôlé proche du français limite les connaissances nécessaires à l'utilisateur pour formuler ses requêtes. Celui-ci doit néanmoins connaître les principes sous-jacents aux systèmes à base de traces : obsels, attributs. Le langage est fondé sur le méta-modèle de traces de kTBS et n'est donc pas spécifique au domaine des EIAH. Il est bien sûr également indépendant du domaine.

Les opérateurs temporels sont des opérateurs de base importants pour les systèmes à base de traces. Or le SPARQL de base ne contient pas d'opérateurs temporels. Nous avons donc repris les propositions de (Settouti, 2011) consistant à utiliser les opérateurs dits de Allen (Allen & Frost, 1983) pour modéliser les opérations sur des intervalles temporels, de manière à pouvoir utiliser dans les requêtes des termes tels que avant/après, premier/dernier, suivi/précédé, etc.

Le langage permet de récupérer un ensemble d'obsels, un ensemble de valeurs ou de calculer une valeur particulière. Dans les phrases, on peut mettre des conditions sur les attributs des obsels, sur les valeurs des attributs, effectuer des calculs, nommer un résultat pour l'utiliser dans une autre requête, compter des résultats. De plus les phrases peuvent se combiner, permettant ainsi à l'utilisateur de procéder par étapes pour construire une requête.

Voici quelques exemples de phrases conformes à la grammaire :

- Je cherche à récupérer les obsels ayant un attribut réponse de valeur fausse.
- Je compte les obsels de type réponse ayant un attribut de valeur fausse, que je nomme B.
- Je veux récupérer les obsels de type A suivis par un obsel de type B, lui-même suivi par un obsel de type C.
- Je veux récupérer les obsels de type note que je nomme Notes. Soit A les attributs résultats de Notes. Soit Moyenne = AVG(A). *Ce qui permet de calculer la moyenne des notes.*
- Je veux récupérer les obsels de type note que je nomme Notes. Parmi Notes, je veux les obsels ayant un attribut de valeur Eric, que je nomme Notes_Eric.

Le lecteur intéressé pourra consulter la grammaire du langage dans (Kong Win Chang, Guin, Lefevre, & Champin, 2014).

Figure 41. Interface graphique de SPARE-LNC permettant de formuler des requêtes conformes à la grammaire du langage.

Nous avons choisi de permettre à l'utilisateur déjà familiarisé avec le langage d'utiliser une interface textuelle pour formuler ses requêtes grâce à ce langage. En effet, pour un utilisateur qui connaît le langage, il peut avec une interface textuelle formuler rapidement

une requête, réutiliser ou modifier une requête, et bénéficier des fonctionnalités classiques d'un outil d'édition de texte (e.g. copier/coller). Cette interface est complétée par des exemples de phrases permettant de rappeler à l'utilisateur les structures de phrases qu'il peut construire. Nous envisageons de plus d'ajouter un processus d'auto-complétion facilitant encore la tâche d'écriture à l'utilisateur.

Cependant, pour un utilisateur qui ne connaît pas du tout le langage, une interface textuelle n'est pas suffisante. Nous avons donc également conçu une interface graphique (cf. **Figure 41**) dans laquelle l'utilisateur dispose d'un ensemble de boutons correspondant aux mots du langage. Seuls sont actifs les boutons permettant d'avancer dans la construction de la requête que l'utilisateur a commencé à formuler en respectant la grammaire. Ce moyen permet de s'assurer que l'utilisateur suit la grammaire et limite les fautes de frappe, mais ralentit la construction de la requête par rapport à l'interface textuelle.

Ces boutons étant liés au langage, ils sont bien évidemment indépendants du modèle de trace. Cependant l'utilisateur utilisera un vocabulaire spécifique à la trace, *via* les noms des types d'obsels, des attributs et des valeurs possibles. Un champ textuel en bas de l'interface lui permet de saisir ces termes spécifiques au modèle de traces. Comme il est probable que l'utilisateur n'ait pas en mémoire les intitulés exacts de tous les types d'obsels, d'attributs et de valeurs présents dans ses traces, nous lui proposons également (dans une partie de l'interface non visible sur la **Figure 41**) une liste des éléments qui peuvent être présents dans ses traces. Nous envisageons pour la suite de ce travail d'utiliser une représentation graphique de la trace en nous appuyant sur des travaux en cours dans l'équipe SILEX relatifs à la visualisation des traces (Mathern & Cordier, 2014).

Une fois que l'utilisateur est satisfait des résultats renvoyés par la requête (ou suite de requêtes, une requête pouvant utiliser le résultat d'une autre requête) qu'il a formulée, il peut la sauvegarder afin de définir une méthode de calcul d'indicateurs qui pourra être utilisée sur d'autres traces relevant du même modèle de traces. Le format de sauvegarde est ainsi un quadruplet *Q* tel que *Q = {n, des, req, m}*, où *n* est le nom donné à la requête, *des* une description textuelle de la requête, *req* la formulation dans le langage et *m* le modèle de traces utilisé.

D'autres travaux destinés à permettre l'élaboration d'indicateurs ont déjà été menés au sein de l'équipe SILEX (Djouad, Mille, & Reffay, 2009). Ces travaux permettent à l'utilisateur d'effectuer des transformations de traces de type filtrage, mais nécessitent l'intervention d'un informaticien pour mettre en œuvre des requêtes plus compliquées faisant intervenir des opérateurs de Allen. Ainsi SPARE-LNC apporte des possibilités de requêtes supplémentaires, et un travail d'intégration des deux approches a d'ailleurs été entrepris.

Pour évaluer la complétude du langage SPARE-LNC, nous l'avons utilisé pour formuler des indicateurs de la littérature, afin de vérifier que ce langage peut répondre à un certain nombre de besoins. Une vérification plus exhaustive sur l'ensemble des indicateurs répertoriés par l'état de l'art du réseau Kaléidoscope (Dimitracopoulou 2004) serait néanmoins nécessaire.

Pour évaluer l'utilisabilité de l'outil développé, des expérimentations avec les utilisateurs ciblés sont également nécessaires. Nous avons pour l'instant travaillé avec Patrice Thiriet, auteur du MOOC FOVEA, qui s'est montré très intéressé par l'outil mais qui a exprimé des réserves sur sa compréhension des éléments présents dans les traces. Les intitulés des types d'obsels et attributs des traces FOVEA sont en effet difficiles à comprendre pour quelqu'un

qui n'a pas participé à la conception du modèle de traces. Un travail de transformation des intitulés des obsels de la trace mené avec l'ingénieur de la trace serait alors un préalable à l'utilisation de SPARE-LNC. Lorsqu'il s'agit de traces existantes en dehors de kTBS, le processus de définition d'un collecteur présenté précédemment peut servir à reformuler les termes utilisés dans les traces en intitulés du modèle de traces plus compréhensibles pour la personne qui va interroger les traces.

Les perspectives concernant ce travail, outre le portage en anglais, consistent à proposer à l'utilisateur des phrases modèles adaptées au domaine des EIAH. En effet, pour l'instant le langage proposé n'est pas spécifique au domaine des EIAH. Nous envisageons également de mettre en œuvre le processus décrit en **Figure 39** d'adaptation de patrons de MCI, en proposant à l'utilisateur des patrons de requêtes s'appuyant sur notre méta-modèle de traces d'EIAH (cf. **Figure 35**).

1.4 Bilan

Les contributions présentées dans cette section ont pour objectif de permettre à un utilisateur non informaticien d'une part d'importer des traces d'EIAH existantes à la plateforme kTBS, et d'autre part d'interroger ces traces afin de définir des méthodes de calcul d'indicateurs permettant l'élaboration de profils d'apprenants.

Par rapport à l'approche scientifique présentée dans le chapitre 1, ces travaux relèvent bien de l'acquisition des connaissances guidée par des méta-modèles.

En effet, la définition du modèle de traces par l'utilisateur est guidée par le méta-modèle de traces. En ce qui concerne la construction d'un collecteur, elle est effectuée en demandant à l'utilisateur d'éliciter des mises en correspondance, à partir desquelles le système suggère à l'utilisateur différentes propositions de généralisation de ces mises en correspondance. On retrouve ici le principe d'acquisition interactive de connaissances à partir d'exemples, que je cherche à mettre en œuvre dans plusieurs de mes travaux.

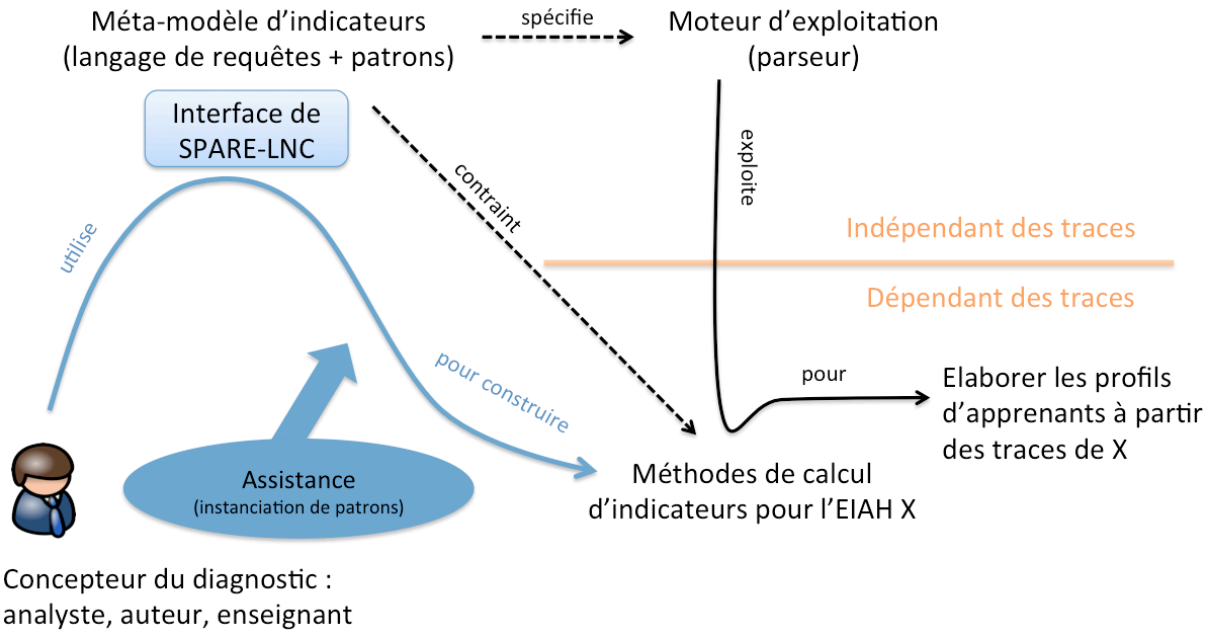

Figure 42. Instanciation de l'approche présentée au chapitre 1 pour la définition de méthodes de calcul d'indicateurs.

La **Figure 42** décrit en quoi les contributions sur la définition de méthodes de calcul d'indicateurs relèvent également de l'approche présentée au chapitre 1. En effet, le langage de requêtes en français associé aux patrons de requêtes forme un méta-modèle d'indicateurs utilisé par l'utilisateur souhaitant analyser les traces pour formuler *via* l'interface de SPARE-LNC une méthode de calcul d'indicateurs (qu'on pourrait assimiler à un modèle d'indicateurs) correspondant à une requête ou à une suite de requêtes exprimées dans le langage SPRARE-LNC. Le parseur de SPARE-LNC permet d'exploiter cette MCI pour exécuter une requête SPARLQL et calculer ainsi un élément du profil de l'apprenant. Une assistance est apportée à l'utilisateur dans sa définition des MCI *via* le mécanisme proposé pour l'instanciation de patrons de MCI.

L'évaluation des contributions décrites dans cette section a été commencée mais mérite d'être poursuivie à travers davantage de tests auprès des utilisateurs. Le projet HUBBLE dont je parlerai dans la conclusion de ce chapitre devrait permettre de mener ces expérimentations et aussi de faire évoluer ces contributions grâce à une collaboration avec plusieurs laboratoires français travaillant sur cette problématique de l'analyse des traces.

2 Élaborer un profil de l'apprenant en se fondant sur un référentiel des connaissances du domaine

Diagnostiquer les compétences et connaissances de l'apprenant à partir des traces de ses interactions avec l'EIAH est une tâche qui peut être modélisée et mise en œuvre grâce à un processus indépendant du domaine, pour peu que l'EIAH repose sur un modèle explicite des connaissances enjeu de l'apprentissage et que les activités proposées à l'apprenant soient liées à ces connaissances. Ainsi la littérature propose des techniques génériques de diagnostic telles que le *Knowledge Tracing* (Corbett & Anderson, 1994) ou le *Constraint-based Modeling* (Ohlsson, 1994).

Cette section présente une technique générique qui permet d'évaluer la maîtrise par l'apprenant de compétences représentées au sein d'une ontologie. Cette technique est fondée sur les travaux de l'équipe METAH du LIG sur la notion de praxéologie (Chaachoua, 2010).

2.1 Cadre des recherches

Le projet "Cartographie des Savoirs" est un Projet Investissements d'Avenir piloté par la société Educlever, qui édite le site Maxicours.com. Financé pour deux ans (2013 et 2014) par l'appel e-Education 2, il vise à faciliter la personnalisation des parcours d'apprentissage, en collaboration avec l'équipe METAH (Modèles et Technologies pour l'Apprentissage Humain) du LIG (Laboratoire d'Informatique de Grenoble) et l'équipe SILEX du LIRIS.

Le financement de ce projet m'a permis de recruter Sonia Mandin sur un contrat post-doctoral. Les contributions présentées dans cette section sont issues de son travail, en collaboration avec les partenaires du projet.

2.2 Question de recherche

Dans le projet "Cartographie des Savoirs", l'objectif est d'évaluer les savoir-faire de l'apprenant tout au long de son cursus scolaire, et de conserver les résultats de ces évaluations à différents stades dans un profil de l'apprenant. Ce profil sera ensuite utilisé

pour améliorer la visibilité des progrès de l'apprenant et pour adapter les parcours qui lui sont proposés au sein de la plateforme.

Pour évaluer les savoir-faire de l'apprenant, il faut pouvoir disposer d'un référentiel dans lequel les savoir-faire des programmes scolaires sont organisés et d'un moyen pour identifier les savoir-faire mis en œuvre dans les productions des élèves. La question à traiter est alors de définir un modèle de diagnostic du taux de maîtrise des savoir-faire à partir des évaluations des productions d'un apprenant.

Dans notre projet, l'équipe METAH du LIG propose de constituer un référentiel des savoir-faire représenté par une ontologie (Chaachoua, 2010). L'ontologie décrit les rapports entre types de tâche, techniques et technologies (cf. descriptions ci-dessous), et leur présence ou non dans des programmes qui sont découpés en un ensemble de disciplines, domaines, secteurs et thèmes. L'ontologie résultante est appelée Ontoprax (Chaachoua, Ferraton, & Desmoulins, 2013). Elle représente une organisation des savoir-faire attendus par l'institution et listés à partir des pratiques observées.

Plus précisément, dans cette ontologie, on définit un *type de tâche* par l'ensemble des tâches réalisables par la (ou les) même(s) *technique*(s) impliquant elle(s)-même(s) la mise en œuvre d'autres types de tâche. Des types de tâche peuvent ainsi être liés entre eux par des relations de *prérequis*. Par exemple, sur la **Figure 43**, pour effectuer le type de tâche "multiplier deux entiers" par la technique 2 qui est celle de la multiplication posée, des types de tâche prérequis sont de poser correctement la multiplication en colonne, de savoir appliquer les tables de multiplication et d'additionner deux entiers.

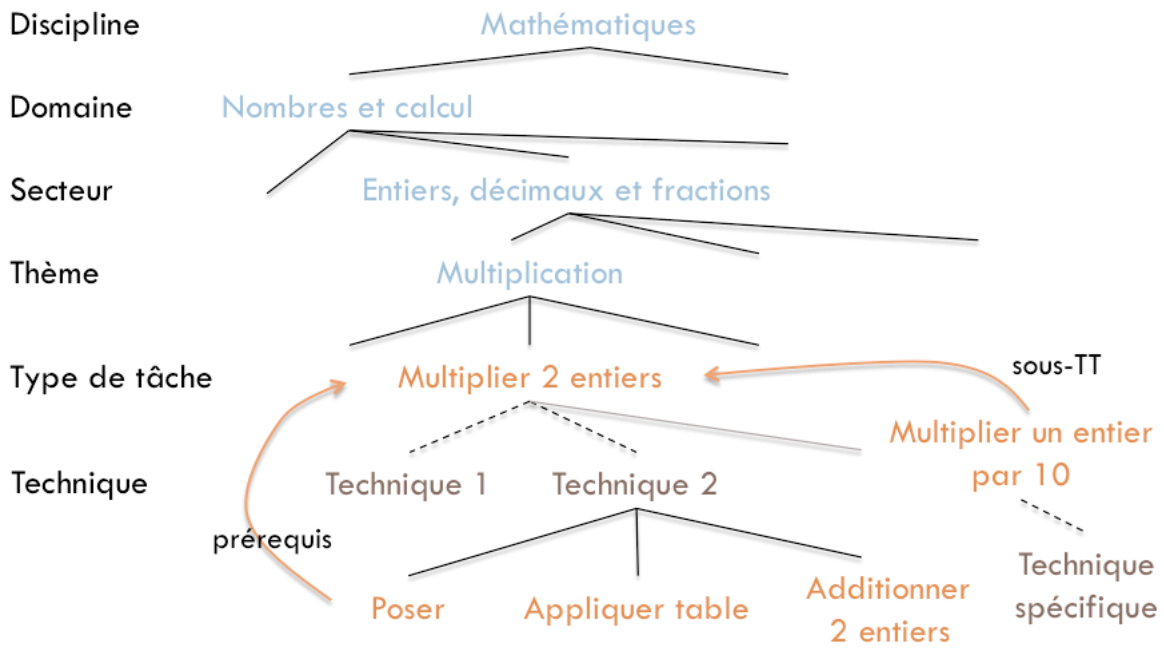

Figure 43. Exemple simplifié permettant de présenter les concepts d'Ontroprax

Il arrive qu'un type de tâche puisse être réalisé par une technique commune à un autre type de tâche en plus de pouvoir l'être par une technique plus spécifique qui lui est propre. En ce cas, on considèrera que ce type de tâche est le *sous-type de tâche* d'un type de tâche plus général. Des types de tâche peuvent donc aussi être liés entre eux par des relations de sous-types de tâche. Dans l'exemple de la **Figure 43**, pour multiplier un entier par 10, la technique

générale de la multiplication posée peut être utilisée, mais une autre technique plus spécifique existe aussi, celle de simplement ajouter un 0 à la fin du multiplié.

Enfin, les *technologies*, éléments théoriques justifiant les techniques, permettent le regroupement des types de tâche dans des ensembles plus larges, les *thèmes* (e.g. techniques opératoires de la multiplication). Ils sont reliés, en fonction de pratiques institutionnelles, à des *secteurs* (e.g. entiers, décimaux et fractions), eux-mêmes assemblés en *domaines* (e.g. nombre et calculs) rattachés à des *disciplines* (e.g. mathématiques) (cf. la partie haute de la **Figure 43**).

L'ontologie Ontoprax est peuplée par l'équipe METAH du LIG et par la société Educlever à partir de l'analyse des programmes et des manuels scolaires. Elle nous fournit un modèle des savoir-faire dont nous cherchons à évaluer la maîtrise. Pour évaluer ces savoir-faire, il faut observer leur mise en œuvre à travers des exercices. Sur Maxicours.com, les exercices sont des questions à choix multiples (QCM) qui traduisent la bonne ou mauvaise maîtrise de types de tâche spécifiques. Ces types de tâche sont à lier au référentiel de connaissances. La société Educlever a donc défini une base de QCM dans laquelle chaque question est identifiée comme nécessitant la réalisation d'un type de tâche précis d'Ontoprax mis en œuvre éventuellement par une technique spécifique. Les items de réponse proposés à chaque QCM sont quant à eux associés à la bonne ou mauvaise réalisation du type de tâche et technique attendus. Certains items erronés peuvent, le cas échéant, être directement associés à la cause supposée de l'erreur (prérequis mal ou non réalisé).

La question de recherche qui se pose à nous est donc d'estimer le niveau de maîtrise de l'élève pour chaque savoir-faire du référentiel. Il s'agit ainsi de passer d'un diagnostic des productions (diagnostic local) à un diagnostic des compétences (diagnostic global). Le modèle de profil utilisé est ici la structure de l'ontologie. Dans ce projet, l'intérêt est en effet porté sur les compétences de l'apprenant, c'est pourquoi le profil est calqué sur l'ontologie, mais cela n'empêche pas d'envisager d'ajouter d'autres éléments au modèle de profil, comme ceux relatifs à son comportement ou à ses préférences.

La **Figure 44** replace cette problématique par rapport à l'approche présentée au chapitre 1. En effet, le modèle Ontoprax fourni par le LIG est un méta-modèle de connaissances qui permet dans un domaine donné de définir un modèle des connaissances du domaine. Dans le cadre du projet "Cartographie des Savoirs", les domaines traités sont ceux du français et des mathématiques enseignés au cycle 3 de l'école élémentaire. Le processus d'acquisition des connaissances est également géré par le LIG, qui est actuellement en train de concevoir une interface permettant à un didacticien ou à un praticien de peupler l'ontologie sans avoir besoin de compétences en informatique. Le processus d'annotation d'activités pédagogiques pour les relier à l'ontologie, également fondé sur le méta-modèle du LIG, est pris en charge par la société Educlever. La contribution que nous proposons dans le cadre de ce projet est la conception d'un processus de calcul du taux de maîtrise de l'apprenant pour chaque élément de l'ontologie. Ce processus présenté dans la sous-section suivante est indépendant du domaine et s'appuie sur le méta-modèle de connaissances. Il permet d'exploiter l'ontologie définie dans un domaine donné X ainsi que les productions des élèves dans ce domaine pour élaborer un profil de chaque apprenant.

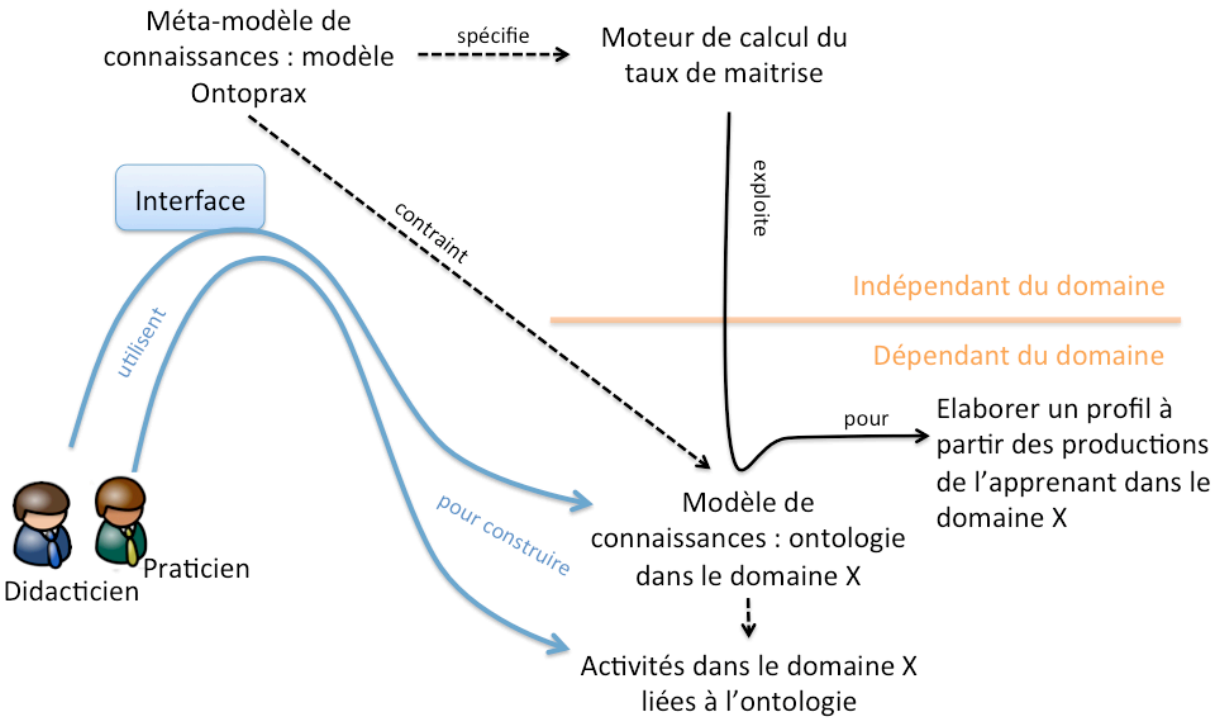

Figure 44. Instanciation de l'approche présentée au chapitre 1 pour l'estimation du taux de maîtrise de types de tâches par un apprenant. *Le méta-modèle Ontoprax ainsi que le processus d'acquisition des connaissances en découlant sont apportés par le LIG. Notre apport concerne le moteur de calcul du taux de maîtrise.*

2.3 Contributions

Le but de la technique de diagnostic définie était de rendre compte dans un profil des taux de maîtrise de types de tâche et techniques évalués à travers des exercices. Nous avons choisi d'utiliser l'ontologie comme structure de base du profil, et de l'enrichir avec les taux de maîtrise à calculer. Nous disposons ainsi dans chaque profil d'éléments (types de tâche, techniques, thèmes, secteurs, domaines et disciplines) auxquels des valeurs de maîtrise sont associées, en stockant l'évolution des valeurs dans le temps. Nous distinguons trois types de valeurs de maîtrise : la valeur de base, la valeur enrichie et la valeur agrégée.

Quand un apprenant réalise un QCM, il est évalué directement sur le type de tâche sollicité (évaluation directe) et peut aussi l'être sur les prérequis à ce type de tâche (évaluation indirecte). La prise en compte ou non d'une évaluation indirecte nous conduit à proposer deux valeurs distinctes pour chaque type de tâche :

La **valeur de base** (VB) correspond à la maîtrise d'un type de tâche d'après des évaluations directes. Ce niveau rend compte de la capacité des élèves à réaliser des exercices demandant la réalisation d'un même type de tâche. Cette valeur est assimilable à une note.

La **valeur enrichie** (VE) est issue d'un processus que nous nommons *propagation descendante*. Ce processus consiste à déterminer les types de tâche qui sont évalués indirectement à partir des QCM. Il aboutit à une valeur VE qui correspond à la maîtrise d'un type de tâche d'après des évaluations directes et indirectes. En d'autres termes, il s'agit d'évaluer pour un type de tâche donné, la capacité des élèves à réaliser l'ensemble des exercices qui demandent sa réalisation ou en nécessitent sa mise en œuvre à travers le

recours à une technique propre à un autre type de tâche. Par exemple, l'aptitude d'un apprenant à effectuer une addition posée peut-être évaluée en présentant directement le type de tâche mais aussi en sollicitant une multiplication dont le multiplicateur est un entier à plusieurs chiffres, type de tâche dont la technique requiert une addition. La valeur enrichie tient également compte de l'évolution des réponses de l'apprenant dans le temps. En effet, sur 10 réponses à un même type de tâche, le taux de maîtrise évalué ne sera pas le même selon que l'apprenant répond juste une fois sur deux ou qu'il répond faux aux 5 premières réponses puis juste aux 5 suivantes. De même, si un TT n'a pas été évalué depuis longtemps, le taux de fiabilité de la maîtrise préalablement calculée doit baisser.

La maîtrise de types de tâche peut aussi tenir compte de la maîtrise d'autres types de tâche plus spécifiques mais partageant une même technique générale. Pour estimer la maîtrise de la multiplication de deux entiers, il faut tenir compte de la capacité à multiplier deux entiers quelconques par une technique générale mais aussi de la capacité à multiplier un entier par 10 en ajoutant simplement un 0 à la fin. L'apprenant maîtrise-t-il les techniques spécifiques à des cas plus particuliers à un type de tâche ? Sait-il choisir la technique la plus appropriée ? Nous introduisons donc une troisième valeur que nous appelons **valeur agrégée** (VA) et qui est le résultat d'un autre processus appelé *propagation ascendante*. L'agrégation peut ensuite se poursuivre au niveau des thèmes avec les VA des types de tâche les plus généraux, puis au niveau des secteurs, des disciplines et des domaines avec les VA des éléments les constituant.

Trois valeurs de maîtrise des savoir-faire (VB, VE et VA) sont ainsi disponibles pour chaque type de tâche et technique évalués. Le lecteur intéressé par une description plus précise du mode de calcul de ces trois valeurs pourra consulter (Mandin & Guin, 2014a, 2014b).

2.4 Évaluation et perspectives

Nous menons actuellement des expérimentations afin d'évaluer notre modèle de calcul des taux de maîtrise. Il s'agit de valider le modèle de diagnostic dans son ensemble mais aussi à chaque étape. Pour ce faire, dans un premier temps, il s'agit de valider le passage des valeurs brutes aux valeurs enrichies (propagation descendante). Les mesures indirectes sont-elles correctement attachées à des types de tâche ? La méthode de prise en compte conjointe des mesures directes et indirectes, tout en s'intéressant à la date de survenue de chacune d'elles, fournit-elle de bons résultats ? Le passage des valeurs enrichies aux valeurs agrégées (propagation ascendante) est-il correct ? Le modèle décrit ici pourra aussi être comparé à d'autres modèles alternatifs afin d'en situer sa performance face à d'autres méthodes. Enfin, une évaluation devra être réalisée pour recueillir les avis d'utilisateurs (enseignants et élèves) sur les résultats finaux proposés dans le profil.

