

---

## **AMBRE-enseignant : un module partenaire de l'enseignant pour créer des problèmes**

**Nathalie Duclosson, Stéphanie Jean-Daubias, Stéphanie Riot**

*LIRIS et ERTé e-praxis  
Université Claude Bernard - Lyon 1  
Nautibus  
43 Boulevard du 11 novembre 1918  
69622 Villeurbanne Cedex*

*{Nathalie.Guin-Duclosson, Stéphanie.Jean-Daubias}@liris.univ-lyon1.fr*

---

*RÉSUMÉ. Si les enseignants utilisent peu d'EIAH avec leurs élèves, c'est peut-être parce qu'ils n'ont pas 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. C'est pourquoi nous proposons dans le cadre du projet AMBRE 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 à l'école primaire. Nous avons, en collaboration avec des enseignants, identifié les fonctionnalités de AMBRE-enseignant nécessaires à l'adaptabilité de l'EIAH. Nous avons en particulier conçu un système à base de connaissances qui permet à l'enseignant de générer les problèmes de son choix pour les proposer à ses élèves dans l'EIAH.*

*MOTS-CLÉS : adaptabilité des EIAH, rôle de l'enseignant, génération de problèmes, connaissances.*

---

## 1. Introduction

Le projet AMBRE (Apprentissage de Méthodes Basé sur le Raisonnement à partir de l'Expérience) est une étude pluridisciplinaire en informatique, sciences cognitives, didactique des mathématiques et sciences de l'éducation, dont le but est de concevoir des environnements d'apprentissage pour l'acquisition de méthodes en résolution de problèmes. Dans chaque domaine d'application, une méthode est fondée sur le classement par l'apprenant des problèmes et des outils de résolution. Nous proposons pour faire acquérir de telles méthodes d'utiliser le raisonnement à partir de cas, paradigme développé en Intelligence Artificielle et issu de recherches en psychologie cognitive sur le raisonnement par analogie. Après l'évaluation d'un premier prototype traitant des problèmes de dénombrement en terminale scientifique, un système complet concernant les problèmes additifs en école primaire, AMBRE-add, a été développé et expérimenté [NOGRY et al. 04].

Afin de favoriser l'intégration de AMBRE à l'enseignement, nous avons conçu un module s'adressant aux enseignants, AMBRE-enseignant. Dans cet article, après avoir brièvement présenté les fonctionnalités de ce module, nous décrivons l'une d'entre elles, dédiée à la génération de problèmes, et nous présentons l'architecture du système à base de connaissances permettant de la mettre en œuvre.

## 2. AMBRE : “which role for the teacher?”

La plupart des EIAH sont aujourd'hui encore centrés sur le duo ordinateur/apprenant masquant ainsi trop souvent le rôle pourtant capital de l'enseignant dans ces environnements [VIVET 90]. L'enseignant peut tout d'abord participer directement à la conception de l'EIAH en tant que concepteur ou partenaire de conception. Il peut également être concepteur d'EIAH en tant qu'utilisateur d'un système auteur. Par ailleurs, l'enseignant endosse généralement le rôle de prescripteur en choisissant le système qu'il fera utiliser à ses élèves. L'enseignant est parfois aussi utilisateur secondaire de l'EIAH utilisé par ses élèves en le paramétrant. Enfin, plus rarement, il est utilisateur principal d'un système s'adressant directement à lui pour l'aider dans sa tâche d'enseignement.

Dans le projet AMBRE, si plusieurs enseignants ont joué un rôle de **partenaires de conception** dans le cadre d'une conception différenciée [JEAN-DAUBIAS 04], l'enseignant a pour rôle majeur celui d'**utilisateur principal** d'un module qui lui est exclusivement destiné. Il pourra ainsi à travers l'environnement proposé tenir explicitement son rôle d'**utilisateur secondaire** du module apprenant de AMBRE, en adaptant et paramétrant l'environnement destiné à ses élèves. Enfin, l'enseignant conserve un rôle de **prescripteur**, car il choisit le logiciel proposé à sa classe.

Pour AMBRE, nous souhaitons proposer à l'enseignant un environnement s'adressant exclusivement à lui et permettant d'intégrer et d'adapter un EIAH AMBRE à sa démarche, à ses stratégies pédagogiques, mais aussi au contexte dans lequel il évolue. Pour cela AMBRE-enseignant comporte cinq fonctionnalités. Le **paramétrage de l'environnement** élève consiste en la personnalisation par

l'enseignant de l'interface du logiciel élève et en la constitution des listes d'élèves. Même si le logiciel destiné aux élèves intègre une batterie d'exercices prédéfinis, AMBRE-enseignant comporte un module de **génération de problèmes**, afin de permettre aux enseignants qui le désirent de faire travailler leurs élèves sur des problèmes dont ils souhaitent préciser les caractéristiques. L'enseignant peut également **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 et en définissant le comportement du logiciel élève en termes d'aide et de diagnostic pour les exercices de la séquence. AMBRE-enseignant permet aussi de **distribuer le travail** aux élèves, c'est-à-dire associer à l'ensemble de la classe ou à chaque élève une ou plusieurs séquences. Enfin, afin que l'enseignant puisse entièrement adapter les problèmes générés au contexte dans lequel ses élèves évoluent, AMBRE-enseignant pourra à terme lui permettre de **gérer les traits de surface** qui seront utilisés dans les exercices : thèmes, objets et personnages spécifiques, actions qui y sont associées, etc.

