



HAL
open science

Le projet **AMBRE** : utiliser le RàPC pour enseigner des méthodes

Nathalie Guin, Stéphanie Jean-Daubias, Sandra Nogry

► To cite this version:

Nathalie Guin, Stéphanie Jean-Daubias, Sandra Nogry. Le projet **AMBRE** : utiliser le RàPC pour enseigner des méthodes. RàPC 2001, 2001, Grenoble, France. hal-01575812

HAL Id: hal-01575812

<https://hal.science/hal-01575812>

Submitted on 21 Aug 2017

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.

Le projet AMBRE : utiliser le RàPC pour enseigner des méthodes

Nathalie Guin-Duclosson, Stéphanie Jean-Daubias et Sandra Nogry

Laboratoire d'Ingénierie des Systèmes d'Information
Université Claude Bernard - Lyon 1
Nautibus, 8 bd Niels Bohr, Campus de la Doua
69622 Villeurbanne Cedex
{nguin, sdaubias, snogry}@bat710.univ-lyon1.fr

Résumé

Dans le cadre des Environnements Informatiques d'Apprentissage Humain (EIAH), nous nous intéressons à l'enseignement de méthodes, fondées sur un classement des problèmes et des outils de résolution. L'objectif du projet AMBRE (Apprentissage de Méthodes Basé sur le Raisonnement à partir de l'Expérience) est de concevoir un EIAH fondé sur le paradigme du raisonnement à partir de cas (RàPC) pour enseigner des méthodes. Notre propos est que l'utilisation de cas pour résoudre des problèmes et la comparaison entre cas peut permettre à l'apprenant d'acquérir des connaissances abstraites (schémas, classes de problèmes), une classe de problèmes étant représentée par un problème prototype.

Mots clés : RàPC, EIAH, enseignement de méthodes, utilisation d'exemples, problèmes prototypes, comparaison de problèmes.

1 Introduction

Nous décrivons dans cet article le projet AMBRE (Apprentissage de Méthodes Basé sur le Raisonnement à partir de l'Expérience), qui a pour objectif la réalisation d'un Environnement Informatique d'Apprentissage Humain (EIAH). Nous ne proposons pas ici un apport à la recherche sur le Raisonnement à Partir de Cas, mais nous présentons une application de ce paradigme de raisonnement au domaine des EIAH. Notre thèse est que l'utilisation du cycle de RàPC peut améliorer l'apprentissage humain de connaissances expertes pour la résolution de problèmes.

Dans une première partie, nous décrivons les connaissances que nous souhaitons faire acquérir à l'apprenant dans cet EIAH, les liens de cette expertise avec le RàPC, ainsi que des travaux existants en EIAH utilisant le RàPC. Dans une deuxième partie, nous présentons la démarche sur laquelle se fonde le projet AMBRE et que nous projetons de mettre en œuvre dans un EIAH. Nous terminons par la description de l'architecture sur laquelle s'appuiera cet EIAH, des

connaissances et des processus de raisonnement nécessaires à sa réalisation.

2 Objectifs et contexte

2.1 Que souhaite-t-on faire acquérir à l'apprenant ?

Des études didactiques proposent d'enseigner explicitement les méthodes qui, dans un domaine restreint, permettent de guider la résolution des problèmes [10][11]. Ces méthodes sont fondées sur un classement des problèmes et des outils de résolution. Elles permettent de choisir la technique de résolution la mieux adaptée à un problème donné.

Dans le projet AMBRE, nous envisageons de nous appuyer sur un solveur de problèmes pour assister l'apprentissage, par exemple pour fournir des explications à l'apprenant. Ce solveur doit donc fonctionner selon les méthodes qu'il souhaite faire acquérir, et non pas selon les méthodes expertes les plus générales du domaine concerné [2]. Il faut ainsi partir des connaissances visées pour la conception du solveur.

Pour représenter ces connaissances, nous nous appuyons sur l'architecture de solveurs SYRCLAD [4], qui permet d'explicitier de manière déclarative une classification de problèmes et les connaissances de reformulation et de résolution qui y sont liées. Pour un domaine donné, l'expert (didacticien, enseignant...) définit une hiérarchie de classes de problèmes. Pour pouvoir utiliser cette hiérarchie pour classer un problème, des experts doivent définir des connaissances de reformulation qui permettent de construire un nouveau modèle du problème (appelé modèle opérationnel) en fonction des attributs discriminants de la hiérarchie. Certaines classes de la hiérarchie, dites opérationnelles, sont suffisamment spécifiques pour qu'on puisse associer à chacune une technique de résolution adaptée aux problèmes qui relèvent de cette classe pour le système. Résoudre un problème consiste alors tout d'abord en une phase d'opérationnalisation, où

le système utilise les connaissances de reformulation et le graphe de classification des problèmes du domaine pour déterminer de quelle classe relève le problème et construire un nouveau modèle de ce problème. La résolution proprement dite consiste alors à appliquer la technique de résolution associée à cette classe sur le modèle ainsi défini pour obtenir une solution au problème posé.