Pour améliorer le contenu du profil, il conviendrait également d'estimer un taux de fiabilité. Ce taux pourrait intégrer la certitude des apprenants dans leurs réponses et leur aptitude à expliquer leurs erreurs (Labat, 2002). Il peut aussi tenir compte d'expositions antérieures des apprenants aux mêmes questions (Montepare, 2005; Wagner-Menghin, Preusche, & Schmidts, 2013). Enfin, le taux de fiabilité des taux de maîtrise pourrait être fonction de la qualité des QCM soumis en s'intéressant au pouvoir distractif des items de réponse proposés (Johnstone, 2003) et en analysant la difficulté et le potentiel des QCM à discriminer la maîtrise d'un type de tâche (McAlpine, 2002).

3 Instancier et comparer des techniques génériques de diagnostic

Le processus de diagnostic présenté dans la section précédente est indépendant du domaine, comme d'autres techniques de diagnostic de la littérature telles que le *Knowledge Tracing* (Corbett & Anderson, 1994) ou le *Constraint-based Modeling* (Ohlsson, 1994). Si un concepteur d'EIAH souhaite utiliser une de ces techniques génériques de diagnostic pour le domaine qui l'intéresse, le coût de mise en œuvre est important et doit être répété pour chacune d'entre elles avant de pouvoir comparer leur résultats pour choisir celle qui est la plus adaptée à ses besoins. Cette section présente des propositions effectuées pour faciliter le travail d'un concepteur de diagnostic dans une telle situation.

3.1 Cadre des recherches

Ces travaux ont été menés par Sébastien Lallé dans le cadre de sa thèse dirigée par Vanda Luengo (LIG). Cette thèse (2010-2013) a été financée par la région Rhône-Alpes dans le cadre du SP 5 (benchmarks et consensus de validation) du projet "Personnalisation des EIAH" du Cluster ISLE.

3.2 Questions de recherche

L'objectif de nos travaux est ici d'assister un utilisateur qui souhaite mettre en place un diagnostic des connaissances et compétences de l'apprenant pour un EIAH donné. Nous supposons que cet utilisateur dispose déjà de traces de l'EIAH et qu'il souhaite donc mettre en œuvre un processus permettant à partir de ces traces d'élaborer un profil de l'apprenant. Nous supposons également que le diagnostic des productions de l'apprenant est déjà intégré aux traces, ce qui nous permet de nous abstraire d'une partie des connaissances qui sont intrinsèquement liées au domaine.

La question étudiée ici est de permettre au concepteur du diagnostic de choisir au mieux parmi les techniques génériques de diagnostic de la littérature. Cela pose donc deux questions :

- Comment instancier ces techniques génériques pour un EIAH et un domaine donné ?
- Comment comparer l'efficacité de ces techniques afin de choisir celle qui répond le mieux aux attentes du concepteur pour cet EIAH et ce domaine ?

Le principal verrou réside dans le fait que la littérature ne propose pas de processus unique permettant d'appliquer plusieurs techniques génériques de diagnostic à un même domaine. Les travaux existants imposent de n'instancier qu'un seul diagnostic, et ne permettent de comparer que des variantes d'un même diagnostic.

Pour lever ce verrou, nous proposons un méta-modèle de techniques de diagnostic, qui permet de représenter de manière unifiée la plupart des techniques génériques de diagnostic de la littérature. Nous associons à ce méta-modèle un processus unique permettant d'instancier différentes techniques de diagnostic à un même domaine. Ce processus s'appuie sur un algorithme d'apprentissage automatique qui permet d'acquérir les connaissances de diagnostic dépendantes du domaine en utilisant une ontologie fournie par le concepteur et en analysant le contenu des traces. L'apprentissage semi-automatique vise à faciliter le travail d'instanciation tout en préservant l'interprétabilité des résultats par un expert humain. Nous proposons également des critères de comparaison qui permettent au concepteur d'évaluer les différentes techniques construites afin de choisir celle qui lui

convient le mieux. Ces propositions ont été mises en œuvre dans la plateforme PlaCID (Platform for Comparing and Instanciating Diagnostics).

La **Figure 45** décrit en quoi ces contributions relèvent bien de l'approche scientifique présentée au chapitre 1. En effet, l'utilisateur (qui est ici le concepteur du diagnostic) utilise la plateforme PlaCID, qui s'appuie sur un méta-modèle de techniques de diagnostic, afin d'éliciter des connaissances relatives au domaine qui l'intéresse et aux traces dont il dispose pour le domaine et l'EIAH concerné. Un algorithme d'apprentissage semi-automatique utilise ces connaissances pour instancier à cet EIAH et à ce domaine plusieurs techniques génériques de diagnostic issues de la littérature. Ces techniques de diagnostic peuvent ensuite être comparées grâce à un processus permettant de les appliquer aux traces et à un ensemble de critères de comparaison. Ce processus permet à l'utilisateur de comparer les performances des différentes techniques sur son domaine et son EIAH, afin de choisir celle qui répond le mieux à ses besoins. À la fin de ce processus, l'utilisateur dispose donc d'une technique de diagnostic implémentée et adaptée à son EIAH.

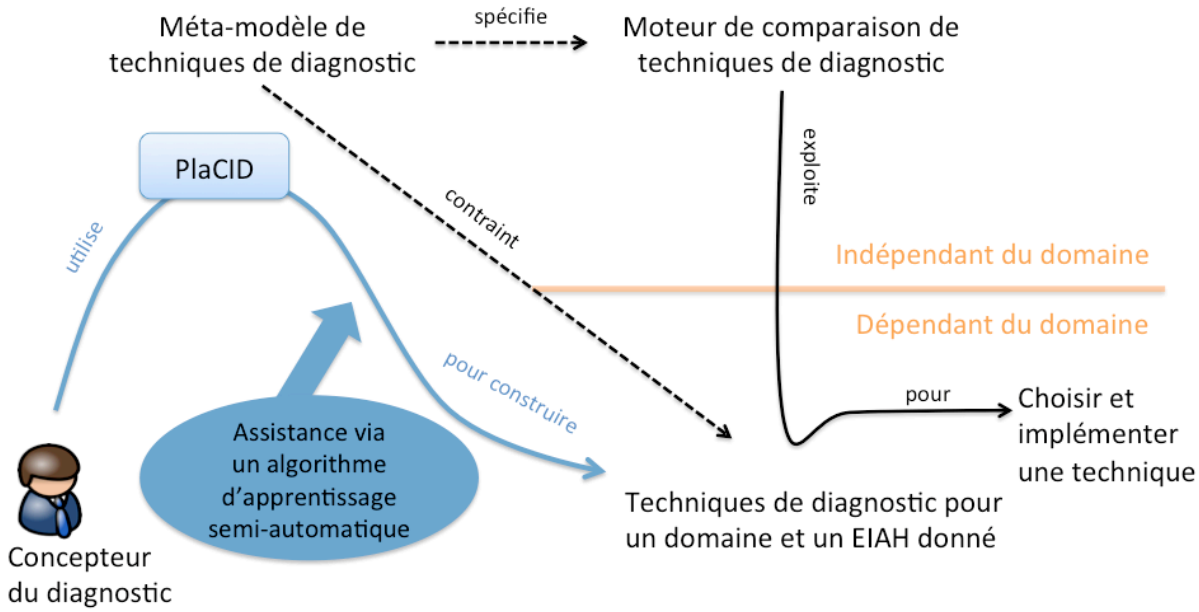

Figure 45. Instanciation de l'approche présentée au chapitre 1 aux questions de l'application à un domaine donné de techniques génériques de diagnostic et de la comparaison des performances de ces techniques instanciées au domaine.

3.3 Contributions

La première contribution de ce travail est un méta-modèle de techniques de diagnostic, établi à partir d'une étude approfondie de l'état de l'art et des techniques de diagnostic qui peuvent s'appliquer à des domaines variés. Nous définissons une **technique générique de diagnostic** comme l'association entre un *modèle de diagnostic* et une *implémentation*. Une technique de diagnostic prend en entrée les traces d'un apprenant enrichies par un diagnostic de ses productions et élabore le profil de l'apprenant défini par le modèle de diagnostic.

Le **modèle de diagnostic**, qui est proche d'un réseau sémantique dans notre méta-modèle, vise à spécifier les éléments requis et les contraintes pour diagnostiquer les connaissances des apprenants. Un modèle de diagnostic est souvent basé sur une théorie de l'apprentissage et représente donc un point de vue ou une approche particulière. Il est défini par l'ensemble des types des variables présentes dans les traces, l'ensemble des types de variables des connaissances à évaluer pour élaborer le profil de l'apprenant, l'ensemble des relations entre ces variables et des contraintes sur ces variables et ces relations.

Le lecteur intéressé trouvera dans (Sébastien Lallé, 2013) une formalisation de ce méta-modèle ainsi que les modèles qui en dérivent pour le *Knowledge tracing* (Corbett & Anderson, 1994) ainsi que ses dérivés *LR-DBN* (Xu & Mostow, 2011) et *Conjunctive Knowledge Tracing* (Koedinger, Pavlik, Stamper, Nixon, & Ritter, 2011), le *Constraint-based* (Ohlsson, 1994), le *Control-based* (Minh Chieu, Luengo, Vadcard, & Tonetti, 2010), et *Item response theory* (Johns, Mahadevan, & Woolf, 2006).

Une **implémentation** permet de mettre en œuvre le modèle de diagnostic grâce à un formalisme calculable, c'est-à-dire de pouvoir lire les traces des apprenants et inférer l'état de leurs connaissances. Il s'agit par exemple de modèles de Markov cachés, de réseaux bayésiens dynamiques, de règles SI / ALORS, de régression linéaire, de la logique floue, etc. Mais toute implémentation ne permet pas nécessairement d'implémenter un modèle de diagnostic donné. Un modèle de Markov caché ne peut par exemple pas implémenter le modèle de diagnostic *Control-based*. On dira qu'un modèle de diagnostic MD et une implémentation I sont compatibles si I peut implémenter toute instance de MD.

Ce méta-modèle de techniques génériques de diagnostic pose une définition commune pour un ensemble de diagnostics très différents et permet de délimiter un sous-ensemble de techniques de diagnostic pour lesquelles nous pouvons étudier l'instanciation et la comparaison de façon générique.

La deuxième contribution de ce travail est en effet une **méthode d'instanciation** de différents diagnostics *via* un algorithme d'apprentissage semi-automatique, guidé par le concepteur du diagnostic. Celui-ci définit une ontologie qui fournit de la sémantique aux traces. L'apprentissage semi-automatique vise à faciliter le travail d'instanciation tout en préservant l'interprétabilité des résultats par un expert humain. La principale innovation est la possibilité d'instancier et d'implémenter avec la même méthode des diagnostics différents de façon indépendante du domaine d'apprentissage.

L'ontologie est construite par le concepteur à partir d'une ontologie par défaut fondée sur CREAM-C (Nkambou, Gauthier, & Frasson, 1997), qui est compatible avec toutes les techniques de diagnostic considérées. Cette ontologie permet de spécifier deux types de variables : les observés que l'on trouve dans les traces, et les connaissances à diagnostiquer, qui sont deux éléments fondamentaux du méta-modèle de techniques de diagnostic. Les variables spécifiées sont ensuite liées aux traces par le concepteur, en associant une variable de l'ontologie à une ou plusieurs variables des traces. Deux niveaux d'ontologies peuvent être construits :

- une ontologie de haut niveau, où les classes de l'ontologie représentent les variables présentes dans les traces ;

- une ontologie détaillée, où les classes peuvent représenter les variables des traces ainsi que des exemples de valeurs de ces variables dans les traces.

L'algorithme semi-automatique est ensuite exécuté afin de construire un ensemble de techniques de diagnostic à partir d'une base de traces d'apprenants et de l'ontologie du concepteur. Cet algorithme suit une démarche de recherche unique fondée sur un score, mais utilise des techniques adaptées aux différents modèles de diagnostic et implémentations considérées. Il en résulte des techniques instanciées, dont le code source peut par exemple être intégré à un EIAH pour diagnostiquer les connaissances des apprenants. L'algorithme semi-automatique est décrit en détail dans (Sébastien Lallé, 2013) et (Sébastien Lallé, Luengo, & Guin, 2013) avec des exemples d'application.

La troisième contribution est un ensemble de **critères de comparaison** pour comparer les résultats des différentes techniques de diagnostic instanciées à un domaine. Les principales innovations sont ici la possibilité de calculer ces critères pour différents diagnostics, et la possibilité de spécifier de nouveaux critères pouvant répondre à des questions de recherche ou des besoins d'experts en EIAH. Ces critères sont des algorithmes appliqués sur les traces dont dispose le concepteur du diagnostic et dont les résultats aident à positionner les techniques instanciées entre elles, par exemple *via* des tests statistiques, des algorithmes de data-mining, des critères de génie logiciel, etc. Ces critères étant calculés pour comparer des techniques instanciées, leurs résultats ne sont valides que pour les traces et le domaine du concepteur. Ce dernier doit analyser et interpréter ces résultats par rapport à ses besoins.

Le processus présenté ici permettant à un concepteur d'appliquer à son EIAH portant sur un domaine donné des techniques génériques de diagnostic, pour ensuite les comparer *via* des critères lui permettant de choisir la technique la plus pertinente par rapport à ses besoins, peut également être utile dans un autre cadre. Il peut en effet servir à un chercheur sur le diagnostic des connaissances qui propose une nouvelle technique, pour instancier cette technique sur plusieurs domaines et pour comparer sur ces domaines cette nouvelle technique aux techniques de la littérature afin de la positionner.

3.4 Évaluations

Nous avons évalué ces contributions à travers deux expérimentations.

La première évaluation a montré la faisabilité de nos propositions, c'est-à-dire la possibilité d'instancier et comparer grâce à la plateforme PlaCID plusieurs techniques de diagnostic fondées sur des modèles différents. Nous avons ainsi instancié quatre techniques différentes (une fondée sur *Constraint-based*, une sur *Control-based* et deux implémentations différentes de *Knowledge Tracing*) sur trois domaines différents (apprentissage de la géométrie, de la lecture de l'anglais, de la chirurgie orthopédique). Nous avons pour cela utilisé trois bases de traces d'apprenants, celles de Geometry Tutor (J. R. Anderson, Boyle, & Yost, 1985), Reading Tutor (Mostow, Aist, & Feltovich, 2001) et TELEOS (Luengo, 2009). Pour chacun de ces domaines, notre algorithme de construction a pu instancier les techniques de diagnostic considérées (Sébastien Lallé, Luengo, et al., 2013; Sébastien Lallé, 2013), et tous les critères de comparaison ont pu être calculés (Sébastien Lallé, Luengo, & Guin, 2012).

La deuxième évaluation a montré la possibilité de spécifier et d'implémenter un nouveau critère de comparaison plus complexe qui portait sur une question de recherche spécifique au domaine des EIAH : l'impact des techniques de diagnostic sur le choix d'un type d'aide à fournir à un apprenant. Ce critère répond à une question de recherche d'un expert du diagnostic des connaissances. Les résultats ont montré d'une part la possibilité de développer un tel critère avec la plateforme PlaCID, et d'autre part que l'impact des techniques de diagnostic s'est révélé significatif sur l'apprentissage de stratégies d'aide dans le domaine de la lecture de l'anglais (Lallé, Mostow, Luengo, & Guin, 2013; Lallé, Mostow, et al., 2013b).

La validation des propositions effectuées dans ce travail permet maintenant d'envisager l'expérimentation de la plateforme PlaCID auprès d'utilisateurs ciblés par cet outil. L'utilisation de la plateforme demande une bonne connaissance du domaine du diagnostic des connaissances ainsi que des compétences en représentation des connaissances et en analyse de données. Les perspectives de ce travail consistent donc à améliorer l'utilisabilité de PlaCID afin de pouvoir mener des expérimentations destinées à mesurer son utilité.

Conclusion et perspectives

Ce chapitre a permis de présenter plusieurs travaux de recherche autour de la thématique de l'élaboration de profils d'apprenants à partir de l'analyse de leurs traces d'interaction avec un EIAH.

Un premier ensemble de travaux est destiné à faciliter l'accès à la plateforme kTBS développée au sein de l'équipe SILEX pour un utilisateur non informaticien comme un enseignant ou un auteur de MOOC. Ces travaux visent à permettre à cet utilisateur d'importer des traces d'EIAH au sein de la plateforme, puis d'interroger ces traces à l'aide d'un langage proche du français pour calculer des indicateurs destinés à alimenter un profil d'apprenant ou à effectuer une analyse des usages.

Dans de nombreux contextes, l'élaboration d'un profil de l'apprenant nécessite l'évaluation de la maîtrise par l'apprenant de compétences liées au domaine d'apprentissage. Nous proposons donc un modèle de diagnostic qui permet d'estimer le taux de maîtrise par un apprenant d'un type de tâche représenté au sein d'une ontologie fondée sur le modèle praxéologique (Chaachoua, 2010). Des expérimentations destinées à évaluer ce modèle sont en cours.

Enfin, une méthode a été présentée pour permettre à un concepteur d'EIAH d'appliquer des techniques génériques de diagnostic à son EIAH portant sur un domaine donné, et pour lequel il dispose de traces. La plateforme PlaCID lui permet de construire plusieurs instances de techniques de diagnostic avant de comparer leur efficacité pour les traces de son EIAH, afin de choisir celle qui convient le mieux à ses besoins. Dans ce travail, effectué par Sébastien Lallé dans le cadre de sa thèse, on retrouve plusieurs problématiques souvent présentes dans mes travaux, telles que l'assistance à l'utilisateur auteur ou concepteur, l'acquisition interactive de connaissances, ou des processus semi-automatiques. L'utilisation de techniques issues de l'apprentissage automatique pour contribuer sur ces problématiques est par contre un élément nouveau, mais qu'il me semble intéressant de développer dans le cadre de mes perspectives de recherche.

Les perspectives de mes recherches sur cette thématique de l'analyse des traces pour l'élaboration de profils d'apprenants prendront place à court terme au sein du projet ANR HUBBLE (HUman oBservatory Based on anaLysis of e-LEarning traces), piloté par Vanda Luengo et visant la création d'un observatoire pour la construction et le partage de processus d'analyse des traces e-learning massives. Ce projet sera l'occasion de mutualiser et finaliser le développement des diverses contributions destinées à faciliter l'utilisation de la plateforme kTBS dans ce cadre, mais aussi de mettre en commun ces travaux avec ceux du LIG autour de la plateforme Undertracks (Bouhineau et al.,2013) et ceux du LIUM autour du langage UTL2 et DCL4UTL (C Choquet & Iksal, 2007; Pham-Thi et al., 2009). Des expérimentations au sein du projet avec des utilisateurs non informaticiens devraient également permettre de mieux évaluer les contributions présentées dans ce chapitre.

Synthèse des contributions

Question de recherche : Interroger des traces pour calculer des indicateurs

- **Contributions théoriques** :
  - Méta-modèle de traces d'EIAH
  - Processus d'importation de traces existantes à kTBS
  - Processus d'adaptation de patrons de méthodes de calcul d'indicateurs
  - Langage naturel contrôlé d'interrogation des traces
- **Systèmes développés** :
  - xCollector
  - SPARE-LNC
- **Déploiement** : SPARE-LNC a été intégré à un outil de visualisation et de transformation de traces développé dans l'équipe SILEX.
- **Co-auteurs** : Lemya Settouti, Vanda Luengo, Mohamed Besnaci, Pierre-Antoine Champin, Bryan Kong Win Chang, Marie Lefevre

Question de recherche : Élaborer un profil de l'apprenant en se fondant sur un référentiel des connaissances du domaine.

- **Contribution théorique** : modèle de calcul de 3 taux de maîtrise par type de tâche décrit dans le référentiel.
- **Système développé** : mise en œuvre par Educlever dans la Cartographie des Savoirs
- **Déploiement** : utilisation de la Cartographie des Savoirs dans les école de la Somme Numérique et au CNED
- **Co-auteur** : Sonia Mandin

Question de recherche : Instancier et comparer des techniques génériques de diagnostic

- **Contributions théoriques** :
  - Méta-modèle de techniques de diagnostic
  - Méthode d'instanciation de techniques de diagnostic à un domaine
  - Critères de comparaison de techniques de diagnostic
- **Système développé** : PlaCID, plateforme permettant l'instanciation et la comparaison de techniques de diagnostic
- **Co-auteurs** : Sébastien Lallé, Vanda Luengo

Pour en savoir plus

• Sur le processus de collecte :

Besnaci, M., Guin, N., & Champin, P.-A. (2014). *Importation de traces existantes dans un système de gestion de bases de traces*. Rapport de recherche LIRIS RR-LIRIS-2014-013

• Sur l'élaboration de profils par analyse de traces et les méta-modèles liés :

Settouti, L., Guin, N., Luengo, V., & Mille, A. (2010). A Trace-Based Learner Modelling Framework for Technology-Enhanced Learning Systems. In *ICALT 2010* (pp. 73–77). Sousse, Tunisia.

• Sur le processus d'adaptation de patrons de requêtes :

Settouti, L., Guin, N., & Luengo, V. (2011). Adaptable and Reusable Query Patterns for Trace-Based Learner Modelling. In *Sixth European Conference on Technology Enhanced Learning: Towards Ubiquitous Learning (EC-TEL 2011)* (pp. 384–397). Palerme, Italie.

• Sur le langage SPARE-LNC pour interroger les traces :

Kong Win Chang, B., Guin, N., Lefevre, M., & Champin, P.-A. (2014). *Conception d'un langage d'interrogation des traces accessible à des non-informaticiens*. Rapport de recherche RR-LIRIS-2014-015

• Sur l'élaboration de profils en lien avec une ontologie du domaine :

Mandin, S., & Guin, N. (2014). *Prise en compte d'une ontologie des savoirs dans la construction d'un profil d'apprenant*. Rapport de recherche RR-LIRIS-2014-014

• Sur l'instanciation de techniques génériques et la comparaison des techniques instanciées :

Lallé, S. (2013). *Assistance à la construction et la comparaison de techniques de diagnostic des connaissances*. Université Joseph Fourier - Grenoble.

Publications liées

Conférences internationales

Lallé, S., Luengo, V., & Guin, N. (2012a). An Automatic Comparison between Knowledge Diagnostic Techniques. In *Intelligent Tutoring Systems* (pp. 622–623). Chania, Crete, Greece.

Lallé, S., Luengo, V., & Guin, N. (2013b). Assistance in building student models using knowledge representation and machine learning. In *Artificial Intelligence in Education* (pp. 754–757). Springer, Lecture Notes in Computer Science.

Lallé, S., Mostow, J., Luengo, V., & Guin, N. (2013). Comparing Student Models in Different Formalisms by Predicting Their Impact on Help Success. In *Artificial Intelligence in Education* (pp. 161–170). Springer.

Mandin, S., & Guin, N. (2014). Basing learner modelling on an ontology of knowledge and skills. In *IEEE International Conference on Advanced Learning Technologies* (pp. 321–323).

Settouti, L., Guin, N., & Luengo, V. (2011). Adaptable and Reusable Query Patterns for Trace-Based Learner Modelling. In *Sixth European Conference on Technology Enhanced Learning: Towards Ubiquitous Learning (EC-TEL 2011)* (pp. 384–397). Palerme, Italie.

Settouti, L., Guin, N., Luengo, V., & Mille, A. (2010). A Trace-Based Learner Modelling Framework for Technology-Enhanced Learning Systems. In *ICALT 2010* (pp. 73–77). Sousse, Tunisia.

Conférences nationales

Lallé, S., Luengo, V., & Guin, N. (2012b). Méthodologie d'assistance pour la comparaison de techniques de diagnostic des connaissances. In *Technologies de l'Information et de la Communication pour l'Enseignement* (pp. 11–19). Lyon, France.

Lallé, S., Luengo, V., & Guin, N. (2013a). Assistance à la conception de techniques de diagnostic des connaissances. In *Environnements Informatiques pour l'Apprentissage Humain (EIAH'2013)* (pp. 203–214). Toulouse, France.

Lallé, S., Mostow, J., Luengo, V., Guin, N., Metah, L. I. G., Fourier, U. J., & Lyon, U. (2013). Comparaison de l'impact de techniques de diagnostic des connaissances sur l'apprentissage d'une stratégie d'aide. In *Journée EIAH&IA 2013*. Toulouse, France.

CHAPITRE B2
GÉNÉRER DES EXERCICES

Résumé

Ce chapitre présente des recherches menées autour de la question de la génération semi-automatique d'exercices et décrit comment l'approche GEPPETO permet d'y répondre. Cette approche peut être utilisée d'une part pour générer des exercices de types génériques (QCM, textes à trous, etc.), ce qui a donné lieu à la réalisation d'un outil auteur de création d'exercices d'évaluation. Elle permet d'autre part de piloter des EIAH existants de manière à générer ou choisir des exercices de types plus spécifiques.

INTRODUCTION 154

1 CADRE DES RECHERCHES 154

2 QUESTIONS DE RECHERCHE : GÉNÉRER DES EXERCICES DE TYPES GÉNÉRIQUES OU SPÉCIFIQUES 155

3 CONTRIBUTION : DES GÉNÉRATEURS D'EXERCICES DE TYPES GÉNÉRIQUES POUR UN OUTIL AUTEUR D'EXERCICES D'ÉVALUATION 156

3.1 L'APPROCHE GEPPETO-P 157
3.2 ASKER : UN OUTIL AUTEUR D'EXERCICES D'ÉVALUATION 157
DIAGNOSTIC DES RÉPONSES DE L'APPRENANT 160
CONNAISSANCES DU DOMAINE 160
3.3 BILAN 161

4 CONTRIBUTION : AGIR SUR DES GÉNÉRATEURS D'EXERCICES INTÉGRÉS À DES EIAH EXISTANTS 163

4.1 L'APPROCHE GEPPETO-S ET SON IMPLÉMENTATION DANS LE LOGICIEL ADAPTE 163
4.2 ILLUSTRATION SUR L'EIAH AMBRE-ADD 164
4.3 BILAN 165

CONCLUSION ET PERSPECTIVES 166

SYNTHÈSE DES CONTRIBUTIONS 167

POUR EN SAVOIR PLUS 167

PUBLICATIONS LIÉES 168

Introduction

Pour personnaliser les EIAH, il faut d'une part disposer d'informations sur l'apprenant, question qui a été étudiée au chapitre précédent, et il faut d'autre part pouvoir agir sur les activités que l'on propose à l'apprenant au sein de l'EIAH. Mes travaux portant essentiellement sur des EIAH dans lesquels on propose à l'apprenant des activités de résolution de problèmes, agir sur les activités de l'EIAH revient à configurer les exercices proposés à l'apprenant. J'utilise le terme d'exercices pour englober des activités allant de la réponse à des QCM jusqu'à la résolution de problèmes nécessitant une démarche construite comme dans AMBRE.

Configurer les exercices à proposer à l'apprenant peut consister à agir sur le type de l'exercice, son contenu, ainsi que sur les fonctionnalités de l'EIAH utilisables par l'apprenant pour réaliser l'exercice. Pour configurer des exercices, on peut choisir en fonction d'un certain nombre de critères parmi un ensemble d'exercices prédéfini, ou générer des exercices en fonction d'un certain nombre de critères sur les exercices à générer, que nous appellerons *contraintes* de génération.

La question de décider comment fixer ces critères en fonction du profil de l'apprenant sera étudiée au chapitre suivant. Celle qui consiste à choisir, à partir de ces critères, des exercices dans une base d'exercices ne pose pas de difficulté particulière. Nous nous intéresserons donc dans ce chapitre uniquement à la question de générer des exercices à partir de contraintes.

Le chapitre A5 a permis d'expliquer les raisons qui nous ont amenés à choisir la génération semi-automatique de problèmes pour le projet AMBRE. Les travaux abordés dans ce chapitre relèvent du même objectif et ont pour ambition de généraliser les principes de GenAMBRE à des types d'exercices plus variés. Les objectifs étant les mêmes, le choix d'une génération semi-automatique est toujours pertinent. En effet, cela permet de laisser l'enseignant intervenir pour le choix des critères que devront respecter les exercices, tout en donnant au système informatique la possibilité de disposer d'une représentation de l'exercice lui permettant de construire les rétroactions de type aide ou diagnostic des réponses de l'apprenant.

1 Cadre des recherches

Les travaux décrits dans ce chapitre ont été effectués dans un premier temps dans le cadre de la thèse de Marie Lefevre (2006-2009), financée par une bourse ministérielle et co-encadrée par Stéphanie Jean-Daubias. Cette thèse a donné lieu à la proposition de l'approche GEPPETO ainsi que ses deux variantes GEPPETO-P et GEPPETO-S.

L'approche GEPPETO-P a ensuite été réutilisée et étendue dans le cadre du projet CLAIRE (Community Learning through Adaptive and Interactive multichannel Resources for Education), un Projet Investissements d'Avenir piloté par la société OpenClassrooms et financé *via* l'appel e-Education 1 de 2012 à 2014. Dans le cadre de ce financement j'ai recruté en contrat post-doctoral Baptiste Cablé, qui a développé des générateurs d'exercices d'évaluation pour l'outil auteur ASKER (Authoring tool for aSsessing Knowledge genErating exeRcises), en collaboration avec Marie Lefevre.

2 Questions de recherche : générer des exercices de types génériques ou spécifiques

Pour adapter des exercices aux besoins de chaque apprenant et conformément aux souhaits de l'enseignant, nous souhaitons permettre à l'enseignant d'exprimer des contraintes sur les exercices à générer. Pour cela, il est nécessaire de disposer d'un modèle des types d'exercices qu'il est possible de générer, afin de connaître le type de contraintes qu'il faut permettre à l'enseignant d'exprimer.