### 3. Un générateur de problèmes pour AMBRE

Nous avons choisi une approche relevant des générateurs dits semi-automatiques, qui construisent eux-mêmes les énoncés des problèmes, mais en laissant intervenir l'utilisateur dans le processus de création. En effet, les générateurs automatiques n'offrant aucune interaction avec l'enseignant ne convenaient pas pour notre objectif. Quant aux générateurs manuels, de type systèmes auteurs, ils ne sont pas aptes à résoudre les problèmes qu'ils permettent de créer et n'offrent pas de fonctionnalité d'aide ou de diagnostic des réponses de l'apprenant. Or un EIAH AMBRE est fondé sur un système à base de connaissances qui s'appuie sur un solveur de problèmes et permet de fournir à l'apprenant de l'aide, un diagnostic de ses réponses ainsi que des explications sur ses erreurs [DUCLOSSON 04]. Les problèmes posés à l'apprenant doivent donc être compréhensibles par le solveur afin que l'EIAH puisse fournir ces fonctionnalités.

Avec AMBRE-enseignant, l'enseignant a la possibilité d'influer sur les problèmes qui seront générés, en spécifiant un ensemble de contraintes sur l'exercice à générer. 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 solveur.

#### 3.1. L'environnement de génération de problèmes pour l'enseignant

Nous avons conçu et développé l'outil de génération de problèmes destiné à l'enseignant pour le domaine des problèmes additifs sur lequel porte l'EIAH AMBRE-add destiné aux élèves de l'école primaire. Les problèmes de ce domaine 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 ? ». Ce domaine a été largement étudié en didactique des mathématiques, et plusieurs classifications de problèmes ont été établies. Celle que nous utilisons dans AMBRE-add est présentée dans [DUCLOSSON 04].

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. Il peut aussi choisir les **valeurs** des données qui seront utilisées dans les problèmes ou définir un intervalle de valeurs pour les éléments et l'écart souhaité entre les deux valeurs, en autorisant ou non la retenue. 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 pour identifier leurs besoins. L'environnement propose des complications de la langue et de l'énoncé lui-même.