La **Figure 46** présente l'approche GEPPETO (GEneric models and processes to Personalize learners' PEdagogical activities according to Teaching Objectives). Cette approche permet d'acquérir des connaissances de plusieurs niveaux, auprès d'experts et auprès d'enseignants, pour générer des exercices.

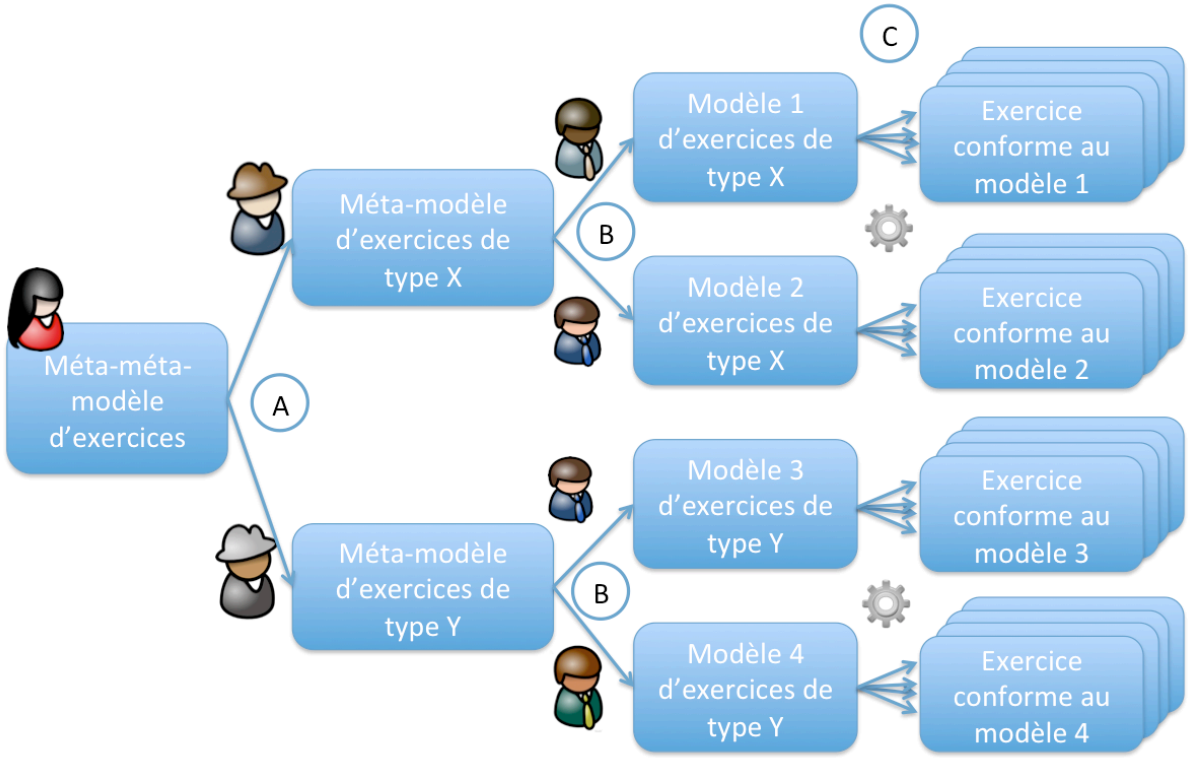

Figure 46. L'approche GEPPETO

Dans GEPPETO, un méta-méta-modèle d'exercices a été défini par Marie Lefevre. Il précise quelles sont les connaissances qu'un expert devra définir pour créer un méta-modèle d'exercices d'un type donné (cf. A sur la **Figure 46**), par exemple un méta-modèle des exercices de type QCM, ou un méta-modèle d'exercices de type AMBRE-add.

Ce méta-modèle d'exercices de type X ou Y permet ensuite à l'enseignant de préciser des contraintes définissant un modèle d'exercices (cf. B sur la **Figure 46**). Selon le type d'exercices, les contraintes ne seront pas les mêmes, c'est donc la raison d'être du méta-modèle d'exercices. En effet, on imagine aisément que les contraintes à fixer pour définir un modèle d'exercices pour AMBRE-add ne sont pas les mêmes que pour définir un modèle de QCM. L'enseignant pourra ainsi créer, à partir du méta-modèle des exercices de AMBRE-add, un modèle d'exercices précisant par exemple la classe de problèmes souhaitée, les

grandeurs numériques à choisir et le niveau de complication attendu, comme il le faisait avec AMBRE-enseignant. Un autre enseignant pourra utiliser le méta-modèle des exercices de type QCM pour définir un modèle d'exercices de type QCM portant par exemple sur un sujet donné et comportant N questions avec pour chacune M propositions avec exactement une réponse juste par question.

À partir de tels modèles d'exercices, des générateurs d'exercices peuvent construire plusieurs exercices conformes à ces modèles (cf. C sur la **Figure 46**). Les générateurs d'exercices permettant d'exploiter ces modèles dépendent du type d'exercices, et donc du méta-modèle d'exercices de type X ou Y.

On constate donc que l'approche GEPPETO requiert deux processus d'acquisition des connaissances :

- Une acquisition des connaissances de l'expert pour la création de méta-modèles d'exercices (cf. A sur la **Figure 46**). Cette acquisition est effectuée une seule fois pour chaque type d'exercices, et nécessite une interface fondée sur le méta-méta-modèle d'exercices.
- Une acquisition des connaissances de l'enseignant pour la création des modèles d'exercices (cf. B sur la **Figure 46**). Cette acquisition est effectuée de manière régulière pour la construction de modèles d'exercices variés, et nécessite également une interface qui, elle, est fondée sur le méta-modèle d'exercices d'un type donné (les contraintes que l'enseignant doit définir dépendant du type d'exercices), ce qui pose donc la question de la génération dynamique d'interfaces à partir de modèles.

Ces deux processus d'acquisition des connaissances seront différents selon que le type d'exercice est générique ou spécifique. Par générique, j'entends un type d'exercice indépendant du domaine et qui peut être appliqué à des domaines variés, comme des QCM ou des textes à trous, et pour lesquels on peut envisager de développer un générateur d'exercices. Par spécifique, j'entends des types d'exercices pour lesquels il existe des EIAH spécifiques, comme AMBRE-add ou Aplusix (Bouhineau & Nicaud, 2006) et où le type de l'exercice est lié au domaine. La question est alors de pouvoir disposer d'un modèle de l'EIAH concerné, pour pouvoir agir sur les exercices posés au sein de l'EIAH.

Ces deux questions de recherche différentes seront donc abordées de manière distincte dans les deux sections suivantes. Dans un cas comme dans l'autre, les processus d'acquisition des connaissances, que ce soit auprès de l'expert ou de l'enseignant, sont dirigés par les modèles et s'inscrivent donc bien dans l'approche présentée au chapitre 1 de ce mémoire. Pour faciliter la compréhension, l'instanciation de cette approche aux questions de recherche explicitées dans ce chapitre sera présentée en fin des sections 3 et 4.

3 Contribution : des générateurs d'exercices de types génériques pour un outil auteur d'exercices d'évaluation

Cette section présente tout d'abord GEPPETO-P, une déclinaison de l'approche GEPPETO appliquée à la génération d'exercices sur papier, avant de décrire ASKER, un outil auteur permettant la création d'exercices interactifs d'évaluation en ligne et s'appuyant sur des générateurs d'exercices issus de l'approche GEPPETO-P.

3.1 L'approche GEPPETO-P

Suite à une étude approfondie d'un corpus d'exercices constitué à partir de manuels scolaires et de feuilles d'exercices de provenances variées, nous avons identifié des invariants dans la structure des exercices posés sur papier, qui nous ont amenés à proposer une typologie d'exercices contenant quinze types d'exercices (Lefevre, Jean-Daubias, & Guin, 2009). Ces types d'exercices, tels que textes à trous, QCM, appariements, groupements, classements, peuvent s'appliquer à des disciplines et à des niveaux très variés.

Ces types génériques d'exercices qui s'appliquent à des domaines variés sont souvent destinés à évaluer la maîtrise par les élèves de connaissances ou compétences. Ils ne semblent en effet pas les plus adaptés à l'apprentissage d'une démarche de résolution de problèmes, pour lesquels il est nécessaire d'accompagner l'apprenant, et que nous traiterons dans la section suivante. Avec un exercice de type générique, l'apprenant doit effectuer la démarche de résolution de manière autonome, mais un diagnostic de la réponse finale est néanmoins intéressant dans un contexte d'auto-évaluation, d'évaluation ou de certification.

Certains types d'exercices pouvant être produits par le même générateur d'exercices (comme par exemple appariements, groupements et classements) sont regroupés en catégories. Ces types et catégories de types d'exercices constituent les méta-modèles d'exercices de l'approche GEPPETO (cf. **Figure 46**). Ces méta-modèles ainsi que le méta-méta-modèle sont indépendants du domaine pour lequel un exercice sera généré. Par exemple le méta-modèle "Identification des parties du texte" précise comment formuler la consigne, comment caractériser les différentes ressources de type texte, et comment décrire les actions à effectuer sur ces ressources pour générer les exercices.

À chacun de ces méta-modèles est associé un générateur d'exercices. Une interface associée au générateur et fondée sur le méta-modèle permet à l'enseignant de définir des contraintes sur les exercices à générer. C'est à ce moment-là, lors de la création du modèle d'exercices (par exemple un texte à trous portant sur la conjugaison des verbes du second groupe à l'imparfait), que l'application à un domaine et à un niveau d'études sera effectuée. Les exercices générés à partir du modèle d'exercices sont donc bien entendu dépendants du domaine.

Les méta-modèles étant tous conformes au méta-méta-modèle, l'ensemble des générateurs d'exercices partage une architecture commune (Lefevre, Jean-Daubias, et al., 2009; Lefevre, 2009). Cette architecture est similaire à l'architecture GenAMBRE, avec des moteurs de génération fondés sur chaque méta-modèle mais indépendants du domaine, et des connaissances spécifiques aux domaines exploitées par les moteurs de génération. Certaines connaissances indépendantes du domaine sont néanmoins utilisées, comme celles permettant d'écrire un énoncé en langue naturelle.

3.2 ASKER : un outil auteur d'exercices d'évaluation

L'approche GEPPETO-P a été réutilisée dans le cadre du projet CLAIRE pour concevoir un outil auteur permettant la création d'exercices destinés à évaluer la maîtrise par les apprenants de connaissances ou compétences. Dans ce projet, il s'agissait de pouvoir générer des exercices interactifs et non des exercices sur papier, mais les types d'exercices génériques de GEPPETO-P convenaient bien aux objectifs du projet et nous nous sommes donc appuyés sur l'architecture proposée pour les générateurs de GEPPETO-P.

Dans le contexte du projet CLAIRE, l'objectif était en effet de permettre à un auteur de contenu pédagogique en ligne (type MOOC) de proposer aux apprenants des exercices d'auto-évaluation. De tels exercices permettent à l'apprenant de tester en autonomie son niveau de maîtrise de ce qu'il a appris dans le cours. Les apprenants peuvent échouer lors des premières tentatives de réponse aux exercices si les connaissances ne sont pas maîtrisées. Il est donc possible qu'un apprenant soit amené à répondre plusieurs fois à un même exercice avant d'obtenir un succès. Afin que l'apprenant ne se laisse pas influencer par ces précédentes résolutions, il est nécessaire que l'exercice d'auto-évaluation soit différent d'une fois à l'autre, tout en évaluant la même connaissance. Toutefois, il ne semble pas raisonnable de demander à l'auteur de rédiger de nombreuses versions différentes de l'énoncé. C'est pourquoi nous avons choisi de recourir à des générateurs d'exercices que l'auteur peut facilement utiliser quel que soit le domaine.

Pour répondre aux besoins d'auto-évaluation identifiés dans le cadre du projet CLAIRE, les propriétés attendues de l'outil auteur étaient les suivantes :

- l'exercice est différent d'une fois sur l'autre ;
- l'auteur est maître du contenu de l'exercice et a la garantie qu'il correspond à ses attentes en matière de contenu pédagogique ;
- les générateurs d'exercices sont utilisables dans un grand nombre de domaines et de niveaux scolaires ;
- le diagnostic de la réponse est fait automatiquement et en temps réel ;
- la construction des exercices avec les générateurs est un gain de temps pour l'auteur par rapport à une création exercice par exercice ;
- la création d'exercices ne demande pas de compétences techniques particulières.

Par rapport aux propriétés attendues, l'approche GEPPETO-P les satisfaisait toutes en dehors de la capacité de diagnostic automatique des réponses. En effet, dans GEPPETO-P une correction était fournie à l'enseignant, mais ce n'est pas suffisant pour effectuer un diagnostic.

Nous avons donc choisi pour l'outil auteur un fonctionnement inspiré de l'approche GEPPETO-P : après avoir choisi un type d'exercice (QCM, appariement, texte à trous…), l'auteur peut créer un modèle d'exercice qui décrit le contenu et la forme de l'exercice, mais sans forcément le contraindre totalement. En exploitant ce modèle, un générateur d'exercices est capable de mettre à disposition de l'apprenant un grand nombre d'exercices différents évaluant la même compétence. Chaque instance d'exercice ainsi générée sera interactive : l'apprenant y répondra en ligne et obtiendra un diagnostic de sa réponse.

Les types d'exercices identifiés par l'approche GEPPETO-P et que nous avons retenus pour le projet CLAIRE sont les suivants :

- identification de parties du texte (inclut le texte à trous),
- organisation d'éléments (ordonnancement, groupement, appariement),
- annotation d'illustration,
- questionnaire à choix multiple (QCM),
- questionnaire à réponses ouvertes et courtes (QROC).

La **Figure 47** présente l'architecture de l'outil auteur ASKER (Authoring tool for aSsessing Knowledge genErating exeRcises). Le bloc du haut est constitué des différents niveaux de représentation des exercices issus de l'approche GEPPETO-P : les méta-modèles de types

d'exercices, les modèles d'exercices, et les instances d'exercices. Dans le bloc central se trouvent les trois mécanismes manipulant ces représentations des exercices. Le bloc inférieur contient les ressources et connaissances utilisées dans le processus de création des exercices.

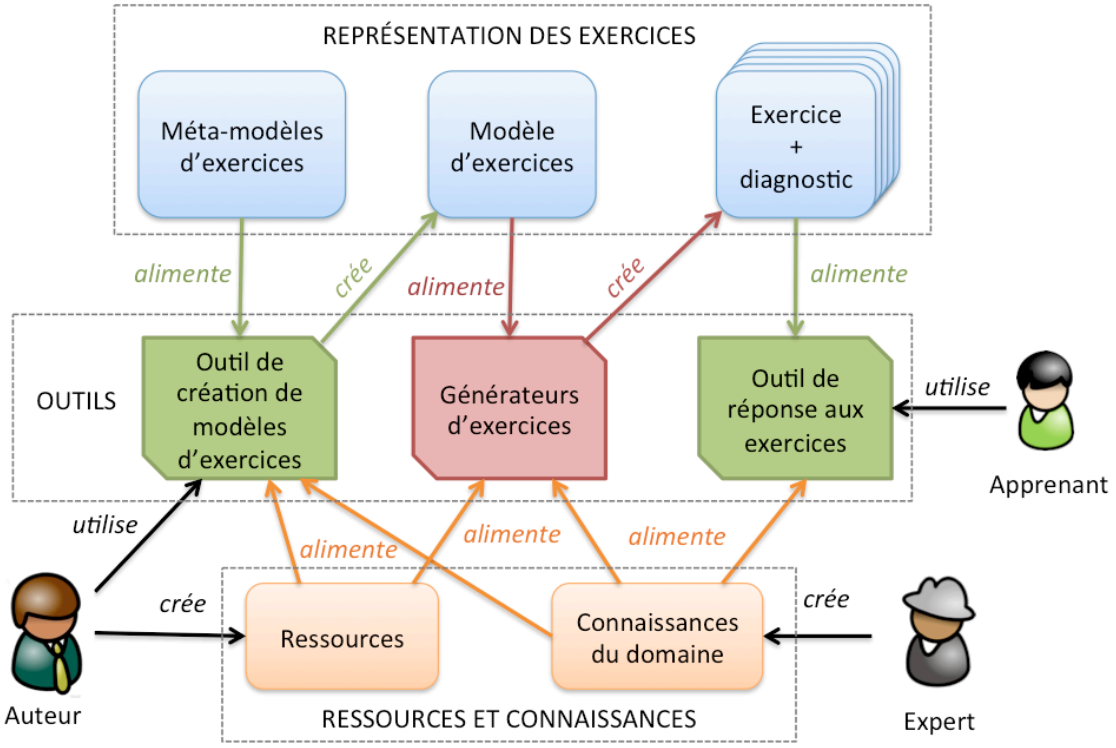

Figure 47. Architecture de l'outil de création d'exercices d'évaluation ASKER

Les *ressources* sont les éléments de base qui sont utilisés pour construire les exercices. Il s'agit, par exemple, de textes, d'images ou de séquences. Chaque ressource dispose de métadonnées la caractérisant pour faciliter les recherches (thème, niveau, etc.) ainsi que de métadonnées enrichissant la ressource pour définir des exercices, comme la légende d'une image ou des annotations sur des zones de l'image.

Les *connaissances du domaine* sont des connaissances relatives à une matière et indépendantes du type d'exercice. Il s'agit par exemple de la liste des mots-clés du langage C, d'une méthode pour détecter le genre d'un nom, de la valeur d'une constante, etc. Ces connaissances peuvent être définies par un expert du domaine.

L'auteur choisit un type d'exercices T. Un expert ayant décrit le méta-modèle MM_T des exercices de type T, l'auteur crée un modèle d'exercices M_T conforme à MM_T à l'aide d'un outil dédié qui s'appuie sur les connaissances sur le type d'exercice T contenues dans MM_T. Cet outil de création de modèles d'exercices facilite la création du modèle, notamment en s'appuyant sur les connaissances du domaine. Il permet également de générer quelques exercices E_T instances de M_T, pour tester si le modèle donne bien lieu aux exercices attendus. Cet outil facilite l'accès et le choix des ressources. Il permet même à l'auteur d'en créer de nouvelles.

Le générateur d'exercices de type T associé à MM_T reçoit donc en entrée un modèle d'exercices M_T qu'il instancie pour donner lieu à un exercice E_T en sortie. L'exercice est conforme au modèle M_T et donc aux choix de l'auteur qui l'a créé. Le générateur ne nécessite aucune intervention humaine. Il dispose de toutes les informations nécessaires dans le modèle d'exercices M_T et fait appel, si nécessaire, aux ressources et aux connaissances du domaine. Le générateur est utilisé à chaque fois que l'on veut obtenir une nouvelle instance d'exercice E_T (qui contient le diagnostic) à partir du modèle M_T.

L'exercice est alors présenté à l'apprenant *via* un outil de résolution d'exercices. Ce dernier met en forme l'exercice, recueille la réponse de l'apprenant et en fournit le diagnostic.

Diagnostic des réponses de l'apprenant

L'approche GEPPETO-P est prévue pour générer des exercices papier, avec une solution fournie à l'enseignant. Dans le cas d'exercices où plusieurs solutions (ou plusieurs syntaxes) sont justes, l'enseignant fait l'effort d'adaptation de la solution pour estimer si la réponse de l'apprenant est juste. Dans ASKER, nous générons des exercices informatisés où l'apprenant répond en ligne et doit obtenir un diagnostic automatique de sa réponse. Il ne suffit pas alors de produire une correction juste : il faut pouvoir reconnaître toutes les bonnes réponses.

Dans le cas d'un exercice où la réponse est unique, il suffit que cette solution soit définie dans le modèle de l'exercice. Le diagnostic consiste alors à analyser la réponse de l'apprenant pour voir si elle est identique à l'unique bonne réponse ou s'il s'agit d'une réponse provoquant un feedback particulier.

Dans le cas où plusieurs bonnes réponses sont possibles, et ce parfois en très grand nombre, nous disposons de deux moyens pour surmonter cette difficulté. Les problèmes simples d'ordre syntaxique sont traités en premier lieu par le système d'acquisition des réponses, qui gère par exemple l'équivalence de notation entre 3.14 et 3,14 ou entre 3E4 et 3.10^4. Pour les problèmes d'ordre sémantique, la solution générée avec l'exercice est en réalité un modèle de solution qui rend plusieurs réponses acceptables. Ces informations sur les réponses acceptables proviennent soit du modèle d'exercice où l'auteur a précisé les variations et tolérances autour de la réponse, soit des ressources employées où les différentes solutions associées sont précisées.

Connaissances du domaine

Les connaissances du domaine sont des connaissances directement liées au domaine et indépendantes du type d'exercices dans lequel elles sont utilisées. Elles peuvent prendre différentes formes selon le type de connaissance : une valeur constante, une formule de calcul, une règle, une énumération.

Ces connaissances sont utilisées à deux niveaux :

D'une part, elles simplifient la création des modèles d'exercices car il est possible d'utiliser ces connaissances plutôt que de les redéfinir pour chaque exercice du même domaine. Dans l'exemple d'un exercice sur le texte d'un programme en C, elles permettent de demander : « crée des trous sur les mots-clés du C » plutôt que : « crée des trous sur les mots for, while, if, then, else, case, end ». En plus de simplifier la création du modèle d'exercices, cela limite les risques d'erreur de la part de l'auteur car en saisissant l'énumération, il peut en oublier ou faire plus facilement des erreurs.

D'autre part, les connaissances du domaine peuvent être utilisées lors du diagnostic de la réponse de l'apprenant.

Les générateurs d'exercices étant indépendants du domaine, les connaissances du domaine n'existent pas au départ. Leur acquisition représente donc un enjeu important. La méthode actuellement utilisée consiste à confier la tâche à un expert qui a la connaissance du domaine ainsi que la maîtrise technique nécessaire pour créer des connaissances, cet expert pouvant bien sûr être aussi un auteur. Cette méthode n'est pas entièrement satisfaisante : la tâche étant relativement fastidieuse, on peut aisément envisager que peu de personnes accepteront de s'y consacrer. L'utilisateur principal d'ASKER étant l'auteur, il serait intéressant que ce soit lui qui construise les connaissances du domaine. Le plus convivial pour lui serait qu'il crée les connaissances du domaine requises au fur et à mesure qu'il en a besoin pendant la définition des modèles d'exercices. Le système pourrait l'assister dans cette tâche en lui proposant une généralisation des informations qu'il renseigne pour créer ses modèles d'exercices. Toujours dans l'exemple des programmes en C, la liste des mots-clés du langage peut être construite automatiquement à partir de textes de programmes en C qui ont déjà été annotés par un auteur. Cette question de recherche fait l'objet de la thèse de Bryan Kong Win Chang qui a commencé en octobre 2014.

3.3 Bilan

L'approche GEPPETO-P, initialement prévue pour la création d'exercices sur papier, a servi de fondement à la conception et au développement d'ASKER, un outil auteur fondé sur des générateurs d'exercices de types génériques.

Si dans le projet CLAIRE cet outil avait pour objectif la création d'exercices d'auto-évaluation par un auteur de MOOC, les exercices créés peuvent également être utilisés dans un cadre plus classique d'évaluation des connaissances. ASKER est ainsi utilisé depuis la rentrée 2014 dans certaines UE d'informatique de la licence STS de l'Université Lyon 1, à la fois pour proposer des exercices d'auto-évaluation aux étudiants, mais aussi pour proposer des versions variées des feuilles de test que les étudiants passent au début de chaque séance de travaux dirigés.

Revenons à présent sur l'approche GEPPETO décrite sur la **Figure 46**. J'y avais mentionné deux processus d'acquisition de connaissances : l'un auprès de l'expert (A) pour la définition de méta-modèles d'exercices d'un type donné, et l'autre auprès de l'enseignant (B) pour la définition de modèles d'exercices. Dans GEPPETO-P comme dans ASKER, il y a bien un processus d'acquisition des connaissances auprès de l'enseignant pour la définition de modèles d'exercices, et ce grâce à des interfaces dépendant des types d'exercices. Par contre, pour définir un nouveau méta-modèle d'exercices correspondant à un nouveau type d'exercices, la définition du méta-modèle et surtout le développement du générateur d'exercices correspondant doivent être effectués par un expert ayant de bonnes compétences en informatique. On peut donc difficilement parler de processus d'acquisition des connaissances à ce niveau-là.

Il existe cependant dans ASKER un processus d'acquisition des connaissances auprès de l'expert, qui concerne les connaissances du domaine. Comme décrit ci-dessus, ce processus n'est pas pour l'instant complètement satisfaisant et un processus d'acquisition de ces connaissances auprès de l'enseignant auteur est à l'étude dans le cadre d'une thèse qui commence.

En ce qui concerne l'évaluation des contributions présentées dans cette section, les générateurs d'ASKER ainsi que l'outil auteur sont fonctionnels. L'outil auteur permet de définir des modèles d'exercices et les générateurs permettent de générer des exercices satisfaisant les contraintes définies par les modèles. Pour ce qui est d'une évaluation de l'outil auteur auprès d'utilisateurs n'ayant pas participé à sa conception, un premier test a été réalisé en juillet 2014 dans le cadre d'un atelier de l'école thématique CNRS "MOOC et EIAH". L'outil auteur a reçu un très bon accueil des participants, même si cela a été pour nous l'occasion d'identifier des problèmes d'utilisabilité des interfaces. À partir de la rentrée 2014, son utilisation dans le cadre de certaines UE d'informatique en licence à l'UCBL sera l'occasion de poursuivre son évaluation. Une utilisation par l'auteur du MOOC FOVEA permettra également d'avoir un retour d'un enseignant non informaticien. L'intégration des générateurs d'exercices d'ASKER dans le cadre de la plateforme CLAIRE utilisée par le site OpenClassrooms (anciennement le Site du Zéro) devrait enfin permettre d'avoir des retours d'usages d'une communauté importante d'auteurs.

La **Figure 48** replace les travaux décrits dans cette section par rapport à l'approche présentée au chapitre 1. Avec l'outil auteur ASKER, un auteur élicite les connaissances nécessaires à la définition de modèles d'exercices. L'interface d'ASKER s'appuie sur des méta-modèles d'exercices de types génériques tels que travail sur texte, organisation d'objets, etc. Ces méta-modèles d'exercices sont définis par des experts et spécifient des générateurs d'exercices qui exploitent les modèles d'exercices définis par l'auteur pour générer des exercices à destination des apprenants. Le travail d'élicitation de connaissances par l'auteur est assisté par des connaissances du domaine dont l'acquisition s'effectuera au fur et à mesure de l'élicitation des modèles d'exercices.

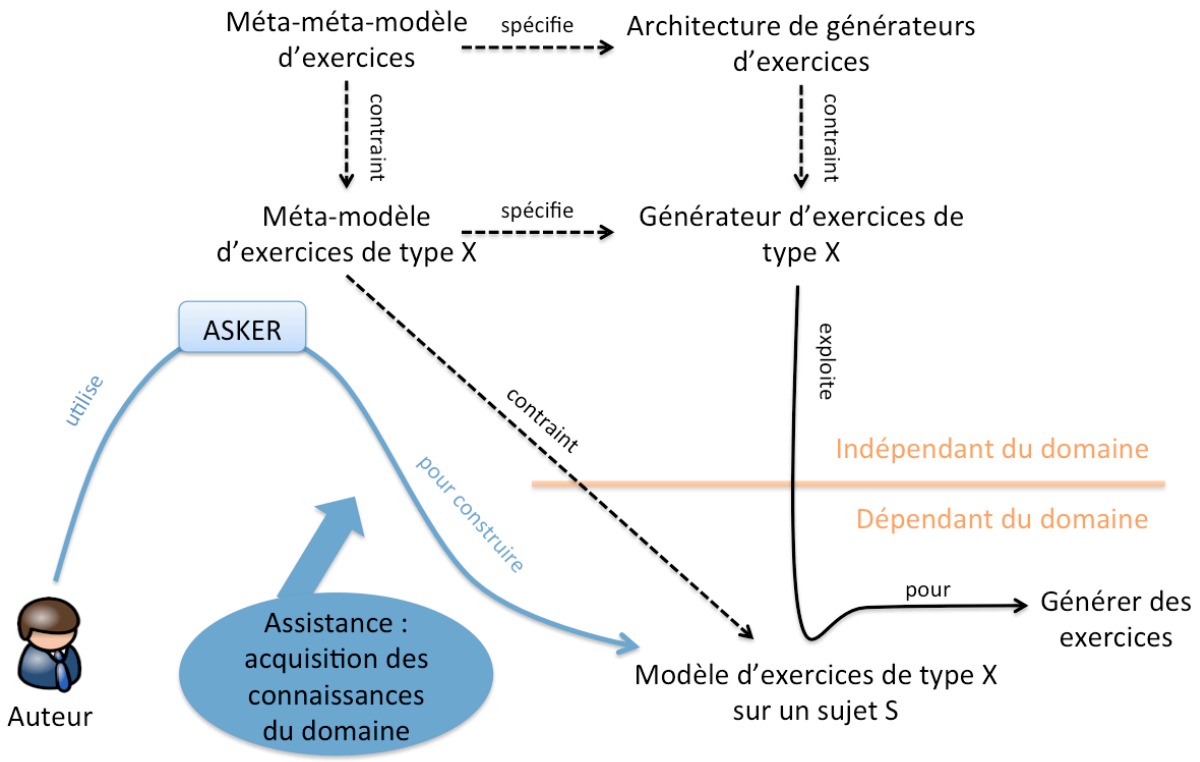

Figure 48. Instanciation de l'approche présentée au chapitre 1 à la question de la génération d'exercices.

4 Contribution : agir sur des générateurs d'exercices intégrés à des EIAH existants

Si l'on souhaite générer des exercices de types plus spécifiques que ceux qui ont fait l'objet de la section précédente, comme par exemple ceux de AMBRE-add, il faudrait un nouveau générateur d'exercices pour chaque type d'exercices. Nous avons donc choisi d'étudier comment utiliser des générateurs d'exercices intégrés à des EIAH existants, ou des bases d'exercices volumineuses intégrées à des EIAH existants, ou des ressources éducatives libres paramétrables, afin de pouvoir personnaliser les exercices proposés à l'apprenant.

4.1 L'approche GEPPETO-S et son implémentation dans le logiciel Adapte

L'approche GEPPETO-S est une déclinaison de l'approche GEPPETO destinée à paramétrer des EIAH existants. Elle a été mise en œuvre dans le logiciel Adapte (Lefevre, Guin, & Jean-Daubias, 2008).

Dans le cadre de sa thèse, Marie Lefevre a constitué un corpus de trente logiciels éducatifs de types variés (tuteurs intelligents, micromondes, simulateurs, exerciseurs), issus de la recherche, du développement libre ou du commerce, destinés à des publics variés et portant sur des domaines divers. Une étude approfondie des logiciels de ce corpus lui a permis d'identifier cinq aspects sur lesquels peut porter la personnalisation d'un EIAH : les activités proposées, la séquence des activités, les fonctionnalités disponibles, l'interface du logiciel et les rétroactions proposées aux apprenants. Ces EIAH peuvent être configurés avant d'être utilisés par un apprenant, *via* un fichier de configuration ou *via* une interface de configuration. Les activités peuvent être générées automatiquement ou choisies dans une liste prédéfinie dépendant éventuellement des contraintes exprimées par le paramétrage.

La question de recherche qui nous intéresse ici est la possibilité d'agir de l'extérieur sur le paramétrage d'un EIAH, et en particulier sur le choix ou la génération des exercices proposés à l'apprenant. L'étude de ce corpus d'EIAH a permis d'élaborer un formalisme unique pour décrire les paramètres intervenant dans la configuration d'un EIAH. L'approche GEPPETO-S est donc fondée sur un méta-modèle (Lefevre, Mille, Jean-Daubias, & Guin, 2009) destiné à guider un expert dans l'élicitation d'un modèle de chaque EIAH sur lequel on cherche à agir. Sur la **Figure 46**, un modèle d'EIAH étant considéré comme un méta-modèle d'exercices, le méta-modèle d'EIAH est représenté comme un méta-méta-modèle d'exercices.

Afin de pouvoir personnaliser un EIAH donné (que nous appellerons X), le logiciel Adapte qui implémente l'approche GEPPETO-S a donc besoin de disposer d'un modèle de cet EIAH. C'est un expert connaissant bien l'EIAH X qui doit définir ce modèle en élicitant d'une part des connaissances pédagogiques sur les exercices de l'EIAH X et d'autre part des connaissances plus techniques sur les fichiers permettant de configurer l'EIAH X.

Pour effectuer cette acquisition de connaissances auprès d'experts, le logiciel Adapte s'appuie sur le méta-modèle d'EIAH pour proposer une interface à l'expert. L'expert utilise cette interface pour définir le type d'exercices proposés dans l'EIAH X, ainsi que la manière dont il est possible de choisir ou de générer les exercices de cet EIAH. Il peut également décrire comment paramétrer les séquences d'exercices ainsi que les fonctionnalités ou l'interface de l'EIAH X. Pour ce qui est des connaissances techniques, l'expert décrit comment agir concrètement sur le système : localisation du système, chemin d'accès au

générateur d'exercices ou à la base d'exercices, place et contenu des fichiers de configuration et règles permettant de remplir ces fichiers.

À partir du modèle ainsi créé pour l'EIAH X, le logiciel Adapte génère dynamiquement une interface spécifique qui permettra à chaque enseignant utilisateur de définir des contraintes sur les exercices qu'il souhaite que ses élèves effectuent avec l'EIAH X. Il pourra faire de même à partir du modèle d'un EIAH Y. Ainsi, le logiciel Adapte permet de fournir aux enseignant un outil unique permettant d'agir sur des logiciels pédagogiques variés. L'enseignant n'a donc plus besoin de maîtriser les différentes interfaces de personnalisation ni le contenu des fichiers de configuration pour adapter des EIAH divers, un seul outil lui permettant d'effectuer les choix adéquats pour chaque environnement.

À partir des contraintes fixées par l'enseignant et des propriétés pédagogiques et techniques du modèle de l'EIAH X, Adapte construit une séquence d'exercices à proposer aux apprenants et crée les fichiers de configurations propres à l'EIAH X ou une liste d'instructions à suivre pour utiliser l'interface de configuration de l'EIAH X.

On retrouve donc bien dans GEPPETO-S deux processus d'acquisition des connaissances (cf. **Figure 46**) :

- un processus d'acquisition des connaissances auprès de l'expert (A), pour éliciter un modèle des types d'exercices que l'on peut effectuer avec un EIAH X. Ce processus s'appuie sur le méta-méta-modèle afin de créer un méta-modèle des exercices de types EIAH-X.
- puis un processus d'acquisition des connaissances auprès de l'enseignant (B), pour éliciter des modèles d'exercices. Ce processus s'appuie sur les méta-modèles d'exercices que sont les modèles d'EIAH.

Le premier processus d'acquisition des connaissances a lieu une seule fois pour chaque EIAH existant. Le second correspond à une utilisation régulière du logiciel Adapte par les enseignants, qui utiliseront ce logiciel pour créer des modèles d'exercices avec plusieurs EIAH existants différents.

4.2 Illustration sur l'EIAH AMBRE-add

Afin d'illustrer en quoi consiste un modèle d'EIAH dans l'approche GEPPETO-S, cette section présente le modèle défini pour l'EIAH AMBRE-add en utilisant le logiciel Adapte. Nous avons souhaité définir ce modèle afin de permettre à un enseignant d'utiliser Adapte pour paramétrer AMBRE-add pour ses élèves. Nous souhaitions reproduire avec Adapte une partie des fonctionnalités du module AMBRE-enseignant (cf. Chapitre A5), mais surtout mettre en place une stratégie de personnalisation pour AMBRE-add qui sera présentée au chapitre suivant.

Nous avons effectué ce travail dans des conditions expérimentales, où j'ai utilisé l'outil Adapte (que je n'avais jamais manipulé auparavant) en tant qu'experte de l'EIAH AMBRE-add, pour définir un modèle de cet EIAH (phase A sur la **Figure 46**).

Le modèle d'un EIAH dans l'approche GEPPETO-S est constitué de quatre parties : les propriétés pédagogiques, les règles pédagogiques, les propriétés techniques et les règles techniques.

Les propriétés pédagogiques sont les caractéristiques des activités pédagogiques proposées dans l'EIAH, et donc ici pour AMBRE-add les caractéristiques des problèmes à résoudre. J'ai

donc défini les éléments des problèmes sur lesquels on peut agir avec le générateur GenAMBRE (cf. Chapitre A5) : la classe du problème, la présence de la retenue dans le calcul, les valeurs des nombres, l'écart entre les nombres, la complexité du vocabulaire de l'énoncé, de la situation décrite dans l'énoncé, le nombre de phrases inutiles, le niveau des phrases inutiles, etc. J'ai également défini des propriétés n'existant pas dans GenAMBRE mais permettant de regrouper les propriétés précédentes, en préparation de la définition d'une stratégie de personnalisation : la difficulté d'une classe de problèmes, la difficulté du calcul, et le niveau de la complication de l'énoncé. L'ensemble de ces propriétés a été séparé en trois catégories : celles liées à la structure du problème (c'est-à-dire sa classe), celles liées au calcul, et celles liées à la complication de l'énoncé.

Les règles pédagogiques permettent de définir des relations entre les valeurs des propriétés pédagogiques. Ainsi, j'ai défini : la difficulté associée à chaque classe de problème ; la difficulté du calcul en fonction des valeurs de trois propriétés pédagogiques que sont la retenue, les valeurs des nombres et l'écart entre les nombres ; et la difficulté de la complication de l'énoncé en fonction des valeurs de sept propriétés pédagogiques comme la complexité du vocabulaire, le nombre de phrases inutiles, etc.

Pour **les propriétés techniques**, j'ai défini l'emplacement de l'exécutable d'AMBRE-add, de l'exécutable du générateur GenAMBRE, du fichier définissant les contraintes de génération qu'il prend en entrée, et des fichiers de séquences à personnaliser.

Les règles techniques m'ont permis de préciser comment, à partir de propriétés pédagogiques fixées dans Adapte, modifier le fichier décrivant les contraintes de génération fournies à GenAMBRE, et comment, à partir des séquences ainsi construites par GenAMBRE, les affecter à chaque apprenant.

4.3 Bilan

L'approche GEPPETO-S a été implémentée dans le logiciel Adapte. Marie Lefevre a utilisé le méta-modèle de GEPPETO-S pour définir les modèles des EIAH constituant son corpus initial. Adapte a également permis de définir les modèles de six EIAH qui ne faisaient pas partie du corpus initial. La définition du modèle d'un EIAH avec Adapte par un expert de cet EIAH peut nécessiter de un à trois jours selon la complexité de l'EIAH à décrire.

Ces modèles d'EIAH sont exploitables par Adapte qui permet aux enseignants de paramétrer des EIAH de différents types (micromonde, tuteur intelligent, simulateur, exerciseur, hypermedia). Il faut néanmoins rappeler que le cadre d'application du méta-modèle et donc de l'approche GEPPETO-S se limite aux EIAH proposant des activités *individuelles* d'apprentissage. La raison de cette limite est que le corpus initial ne comprenait pas de logiciel permettant un apprentissage collectif ou collaboratif. Toutefois, le méta-modèle peut être complété pour considérer de nouveaux types d'EIAH ou de nouveaux types de paramètres afin de traiter d'éventuelles facettes non couvertes.

La **Figure 49** présente comment GEPPETO-S s'inscrit dans l'approche scientifique décrite au Chapitre 1. En effet, l'expert d'un EIAH X définit un modèle de cet EIAH en s'appuyant sur un méta-modèle, *via* le logiciel Adapte. Un moteur exploite le modèle de l'EIAH X ainsi défini pour générer dynamiquement *via* Adapte une interface permettant à l'enseignant de définir un modèle d'exercices avec l'EIAH X. Un moteur sous-jacent au logiciel Adapte exploite de tels modèles d'exercices pour générer ou choisir des séquences d'exercices avec l'EIAH X.

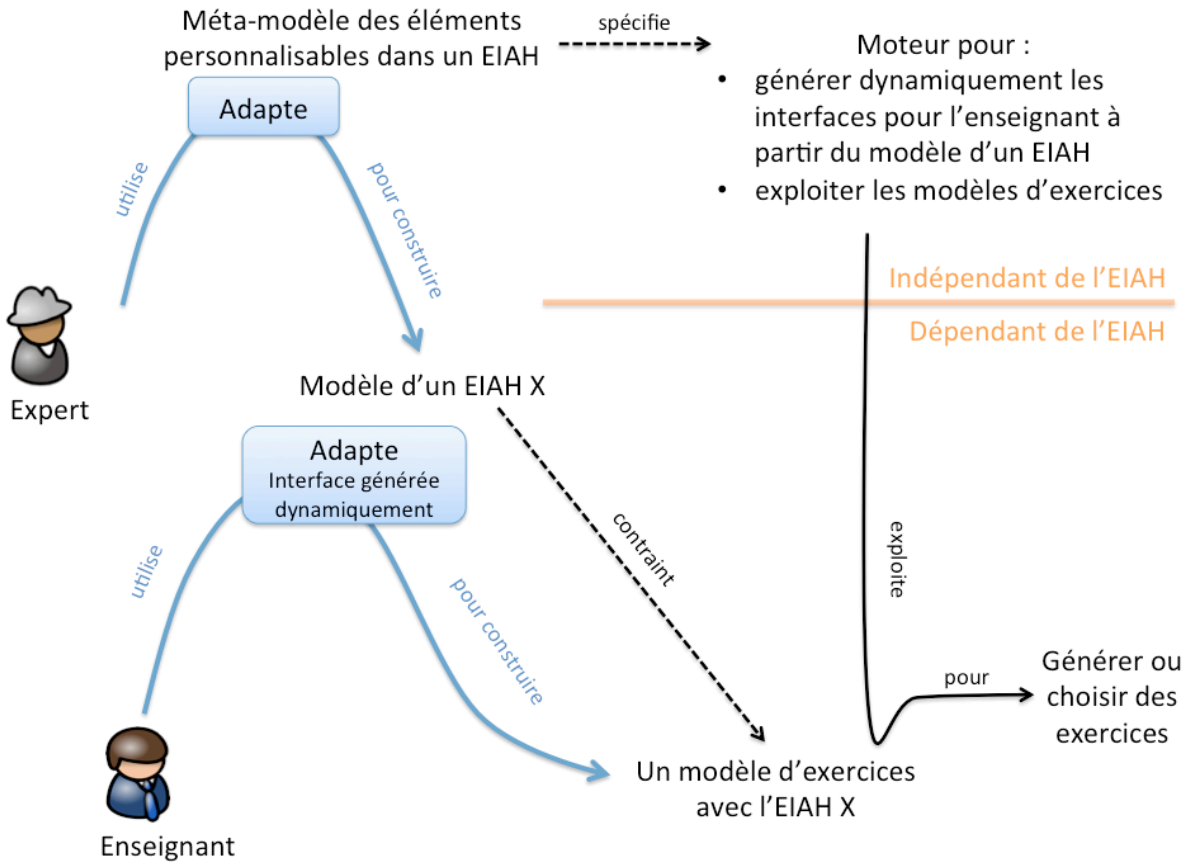

Figure 49. Instanciation de l'approche présentée au chapitre 1 à la question du pilotage d'EIAH existants pour générer ou choisir des activités.

Conclusion et perspectives

Ce chapitre a permis de présenter nos propositions pour la génération semi-automatique d'exercices, qui s'appuient sur l'approche GEPPETO. Cette approche est fondée sur deux processus d'acquisition des connaissances : l'un auprès de l'expert pour acquérir des connaissances sur des types d'exercices, ce qui donne lieu à la définition de méta-modèles d'exercices, et l'autre auprès de l'enseignant pour acquérir des connaissances sur des modèles d'exercices, qui permettent le choix ou la génération d'exercices.

Cette approche a été déclinée de manière à pouvoir traiter à la fois des types d'exercices génériques, souvent utilisés pour l'évaluation des compétences de l'apprenant, et des types d'exercices spécifiques, pour lesquels nous nous appuyons sur des EIAH existants que l'on permet à l'enseignant de configurer.

Une première perspective concernant les générateurs d'exercices de l'outil auteur ASKER consiste à les faire évoluer dans le cadre du projet "Cartographie des Savoirs". En effet, le référentiel des compétences du domaine représenté par l'ontologie Ontoprax fait intervenir des variables concernant les types de tâches, qui pourraient être exploitées par les générateurs d'exercices. Il s'agit en effet d'une représentation des connaissances du domaine qui permettrait d'intégrer une dimension sémantique nettement plus importante aux générateurs actuels.

Une deuxième perspective de recherche porte sur l'assistance qu'il est possible d'apporter à l'enseignant qui élicite des modèles d'exercices. Nous souhaitons tout d'abord assister l'enseignant dans la définition de connaissances du domaine lorsque l'on ne dispose pas d'un référentiel. Lorsque l'on utilise un EIAH existant, les connaissances du domaine y sont en général déjà représentées, mais ce n'est pas le cas pour les générateurs d'exercices de types génériques de l'outil auteur ASKER. Une thèse commence donc sur la question de l'utilisation des traces d'activités de l'enseignant pour l'acquisition interactive des connaissances du domaine. Le Raisonnement à Partir de l'Expérience Tracée développé au sein de l'équipe SILEX (Cordier et al., 2013) pourrait également permettre au système d'assister le processus de création des modèles par l'enseignant, en lui proposant des améliorations afin que ces modèles correspondent mieux à ses besoins. En effet, dans le cas de générateurs d'exercices de types génériques ou de générateurs intégrés à des EIAH existants, l'enseignant peut vérifier les exercices générés à partir d'un modèle d'exercices. Si on lui donne la possibilité de réfuter des instances d'exercices, le système peut s'appuyer sur ce retour de l'enseignant pour ajuster le modèle d'exercices (Cordier, Lefevre, Jean-Daubias, & Guin, 2010).

Synthèse des contributions

Question de recherche : Générer des exercices à partir de contraintes fixées par l'enseignant

- **Contribution théorique** : l'approche GEPPETO, avec ses deux déclinaisons : GEPPETO-P pour les types génériques d'exercices et GEPPETO-S pour piloter des EIAH
- **Système développé** :
  - ASKER, outil auteur de génération d'exercices
  - Adapte, qui permet de définir des modèles d'exercices pour des EIAH existants
- **Déploiement** : ASKER est utilisé en première année de licence à l'Université Lyon 1 et devrait également être intégré à la plateforme CLAIRE (OpenClassrooms)
- **Co-auteurs** : Marie Lefevre, Stéphanie Jean-Daubias, Baptiste Cablé

Pour en savoir plus

• Sur la catégorisation d'exercices de type générique et l'approche GEPPETO-P :

Lefevre, M. (2009). *Processus unifié pour la personnalisation des activités pédagogiques : méta-modèle, modèles et outils*. Université Claude Bernard Lyon 1.

Lefevre, M., Jean-Daubias, S., & Guin, N. (2009). Generation of pencil and paper exercises to personalize learners' work sequences: typology of exercises and meta-architecture for generators. In *E-Learn 2009 (World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education)*. Vancouver, Canada.

• Sur l'outil auteur ASKER :

Cablé, B., Guin, N., & Lefevre, M. (2013). Un outil auteur pour une génération semi-automatique d'exercices d'auto-évaluation. In *EIAH 2013* (pp. 155–166). Toulouse, France.

• Sur la représentation et l'acquisition des connaissances dans ASKER :

Cablé, B., Lefevre, M., & Guin, N. (2013). Comment représenter les connaissances dans un générateur semi-automatique d'exercices d'auto-évaluation ? In *Journée EIAH&IA 2013*. Toulouse, France.

• Sur GEPPETO-S :

Lefevre, M. (2009). *Processus unifié pour la personnalisation des activités pédagogiques : méta-modèle, modèles et outils*. Université Claude Bernard Lyon 1.

Lefevre, M., Mille, A., Jean-Daubias, S., & Guin, N. (2009). A Meta-Model to Acquire Relevant Knowledge for Interactive Learning Environments Personalization. In *Adaptive 2009*. Athènes, Grèce.

• Sur le logiciel Adapte :

Lefevre, M. (2009). *Processus unifié pour la personnalisation des activités pédagogiques : méta-modèle, modèles et outils*. Université Claude Bernard Lyon 1.

Lefevre, M., Cordier, A., Jean-Daubias, S., & Guin, N. (2009). A Teacher-dedicated Tool Supporting Personalization of Activities. In AACE (Ed.), *ED-MEDIA 2009 - World Conference on Educational Multimedia, Hypermedia & Telecommunications* (pp. 1136–1141). Honolulu, Hawaii.

• Sur son application à AMBRE-add :

Guin, N., & Lefevre, M. (2013). From a customizable ITS to an adaptive ITS. In *AIED 2013* (pp. 141–150). Memphis, USA.

Publications liées

Conférences internationales

Cablé, B., Guin, N., & Lefevre, M. (2013a). An authoring tool for semi-automatic generation of self-assessment exercises. In *AIED 2013* (pp. 679–682). Memphis, USA.

Guin, N., & Lefevre, M. (2013). From a customizable ITS to an adaptive ITS. In *AIED 2013* (pp. 141–150). Memphis, USA.

Jean-Daubias, S., Lefevre, M., & Guin, N. (2009). Generation of exercises within the PERLEA project. In *2nd Workshop on Question Generation, International Conférence on Artificial Intelligence in Education (AIED 2009)* (pp. 38–42). Brigton, Grande-Bretagne: AIED 2009 Workshops Proceedings 1.

Lefevre, M., Guin, N., & Jean-Daubias, S. (2012). A Generic Approach for Assisting Teachers During Personalization of Learners' Activities. In *International Workshop on Personalization Approaches in Learning Environments (PALE) - Conference User Modeling, Adaptation, and Personalization* (pp. 35–40).

Lefevre, M., Jean-Daubias, S., & Guin, N. (2009). Supporting Acquisition of Knowledge to Personalize Interactive Learning Environments through a Meta-Model. In *17th International Conference on Computers in Education (ICCE 2009)*. Hong Kong.

Lefevre, M., Mille, A., Jean-Daubias, S., & Guin, N. (2009). A Meta-Model to Acquire Relevant Knowledge for Interactive Learning Environments Personalization. In *Adaptive 2009*. Athènes, Grèce.

Conférences nationales

Cablé, B., Guin, N., & Lefevre, M. (2013b). Un outil auteur pour une génération semi-automatique d'exercices d'auto-évaluation. In *EIAH 2013* (pp. 155–166). Toulouse, France.

Cablé, B., Lefevre, M., & Guin, N. (2013). Comment représenter les connaissances dans un générateur semi-automatique d'exercices d'auto-évaluation ? In *Journée EIAH&IA 2013*. Toulouse, France.

CHAPITRE B3
PILOTER LA GÉNÉRATION D'EXERCICES À PARTIR DES PROFILS

Résumé

Ce chapitre étudie comment permettre à un enseignant de définir la manière dont les activités doivent être affectées de manière personnalisée aux apprenants, en fonction du contenu de leurs profils.

Il présente le modèle PERSUA2, qui permet à l'enseignant de définir une stratégie de personnalisation qu'un moteur exploite pour affecter des activités personnalisées aux apprenants. Il montre également comment nous avons utilisé ce modèle pour rendre l'EIAH AMBRE-add adaptatif.

Deux évolutions du modèle PERSUA2 sont ensuite proposées : l'une adaptée au contexte des MOOCs, et l'autre destinée à un EIAH fondé sur un référentiel des compétences.

INTRODUCTION 172

1 CADRE DES RECHERCHES 172

2 QUESTIONS DE RECHERCHE : COMMENT PERMETTRE À L'ENSEIGNANT D'ÉLICITER UNE STRATÉGIE DE PERSONNALISATION ? COMMENT APPLIQUER UNE TELLE STRATÉGIE POUR PROPOSER DES ACTIVITÉS PERSONNALISÉES ? 173

3 CONTRIBUTIONS : LE MODÈLE PERSUA2 ET SES APPLICATIONS 174

3.1 LE MODÈLE PERSUA2 174
3.2 ILLUSTRATION SUR L'EIAH AMBRE-ADD 176
3.3 ADAPTATION DU MODÈLE PERSUA2 POUR LES MOOCS 178
3.4 ÉVOLUTION DU MODÈLE PERSUA2 DANS LE CADRE DU PROJET "CARTOGRAPHIE DES SAVOIRS" 180

CONCLUSION ET PERSPECTIVES 182

SYNTHÈSE DES CONTRIBUTIONS 184

POUR EN SAVOIR PLUS 184

PUBLICATIONS LIÉES 185

Introduction

Nous avons vu dans le chapitre B1 des outils permettant d'obtenir un profil de chaque apprenant, et dans le chapitre B2 des outils permettant de proposer à l'apprenant des exercices répondant à certaines contraintes. Si un enseignant souhaite à partir d'un profil d'apprenant lui proposer des exercices personnalisés, il peut alors le faire, mais cela lui demanderait d'étudier chaque profil d'apprenant pour ensuite définir des contraintes sur les exercices à lui proposer, ce qui représente une charge de travail beaucoup trop lourde. Nous souhaitons donc mettre en place un processus automatique qui, à partir du contenu du profil de chaque apprenant, définisse les contraintes nécessaires à la constitution d'une séquence d'exercices personnalisée. La problématique étudiée dans ce chapitre est ainsi celle de l'adaptativité des EIAH.

La manière dont le système s'adapte à l'apprenant ne devrait pas dépendre seulement du profil de l'apprenant, mais aussi des objectifs de l'enseignant et du contexte d'apprentissage (Luengo, 2009, p. 12). C'est pourquoi il est important d'associer l'enseignant à la mise en place du processus de personnalisation. De plus, deux enseignants différents n'auront peut-être pas les mêmes attentes sur la manière dont un EIAH donné doit s'adapter à l'apprenant.

Notre objectif est donc de proposer des outils permettant à l'enseignant d'exprimer une stratégie de personnalisation que le système puisse ensuite mettre en œuvre de manière automatique.

1 Cadre des recherches

Les travaux décrits dans ce chapitre ont été effectués dans un premier temps dans le cadre de la thèse de Marie Lefevre (2006-2009), financée par une bourse ministérielle et co-encadrée par Stéphanie Jean-Daubias. Cette thèse a donné lieu à la proposition du modèle PERSUA2 mis en œuvre dans le logiciel Adapte.

Ce modèle a ensuite été adapté au contexte particulier des MOOCS dans le cadre du stage de Master 2 Recherche en informatique de Florian Clerc (2014), encadré par Marie Lefevre, en collaboration avec Jean-Charles Marty, et financé par le projet COAT (Connaissances Ouvertes à Tous).

Le modèle PERSUA2 est par ailleurs réutilisé et adapté en collaboration avec Sonia Mandin et Marie Lefevre dans le cadre du projet "Cartographie des Savoirs", un Projet Investissements d'Avenir piloté par la société Educlever, qui édite le site Maxicours.com. Financé pour deux ans (2013 et 2014) par l'appel e-Education 2, ce projet vise à faciliter la personnalisation des parcours d'apprentissage, en collaboration avec l'équipe METAH (Modèles et Technologies pour l'Apprentissage Humain) du LIG (Laboratoire d'Informatique de Grenoble) et l'équipe SILEX du LIRIS.

2 Questions de recherche : comment permettre à l'enseignant d'éliciter une stratégie de personnalisation ? Comment appliquer une telle stratégie pour proposer des activités personnalisées ?

Pour permettre à un enseignant d'éliciter sa stratégie de personnalisation, nous avons défini dans le cadre de la thèse de Marie Lefevre le modèle PERSUA2 (PERSonnalisation Unifiée des Activités d'Apprentissage), qui propose un formalisme permettant de décrire une stratégie de personnalisation. Le logiciel Adapte s'appuie sur ce modèle pour proposer à l'enseignant une interface lui permettant d'éliciter sa stratégie. Pour que le système puisse mettre en œuvre la stratégie de personnalisation définie par l'enseignant, un moteur associé au modèle PERSUA2 permet d'exploiter les stratégies de personnalisation.

La **Figure 50** présente comment cette approche s'inscrit dans la démarche présentée au chapitre 1 de ce mémoire. Il s'agit en effet de permettre à un enseignant de définir un modèle de personnalisation. L'élicitation d'un tel modèle s'appuie sur le modèle PERSUA2, et est effectuée par l'enseignant *via* l'interface du logiciel Adapte. PERSUA2 est donc un méta-modèle de personnalisation. Un moteur associé à ce méta-modèle permet d'exploiter le modèle de personnalisation défini par l'enseignant afin de proposer aux élèves des activités adaptées à leurs profils.

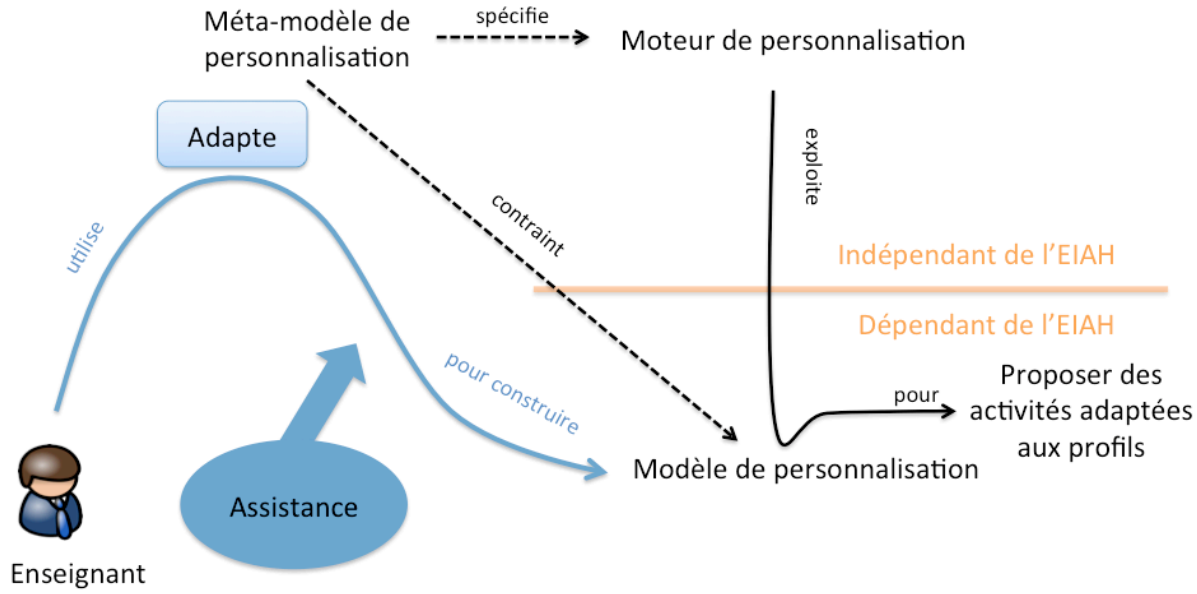

Figure 50. Instanciation de l'approche présentée au chapitre 1 à la question de l'élicitation de stratégies de personnalisation.