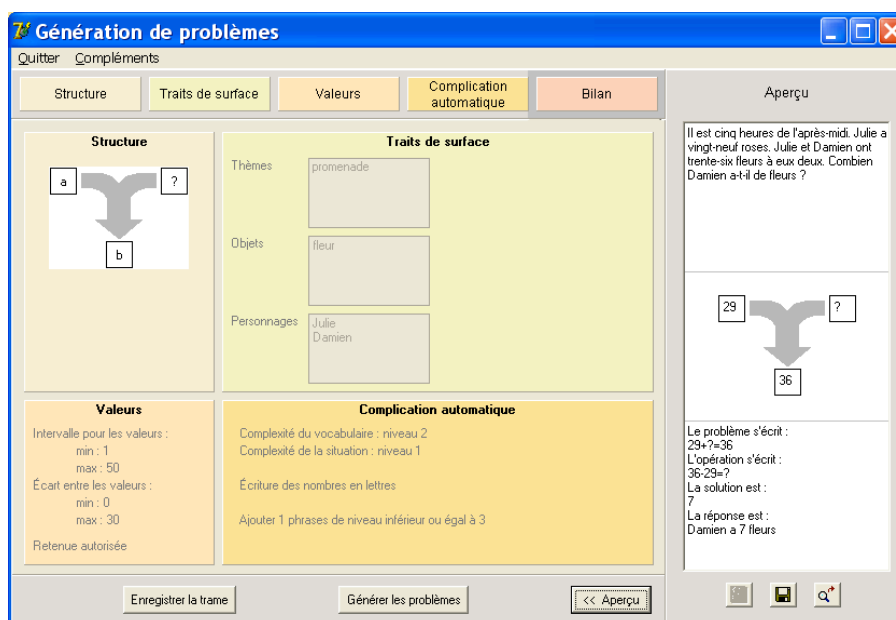


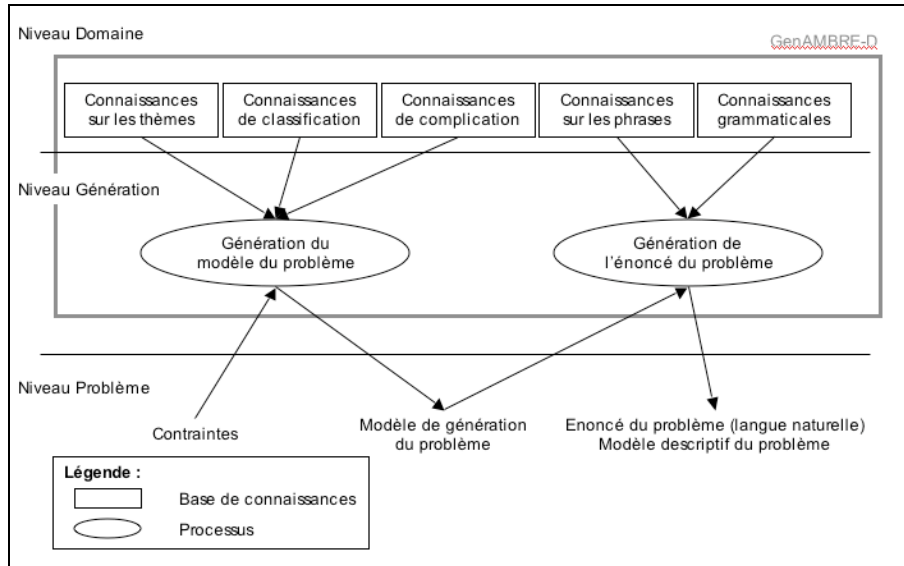
Figure 1. L'écran de bilan du module enseignant de AMBRE-add.

Aucune des contraintes n'est obligatoire pour la création d'exercices. Les contraintes non spécifiées par l'enseignant seront définies aléatoirement par le système. La **Figure 1** présente l'écran de bilan du module enseignant de AMBRE-add. Cet écran résume les contraintes définies par l'enseignant pour les quatre catégories et présente un exemple de problème qui pourra être généré à partir de ces contraintes.

### 3.2. L'architecture GenAMBRE

Le processus 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 une formulation de l'énoncé appelée modèle descriptif du problème, destinée au résolveur de problèmes de AMBRE.



**Figure 2.** L'architecture GenAMBRE

L'architecture du générateur de problèmes pour AMBRE est présentée en **Figure 2** et chacun de ses constituants est présenté dans la suite de l'article. Pour un domaine D (par exemple celui des problèmes additifs), il faut définir les cinq bases de connaissances du « niveau domaine » en utilisant les formalismes de représentation des connaissances indépendants du domaine imposés par le « niveau génération ». Le processus de génération du problème s'effectue en deux temps : le système 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.

**Connaissances de classification** Dans chaque domaine d'application, un expert fournit au résolveur de AMBRE, et donc par extension au générateur GenAMBRE, un graphe de classification de problèmes. Cette hiérarchie de classes est utilisée par le résolveur pour classer le problème.

**Connaissances sur les thèmes** Pour pouvoir générer un problème, il faut connaître le thème concerné et les traits de surface qui lui sont associés (par exemple les objets, personnages et actions). Les connaissances sur les thèmes sont actuellement fournies par l'expert, mais elles pourront à terme être créées par l'enseignant lui-même, via le module de gestion des traits de surface de AMBRE-enseignant.

**Connaissances de complication** 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 est-il possible de changer l'ordre des phrases utiles du problème ? Quelles phrases inutiles peut-on ajouter aux problèmes et où les placer ?

**Processus de génération du modèle du problème** À partir des trois bases de connaissances décrites précédemment et des contraintes entrées par l'enseignant, le système doit générer ce que nous appelons le modèle de génération du problème. Ce modèle est en fait un modèle descriptif étendu, puisqu'il précise également la classe du problème ou encore son thème. Pour construire ce modèle, le processus complète les contraintes définies par l'enseignant, en choisissant notamment aléatoirement des « valeurs » pour celles qui n'ont pas été précisées.

**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.

**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. Les connaissances sur les phrases permettent donc d'associer à la classe du problème les structures de phrases utilisables, et les éléments du problème associés.

**Processus de génération de l'énoncé du problème** Pour générer un énoncé en langue naturelle, le processus 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 système de prendre les décisions d'ordre conceptuel (décider quoi dire), puis le processus passe à l'étape de génération de texte, et met donc en place les traitements syntaxiques (décider comment le dire), lexicaux et morphologiques (décider comment l'écrire).

#### 4. Références

[DUCLOSSON 04] Duclosson, N., « Représentation des connaissances dans l'EIAH AMBRE-add », Conférence TICE'2004, Compiègne, 20-22 octobre 2004, p. 164-171.

[JEAN-DAUBIAS 04] Jean-Daubias, S., « De l'intégration de chercheurs, d'experts, d'enseignants et d'apprenants à la conception d'EIAH », Conférence TICE'2004, Compiègne, 20-22 octobre 2004, p. 290-297.

[NOGRY et al. 04] Nogry, S., Jean-Daubias, S., Duclosson, N., « ITS Evaluation in Classroom: The Case of AMBRE-AWP », ITS'2004 proceedings, Springer, 2004, Lecture Notes in Computer Science vol. 3220, p. 511-520.

[VIVET 90] Vivet, M., « Uses of ITS: Which role for the teacher? », New Directions for Intelligent Tutoring Systems, NATO ASI series, Vol. F91, Springer-verlag, Sintra, 1990.

Une version longue de cet article est disponible sur la page web du projet AMBRE : <http://bat710.univ-lyon1.fr/~nducloss/ambre.html>