Nous souhaitons également apporter une assistance à l'enseignant pour définir un modèle de personnalisation. Cette assistance pourra consister à proposer des versions spécialisées du méta-modèle, appliquées à un cadre particulier, par exemple celui des MOOCs (cf. 3.3).

3 Contributions : le modèle PERSUA2 et ses applications

Dans cette section, je présente d'abord le modèle PERSUA2 et le processus de personnalisation mis en œuvre par le moteur associé, avant d'illustrer comment nous l'avons utilisé pour rendre l'EIAH AMBRE-add adaptatif. Je décris ensuite une adaptation du modèle PERSUA2 au contexte particulier des MOOCs. La dernière sous-section présente comment nous avons enrichi le modèle PERSUA2 pour proposer un moteur de personnalisation des parcours dans le cadre d'un EIAH fondé sur une ontologie des savoirs.

3.1 Le modèle PERSUA2

Pour acquérir une stratégie de personnalisation auprès d'un enseignant, il est nécessaire de connaître les critères d'individualisation qu'il souhaite mettre en œuvre. Ces critères portent d'une part sur la manière dont il souhaite affecter une activité à un apprenant, et d'autre part sur le contexte dans lequel les apprenants vont utiliser la séquence de travail qui leur est destinée.

Selon le méta-modèle PERSUA2, un modèle de personnalisation est constitué de deux éléments : une *stratégie de personnalisation,* qui permet d'affecter une activité à un apprenant et un *contexte d'utilisation* de cette stratégie de personnalisation.

Nous définissons une **stratégie de personnalisation** comme un ensemble de règles d'affectation précisant quelles activités choisir ou générer en fonction du contenu du profil de l'apprenant. Pour définir une telle stratégie de personnalisation, il faut donc disposer d'une part de profils d'apprenants sur lesquels on peut définir des contraintes, et d'autre part de modèles d'exercices ou plus généralement de modèles d'activités, issus par exemple de l'approche GEPPETO (cf. Chapitre B2).

Chaque règle d'affectation est de la forme : SI <contraintes sur le profil> ALORS <proposer ce(s) modèles(s) d'exercices> SINON <proposer ce(s) (autres) modèles(s) d'exercices>, la partie SINON étant optionnelle. Les contraintes sur le profil permettent de poser des conditions sur certaines valeurs du profil de l'apprenant.

Une priorité est associée à chaque règle, dans le cas où plusieurs règles peuvent s'appliquer. Ces priorités permettent d'ordonner les règles d'affectation constituant une stratégie de personnalisation.

Un **contexte d'utilisation** rassemble quant à lui un ensemble d'informations permettant de caractériser la situation pédagogique prévue par l'enseignant et dans laquelle se trouve un apprenant lorsque qu'il effectue la séance de travail. Ces informations portent, entre autres, sur la liste des élèves pour lesquels une personnalisation des séances est souhaitée, mais aussi sur des caractéristiques plus générales de la séquence d'activités (nombre d'exercices, durée, etc.).

Une définition formelle du modèle PERSUA2 peut être consultée dans (Lefevre, Guin, & Jean-Daubias, 2012).

L'indépendance des deux parties du modèle de personnalisation permet d'associer une même stratégie pédagogique à plusieurs contextes d'utilisation et inversement. Ainsi l'enseignant peut définir une unique stratégie de personnalisation qui explicite comment il estime souhaitable d'affecter les activités, et l'utiliser pour des séances qui n'ont pas toutes la même durée et ne concernent pas les mêmes élèves à chaque fois. Inversement, pour un même groupe d'élèves et des séances de durées identiques, l'enseignant pourra choisir

d'utiliser des stratégies de personnalisation différentes, selon le contenu sur lequel il veut travailler pour une séance donnée, chacune pouvant refléter un objectif pédagogique différent.

Le méta-modèle PERSUA2 a été mis en œuvre dans le logiciel Adapte, qui offre à l'enseignant une interface lui permettant de définir un modèle de personnalisation. Chaque modèle de personnalisation défini selon le méta-modèle PERSUA2 peut être partagé et réutilisé par d'autres enseignants. Nous avons associé au modèle PERSUA2 un moteur capable d'exploiter de tels modèles de personnalisation.

Processus de personnalisation

Le moteur de personnalisation prend en entrée trois éléments : les profils des apprenants, la stratégie de personnalisation de l'enseignant et le contexte d'utilisation de la séquence d'activités à personnaliser. Il fournit en sortie deux types d'éléments : les séquences d'activités adaptées au profil de chaque apprenant et un bilan des propositions à destination de l'enseignant.

Le processus mis en œuvre par le moteur est constitué de trois étapes :

1. L'application des règles d'affectation : pour chaque apprenant, le moteur évalue chaque règle d'affectation. Si les prémisses de la règle sont vérifiées par le profil de l'apprenant, alors on ajoute le(s) modèle(s) d'exercices à proposer à une liste de modèles. Si ce n'est pas le cas et qu'il existe une partie *Sinon* à la règle, d'autres modèles d'exercices sont ajoutés. Dans cette liste de modèles, à chaque modèle est associé le degré de priorité de la règle d'affectation qui a permis de l'ajouter.
2. La création des listes d'activités à générer : à partir de la liste de modèles d'exercices à proposer à l'apprenant, des degrés de priorité, et des contraintes du contexte d'utilisation (durée, nombre d'exercices), le moteur choisit quels modèles d'exercices utiliser et éventuellement combien d'exercices doivent être proposés pour chaque modèle.
3. La génération des activités : à partir de bases d'exercices ou de générateurs d'exercices, et de la liste issue de l'étape précédente, les activités sont construites pour chaque apprenant. On peut utiliser pour cette étape les générateurs issus de l'approche GEPPETO-P ou des EIAH pilotés par l'approche GEPPETO-S (cf. Chapitre B2).

En utilisant ce processus de personnalisation, le moteur crée autant de séances de travail personnalisées qu'il y a de profils d'élèves, ainsi qu'un bilan pour l'enseignant. (Lefevre et al., 2012) décrit de manière plus détaillée ce processus et un exemple illustrant son fonctionnement avec une stratégie de personnalisation et deux profils d'élèves.

Évaluation des contributions

La mise en œuvre du modèle PERSUA2 et du moteur de personnalisation au sein du logiciel Adapte a d'une part montré leur faisabilité technique et nous a d'autre part permis de les évaluer par des mises en situation concrètes impliquant des enseignants. Dans ces expérimentations, décrites dans (Lefevre, 2009), nous avons proposé à des enseignants n'étant pas intervenus dans l'élaboration de cette contribution et ne maitrisant pas particulièrement l'outil informatique de définir sur papier des modèles de personnalisation correspondant à leurs besoins. Nous leur avons ensuite demandé de retranscrire ces

modèles dans le logiciel Adapte. Ces expérimentations ont montré que le modèle PERSUA2 permettait bien à des enseignants de définir des stratégies de personnalisation, et que le moteur de personnalisation associé permettrait de construire des séquences personnalisées conformes aux stratégies de personnalisation définies.

Positionnement

Le modèle PERSUA2 permet à des enseignants de définir des stratégies de personnalisation destinées à adapter des séances de travail proposées aux élèves. D'autres travaux de recherche permettent à des enseignants de faire cela, en particulier au sein d'outils auteurs (Ainsworth, 2000; Cristea, 2005; Murray, 2003b; Van Marcke, 1998), mais le fait que la création de la stratégie de personnalisation soit liée à la création de l'EIAH implique que si l'on veut pouvoir disposer de plusieurs stratégies pédagogiques, il y aura autant de versions du logiciel que de stratégies pédagogiques.

Certains outils ont choisi de séparer la définition du contenu pédagogique de l'EIAH des stratégies de personnalisation (Cristea & Calvi, 2003; Murray, 2003c), ce qui permet d'envisager plusieurs stratégies pour un même EIAH, mais ces stratégies sont définies par le concepteur au moment de la création de l'EIAH et sont incluses dans l'EIAH créé. Il n'est pas possible à un enseignant de définir une nouvelle stratégie de personnalisation pour un EIAH existant. Adapte permet de le faire essentiellement grâce à l'association du modèle PERSUA2 et de l'approche GEPPETO-S, qui permet d'agir de l'extérieur sur un EIAH existant (cf. Chapitre B2). Nous allons à présent en voir une illustration pour l'EIAH AMBRE-add.

3.2 Illustration sur l'EIAH AMBRE-add

AMBRE-add (cf. Chapitre A3) est un EIAH adaptable, puisque *via* le module AMBRE-enseignant et le générateur GenAMBRE (cf. Chapitre A5), l'enseignant peut créer des séquences de travail individualisées avec l'EIAH. De plus, un module de calcul de profils à partir des traces permet d'évaluer les compétences des élèves pour les différentes classes de problèmes ou les différentes étapes du cycle AMBRE.

L'objectif du travail décrit ici était de transformer cet EIAH adaptable en EIAH adaptatif, c'est-à-dire qui adapte automatiquement les séquences de travail proposées aux élèves à partir de leurs profils. Nous avons donc choisi d'utiliser Adapte pour définir une stratégie de personnalisation. Nous avons effectué ce travail dans les mêmes conditions expérimentales que celui décrit dans le chapitre B2, où je me suis placée en tant qu'experte de l'EIAH AMBRE, utilisant Adapte pour la première fois. Nous sommes par conséquent ici dans un cas particulier de l'utilisation d'Adapte, puisque c'est le concepteur de l'EIAH qui définit une stratégie de personnalisation et pas un enseignant. Cependant il est toujours possible qu'un enseignant modifie cette stratégie de personnalisation s'il le souhaite.

Le chapitre B2 a permis de présenter comment j'ai en tant qu'expert de AMBRE-add utilisé l'approche GEPPETO-S *via* le logiciel Adapte pour définir un modèle de l'EIAH AMBRE-add, décrivant comment on peut agir sur les activités proposées à l'élève. Pour pouvoir définir une stratégie de personnalisation suivant le modèle PERSUA2, il faut effectivement disposer de modèles d'activités (pour la partie *Alors* des règles), mais il faut également disposer de profils d'apprenants (pour la partie *Si* des règles). Adapte est un module de l'environnement EPROFILEA (Jean-Daubias et al., 2009), qui permet à un enseignant de gérer des profils d'apprenants issus de sources diverses, quels que soient la discipline ou le niveau d'enseignement. EPROFILEA est constitué de deux parties : la première est destinée à

obtenir des profils utilisables au sein de l'environnement, la seconde permet de les exploiter, en particulier en utilisant Adapte.

Avant de pouvoir utiliser Adapte, j'ai donc dû définir le procédé d'importation des profils existants pour AMBRE-add au sein d'EPROFILEA (Guin & Lefevre, 2013). Ceci s'effectue en deux étapes au sein d'EPROFILEA : la définition d'un modèle de profil conforme à l'environnement EPROFILEA (Jean-Daubias et al., 2009), puis la création d'un procédé de transformation des profils existants en des profils respectant le dit modèle (Lefevre & Jean-Daubias, 2012).

Pour définir une stratégie de personnalisation pour AMBRE-add, j'ai dans un premier temps défini une stratégie sur papier, en utilisant des règles mais sans utiliser Adapte. L'objectif était de ne pas me limiter aux fonctionnalités d'Adapte (que je n'avais jamais utilisé au moment d'effectuer ce travail). J'ai ainsi mis quarante minutes pour définir un ensemble de dix règles. Deux d'entre elles concernent le niveau de lecture de l'apprenant (par exemple : SI niveau de lecture = très faible ALORS ne jamais proposer de niveau de complication de l'énoncé supérieur à 1), trois concernent le niveau de calcul de l'apprenant (par exemple : SI calcul en général = partiellement maîtrisé ou maîtrisé ALORS proposer une difficulté de calcul supérieure à 2). Les cinq autres règles sont relatives à la difficulté des classes de problèmes proposées (par exemple : SI classes très faciles = maîtrisé et classes faciles = partiellement maîtrisé ALORS proposer classes très faciles avec complication = 2 et/ou classes faciles ou difficiles avec complication = 1).

J'ai ensuite tenté de définir avec l'outil Adapte cette stratégie de personnalisation explicitée sur papier. Adapte utilise le modèle de l'EIAH AMBRE-add présenté au chapitre B2 pour générer une interface permettant de définir des règles d'affectation d'activités utilisant les modèles d'activités de AMBRE-add.

Lors de la retranscription de ma stratégie pédagogique sur papier dans Adapte, j'ai été confrontée à plusieurs difficultés (Guin & Lefevre, 2013), que j'ai contournées en multipliant le nombre de règles. Par exemple, comme il n'est pas possible d'utiliser un "OU" dans la partie ALORS d'une règle dans Adapte (par exemple classe facile OU difficile), il faut créer deux règles.

J'ai ainsi défini en tant que concepteur de l'EIAH une stratégie de personnalisation. Chaque enseignant pourra y associer un ou plusieurs contextes d'utilisation, pour préciser la durée de la séance souhaitée et les élèves qui devront effectuer cette séance. Je pourrais également définir un contexte d'utilisation par défaut pour les sessions se déroulant en dehors du contexte scolaire, et donc sans enseignant.

Synthèse

En utilisant l'outil Adapte, j'ai ainsi pu en tant que concepteur de AMBRE-add définir des règles permettant de rendre AMBRE-add adaptatif. Pour cela, trois étapes ont été nécessaires :

- Deux heures pour définir les mécanismes d'importation des profils AMBRE-add dans l'environnement Eprofilea supportant l'outil Adapte ;
- Deux heures pour définir le modèle de l'EIAH AMBRE-add, c'est-à-dire les connaissances permettant à Adapte (*via* l'approche GEPPETO) de connaître et d'agir sur le paramétrage de l'EIAH AMBRE-add ;

- Une heure et demie pour définir une stratégie de personnalisation qui sera proposée aux enseignants.

Le couple AMBRE-add/Adapte ainsi formé représente à présent un EIAH adaptatif, au sens qu'il s'adapte automatiquement à chaque apprenant, et cette adaptativité est de plus adaptable par chaque enseignant. En effet, un enseignant pourra soit utiliser la stratégie de personnalisation proposée, soit modifier la stratégie en fonction de ses besoins, soit redéfinir sa propre stratégie de personnalisation en suivant le modèle PERSUA2.

Cette étude montre l'intérêt de l'outil Adapte qui combine l'approche GEPPETO et le modèle PERSUA2. Il peut ainsi être utilisé pour rendre adaptatif un EIAH adaptable existant, quelle que soit l'origine de cet EIAH et la plateforme auteur avec laquelle il a éventuellement été conçu. Le fait de pouvoir utiliser un seul système pour rendre adaptatifs plusieurs EIAH représente également un gain de temps pour un concepteur qui n'a pas à s'approprier plusieurs outils.

Ce travail a de plus permis de recueillir un retour d'expérience sur l'utilisation de l'outil Adapte. Certains problèmes d'utilisabilité ont été identifiés, en particulier pour l'importation des profils et la définition de la stratégie pédagogique, ce qui permet de proposer des améliorations à l'outil.

Une prochaine étape de ce travail pourrait consister à réaliser des expérimentations en classe de cette version adaptative de AMBRE-add. En effet, les dernières expérimentations de la version classique de l'EIAH ont semblé montrer que c'est la personnalisation de l'EIAH qui permettrait d'améliorer le gain du cycle AMBRE sur l'apprentissage. On pourrait ainsi étudier si une version adaptative suscite plus de satisfaction ou d'intérêt de la part des élèves et des enseignants qu'une version classique, et si cela apporte un gain sur l'apprentissage. Il serait également intéressant d'étudier dans quelle mesure les enseignants adaptent l'adaptabilité de l'EIAH, c'est-à-dire personnalisent la stratégie d'adaptation en fonction de leurs besoins.

3.3 Adaptation du modèle PERSUA2 pour les MOOCs

Les MOOCs (Massive Online Open Courses) constituent un type particulier d'EIAH dans lequel la personnalisation prend une place particulièrement importante. En effet, une des principales caractéristiques de ces EIAH est le caractère très diversifié des apprenants inscrits aux cours, du fait de leur caractère ouvert. Personnaliser le parcours de chaque apprenant pourrait constituer une réponse à la faible rétention constatée dans les MOOCs, un grand nombre d'apprenants s'inscrivant mais ne suivant pas ensuite le MOOC ou seulement pendant peu de temps. Le tutorat utilisé traditionnellement en e-learning n'étant pas envisageable étant donné le nombre d'apprenants (souvent plusieurs milliers), une personnalisation automatique mais pilotée par l'auteur du MOOC semble être une approche pertinente. Nous ne cherchons pas ici à détecter une situation de démotivation pour proposer une remédiation (Miranda, Mangioni, Orciuoli, Loia, & Salerno, 2014) mais bien à proposer à l'auteur du MOOC de concevoir des parcours personnalisés afin de prévenir la démotivation.

Nous avons donc étudié comment utiliser le modèle PERSUA2 dans le cadre des MOOCs. Nous avons complété ce modèle afin qu'il soit mieux adapté à ce cas particulier d'EIAH, et nous avons défini des patrons de modèles, afin de faciliter la tâche de l'enseignant (il peut s'agir de l'auteur du MOOC ou plus généralement de l'équipe pédagogique).

Dans le modèle PERSUA2, l'enseignant doit définir une stratégie de personnalisation ainsi qu'un contexte d'utilisation. Dans son adaptation aux MOOC, $PERSUA2_{MOOC}$, on retrouve ces deux éléments (cf. **Figure 51**), le contexte d'utilisation décrivant les contraintes de la séquence de travail avec le MOOC (qui dure en général une semaine), par exemple le nombre d'heures de travail. La nouveauté par rapport au modèle PERSUA2 est le *Contexte d'Utilisation "Live"*. Il permet d'obtenir des informations qu'on ne peut avoir qu'au moment où l'apprenant se connecte à la plateforme de MOOC, comme le matériel utilisé par l'apprenant pour se connecter, la bande passante dont il dispose, ou le nombre d'apprenants connectés sur le MOOC à ce moment-là.

À partir des informations du contexte d'utilisation "live" et des éléments du profil de l'apprenant, le système va appliquer les règles d'affectation de la stratégie de personnalisation définie par l'enseignant, afin de choisir les activités à proposer à l'apprenant. Le nombre d'activités sera ensuite fixé en prenant en compte les contraintes liées au contexte d'utilisation de la séquence.

L'autre nouveauté par rapport au modèle PERSUA2 réside dans le fait que le système ne construit pas une séquence de travail pour l'apprenant, mais seulement une *boussole*, c'est-à-dire une recommandation d'activités destinée à l'apprenant. En effet, dans le contexte très ouvert et libre des MOOCS, il ne nous paraissait pas judicieux de contraindre l'apprenant à suivre un scénario fixé pour lui, mais plutôt de lui recommander un scénario d'apprentissage.

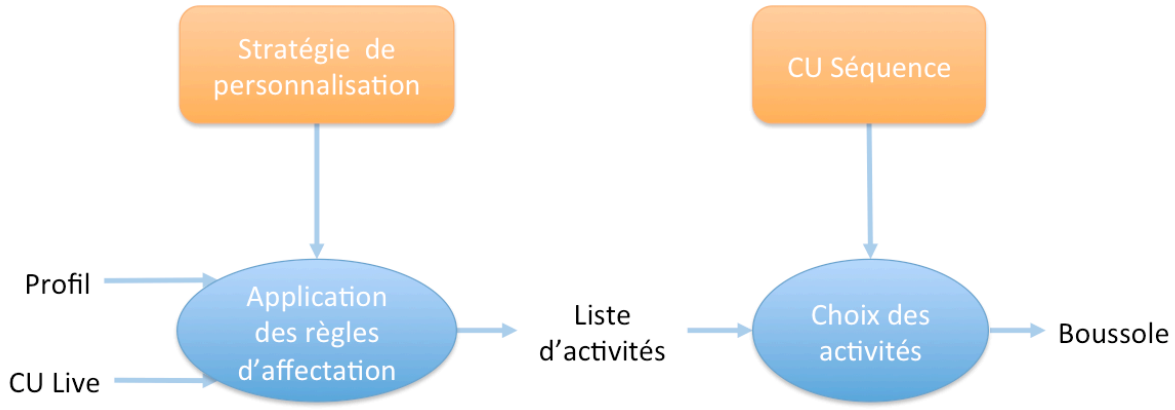

Figure 51. Processus de personnalisation fondé sur le modèle $PERSUA2_{MOOC}$

Nous avons vu qu'une stratégie de personnalisation est constituée de règles d'affectation permettant de choisir des activités proposées à l'apprenant en fonction des éléments de son profil. L'écriture de telles règles nécessite donc de disposer d'un modèle des profils d'apprenants et d'un modèle des activités proposées au sein du MOOC. Afin de faciliter la définition de ces modèles, nous avons défini des patrons de modèles adaptés au contexte des MOOCs.

Une étude de onze plateformes de MOOCs (Clerc, Lefevre, Guin, & Marty, 2014) a permis de recenser d'une part les fonctionnalités souvent présentes, afin de définir un patron de modèle des activités à personnaliser, et d'autre part les informations souvent collectées sur les apprenants, afin de définir un patron de modèle de profils d'apprenants.

Le patron de modèle de profils est ainsi divisé en catégories, telles que l'utilisation des ressources par l'apprenant, ses habitudes de travail sur le MOOC, sa manière de participer aux activités collectives, ses connaissances et compétences, et les informations qui ne peuvent pas être déduites des traces mais lui sont demandées lors de l'inscription. Ce patron de modèle (décrit dans (Clerc et al., 2014)) est destiné à être instancié tout d'abord par l'administrateur d'une plateforme de MOOC, en fonction des éléments tracés par la plateforme, et ensuite par l'équipe pédagogique du MOOC, qui précisera les éléments qu'elle souhaite connaître dans l'utilisation du MOOC par l'apprenant.

Le patron de modèle des fonctionnalités d'un MOOC (également décrit dans (Clerc et al., 2014)), issu de l'approche GEPPETO, permet à l'administrateur de la plateforme de la décrire en instanciant les fonctionnalités prévues qui, pour aider l'utilisateur, sont regroupées en catégories : apprentissage, échange avec les autres apprenants, évaluation, navigation, suivi de la progression, etc. L'équipe pédagogique caractérisera ensuite les activités d'un MOOC en particulier.

Une implémentation a été effectuée, fondée sur ces deux patrons de modèles et sur le modèle PERSUA2$_{MOOC}$. Le logiciel permet à un enseignant de définir une stratégie de personnalisation et de la tester sur des profils dont il peut préciser les valeurs. Le système construit pour chaque apprenant une boussole, c'est-à-dire une liste personnalisée d'activités recommandées.

Le modèle PERSUA2$_{MOOC}$ a été évalué d'une part en collaborant avec les auteurs du MOOC FOVEA (FOVEA, 2014) proposé à l'Université Lyon 1, et d'autre part à travers deux questionnaires destinés à des concepteurs de MOOC (Clerc et al., 2014). Ces évaluations semblent indiquer que ce modèle permet à une équipe pédagogique de mettre en place une stratégie de personnalisation pour un MOOC. Une utilisation en conditions réelles devrait être mise en place pour la prochaine édition du MOOC FOVEA.

3.4 Évolution du modèle PERSUA2 dans le cadre du projet "Cartographie des savoirs"

Dans le cadre du projet "Cartographie des savoirs", le LIG modélise les savoirs et savoir-faire d'un domaine à l'aide d'une ontologie fondée sur la notion de type de tâche (TT) (cf. Chapitre B1 section 2). Le chapitre B1 a décrit comment nous utilisons cette ontologie pour élaborer un profil de l'apprenant décrivant le taux de maîtrise de l'apprenant pour les différents TT. Les ressources pédagogiques proposées par le site Maxicours sont étiquetées par ces TT. Un des principaux objectifs du projet est la personnalisation des parcours d'apprentissage. Les profils comme les ressources étant liés aux TT, il nous a paru pertinent d'utiliser le modèle PERSUA2 pour définir des stratégies de personnalisation permettant de choisir des activités adaptées à chaque profil d'apprenant.

Dans le modèle PERSUA2, les règles d'affectation sont liées au domaine enseigné. On a par exemple pour AMBRE-add défini des règles comme « SI classes très faciles = maîtrisé et classes faciles = partiellement maîtrisé ALORS proposer classes très faciles avec complication = 2 et/ou classes faciles ou difficiles avec complication = 1 », liées à l'EIAH AMBRE-add. Dans le projet "Cartographie des savoirs", du fait de l'existence du référentiel, les règles d'affectation seront indépendantes du domaine, par exemple « SI un TT est non-acquis ou en-cours-d'acquisition ALORS proposer une activité de remédiation sur ce TT ». Étant donné le grand nombre de TT dans l'ontologie, cette règle va s'appliquer à plusieurs TT. Pour

effectuer des choix parmi toutes les activités proposées par les règles, une priorité sur les règles n'est alors plus suffisante, puisqu'il s'agit ici d'une seule règle. Nous avons donc modifié la définition d'une stratégie pédagogique, qui est ici composée d'un ensemble de règles pédagogiques, et d'un ensemble de règles d'arbitrage. Ces règles d'arbitrage sont destinées à choisir les activités à proposer en fonction du contexte d'utilisation (par exemple la durée de la séance de travail) et de l'historique de l'élève. Dans notre exemple, une règle d'arbitrage pourrait être de donner priorité aux TT les plus "bas" dans la hiérarchie des prérequis.

Dans le projet, les stratégies de personnalisation sont pour l'instant définies par les pédagogues de la société Educlever. Il est prévu par la suite de permettre aux enseignants utilisateurs de la plateforme d'agir sur ces stratégies *via* un éditeur. Les pédagogues ont exprimé le souhait de définir plusieurs stratégies pédagogiques correspondant à différentes intentions, comme cela est possible avec PERSUA2. Il était de plus souhaité que le système choisisse automatiquement les stratégies pertinentes en fonction d'un moment pédagogique (évaluation, remédiation, formation, révision) défini par l'enseignant utilisant la plateforme.

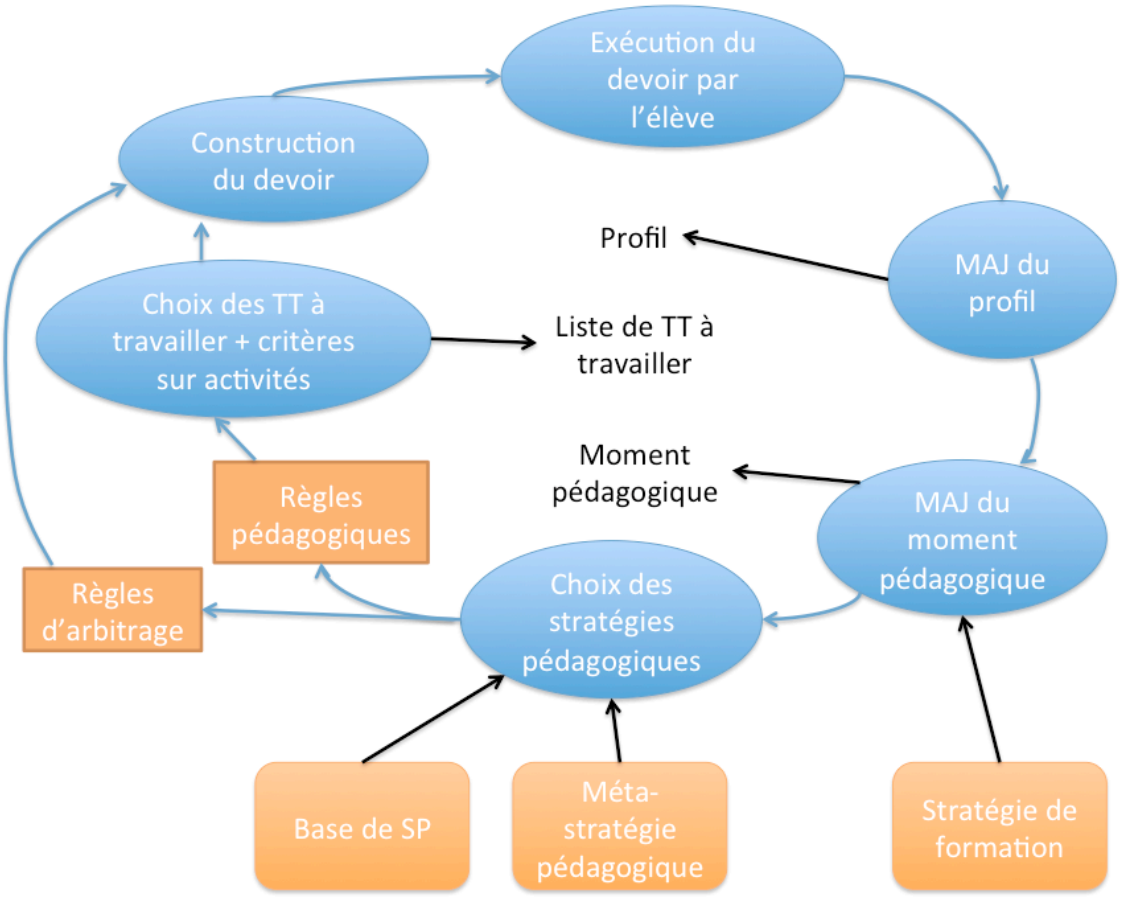

Figure 52. Processus de personnalisation en lien avec un référentiel des compétences

Nous avons donc introduit la notion de *méta-stratégie pédagogique*, un ensemble de règles qui permet de choisir les stratégies pédagogiques pertinentes en fonction du moment pédagogique. Cela nous a également amené à proposer la notion de *stratégie de formation*, un ensemble de règles qui permet au système de passer d'un moment pédagogique à un

autre en fonction de l'évolution du profil de l'apprenant (par exemple de l'évaluation à la remédiation).

La **Figure 52** présente une version simplifiée du processus de personnalisation ainsi construit. L'enseignant qui souhaite faire travailler ses élèves avec la plateforme choisit un type de tâche ou un sujet (ensemble de TT) à faire travailler, ainsi qu'un moment pédagogique (évaluation, remédiation…). Le système utilise la méta-stratégie pour choisir dans la base de stratégies pédagogiques celles qui sont pertinentes pour le moment pédagogique choisi. Il utilise les règles pédagogiques de ces stratégies pour construire une liste de TT à travailler ainsi que les types d'activités à proposer sur chacun de ces TT. Il utilise ensuite les règles d'arbitrage des même stratégies, l'historique de l'élève et le contexte d'utilisation (durée de la séance de travail) pour construire un devoir personnalisé pour chaque élève. L'élève effectue alors sa séance de travail, après quoi le système met à jour son profil (cf. Chapitre B1), puis utilise la stratégie de formation pour mettre à jour le moment pédagogique ainsi que la liste des TT à travailler. Par exemple, si le travail de l'apprenant portait sur une remédiation sur un TT, le système passe à une évaluation pour vérifier que le TT est acquis, puis au prochain cycle si c'est bien le cas, il choisira de passer à un TT dont le précédent est un prérequis.

Les règles nécessaires à ce moteur de personnalisation (règles pédagogiques, règles d'arbitrage, méta-stratégie et stratégie de formation) sont définies par les pédagogues d'Educlever. Une implémentation du moteur de personnalisation est en cours, ce qui permettra à court terme une évaluation des propositions sur les terrains du projet, c'est-à-dire les établissements de la Somme Numérique et le CNED.

Conclusion et perspectives

Ce chapitre a présenté le modèle PERSUA2, permettant à un enseignant d'expliciter une stratégie de personnalisation constituée par un ensemble de règles décrivant quelles activités proposer à un apprenant en fonction de son profil.

Après avoir décrit le modèle PERSUA2, et le processus de personnalisation qui lui est associé, j'ai montré avec l'EIAH AMBRE-add comment PERSUA2, *via* le logiciel Adapte qui le met en œuvre, peut être utilisé par le concepteur d'un EIAH adaptable pour le rendre adaptatif.

Les évaluations du modèle PERSUA2 ont montré qu'il permet à des enseignants de définir des stratégies de personnalisation, et que le moteur de personnalisation associé construit effectivement des séquences de travail adaptées aux apprenants et respectant les souhaits de l'enseignant. Nous étudions actuellement comment assister l'enseignant dans la définition de ses stratégies pédagogiques (cf. **Figure 50**).

Une première piste de recherche consiste à concevoir des versions spécialisées du modèle PERSUA2, adaptées à des contextes particuliers. C'est par exemple ce que nous avons fait avec le modèle $\text{PERSUA2}_{\text{MOOC}}$, qui propose à l'enseignant de compléter des patrons de modèles pour les profils et les activités, et à partir desquels on pourrait également proposer des patrons de règles d'affectation.

Une deuxième piste de recherche consiste à appliquer le Raisonnement à Partir de l'Expérience Tracée à la question de la définition de stratégies de personnalisation (Cordier et al., 2010). Il s'agirait en effet d'exploiter les traces de l'enseignant qui travaille avec le

système pour lui faciliter certaines écritures répétitives de règles d'affectation. On pourrait également exploiter un avis de l'enseignant qui, en testant sa stratégie, peut estimer certaines activités comme non adaptées aux profils, et lui proposer alors des modifications dans sa stratégie qui permettraient d'obtenir des résultats correspondant mieux à ses attentes. L'analyse des traces des apprenants travaillant avec l'EIAH pourrait aussi être utilisée par le système pour signaler à l'enseignant les stratégies de personnalisation qui conduisent à un bon apprentissage et celles qui semblent moins efficaces, et pour lui suggérer des améliorations à apporter à ses stratégies.

Dans le cadre des MOOCs, où l'apprenant est très autonome et souhaite pouvoir être libre de construire lui-même son apprentissage, l'on peut même envisager de recueillir l'avis de l'apprenant sur le parcours qui lui est proposé pour modifier (et donc personnaliser) la stratégie de personnalisation prévue par l'enseignant.

L'outil mettant en œuvre le modèle PERSUA2$_{MOOC}$ va être à court terme couplé au langage SPARE-LNC (cf. Chapitre B1) qui permet d'élaborer les profils des apprenants en définissant des méthodes de calcul d'indicateurs, ainsi qu'aux générateurs d'exercices de l'outil auteur ASKER (cf. Chapitre B2), ce qui permettra d'obtenir pour le contexte des MOOCs un environnement mettant en œuvre l'ensemble du cycle de personnalisation présenté en introduction de la partie B.

Nous conduisons également des travaux dans lesquels nous faisons évoluer le modèle PERSUA2 pour répondre à des besoins plus spécifiques, comme dans le projet "Cartographie des savoirs" dans lequel plusieurs niveaux de règles (règles pédagogiques, règles d'arbitrage, méta-stratégie et stratégie de formation) permettent de répondre finement aux intentions pédagogiques des concepteurs de la personnalisation. Il reste encore à étudier dans quelle mesure il sera possible de donner la main sur ces différents niveaux de règles à un enseignant qui se positionne en simple utilisateur de la plateforme pour ses élèves.

Synthèse des contributions

Question de recherche : Adapter les activités au profil de l'apprenant, en fonction d'une stratégie pédagogique choisie par l'enseignant.

- **Contributions théoriques** :
  - Méta-modèle PERSUA2
  - $PERSUA2_{MOOC}$, une spécialisation du modèle PERSUA2 au contexte des MOOCs
  - Patron de modèles de profils et patron de modèles d'activités adaptés aux MOOCS.
  - $PERSUA2_{Carto}$, une évolution du modèle PERSUA2 pour des EIAH fondés sur un référentiel des compétences.
- **Systèmes développés** :
  - Adapte, qui met en œuvre le modèle PERSUA2 et un moteur de personnalisation.
  - Une implémentation de $PERSUA2_{MOOC}$ et des patrons de modèles de profils et d'activités, pour une plateforme de personnalisation des MOOCs.
  - Un moteur de personnalisation pour la Cartographie des Savoir fondé sur $PERSUA2_{Carto}$ est en cours d'implémentation par Educlever.
- **Déploiement** :
  - La plateforme de personnalisation des MOOCs sera utilisée pour la prochaine édition du MOOC FOVEA.
  - Le moteur de personnalisation fondé sur $PERSUA2_{Carto}$ sera déployé à partir de novembre 2014 sur la Cartographie des Savoirs, pour la Somme Numérique.
- **Co-auteurs** : Marie Lefevre, Stéphanie Jean-Daubias, Florian Clerc, Sonia Mandin

Pour en savoir plus

• Sur le modèle PERSUA2 et le moteur de personnalisation :

Lefevre, M., Guin, N., & Jean-Daubias, S. (2012). Personnaliser des activités pédagogiques de manière unifiée : une solution à la diversité des dispositifs. *Revue Des Sciences et Technologies de l'Information et de La Communication Pour l'Education et La Formation (STICEF)*, *19*.

• Sur l'utilisation de PERSUA2 et Adapte pour personnaliser AMBRE-add :

Guin, N., & Lefevre, M. (2013). From a customizable ITS to an adaptive ITS. In *AIED 2013* (pp. 141–150). Memphis, USA.

• Sur $PERSUA2_{MOOC}$:

Clerc, F., Lefevre, M., Guin, N., & Marty, J.-C. (2014). *Mise en Place de la Personnalisation Dans le Cadre des MOOCs*. Rapport de recherche LIRIS RR-LIRIS-2014-012.

Publications liées

Conférences internationales

Guin, N., & Lefevre, M. (2013). From a customizable ITS to an adaptive ITS. In *AIED 2013* (pp. 141–150). Memphis, USA.

Lefevre, M., Guin, N., & Jean-Daubias, S. (2008). Adapte, a tool for the teacher to personalize activities. In *9th International Conference on Intelligent Tutoring Systems (ITS'2008)* (pp. 699–701). Montréal, Canada.

Lefevre, M., Jean-Daubias, S., & Guin, N. (2012). An approach for unified personalization of learning. In *International Workshop on Personalization Approaches in Learning Environments (PALE) - Conference User Modeling, Adaptation, and Personalization* (pp. 5–10).

Revues nationales

Lefevre, M., Broisin, J., Butoianu, V., Daubias, P., Daubigney, L., Greffier, F., Guin, N., Jean-Daubias S., Monod-Ansaldi, R., Terrat, H. (2012). Personnalisation de l'apprentissage : comparaison des besoins et approches à travers l'étude de quelques dispositifs. *Revue Des Sciences et Technologies de l'Information et de La Communication Pour l'Education et La Formation (STICEF)*.

Lefevre, M., Guin, N., & Jean-Daubias, S. (2012). Personnaliser des activités pédagogiques de manière unifiée : une solution à la diversité des dispositifs. *Revue Des Sciences et Technologies de l'Information et de La Communication Pour l'Education et La Formation (STICEF)*, *19*.

Conférences nationales

Lefevre, M., Cordier, A., Jean-Daubias, S., & Guin, N. (2011). Quels modèles de connaissances pour une personnalisation unifiée de l'apprentissage ? In *IC 2011* (pp. 657–672). Chambéry, France.

Lefevre, M., Jean-Daubias, S., & Guin, N. (2011a). Adapte, un logiciel pour aider l'enseignant à proposer des activités personnalisées à chacun de ses apprenants. In *Atelier "Personnalisation de l'apprentissage : quelles approches pour quels besoins ?", EIAH 2011*. Mons, Belgique.

Lefevre, M., Jean-Daubias, S., & Guin, N. (2011b). PERSUA2, un modèle pour unifier le processus de personnalisation des activités d'apprentissage. In *EIAH 2011* (pp. 369–380). Mons, Belgique.

CONCLUSION ET PERSPECTIVES SUR LA PARTIE B

1 Bilan

La problématique qui a été abordée dans cette partie B est celle de la personnalisation des EIAH. Trois questions de recherche ont été discutées : celle de l'élaboration d'un profil de chaque apprenant à partir de l'analyse de ses traces d'interaction avec l'EIAH (modélisation de l'apprenant), celle de la génération semi-automatique d'exercices, et celle de la personnalisation des activités à partir des profils, selon une stratégie définie par l'enseignant.

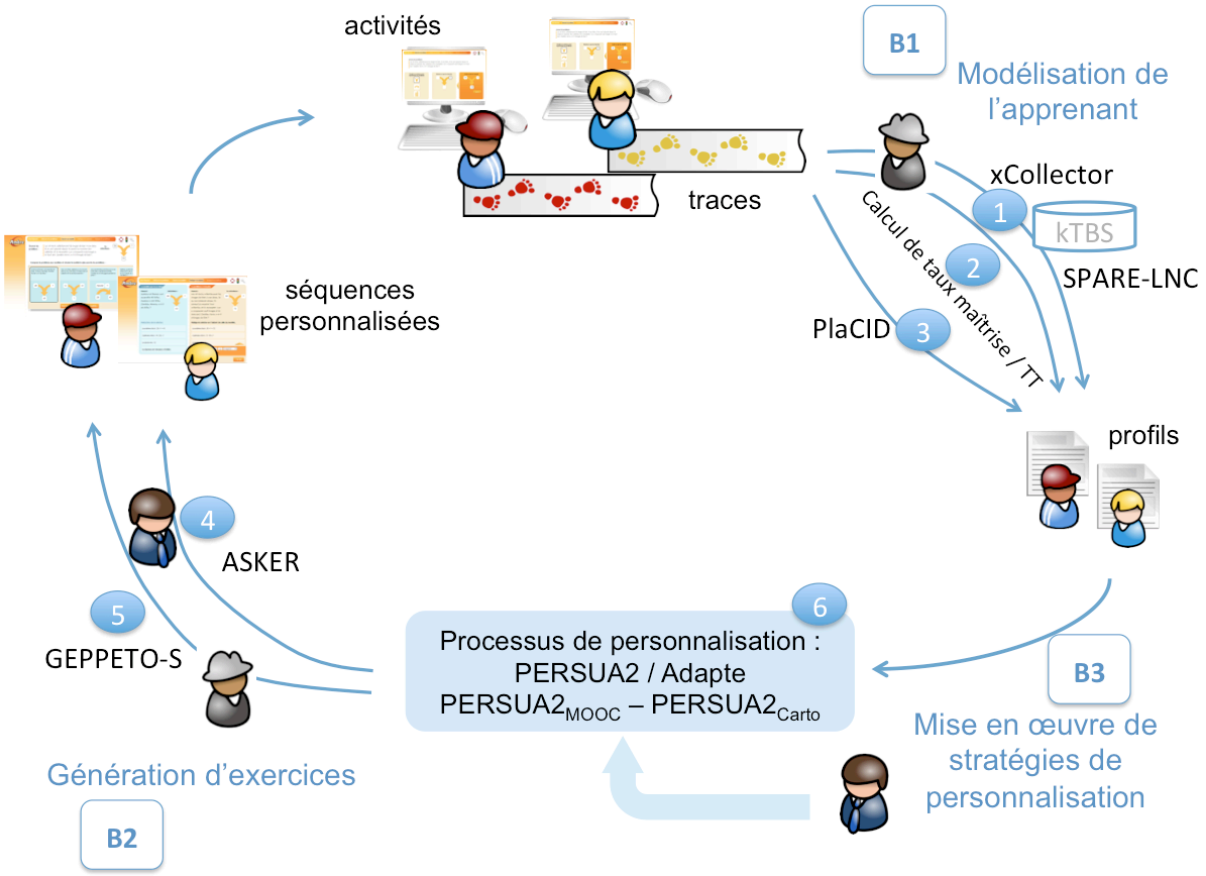

Figure 53. Contributions à la question de la personnalisation des EIAH

La **Figure 53** reprend la figure **Figure 34** présentée dans l'introduction de la partie B et synthétise les contributions apportées sur ces trois questions de recherche.

Concernant la **modélisation de l'apprenant**, xCollector permet à un utilisateur d'importer dans la plateforme kTBS des traces d'EIAH existantes. SPARE-LNC permet à un analyste ou un enseignant d'interroger des traces stockées dans kTBS en utilisant un langage naturel contrôlé, de manière à élaborer des indicateurs constituant un profil de l'apprenant (cf. (1) sur la **Figure 53**).

Un autre modèle proposé dans le cadre du projet “Cartographie des savoirs” permet d’élaborer des profils d’apprenant en calculant des taux de maîtrise de différentes natures pour des types de tâches représentés au sein d’une ontologie (cf. (2) sur la **Figure 53**).

Enfin la plateforme PlaCID permet l’instanciation et la comparaison de techniques génériques de diagnostic, afin de permettre au concepteur du diagnostic de choisir la technique de diagnostic la plus pertinente par rapport à son EIAH et à ses besoins (cf. (3) sur la **Figure 53**).

Pour ce qui est de la **génération d’exercices**, l’approche GEPPETO a donné lieu d’une part à la réalisation de l’outil auteur ASKER (cf. (4) sur la **Figure 53**), qui permet à un enseignant de définir des modèles d’exercices qui seront utilisés par des générateurs pour construire des exercices respectant les critères fixés par chaque modèle d’exercices. Les types d’exercices générés par cet outil sont ceux qui peuvent être appliqués à un grand nombre de domaines : QCM, appariements, groupements, textes à trous, etc.

L’approche GEPPETO permet d’autre part de personnaliser des EIAH existants, qui peuvent proposer des types d’exercices plus spécifiques. En effet, l’approche GEPPETO-S (cf. (5) sur la **Figure 53**) permet à un expert de définir un modèle d’un EIAH existant, modèle qui permet ensuite au système de générer pour l’enseignant une interface spécifique lui permettant de caractériser les exercices à effectuer au sein de cet EIAH.

Enfin, pour ce qui est de la **personnalisation des activités à partir des profils** (cf. (6) sur la **Figure 53**), le modèle PERSUA2 permet à un enseignant de définir une stratégie de personnalisation qui pourra être mise en œuvre par le système Adapte. Une telle stratégie est constituée d’un ensemble de règles décrivant quels modèles d’exercices proposer à l’apprenant en fonction des valeurs des éléments de son profil.

Une version de ce modèle adaptée aux MOOCS a été définie, modèle auquel ont été associés des patrons de modèles de profils et d’activités, qui pourront être instanciés par un auteur de MOOC. L’outil rassemblant ces trois modèles permet de personnaliser des MOOCs.

Le modèle PERSUA2 a d’autre part été étendu pour permettre la personnalisation de parcours au sein d’un EIAH fondé sur une cartographie des savoirs. Les règles d’affectation des activités en fonction du profil, définies par le pédagogue, ne dépendent alors plus du domaine et nécessitent l’ajout de règles d’arbitrages. Une méta-stratégie permet de choisir entre plusieurs stratégies de personnalisation, en fonction d’une intention pédagogique qui pourra évoluer au cours de l’apprentissage, selon une stratégie de formation.

Dans l’ensemble de ces contributions, l’élicitation des connaissances par l’enseignant ou l’expert est rendue possible par des interfaces s’appuyant sur des méta-modèles. Les modèles issus de l’acquisition des connaissances sont ensuite exploités par des moteurs. L’élicitation des connaissances par l’utilisateur ne nécessite pas de compétence en informatique, et nous cherchons de plus à faciliter la tâche de cet utilisateur en lui proposant des patrons ou des modèles pré-remplis lorsque cela est possible. Grâce à cette approche fondée sur des méta-modèles, les contributions ne sont pas dépendantes du domaine des EIAH que l’on souhaite personnaliser.

2 Perspectives

Ces contributions à la question de la personnalisation des EIAH permettent de mettre en œuvre l'ensemble du cycle de personnalisation. Nos travaux sur cette problématique vont se poursuivre, essentiellement au sein de deux contextes : celui des MOOCs et celui de la Cartographie des Savoirs.

Au sein du projet "Cartographie des Savoirs", nous souhaitons profiter de l'existence d'Ontoprax (Chaachoua et al., 2013), qui apporte une représentation des connaissances du domaine que nous souhaitons pouvoir exploiter dans notre approche de la personnalisation. Nous allons poursuivre au sein de ce projet nos travaux sur l'élaboration de profils d'apprenants, en affinant les calculs des taux de maîtrise et en proposant plusieurs types de profils adaptés à plusieurs acteurs et plusieurs contextes. Nous allons également évaluer et affiner le modèle issu de PERSUA2 pour ce projet, et coupler les générateurs d'exercices de l'outil auteur ASKER aux variables de l'ontologie, pour achever de mettre en œuvre le cycle complet de la personnalisation dans le cadre de ce projet. Ce projet offre à la fois une possibilité de collaborer avec le LIG et de bénéficier d'une représentation des connaissances du domaine, mais aussi la présence de pédagogues intéressés par la mise en place de la personnalisation ainsi que des terrains d'expérimentation à grande échelle.

Dans le contexte des MOOCS, nous allons poursuivre dans le cadre du projet HUBBLE nos travaux sur l'élaboration des profils, en expérimentant en vraie grandeur les outils développés autour de la plateforme kTBS, pour la collecte (xCollector et TraceMe) et l'interrogation des traces (SPARE-LNC), et en les combinant avec les approches des laboratoires LIG et LIUM. Coupler ces outils d'élaboration du profil de l'apprenant avec $PERSUA2_{MOOC}$ et ASKER permettra d'offrir à un auteur de MOOC une plateforme complète lui donnant la possibilité de mettre en œuvre un enseignement personnalisé. L'ouverture du processus de personnalisation à l'apprenant est également une perspective de recherche intéressante dans le cadre des MOOCs.

Dans ces deux contextes dans lesquels nous travaillerons sur la personnalisation, la principale question que nous aurons à résoudre est celle de l'assistance que nous pouvons apporter à l'enseignant pour lui faciliter au maximum la tâche de mise en œuvre de la personnalisation. Cette assistance pourra être apportée d'une part en proposant à l'utilisateur des patrons (de requêtes, d'indicateurs, de profils, de règles d'affectation, d'exercices), et d'autre part en utilisant le raisonnement à partir de l'expérience tracée (Cordier et al., 2013). Une première thèse qui commence cet automne traitera par exemple dans le cadre des MOOCs de l'assistance que l'on peut apporter à un auteur utilisant ASKER, pour éliciter les connaissances du domaine au fur et à mesure qu'il définit des modèles d'exercices. Un autre sujet de thèse à venir serait d'étudier comment aider l'enseignant à améliorer le processus de personnalisation, à partir de l'observation de l'efficacité de ce processus sur l'apprentissage.

CONCLUSION ET PERSPECTIVES DE RECHERCHE

1 Bilan

Ce mémoire effectue une synthèse des recherches menées depuis 1997 (année de soutenance de ma thèse), sur la question générale de l'élicitation des connaissances dans le cadre de l'ingénierie des EIAH. Deux thématiques de recherche ont été traitées : l'enseignement de méthodes et la personnalisation des EIAH. Les contributions à l'élicitation des connaissances dans ces deux thématiques sont des modèles et outils permettant à un utilisateur humain d'éliciter les connaissances nécessaires au système pour proposer à l'apprenant un contenu pédagogique personnalisé, que ce soit un exercice, une rétroaction ou une recommandation.

Dans l'ensemble de ces travaux, l'approche commune choisie pour répondre à la problématique de l'élicitation des connaissances est de proposer, pour chacune des questions de recherche abordées, un méta-modèle des connaissances à acquérir, indépendant du domaine d'apprentissage. Ce méta-modèle permet de guider l'utilisateur humain (concepteur, expert, auteur, enseignant) dans la définition d'un modèle de connaissances, qui sera lui dépendant du domaine. Le méta-modèle proposé permet également de définir un moteur de raisonnement associé, capable d'exploiter tout modèle de connaissances conforme au méta-modèle. Ce moteur de raisonnement exploite le modèle de connaissances défini par l'utilisateur pour fournir des réponses aux requêtes induites par l'accompagnement par l'EIAH d'une tâche d'apprentissage.

La manière dont le méta-modèle guide l'élicitation des connaissances a évolué au cours du temps : dans mes premiers travaux, le méta-modèle spécifiait comment un expert devait représenter les connaissances qu'il élicitait ; actuellement nous concevons des interfaces fondées sur les méta-modèles pour assister le processus d'acquisition des connaissances auprès de l'utilisateur.

Les conclusions des parties A et B constituent un bilan des contributions apportées respectivement à l'enseignement de méthodes et à la personnalisation des EIAH. En ce qui concerne l'enseignement de méthodes, les architectures proposées, rassemblant méta-modèles et moteurs de raisonnement, permettent de définir, dans un domaine donné, une méthode de résolution de problèmes et les connaissances destinées à accompagner l'élève dans son apprentissage de la méthode. Pour ce qui est de la personnalisation des EIAH, nous avons proposé des méta-modèles et des outils fondés sur ces méta-modèles, outils destinés à un utilisateur ne possédant pas forcément de compétences poussées en informatique, comme un enseignant ou un auteur de MOOC. Ces outils lui permettent de mettre en place un processus de personnalisation complet, en définissant d'une part quels indicateurs calculer à partir des traces d'interactions des apprenants, en définissant d'autre part des modèles d'exercices permettant la génération d'activités répondant à des besoins spécifiques, et en précisant enfin selon quelle stratégie affecter des exercices adaptés au profil de chaque apprenant.

La **Figure 54** synthétise les résultats de ces travaux de recherche en les replaçant dans le temps. Le lecteur pourra trouver des précisions sur les contributions mentionnées dans les conclusions des parties A et B, dans lesquelles il est également mentionné les chapitres

décrivant ces contributions. La **Figure 54** fait également apparaître les collègues et étudiants qui ont contribué à ces recherches, et les projets collaboratifs financés qui ont permis de les mener.

Ce mémoire montre les liens existant dans mes travaux entre les deux thématiques de recherche que sont l'enseignement de méthodes et la personnalisation des EIAH. En effet, les travaux menés dans le cadre du projet AMBRE sur la génération de problèmes (architecture GenAMBRE) ou sur l'élaboration de profils d'apprenants ont motivé la généralisation des approches choisies et ont conduit à effectuer des recherches proposant des modèles indépendants de l'EIAH à personnaliser. L'EIAH AMBRE-add permet inversement de tester certains outils développés pour la personnalisation des EIAH, comme Adapte (GEPPETO-S et PERSUA2) ou SPARE-LNC.

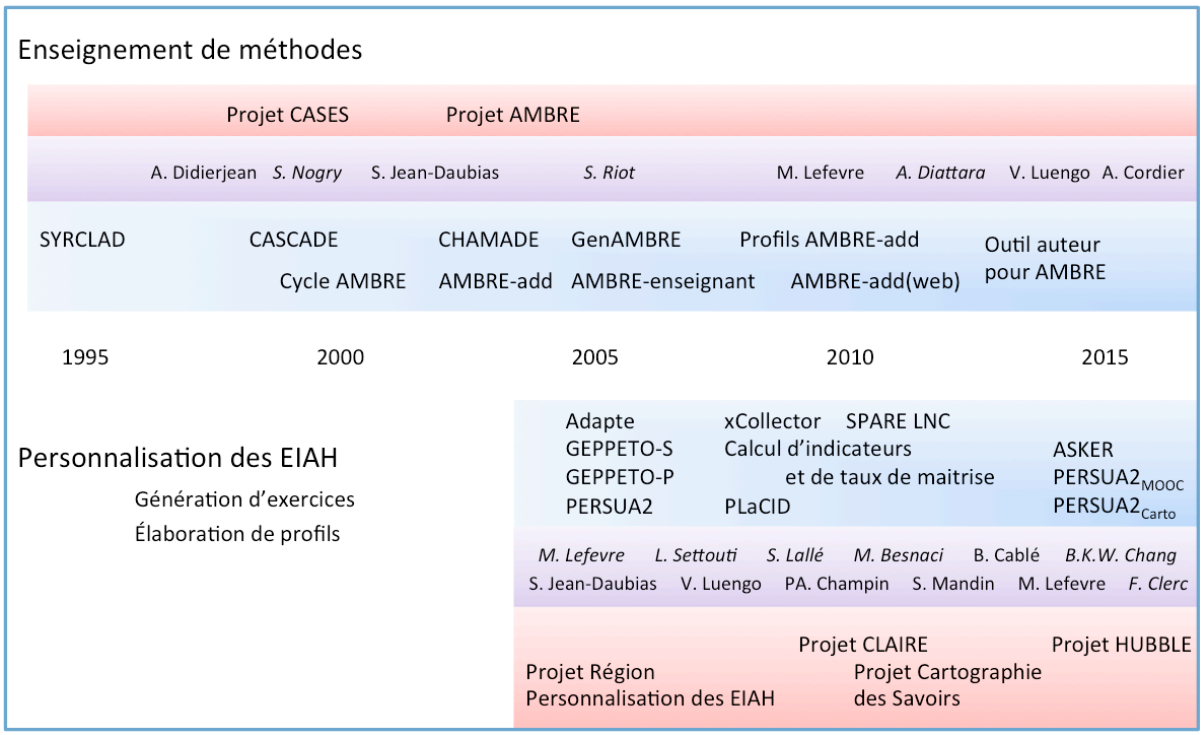

Figure 54. Synthèse des travaux de recherche sur l'enseignement de méthodes et la personnalisation des EIAH. *Les contributions sont indiquées sur fond bleu, les projets collaboratifs financés sur fond rouge, et les collègues impliqués sur fond violet (les doctorants et étudiants de master encadrés apparaissent en italique).*

2 Perspectives

Les conclusions des parties A et B décrivent respectivement les perspectives de recherche envisagées, pour l'enseignement de méthodes et pour la personnalisation des EIAH.

En ce qui concerne **l'enseignement de méthodes**, notre objectif est de concevoir un outil auteur permettant à un utilisateur souhaitant élaborer un EIAH AMBRE de constituer les systèmes à base de connaissances sous-jacents aux EIAH AMBRE. Nous souhaitons permettre à cet utilisateur de définir les connaissances à l'aide d'exemples que le système généralisera. Il s'agit de mettre en œuvre un processus d'acquisition interactive des

connaissances. L'approche *centrée connaissances* du projet AMBRE s'inscrit dans le même courant que celle des *Model Tracing Tutors* développés à l'Université Carnegie Mellon à Pittsburgh, même si les modèles cognitifs sous-jacents sont très différents. Le succès de ces tuteurs intelligents et des outils auteurs CTAT permettant de les créer renforce ma conviction qu'une approche *centrée connaissances* est souhaitable en EIAH. Le modèle ACT-R sur lequel se fondent les EIAH développés à Pittsburgh considère que la tâche de résolution de problèmes s'effectue pas à pas, un élément de connaissance étant mobilisé à chaque pas de résolution, et que l'ensemble des connaissances du domaine peut s'exprimer sous forme de règles de production. Ce modèle assez restrictif limite les domaines pour lesquels on peut concevoir des *Model Tracing Tutors*. L'enseignement de méthodes propose un modèle cognitif dans lequel le sujet raisonne globalement sur le problème à résoudre. C'est également un modèle restrictif puisqu'il s'applique uniquement aux domaines dans lesquels on peut établir une classification des problèmes à résoudre, mais il pourrait compléter l'approche de Carnegie Mellon. Un des objectifs de la thèse d'Awa Diattara sur la conception d'un outil auteur pour AMBRE est d'intégrer ce travail à la plateforme CTAT.

Pour ce qui est de la **personnalisation des EIAH**, nos perspectives s'inscrivent dans deux contextes particulièrement intéressants par rapport à cette problématique : celui des MOOCs et celui de la Cartographie des Savoirs.

Les **MOOCs** forment en effet un bon terrain, que ce soit pour l'élaboration des profils d'apprenants à partir des traces, pour la génération semi-automatique d'exercices ou pour la personnalisation des activités à partir des profils. En effet, la diversité des publics rend le besoin de personnalisation particulièrement prégnant, comme le montrent les publications récentes traitant de la personnalisation des MOOCs (Collet, 2014; Miranda et al., 2014). De plus, la masse des étudiants à former rend nos approches semi-automatiques particulièrement pertinentes. Les travaux à venir dans ce cadre (projet HUBBLE) ont pour objectif de mutualiser les résultats de nos travaux et de les confronter à une utilisation en conditions réelles. Des recherches commencent également (thèse de B. Kong Win Chang) sur la manière dont on peut exploiter les traces de l'enseignant utilisant les outils que nous lui proposons pour l'assister dans sa tâche, en mêlant acquisition interactive de connaissances et découverte de connaissances.

Une des faiblesses des MOOCs réside dans le manque de représentation des connaissances du domaine. C'est pourquoi notre collaboration avec le LIG sur la **Cartographie des Savoirs** est, au-delà du projet en cours, une perspective également très intéressante, car la sémantique apportée par le référentiel de compétences Ontoprax (Chaachoua et al., 2013) nous permet d'aller plus loin dans les modèles que nous proposons, que ce soit pour l'élaboration des profils d'apprenants, pour la génération semi-automatique d'exercices ou pour la personnalisation des activités. Le peuplement de l'ontologie Ontoprax est un travail très lourd pour les didacticiens, mais il me semble que ce modèle de représentation des connaissances proposé par le LIG pourrait servir de socle pour la conception d'EIAH centrés connaissances au niveau Rhône-Alpin, comme le modèle ACT-R l'a été pour les *Model Tracing Tutors* à Carnegie Mellon University.

Le projet AMBRE sur l'enseignement de méthodes continuera à constituer l'un des premiers terrains d'évaluation de nos travaux sur la personnalisation des EIAH. Ainsi, en nous fondant sur les méta-modèles décrivant les connaissances nécessaires pour un EIAH AMBRE, nous pourrons proposer des patrons de modèles de profils pour les EIAH AMBRE, en utilisant les notions de classes de problèmes et d'étapes de résolution. Il sera également possible de

définir des patrons de requêtes sur les traces et des patrons de méthodes de calcul d'indicateurs, fondées sur les éléments des méta-modèles de connaissances de AMBRE, afin d'alimenter les profils d'apprenants. Des patrons de stratégies de personnalisation fondées sur la notion de classe de problèmes pourraient également être pertinents. L'ensemble de ces patrons de modèles de connaissances, adaptés au projet AMBRE, permettront d'une part de faciliter la personnalisation des EIAH AMBRE issus de l'outil auteur en cours de définition, et d'autre part d'évaluer l'évolution de notre approche visant à faciliter l'élicitation de connaissances *via* la définition de patrons de modèles de connaissance.

3 Quelle évolution pour l'élicitation des connaissances en EIAH ?

Pour la plupart des questions de recherche étudiées, l'approche décrite dans ce mémoire consiste à faire intervenir l'enseignant ou le concepteur pour éliciter les connaissances nécessaires à l'EIAH, alors que d'autres approches préfèrent utiliser des techniques d'IA comme l'apprentissage automatique pour découvrir ces connaissances. Par exemple, nous proposons aux auteurs de MOOCs de définir des stratégies de personnalisation, alors que le projet POEMs (Collet et al., 2013) s'appuie sur des algorithmes de colonies de fourmis pour construire un parcours adapté au profil d'un apprenant. Même si les techniques d'IA peuvent s'avérer efficaces, je suis toujours convaincue qu'il est primordial d'impliquer l'enseignant dans la conception ou la configuration des EIAH. En effet, à l'heure d'internet, des réseaux sociaux et de la connaissance partagée, nombreux sont les enseignants à souhaiter concevoir leurs ressources pédagogiques (Caron, 2007) ou réutiliser celles créées par d'autres (Quentin & Bruillard, 2009). Dans les travaux qui sont décrits dans ce mémoire, l'enseignant est souvent le concepteur d'un EIAH, d'une technique de diagnostic, l'auteur de modèles d'exercices ou de MOOCs, c'est-à-dire un enseignant prêt à s'impliquer et à passer du temps à éliciter des connaissances. L'enseignant souhaitant utiliser un EIAH ou une ressource pédagogique en classe avec ses élèves n'aura pas la possibilité de consacrer un tel temps à éliciter des connaissances, mais pourra utiliser AMBRE-enseignant, ou utiliser des modèles d'exercices pour générer des exercices, avant de créer des modèles d'exercices lui-même, sans pour autant aller jusqu'à éliciter les connaissances du domaine qui auront pu être définies par un autre enseignant. Il existe ainsi un continuum dans l'implication des enseignants utilisant nos outils. Dans certains contextes comme celui des MOOCs, impliquer l'apprenant dans la conception du contenu pédagogique est également à envisager. En effet, l'apprenant pourrait intervenir dans le processus de personnalisation en modifiant la stratégie de personnalisation choisie pour lui par l'enseignant. La création de modèles d'exercices destinés à des pairs pourrait également être une activité pédagogique très intéressante, évaluant la compréhension des concepts du domaine nécessaires à la conception d'exercices.

Les outils que nous proposons aux utilisateurs qui ont à éliciter des connaissances sont fondés sur des méta-modèles qui sont indépendants du domaine à enseigner. La généricité est un élément très important de tous les travaux menés depuis ma thèse, parce qu'elle permet d'appliquer les solutions apportées aux questions de recherche à des contextes variés. Cette indépendance au domaine et au contexte peut néanmoins s'avérer gênante pour l'utilisateur qui manque de repère et ne sait pas forcément comment commencer à éliciter des connaissances. C'est pourquoi nous proposons de plus en plus des versions spécialisées des méta-modèles définis, adaptés à des contextes particuliers, comme par

exemple avec PERSUA2$_{MOOC}$, ou des patrons (par exemple de méthodes de calcul d'indicateurs) que l'utilisateur doit instancier pour définir ses modèles.

Si dans mes travaux de thèse ou ceux des années qui ont suivi les méta-modèles proposés étaient destinés à préciser à un expert comment décrire les connaissances attendues, dans nos travaux des dix dernières années, ces méta-modèles ont servi à concevoir des interfaces permettant à un utilisateur ne maitrisant pas nécessairement l'informatique d'éliciter les connaissances attendues. L'élicitation des connaissances dans mes travaux a ainsi évolué de la représentation à l'acquisition des connaissances. Dans les travaux en cours et à venir, cette acquisition des connaissances devient de plus en plus interactive et évolue aussi vers la découverte de connaissances. Par exemple, dans l'outil auteur pour AMBRE, nous envisageons d'inviter l'utilisateur à décrire des exemples que le système généralisera pour proposer à l'utilisateur une connaissance à valider. Nous envisageons également d'utiliser l'analyse des traces d'interaction d'un enseignant qui définit des modèles d'exercices pour découvrir des connaissances du domaine que le système lui proposera pour validation. La même approche pourrait servir pour concevoir ou améliorer des stratégies de personnalisation.

Comme le dit Vanda Luengo (Luengo, 2009, p. 45), il est nécessaire de trouver un équilibre entre un modèle centré *expert* et un modèle centré *données*. On retrouve cette distinction dans la différence entre la communauté LAK (Learning Analytics and Knowledge) et la communauté EDM (Educational Data Mining). Ces deux communautés de recherche ont des objectifs et des objets d'études similaires, mais des approches différentes. La première cherche à analyser les données pour proposer à l'utilisateur humain des outils de décision, alors que la seconde s'intéresse davantage à la découverte automatique des connaissances et à une prise de décision par le système. Mon approche est ainsi plus proche de celle de la communauté LAK, que ce soit pour l'analyse des traces ou pour d'autres questions de recherche.

Ma réticence vis à vis de l'apprentissage automatique vient du fait que la fiabilité des connaissances apprises n'est pas totale, or il ne me paraît pas admissible dans le contexte des EIAH que le système renvoie à l'apprenant une connaissance erronée. Il est donc nécessaire que les connaissances apprises soient validées par un expert, ce qui nécessite que le processus d'apprentissage des connaissances soit compréhensible pour l'expert. De même, le fait qu'il est presque impossible d'expliquer à un enseignant comment a été construite une connaissance apprise automatiquement par le système ne facilite pas la confiance et l'appropriation des EIAH par les enseignants, qui pour la plupart ont encore une réaction méfiante vis-à-vis de ces outils.

Les techniques d'apprentissage automatique peuvent néanmoins être efficaces quand elles sont intégrées à un processus interactif d'élicitation des connaissances, comme nous l'avons constaté dans le travail de thèse de Sébastien Lallé. Comme G. Siemens et R. Baker qui défendent un rapprochement entre les communautés LAK et EDM (Siemens & Baker, 2012), j'ai la volonté d'intégrer davantage les techniques d'apprentissage automatique à mes travaux, mais en restant dans un processus interactif d'élicitation des connaissances. N'étant pas moi-même chercheur en apprentissage automatique, je souhaite collaborer avec des collègues dont c'est la spécialité. Nous avons par exemple commencé une collaboration avec les laboratoires LIMSI (Laboratoire d'Informatique pour la Mécanique et les Sciences de

l'Ingénieur) et LIP6 (Laboratoire d'Informatique de Paris 6) dans le cadre de nos travaux sur la génération semi-automatique d'exercices. En effet, des techniques issues de l'analyse de la langue naturelle et de l'apprentissage automatique pourraient être utilisées pour exploiter des supports de cours afin de générer des questions, diagnostiquer les réponses de l'apprenant, et découvrir les connaissances du domaine nécessaires aux générateurs d'exercices de l'outil auteur ASKER.

RÉFÉRENCES BIBLIOGRAPHIQUES

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, *7*(1), 39–59.

Ainsworth, S. (2000). Redeem : ITS authoring tools and human teaching. In *Workshop on Modelling Human Teaching Tactics - ITS2000*. Montréal, Canada.

Alamargot, D. (2001). L'acquisition des connaissances. In C. Golder & D. Gaonac'h (Eds.), *Enseigner à des adolescents. Manuel de Psychologie* (pp. 78–113). Hachette Education.

Aleven, V., McLaren, B., Sewall, J., & Koedinger, K. R. (2006). The cognitive tutor authoring tools (CTAT): Preliminary evaluation of efficiency gains. In *Intelligent Tutoring Systems* (pp. 61–70). Taiwan: Springer.

Aleven, V., Mclaren, B., Sewall, J., & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, *19*(2), 105–154.

Allen, P., & Frost, S. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, *26*(11), 832–843.

Anderson, J. (1983). *The architecture of cognition*. Cambridge, Harvard University Press.

Anderson, J., Boyle, C., & Yost, G. (1985). The Geometry Tutor. In *International Joint Conferences on Artificial Intelligence*. Los Angeles, USA.

Anderson, J., & Pelletier, R. (1991). A development system for model tracing tutors. In *International Conference in the Learning Sciences*. Northwestern University, USA.

Anderson, J. R., Boyle, C., & Yost, G. (1985). The Geometry Tutor. In *International Joint Conferences on Artificial Intelligence* (pp. 1–7).

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive Tutors : Lessons Learned. *Journal of the Learning Sciences*, *4*(2), 167–207.

Aussenac-Gilles, N. (2005). *Méthodes ascendantes pour l'ingénierie des connaissances*. Mémoire d'HDR, Université Toulouse 3.

Bachimont, B. (2004). *Arts et sciences du numérique : Ingénierie des connaissances et critique de la raison computationnelle*. Mémoire d'HDR, Université de Technologie de Compiègne.

Baker, R., Corbett, A., Koedinger, K. R., Evenson, S., Roll, I., Wagner, A., … Beck, J. (2006). Adapting to when students game an intelligent tutoring system. In *Intelligent Tutoring Systems* (pp. 392–401). Taiwan: Springer.

Baulac, Y. (1990, February 7). *Un micromonde de géométrie, Cabri-géomètre*. Thèse de doctorat, Université Joseph Fourrier - Grenoble 1.

Besnaci, M., Guin, N., & Champin, P.-A. (2014). *Importation de traces existantes dans un système de gestion de bases de traces*. Rapport de recherche LIRIS.

Bielaczyc, K., Pirolli, P., & Brown, A. L. (1995). Training in self explanation and self- regulation strategies: investigating the effect of knowledge acquisition activities on problem solving. *Journal of Educational Psychology*, *86*, 122–133.

Blessing, S. B. (1997). A programming by demonstration authoring tool for model-tracing tutors. *International Journal of Artificial Intelligence in Education*, *8*(3-4), 233–261.

Blessing, S., & Ross, B. (1996). Content effects in problem categorization and problem solving. *Journal of Experimental Psychology : Learning, Memory and Cognition*, *23*, 932–945.

Bouhineau, D., Luengo, V., Mandran, N., Ortega, M., & Waleman, C. (2013). Conception et mise en place d'un entrepôt de traces et processus de traitement EIAH : UnderTracks. In *Environnements Informatiques pour l'Apprentissage Humain* (pp. 41–42). Toulouse, France.

Bouhineau, D., & Nicaud, J.-F. (2006). Aplusix, un EIAH de l'algèbre. In M. Grandbastien & J.-M. Labat (Eds.), *Environnements informatiques pour l'apprentissage humain (Traité IC2, série Cognition et traitement de l'information)* (pp. 333–350).

Breuker, J., & Velde, W. Van de. (1994). *CommonKADS library for expertise modelling: reusable problem solving components*. IOS Press, Amsterdam.

Bruillard, E., Delozanne, E., Leroux, P., Delannoy, P., Dubourg, X., Jacoboni, P., … Teutsch, P. (2000). Quinze ans de recherche informatique sur les sciences et techniques éducatives au LIUM . *Sciences et Techniques Éducatives*, *7*(1), 87–145.

Brusilovsky, P., Schwarz, E., & Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. In *Intelligent Tutoring Systems* (Vol. 1086, pp. 261–269). Berlin: Springer Verlag.

Caron, P. (2007). Bricoles: une approche dispositive des applications Web 2.0 utilisables pour enseigner. In *Environnements Informatiques pour l'Apprentissage Humain* (pp. 137–142). Lausane, Suisse.

Carrol, W. (1994). Using worked examples as an instructional support in the algebra classroom. *Journal of Educational Psychology*, *86*, 360–367.

Cauzinille-Marmèche, E., & Didierjean, A. (1999). Raisonnement par analogie et généralisation des connaissances. In G. Netchine-Grynberg (Ed.), *Développement et fonctionnement cognitif: vers une intégration* (pp. 125–152). Presses universitaires de France.

Chaachoua, H. (2010). *La praxéologie comme modèle didactique pour la problématique EIAH.Etude de cas : la modélisation des connaissances des élèves.* Mémoire d'HDR, Université Joseph Fourrier, Grenoble.

Chaachoua, H., Ferraton, G., & Desmoulins, C. (2013). Utilisation du modèle praxéologique de référence dans un environnement informatique d'apprentissage humain. In *4ème Congrès International de la Théorie Anthropologique du Didactique*. Toulouse, France.

Champin, P., Mille, A., & Prié, Y. (2013). Vers des traces numériques comme objets informatiques de premier niveau : une approche par les traces modélisées. *Intellectica*, *59*(1).

Chi, M., Bassok, M., Lewis, M., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, *13*, 145–182.

Chi, M., Feltovich, P., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, *5*(2), 121–152.

Choquet, C. (2007). *Ingénierie et réingénierie des EIAH : l'approche REDiM*. Mémoire d'HDR, Université du Maine, Le Mans.

Choquet, C., & Iksal, S. (2007). Modélisation et construction de traces d'utilisation d'une activité d'apprentissage: une approche langage pour la réingénierie d'un EIAH. *Sciences et Technologies de l'Information et de La Communication Pour l'Education et La Formation*, *14*.

Clerc, F., Lefevre, M., Guin, N., & Marty, J.-C. (2014). *Mise en Place de la Personnalisation Dans le Cadre des MOOCs*. Rapport de recherche LIRIS.

Cogne, A., David, J.-P., & Lacombe, C. (1998). Production d'exercices hypermédias et mise en œuvre pédagogique. In *8èmes Journées Informatique et Pédagogie des Sciences Physiques*. Montpellier.

Collet, P. (2014). POEM. Retrieved from http://audiovideocast.unistra.fr/avc/courseaccess?id=9443

Collet, P., Louça, J., Johnson, J., Bourgine, P., Diouf, I., Parrend, P., … Cointet, J. (2013). *Personalised Open Education for the Masses (POEM) for a 4P Education within the CS-DC UNESCO UniTwin*.

Cooper, G., & Sweller, J. (1987). The effects of schema acquisition and rule automation on mathematical problem-solving transfer. *Journal of Educational Psychology*, *79*, 347–362.

Corbett, A., & Anderson, J. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253–278.

Corbett, A. T., Koedinger, K. R., & Anderson, J. (1997). Intelligent tutoring systems. In T. K. Helander & P. Landauer (Eds.), *Handbook of Human-Computer Interaction*. Elsevier Science.

Cordier, A., Lefevre, M., Champin, P.-A., Georgeon, O., & Mille, A. (2013). Trace-Based Reasoning — Modeling Interaction Traces for Reasoning on Experiences. In *Florida Artificial Intelligence Research Society*. St. Pete Beach, USA.

Cordier, A., Lefevre, M., Jean-Daubias, S., & Guin, N. (2010). Concevoir des assistants intelligents pour des applications fortement orientées connaissances : problématiques , enjeux et étude de cas. In *21èmes Journées Francophones d'Ingénierie des Connaissances*. Nîmes, France.

Cristea, A. (2005). Authoring of Adaptative Hypermedia. *Educational Technology & Society*, *8*(3), 6–8.

Cristea, A., & Calvi, L. (2003). The three Layers of Adaptation Granularity. In *User Modeling* (pp. 4–14). Johnstown, USA: Springer.

Cummins, D. (1992). Role of Analogical Reasoning in the Induction of Problem Categories. *Journal of Experimental Psychology : Learning, Memory, and Cognition*, *5*, 1103–1124.

d'Ham, C., & Girault, I. (2009). Scaffolding the students' activity of experimental design with a dedicated software: copex-chimie. In *European Science Education Research Association*.

David, J., Guilloux, C., & Flament, A. (2002). A learning objects generator with xml-xslt technology. In *TICE2002, session ARIADNE*.

De Jong, T., & Ferguson-Hessler, M. (1996). Types and qualities of knowledge. *Educational Psychologist*, *31*(2), 105–113.

Diagne, F. (2009). *Instrumentation de la supervision de l'apprentissage par la réutilisation d'indicateurs : Modèles et Architecture*. Thèse de doctorat, Université Joseph Fourrier, Grenoble.

Didierjean, A. (2001). Apprendre à partir d'exemples: Abstraction de règles et/ou mémoire d'exemplaires ? *L'Année Psychologique*, *101*, 325–348.

Didierjean, A. (2003). s Case-based Reasoning a Source of Knowledge Generalization ? *European Journal of Cognitive Psychology*, *15*, 435–453.

Didierjean, A., & Cauzinille-Marmèche, E. (1997). Eliciting self- explanations improves problem solving: What processes are involved ? *Current Psychology of Cognition*, *13*(3), 325–351.

Dimitrakopoulou, A. (2004). *State of the art on Interaction and Collaboration Analysis*.

Djouad, T., Mille, A., & Reffay, C. (2009). Ingénierie des indicateurs d'activités à partir de traces modélisées pour un Environnement Informatique d'Apprentissage Humain. *Sciences et Technologies de l'Information et de La Communication Pour l'Education et La Formation*, *16*.

FOVEA. (2014). http://www.univ-lyon1.fr/formation/fovea-un-mooc-d-anatomie-base-sur-la-technologie-3d-761051.kjsp#.VCL01it_tg1.

Gendron, E. (2010). *Cadre conceptuel pour l'élaboration d'indicateurs de collaboration à partir des traces d'activité*. Thèse de doctorant, Université Claude Bernanrd - Lyon 1.

Gentner, D., Rattermann, M., & Forbus, K. (1993). The roles of similarity in transfer: Separating retrievability from inferential soundness. *Cognitive Psychology*, *25*(4), 525–575.

Gick, M., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology*, *12*, 306–355.

Gick, M., & Holyoak, K. J. (1983). Schema Induction and Analogical Transfer. *Cognitive Psychology*, *15*, 1–38.

Gick, M., & Paterson, K. (1992). Do contrasting examples facilitate schema induction and analogical transfer? *Canadian Journal of Psychology*, *46*, 539–550.

Ginon, B. (2014). *Modèles et outils génériques pour mettre en place des systèmes d'assistance épiphytes*. Thèse de doctorat, Université de Lyon.

Ginon, B., Jean-Daubias, S., & Lefevre, M. (2011). Evolutive learners profiles. *ED-MEDIA 2011 - World Conference on Educational Multimedia, Hypermedia & Telecommunications*.

Grandbastien, M. (1999). Teaching Expertise is at the Core of ITS Research. *International Journal of Artificial Intelligence in Education*, *10*, 335–349.

Greeno, J., & Riley, M. (1987). Processes and development of understanding, Metacognition, motivation and understanding. In F. E. Weinert & R. H. Kluwe (Eds.), *Metacognition, motivation, and understanding* (pp. 289–313). Lawrence Erlbaum.

Guin, N., & Lefevre, M. (2013). From a customizable ITS to an adaptive ITS. In *Artificial Intelligence in Education* (pp. 141–150). Memphis, USA.

Guin, N., Lefevre, M., & Jean-Daubias, S. (2011). Personnalisation de l'apprentissage dans l'EIAH AMBRE-add. In *Atelier "Personnalisation de l'apprentissage : quelles approches pour quels besoins ?", EIAH 2011*. Mons, Belgique.

Guin-Duclosson, N., Jean-Daubias, S., & Nogry, S. (2002). The AMBRE ILE: How to Use Case-Based Reasoning to Teach Methods. In *Intelligent Tutoring Systems* (Vol. 2363, pp. 782–791). San Sebastian (Espagne) & Biarritz (France): Springer, Lecture Notes in Computer Science.

Guin-Duclosson, N., & Nova, N. (2002). Utiliser des connaissances abstraites ou contextualisées pour proposer différents types d'aide. In *Technologies de l'information et de la connaissance dans l'enseignement supérieur et l'industrie* (pp. 195–202). Lyon, France.

Holyoak, K., & Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory & Cognition*, *15*(4), 332–340.

Hsieh, P., Halff, H., & Redfield, C. (1999). Four easy pieces: Development systems for knowledge-based generative instruction. *International Journal of Artificial Intelligence in Education*, *10*, 1–45.

Jean-Daubias, S. (2004). De l'intégration de chercheurs, d'experts, d'enseignants et d'apprenants à la conception d'un EIAH. In *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie* (pp. 290–297). Compiègne, France.

Jean-Daubias, S. (2011). *Ingénierie des profils d'apprenants*. Mémoire d'HDR, Université Claude Bernanrd - Lyon 1.

Jean-Daubias, S., Eyssautier-Bavay, C., & Lefevre, M. (2009). Modèles et outils pour rendre possible la réutilisation informatique de profils d'apprenants hétérogènes. *Sciences et Technologies de l'Information et de La Communication Pour l'Education et La Formation*, *16*, 171–208.

Johns, J., Mahadevan, S., & Woolf, B. (2006). Estimating student proficiency using an item response theory model. In *Intelligent Tutoring Systems* (pp. 473–480). Taiwan: Springer.

Johnstone, A. (2003). *Effective practice in objective assessment*. *LTSN Physical Sciences Centre*. LTSN Physical Sciences.

Jong, T. De, & Joolingen, W. Van. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research*, *68*(2), 179–201.

Kleppe, A., Warmer, J., & Bast, W. (2003). *MDA explained: the model driven architecture: practice and promise*. Addison-Wesley.

Koedinger, K., Aleven, V., Hefferman, T., McLaren, B., & Hockenberry, M. (2004). Opening the door to non-programmers: Authoring intelligent tutor behavior by demonstration. In *Intelligent Tutoring Systems* (pp. 162–174). Maceio, Brésil: Springer.

Koedinger, K., Pavlik, J., Stamper, J., Nixon, T., & Ritter, S. (2011). Avoiding Problem Selection Thrashing with Conjunctive Knowledge Tracing. In *Educational Data Mining* (pp. 91–100). Eindhoven , The Netherlands.

Kolodner, J. (1993). *Case-Based Reasoning*. Morgan Kaufman.

Kong Win Chang, B., Guin, N., Lefevre, M., & Champin, P.-A. (2014). *Conception d'un langage d'interrogation des traces accessible à des non-informaticiens*. Rapport de recherche LIRIS.

Labat, J.-M. (2002). EIAH : quel retour d'informations pour le tuteur ? In *Technologies de l'information et de la connaissance dans l'enseignement supérieur et l'industrie* (pp. 81–87). Lyon, France.

Laforcade, P., Nodenot, .Thierry, & Sallaberry, C. (2005). Un langage de modélisation pédagogique basé sur UML. *Sciences et Technologies de l'Information et de La Communication Pour l'Education et La Formation*, *12*, 89–116.

Lallé, S. (2013). *Assistance à la construction et la comparaison de techniques de diagnostic des connaissances*. Thèse de doctorat, Université Joseph Fourier - Grenoble.

Lallé, S., Luengo, V., & Guin, N. (2012). Méthodologie d'assistance pour la comparaison de techniques de diagnostic des connaissances. In *Technologies de l'Information et de la Communication pour l'Enseignement* (pp. 11–19). Lyon, France.

Lallé, S., Luengo, V., & Guin, N. (2013). Assistance à la conception de techniques de diagnostic des connaissances. In *Environnements Informatiques pour l'Apprentissage Humain* (pp. 203–214). Toulouse, France.

Lallé, S., Mostow, J., Luengo, V., & Guin, N. (2013). Comparing Student Models in Different Formalisms by Predicting Their Impact on Help Success. In *Artificial Intelligence in Education* (pp. 161–170). Memphis, USA: Springer.

Lallé, S., Mostow, J., Luengo, V., Guin, N., Metah, L. I. G., Fourier, U. J., & Lyon, U. (2013). Comparaison de l'impact de techniques de diagnostic des connaissances sur l'apprentissage d'une stratégie d'aide. In *Journée EIAH&IA 2013*. Toulouse, France.

Leake, D. (1996). *Case-based reasoning: Experiences, lessons and future directions*. MIT Press.

Lefevre, M. (2009). *Processus unifié pour la personnalisation des activités pédagogiques : méta-modèle, modèles et outils*. Thèse de doctorat, Université Claude Bernard Lyon 1.

Lefevre, M., Guin, N., & Jean-Daubias, S. (2008). Adapte, a tool for the teacher to personalize activities. In *9th International Conference on Intelligent Tutoring Systems (ITS'2008)* (pp. 699–701). Montréal, Canada.

Lefevre, M., Guin, N., & Jean-Daubias, S. (2012). Personnaliser des activités pédagogiques de manière unifiée : une solution à la diversité des dispositifs. *Revue Des Sciences et Technologies de l'Information et de La Communication Pour l'Education et La Formation (STICEF)*, *19*.

Lefevre, M., & Jean-Daubias, S. (2012). Intégration de données hétérogènes : un exemple pour la constitution de profils d'apprenants. In *Technologies de l'Information et de la Communication pour l'Enseignement* (pp. 147–157). Lyon, France.

Lefevre, M., Jean-Daubias, S., & Guin, N. (2009). Generation of pencil and paper exercises to personalize learners' work sequences: typology of exercises and meta-architecture for generators. In *E-Learn 2009 (World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education)*. Vancouver, Canada.

Lefevre, M., Mille, A., Jean-Daubias, S., & Guin, N. (2009). A Meta-Model to Acquire Relevant Knowledge for Interactive Learning Environments Personalization. In *Adaptive 2009*. Athènes, Grèce.

Lenne, D. (2009). *Modélisation des connaissances et de l'interaction. Application aux Environnements Informatiques pour l'Apprentissage Humain*. Mémoire d'HDR, Université de Technologie de Compiègne.

Leroux, P. (2002). *Machines partenaires des apprenants et des enseignants - Étude dans le cadre d'environnements supports de projets pédagogiques*. Mémoire d'HDR, Université du Maine, Le Mans.

Luengo, V. (2009). *Les rétroactions épistémiques dans les Environnements Informatiques pour l'Apprentissage Humain*. Mémoire d'HDR, Université Joseph Fourrier - Grenoble 1.

Major, N. (1995). Modelling Teaching Strategies. *International Journal of Artificial Intelligence in Education*, *6*(2/3), 117–152.

Major, N., Ainsworth, S., & Wood, D. (1997). REDEEM: Exploiting symbiosis between psychology and authoring environments. *International Journal of Artificial Intelligence in Education*, *8*(3-4), 317–340.

Mandin, S., & Guin, N. (2014a). Basing learner modelling on an ontology of knowledge and skills. In *IEEE International Conference on Advanced Learning Technologies* (pp. 321–323). Athènes, Grèce.

Mandin, S., & Guin, N. (2014b). *Prise en compte d'une ontologie des savoirs dans la construction d'un profil d'apprenant*. Rapport de recherche LIRIS.

Marty, J.-C., & Mille, A. (2009). *Analyse de traces et personnalisation des environnements informatiques pour l'apprentissage humain*. *Traité IC2 : Informatique et systèmes d'information*. Hermès Sciences.

Mathern, B., & Cordier, A. (2014). Samotraces. https://github.com/bmathern/samotraces.js

Matsuda, N., Keiser, V., Raizada, R., Tu, A., Stylianides, G., Cohen, W., & Koedinger, K. R. (2010). Learning by teaching simstudent: technical accomplishments and an initial use with students. In *Intelligent Tutoring Systems* (pp. 317–326). Springer.

McAlpine, M. (2002). *Principles of assessment*. University of Luton.

Mille, A. (1998). *Mémoire d'HDR*. Université Lyon 1.

Minh Chieu, V., Luengo, V., Vadcard, L., & Tonetti, J. (2010). Student modeling in orthopedic surgery training: Exploiting symbiosis between temporal bayesian networks and fine-grained didactic analysis. *International Journal of Artificial Intelligence in Education*, *20*(3), 269–301.

Miranda, S., Mangioni, G., Orciuoli, F., Loia, V., & Salerno, S. (2014). The SIRET training platform: Facing the dropout phenomenon of MOOC environments. In *EMOOCs 2014* (pp. 107–113).

Mitrovic, A., Suraweera, P., Martin, B., Zakharov, K., Milik, N., & Holland, J. (2006). Authoring constraint-based tutors in ASPIRE. In *Intelligent Tutoring Systems* (pp. 41–50). Taiwan: Springer.

Montepare, J. (2005). A self-correcting approach to multiple choice tests. *Observer*, *18*(10).

Mostow, J., Aist, G., & Feltovich, K. F. & P. (2001). Evaluating tutors that listen: An overview of Project LISTEN. In *Smart Machines in Education* (pp. 169–234). Menlo Park, CA: MIT/AAAI Press.

Mostow, J., Beck, J., Bey, J., Cuneo, A., Sison, J., Tobin, B., & Valeri, J. (2004). Using automated questions to assess reading comprehension, vocabulary, and effects of tutorial interventions. *Technology, Instruction, Cognition and Learning*, *2*, 97–134.

Munro, A., Johnson, M., & Pizzini, Q. (1997). Authoring simulation-centered tutors with RIDES. *International Journal of Artificial Intelligence in Education*, *8*(3-4), 284–316.

Murray, T. (1998). Authoring knowledge-based tutors: Tools for content, instructional strategy, student model, and interface design. *The Journal of the Learning Sciences*, *7*(1), 5–64.

Murray, T. (2003a). An Overview of Intelligent Tutoring Systems Authoring Tools: Updated Analysis of the State of the Art. In Murray, Ainsworth, & Blessing (Eds.), *Authoring Tools for Advanced Technology Learning Environments* (pp. 493–546). Kluwer Academic Publisher.

Murray, T. (2003b). Eon: Authoring Tools for Content, Instructional Strategy, Student Model, and Interface Design. In Murray, Ainsworth, & Blessing (Eds.), *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher.

Murray, T. (2003c). Principles for Pedagogy-oriented Knowledge Based Tutor Authoring Systems: Lessons Learned and a Design Meta-Model. In Murray, Blessing, & Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher.

Nanard, M., & Nanard, J. (2002). EadGen, un environnement ouvert de production en ligne basé sur XML et XSL-T. In *TICE 2002, session ARIADNE*. Lyon, France.

Nicaud, J.-F. (1994). Modélisation en EIAO, les modèles d'APLUSIX. *Recherches En Didactique Des Mathématiques*, *14*(1.2), 76–112.

Nigro, J.-M. (1995). *La réalisation et la conception d'un générateur automatique de commentaires : le système GénéCom. Application au jeu du Tarot*. Thèse de doctorat, Université Paris 6.

Nkambou, R., Gauthier, G., & Frasson, C. (1997). Un modèle de représentation des connaissances relatives au contenu dans un système tutoriel intelligent. *Sciences et Techniques Educatives*, *4*(3), 299–330.

Nodenot, T. (2005). *Contribution à l'ingénierie dirigée par les modèles en EIAH : le cas des situations-problèmes coopératives*. Mémoire d'HDR, Université de Pau et des Pays de l'Adour.

Nogry, S. (2003). Comment faire apprendre des connaissances abstraites à partir d'exemples : application au projet AMBRE. In *EIAH 2003, session jeunes chercheurs* (pp. 67–70).

Nogry, S., & Didierjean, A. (2006). Apprendre à partir d'exemples : interactions entre présentation du matériel, activités des apprenants et processus cognitifs. *L'Année Psychologique*, *106*(1), 105–128.

Ohlsson, S. (1994). Constraint-based student modeling. In *Student modelling: the key to individualized knowledge-based instruction* (pp. 167–189). Springer Berlin Heidelberg.

OMG. (2008). Meta object facility specification 2.0. *Final Adopted Specification (November 2005)*.

Paquette, G. (2007). Graphical Ontology Modeling Language for Learning Environments. *Technology, Instruction, Cognition and Learning*, *5*, 133–168.

Pernin, J., Emin, V., & Guéraud, V. (2009). Intégration de la dimension utilisateur dans la conception de systèmes pour l'apprentissage: Scénarisation pédagogique dirigée par les intentions. *Ingénierie Des Systèmes d'Information*, *14*(3), 9–30.

Pham-Thi, N., Iksal, S., Choquet, C., & Klinger, E. (2009). Dcl4utl: Une proposition de langage de calcul déclaratif pour le processus d'analyse de traces d'apprentissage. In *Environnements Informatiques pour l'Apprentissage Humain* (pp. 29–36). Le Mans, France.

Piaget, J. (1975). *L'équilibration des structures cognitives. Problème central du développement.* Paris, France: PUF.

Pitrat, J. (1990). *Métaconnaissance, futur de l'intelligence artificielle*. Hermes.

Quentin, I., & Bruillard, E. (2009). Le fonctionnement de Sésamath: une étude exploratoire. In *Colloque Echanger pour apprendre en ligne*.

Quilici, J. H., & Mayer, R. E. (1996). Role of examples in how students learn to categorize statistics word problems. *Journal of Educational Psychology*, *88*, 144–161.

Quilici, J. H., & Mayer, R. E. (2002). Teaching students to recognize structural similarities between statistics word problems. *Applied Cognitive Psychology*, *16*, 325–342.

Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. *Cognitive Science*, *21*, 1–29.

Riley, M., Greeno, J., & Heller, J. (1983). Development of children's problem-solving ability in arithmetic. In H. P. Ginsburg (Ed.), *The development of mathematical thinking*. Academic Press.

Rogalski, M. (1990). Enseigner des méthodes en mathématiques. *Enseigner Autrement Les Mathématiques En Deug A Première Année, Bulletin Inter-Irem*, 65–79.

Rogalski, M. (1994). Les concepts de l'EIAO sont-ils indépendants du domaine? L'exemple de l'enseignement. *Recherches En Didactique Des Mathématiques*, *14*(1-2), 43–66.

Ross, B. H. (1987). This is Like That : the Use of Earlier Problems and the Separation of Similarity Effects. *Journal of Experimental Psychology : Learning, Memory and Cognition*, *13*, 629–639.

Ross, B. H. (1989). Distinguishing Types of Superficial Similarities : Different Effects on the Access and Use of Earlier Problems. *Journal of Experimental Psychology : Learning, Memory and Cognition*, *15*, 456–468.

Ross, B., & Kennedy, P. (1990). Generalizing from the use of earlier examples in problem solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *16*, 42–55.

Sander, E., & Richard, J.-F. (1997). nalogical Transfer as Guided by an Abstraction Process : the Case of Learning by Doing in Text Editor. *Journal of Experimental Psychology : Learning, Memory and Cognition*, *23*, 1459–1483.

Schoenfeld, A. (1985). *Mathematical problem solving*. New York, USA: Academic Press.

Selva, T. (2002). Génération automatique d'exercices contextuels de vocabulaire. In *Traitement Automatique du Langage Naturel* (pp. 185–194).

Settouti, L., Guin, N., & Luengo, V. (2011). Adaptable and Reusable Query Patterns for Trace-Based Learner Modelling. In *Sixth European Conference on Technology Enhanced Learning: Towards Ubiquitous Learning* (pp. 384–397). Palerme, Italie.

Settouti, L., Guin, N., Luengo, V., & Mille, A. (2010). A Trace-Based Learner Modelling Framework for Technology-Enhanced Learning Systems. In *ICALT 2010* (pp. 73–77). Sousse, Tunisia.

Settouti, L. S. (2011). *Systèmes à Base de Traces Modélisées : Modèles et Langages pour l'exploitation des traces d'Interactions*. Thèse de doctorat, Université Lyon1.

Shute, V. J., & Zapata-Rivera, J. D. (2007). Adaptive Technologies. In *Handbook of Research on Educational Communications and Technology* (pp. 277–294).

Siemens, G., & Baker, R. (2012). Learning analytics and educational data mining: towards communication and collaboration. In *International conference on Learning Analytics and Kowledge* (pp. 252–254). ACM.

Studer, R., Benjamins, V., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, *25*(1), 161–197.

Tchounikine, P. (2009). *Précis de recherche en ingénierie des EIAH*. en ligne sur le Web.

Tisseau, G. (1990). *Modélisation à partir d'un énoncé informel : Le système Modelis. Application à des exercices de thermodynamique*. Thèse de doctorat, Université Paris 6.

Van Marcke, K. (1998). GTE: An epistemological approach to instructional modeling. Instructional Science. *Instructional Science*, *26*(3-4), 147–191.

Vergnaud, G. (1982). A classification of cognitive tasks and operations of thought involved in addition and subtraction problems. In *Addition and subtraction: A cognitive perspective* (pp. 39–58). Lawrence Erlbaum.

Wagner-Menghin, M., Preusche, I., & Schmidts, M. (2013). The Effects of Reusing Written Test Items: A Study Using the Rasch Model. *International Scholarly Research Notes*.

Ward, M., & Sweller, J. (1990). Structuring effective worked examples. *Cognition and Instruction*, *7*, 1–39.

Weil-Barais, A., & Dubois, D. (1994). *L'homme cognitif*. Presses universitaires de France.

Willis, G. B., & Fuson, K. C. (1988). Teaching children to use schematic drawings to solve addition and subtraction word problems. *Journal of Educational Psychology*, *80*, 190–201.

Winograd, T. (2006). Shifting viewpoints: Artificial intelligence and human–computer interaction. *Artificial Intelligence*, *170*(18), 1256–1258.

Woolf, B. P. (2009). *Building Intelligent Interactive Tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.

Xu, Y., & Mostow, J. (2011). Using Logistic Regression to Trace Multiple Sub-skills in a Dynamic Bayes Net. In *Educational Data Mining* (pp. 241–246). Eindhoven, The Netherlands.

INDEX DES ILLUSTRATIONS

Figure 1 : Synopsis des questions de recherche 21

Figure 2 : Approche scientifique 25

Figure 3. Premières questions de recherche sur l'enseignement de méthodes 36

Figure 4. Outils pour la conception d'un EIAH AMBRE 37

Figure 5. Instanciation de l'approche générale présentée dans le chapitre 1 pour la question de l'élicitation des connaissances d'une méthode de résolution de problèmes. 41

Figure 6. L'architecture SYRCLAD 42

Figure 7. Schéma XML décrivant une classe de problèmes 44

Figure 8. Extrait des connaissances de classification pour les problèmes additifs 45

Figure 9. Instanciation de l'approche générale présentée dans le chapitre 1 pour l'élicitation de connaissances de résolution de problèmes combinant des classes et des cas. 55

Figure 10. Modèle des connaissances de l'expert pour les problèmes d'alignement aux échecs 59

Figure 11. Méta-modèles des connaissances contextualisées et leur utilisation dans CASCADE pour résoudre des problèmes. 61

Figure 12. Le cycle AMBRE : cycle du RàPC adapté au projet AMBRE 68

Figure 13. Présentation d'un problème-type résolu. 72

Figure 14. Étape de reformulation du problème 73

Figure 15. Étape de choix d'un problème-type 74

Figure 16. Résolution du problème par analogie 75

Figure 17. Exemple de fenêtre d'aide à deux niveaux pour l'étape d'adaptation. 83

Figure 18. Aide à l'analogie dans l'étape d'adaptation 84

Figure 19. Exemple de message d'explication généré à partir de la réponse de l'élève 85

Figure 20. Instanciation de l'approche générale présentée dans le chapitre 1 à la question de l'élicitation des connaissances permettant d'accompagner l'apprenant dans son apprentissage avec un EIAH AMBRE. 86

Figure 21. Les différentes bases de connaissances de l'architecture CHAMADE 87

Figure 22. Méta-modèle de connaissances permettant le diagnostic de la réponse de l'apprenant et la construction d'une rétroaction appropriée. 89

Figure 23. Architecture CHAMADE 89

Figure 24. Production d'informations pour l'interface 90

Figure 25. Production d'aide 91

Figure 26. Diagnostic des réponses et production d'explications 92

Figure 27. Instanciation de l'approche générale présentée dans le chapitre 1 à la question de l'élicitation de connaissances de génération de problèmes. 97

Figure 28. Utilisation de AMBRE-enseignant 97

Figure 29. L'écran de bilan du module de génération de problèmes de AMBRE-enseignant 102

Figure 30. L'architecture GenAMBRE 103

Figure 31. Instanciation de l'approche générale présentée dans le chapitre 1 à la question de l'acquisition des connaissances nécessaires à un EIAH AMBRE 109

Figure 32. Interface de construction du graphe de classification 113

Figure 33. Les systèmes à base de connaissances de AMBRE 116

Figure 34. Cycle de personnalisation des EIAH 123

Figure 35. Un méta-modèle des traces d'EIAH 130

Figure 36. Processus d'importation de traces externes dans kTBS. 131

Figure 37. Mise en correspondance avec xCollector entre un élément de la trace externe et un élément du modèle de traces. 132

Figure 38. Processus de définition de méthodes de calcul d'indicateurs permettant d'alimenter le profil 133

Figure 39. Processus d'adaptation d'un patron de MCI à un EIAH X 134

Figure 40. Processus d'interrogation des traces *via* SPARE-LNC 135

Figure 41. Interface graphique de SPARE-LNC permettant de formuler des requêtes conformes à la grammaire du langage. 136

Figure 42. Instanciation de l'approche présentée au chapitre 1 pour la définition de méthodes de calcul d'indicateurs 138

Figure 43. Exemple simplifié permettant de présenter les concepts d'Ontroprax 140

Figure 44. Instanciation de l'approche présentée au chapitre 1 pour l'estimation du taux de maîtrise de types de tâches par un apprenant 142

Figure 45. Instanciation de l'approche présentée au chapitre 1 aux questions de l'application à un domaine donné de techniques génériques de diagnostic et de la comparaison des performances de ces techniques instanciées au domaine. 145

Figure 46. L'approche GEPPETO 155

Figure 47. Architecture de l'outil de création d'exercices d'évaluation ASKER 159

Figure 48. Instanciation de l'approche présentée au chapitre 1 à la question de la génération d'exercices. 162

Figure 49. Instanciation de l'approche présentée au chapitre 1 à la question du pilotage d'EIAH existants pour générer ou choisir des activités. 166

Figure 50. Instanciation de l'approche présentée au chapitre 1 à la question de l'élicitation de stratégies de personnalisation 173

Figure 51. Processus de personnalisation fondé sur le modèle PERSUA2$_{MOOC}$..... 179

Figure 52. Processus de personnalisation en lien avec un référentiel des compétences..... 181

Figure 53. Contributions à la question de la personnalisation des EIAH 187

Figure 54. Synthèse des travaux de recherche sur l'enseignement de méthodes et la personnalisation des EIAH. 192

TABLES DES MATIÈRES DÉTAILLÉE

INTRODUCTION 7

PRÉSENTATION DU PLAN DU MÉMOIRE 9

CHAPITRE 1 : PROBLÉMATIQUE GÉNÉRALE ET THÉMATIQUES DE RECHERCHE 11

INTRODUCTION 12
1 DOMAINE D'ÉTUDE : DE QUELS EIAH PARLONS-NOUS ? 12
1.1 QUELLES SITUATIONS PÉDAGOGIQUES ? 12
Quels apprenants ? 12
Quels contextes ? 12
Quelles activités pour l'apprenant ? 13
1.2 QUEL RÔLE POUR L'ENSEIGNANT ? 13
1.3 DES EIAH CENTRÉS CONNAISSANCES 14
1.4 SYNTHÈSE 15
2 THÉMATIQUES DE RECHERCHE 15
2.1 ENSEIGNEMENT DE MÉTHODES 15
Comment représenter les connaissances d'une méthode pour permettre à un système d'appliquer cette méthode pour résoudre des problèmes ? 15
Comment résoudre des problèmes en utilisant à la fois des connaissances issues d'une méthode et des connaissances plus contextualisées ? 16
Comment enseigner une méthode de résolution de problèmes ? 16
Comment diagnostiquer les réponses de l'apprenant, et lui fournir aide et explications sur ses erreurs ? 16
Comment permettre à l'enseignant d'agir sur l'EIAH ? Comment générer des problèmes ? 17
Comment permettre à un enseignant de concevoir un EIAH AMBRE ? 17
2.2 PERSONNALISATION DES EIAH 17
Comment élaborer des profils d'apprenants en analysant les traces d'interactions ? 18
Comment générer des exercices ? 19
Comment personnaliser les exercices en fonction des profils d'apprenants ? 19
2.3 LIEN ENTRE LES DEUX THÉMATIQUES DE RECHERCHE 20
3 PROBLÉMATIQUE GÉNÉRALE : L'ÉLICITATION DES CONNAISSANCES 21
De quelles connaissances parle-t-on ? 22
4 APPROCHE SCIENTIFIQUE 25
4.1 S'APPUYER SUR DES MÉTA-MODÈLES POUR GUIDER L'ÉLICITATION DES CONNAISSANCES 25
4.2 QUELLE ASSISTANCE À CE PROCESSUS D'ÉLICITATION DES CONNAISSANCES ? 26
4.3 POSITIONNEMENT 26
4.4 DÉMARCHE DE RECHERCHE 28
SYNTHÈSE 30

PARTIE A : ENSEIGNEMENT DE MÉTHODES 33

INTRODUCTION 35

CHAPITRE A1 : ÉLICITER LES CONNAISSANCES D'UNE MÉTHODE DE RÉSOLUTION DE PROBLÈMES 39

INTRODUCTION 40
1 CADRE DES RECHERCHES 40
2 QUESTION DE RECHERCHE : ÉLICITER LES CONNAISSANCES D'UNE MÉTHODE DE RÉSOLUTION DE PROBLÈMES 41

3 CONTRIBUTION : L'ARCHITECTURE SYRCLAD 42
3.1 TROIS MÉTA-MODÈLES DE CONNAISSANCE 43
Connaissances de classification 43
Connaissances de reformulation 45
Connaissances de résolution 46
3.2 PROCESSUS DE RÉSOLUTION DE PROBLÈMES 46
3.3 ÉVALUATION DE LA PROPOSITION 47
CONCLUSION 48
SYNTHÈSE DES CONTRIBUTIONS 50
POUR EN SAVOIR PLUS 50
PUBLICATIONS LIÉES 50

CHAPITRE A2 : MODÉLISER UN PROCESSUS DE RÉSOLUTION DE PROBLÈMES MOBILISANT CONJOINTEMENT DES CLASSES ET DES CAS 53

INTRODUCTION 54
1 CADRE DES RECHERCHES : LE PROJET CASES 54
2 QUESTION DE RECHERCHE : MODÉLISER UN PROCESSUS DE RÉSOLUTION DE PROBLÈMES MOBILISANT CONJOINTEMENT DES CLASSES ET DES CAS 54
3 CONTRIBUTION : L'ARCHITECTURE CASCADE 56
3.1 EXPÉRIMENTATIONS PRÉLIMINAIRES 56
3.2 MÉTA-MODÈLES DE CONNAISSANCE 58
3.3 PROCESSUS DE RAISONNEMENT 61
Mémorisation d'un problème résolu 62
3.4 ÉVALUATION DE L'ARCHITECTURE 62
CONCLUSION 63
SYNTHÈSE DES CONTRIBUTIONS 64
POUR EN SAVOIR PLUS 64
PUBLICATIONS LIÉES 64

CHAPITRE A3 : AMBRE, ENSEIGNER UNE MÉTHODE À PARTIR D'EXEMPLES 65

INTRODUCTION 66
1 CADRE DES RECHERCHES : LE PROJET AMBRE 66
2 QUESTION DE RECHERCHE : COMMENT ENSEIGNER UNE MÉTHODE ? 67
3 CONTRIBUTION : LE CYCLE AMBRE 67
4 APPLICATIONS : LES EIAH AMBRE 69
4.1 PREMIER PROTOTYPE ET PREMIÈRE ÉVALUATION 69
4.2 ÉTUDES EN PSYCHOLOGIE COGNITIVE ET RECOMMANDATIONS POUR LE PROJET AMBRE 70
4.3 L'EIAH AMBRE-ADD 71
Le domaine des problèmes additifs 71
Présentation des problèmes résolus 72
Reformuler le problème 73
Choisir un problème-type 73
Adapter la solution du problème-type 74
Classer le problème 75
5 ÉVALUATION DU CYCLE AMBRE 75
CONCLUSION ET PERSPECTIVES 77
SYNTHÈSE DES CONTRIBUTIONS 78
POUR EN SAVOIR PLUS 78
PUBLICATIONS LIÉES 79

CHAPITRE A4 : ÉLICITER LES CONNAISSANCES PERMETTANT D'ACCOMPAGNER L'APPRENANT 81

INTRODUCTION 82

1 CADRE DES RECHERCHES : LE PROJET AMBRE 82
2 QUESTION DE RECHERCHE : ÉLICITER LES CONNAISSANCES PERMETTANT DE DIAGNOSTIQUER LES RÉPONSES DE L'APPRENANT, ET DE LUI FOURNIR AIDE ET EXPLICATIONS SUR SES ERREURS 82
2.1 ACCOMPAGNER L'APPRENANT DANS SA RÉSOLUTION DE PROBLÈMES 82
2.2 ÉLICITER LES CONNAISSANCES PERMETTANT D'ACCOMPAGNER L'APPRENANT 85
3 CONTRIBUTION : L'ARCHITECTURE CHAMADE 86
3.1 MÉTA-MODÈLES DE CONNAISSANCES 87
3.2 PROCESSUS DE RAISONNEMENT 90
Production d'informations pour l'interface 90
Production d'aide 91
Diagnostic des réponses et production d'explications 91
CONCLUSION ET PERSPECTIVES 92
SYNTHÈSE DES CONTRIBUTIONS 93
PUBLICATIONS LIÉES 93

CHAPITRE A5 : PERMETTRE À L'ENSEIGNANT D'ADAPTER L'EIAH À SES BESOINS 95

INTRODUCTION 96
1 CADRE DES RECHERCHES : LE PROJET AMBRE 96
2 QUESTIONS DE RECHERCHE : COMMENT PERMETTRE À L'ENSEIGNANT D'AGIR SUR L'EIAH ? COMMENT GÉNÉRER DES PROBLÈMES ? 96
3 CONTRIBUTION : AMBRE-ENSEIGNANT 98
4 CONTRIBUTION : GENAMBRE, VERS LA GÉNÉRATION DE PROBLÈMES 99
4.1 QUEL TYPE DE GÉNÉRATEUR POUR AMBRE ? 99
4.2 L'ENVIRONNEMENT DE GÉNÉRATION DE PROBLÈMES VU PAR L'ENSEIGNANT 100
4.3 L'ARCHITECTURE GENAMBRE 102
Méta-modèles de connaissances 103
Processus de raisonnement 104
5 ÉVALUATION DES CONTRIBUTIONS 105
CONCLUSION ET PERSPECTIVES 105
SYNTHÈSE DES CONTRIBUTIONS 106
POUR EN SAVOIR PLUS 106
PUBLICATIONS LIÉES 106

CHAPITRE A6 : VERS UN OUTIL AUTEUR POUR AMBRE 107

INTRODUCTION 108
1 CADRE DES RECHERCHES 108
2 QUESTION DE RECHERCHE : ACQUÉRIR LES CONNAISSANCES NÉCESSAIRES À UN EIAH AMBRE 108
3 ÉTAT DE L'ART 109
4 DÉMARCHE ENVISAGÉE 111
CONCLUSION ET PERSPECTIVES 113
PUBLICATIONS LIÉES 114

CONCLUSION ET PERSPECTIVES SUR LA PARTIE A 115

1 BILAN 115
2 PERSPECTIVES POUR LE PROJET AMBRE 116
3 VERS LA PERSONNALISATION DES EIAH 117

PARTIE B : PERSONNALISATION DES EIAH 121

INTRODUCTION 122

CHAPITRE B1 : ÉLABORER UN PROFIL DE L'APPRENANT À PARTIR DE L'ANALYSE DE SES TRACES 125

INTRODUCTION **127**
1 INTERROGER DES TRACES POUR CALCULER DES INDICATEURS **128**
1.1 CADRE DES RECHERCHES 128
1.2 QUESTIONS DE RECHERCHE 129
1.3 CONTRIBUTIONS 129
Importer des traces existantes au sein de kTBS 129
Analyser les traces pour élaborer des profils 132
Permettre à un utilisateur non informaticien d'interroger des traces 134
1.4 BILAN 138
2 ÉLABORER UN PROFIL DE L'APPRENANT EN SE FONDANT SUR UN RÉFÉRENTIEL DES CONNAISSANCES DU DOMAINE **139**
2.1 CADRE DES RECHERCHES 139
2.2 QUESTION DE RECHERCHE 139
2.3 CONTRIBUTIONS 142
2.4 ÉVALUATION ET PERSPECTIVES 143
3 INSTANCIER ET COMPARER DES TECHNIQUES GÉNÉRIQUES DE DIAGNOSTIC **144**
3.1 CADRE DES RECHERCHES 144
3.2 QUESTIONS DE RECHERCHE 144
3.3 CONTRIBUTIONS 145
3.4 ÉVALUATIONS 147
CONCLUSION ET PERSPECTIVES **148**
SYNTHÈSE DES CONTRIBUTIONS **149**
POUR EN SAVOIR PLUS **150**
PUBLICATIONS LIÉES **150**

CHAPITRE B2 : GÉNÉRER DES EXERCICES **153**

INTRODUCTION **154**
1 CADRE DES RECHERCHES **154**
2 QUESTIONS DE RECHERCHE : GÉNÉRER DES EXERCICES DE TYPES GÉNÉRIQUES OU SPÉCIFIQUES **155**
3 CONTRIBUTION : DES GÉNÉRATEURS D'EXERCICES DE TYPES GÉNÉRIQUES POUR UN OUTIL AUTEUR D'EXERCICES D'ÉVALUATION **156**
3.1 L'APPROCHE GEPPETO-P 157
3.2 ASKER : UN OUTIL AUTEUR D'EXERCICES D'ÉVALUATION 157
Diagnostic des réponses de l'apprenant 160
Connaissances du domaine 160
3.3 BILAN 161
4 CONTRIBUTION : AGIR SUR DES GÉNÉRATEURS D'EXERCICES INTÉGRÉS À DES EIAH EXISTANTS **163**
4.1 L'APPROCHE GEPPETO-S ET SON IMPLÉMENTATION DANS LE LOGICIEL ADAPTE 163
4.2 ILLUSTRATION SUR L'EIAH AMBRE-ADD 164
4.3 BILAN 165
CONCLUSION ET PERSPECTIVES **166**
SYNTHÈSE DES CONTRIBUTIONS **167**
POUR EN SAVOIR PLUS **167**
PUBLICATIONS LIÉES **168**

CHAPITRE B3 : PILOTER LA GÉNÉRATION D'EXERCICES À PARTIR DES PROFILS **171**

INTRODUCTION **172**
1 CADRE DES RECHERCHES **172**
2 QUESTIONS DE RECHERCHE : COMMENT PERMETTRE À L'ENSEIGNANT D'ÉLICITER UNE STRATÉGIE DE PERSONNALISATION ? COMMENT APPLIQUER UNE TELLE STRATÉGIE POUR PROPOSER DES ACTIVITÉS PERSONNALISÉES ? **173**
3 CONTRIBUTIONS : LE MODÈLE PERSUA2 ET SES APPLICATIONS **174**
3.1 LE MODÈLE PERSUA2 174

Processus de personnalisation 175
Évaluation des contributions 175
Positionnement 176
3.2 ILLUSTRATION SUR L'EIAH AMBRE-ADD 176
Synthèse 177
3.3 ADAPTATION DU MODÈLE PERSUA2 POUR LES MOOCS 178
3.4 ÉVOLUTION DU MODÈLE PERSUA2 DANS LE CADRE DU PROJET "CARTOGRAPHIE DES SAVOIRS" 180
CONCLUSION ET PERSPECTIVES 182
SYNTHÈSE DES CONTRIBUTIONS 184
POUR EN SAVOIR PLUS 184
PUBLICATIONS LIÉES 185

CONCLUSION ET PERSPECTIVES SUR LA PARTIE B 187

1 BILAN 187
2 PERSPECTIVES 189

CONCLUSION ET PERSPECTIVES DE RECHERCHE 191

1 BILAN 203
2 PERSPECTIVES 204
3 QUELLE ÉVOLUTION POUR L'ÉLICITATION DES CONNAISSANCES EN EIAH ? 206

RÉFÉRENCES BIBLIOGRAPHIQUES 197

INDEX DES ILLUSTRATIONS 211