Le projet AMBRE a pour objectif de concevoir un EIAH dont le but est de faire acquérir à l'apprenant une méthode de résolution fondée sur une hiérarchie de classes de problèmes et sur des techniques de résolution associées à ces classes. Cependant il ne semble pas approprié de présenter explicitement une telle hiérarchie à l'apprenant, car les termes définissant les classes et les techniques de résolution ne sont pas forcément utilisés institutionnellement, et ne sont pas connus des apprenants. Nous envisageons par conséquent une démarche où l'apprenant est plus actif, et dans laquelle il se construit sa propre méthode de résolution, en représentant une classe de problème par un problème prototype. Nous pensons que le paradigme du raisonnement à partir de cas peut aider l'apprenant dans l'acquisition d'une telle méthode.

2.2 Liens avec le Raisonnement à Partir de Cas

Le Raisonnement à Partir de Cas peut être décrit par un ensemble d'étapes séquentielles (élaborer, remémorer, adapter, tester/réviser, mémoriser) que l'on représente souvent par un cycle [9].

Dans le cadre d'une méthode de résolution implémentée grâce à l'architecture SYRCLAD, l'étape de *reformulation du problème* peut être vue comme une étape d'**élaboration d'un cas cible**, même si cette étape est alors plus poussée puisqu'elle utilise des connaissances de reformulation et des connaissances de classification¹. On peut également remplacer le *classement du problème* par une étape de **remémoration d'un cas source**, le cas source alors remémoré correspondant au cas prototype de la classe de problème qu'aurait déterminé un résolveur SYRCLAD. **Adapter la solution** de ce cas source au cas cible est alors équivalent à *appliquer la technique de résolution* associée à la classe opérationnelle que représente ce prototype. Pour la **mémorisation du problème résolu** on peut envisager que le résolveur SYRCLAD *rattache ce nouveau cas* à une classe de problèmes

¹ Nous sommes bien ici dans le cas d'une reformulation a priori du problème, reformulation dont le but est d'élaborer un cas en construisant les indices qui permettront la remémoration. Notre travail se distingue par ce fait des reformulations définies dans [8], où l'appariement du problème cible avec un problème source est effectué en définissant un chemin de similarité composé d'une séquence de reformulations, ce chemin de similarité permettant d'effectuer l'adaptation de la solution source par une séquence analogue de reformulations.

Notre objectif est de faire acquérir une méthode de résolution fondée sur une hiérarchie de classes de problèmes en présentant à l'apprenant des exemples de problèmes résolus et en guidant la résolution de nouveaux problèmes grâce à un raisonnement à partir de cas. On souhaite que les connaissances de l'« élève-expert » après l'apprentissage puissent se modéliser comme sur la figure 1 : à partir d'une base de cas, l'apprenant a construit des classes opérationnelles représentées par un problème prototype, ces classes de problèmes étant éventuellement organisées par une hiérarchie.

2.3 Des utilisations du RàPC en EIAH

Le RàPC est une technique utilisable dans le domaine des EIAH. Le RàPC peut intervenir dans diverses composantes d'un EIAH.

La modélisation des connaissances de l'apprenant peut être réalisée grâce au RàPC [13], de même que l'évaluation de l'apprenant [6]. Par comparaison du modèle de l'apprenant avec des évaluations d'autres apprenants (formant une base de cas), le RàPC peut également permettre la sélection d'une stratégie d'apprentissage [3] ou la construction d'un chemin dans un hypertexte [5].

L'utilisation la plus proche de notre propos est l'enseignement à partir de cas ou Case-Based Teaching [12]. Les systèmes fondés sur cette stratégie d'apprentissage proposent un cas proche à l'apprenant lorsque celui-ci est en difficulté lors de la résolution d'un problème, ou lorsqu'il est face à un problème qu'il n'a jamais rencontré (d'un nouveau domaine ou d'un nouveau type). Dans ces systèmes, on peut trouver différents niveaux d'interactivité entre l'apprenant et l'environnement informatique [14]. L'apprenant peut demander au système de lui retrouver un exemple similaire, de lui expliquer comment ce cas a été résolu ; le système peut aussi lui proposer la résolution complète de son exercice comme dans CATO [1].

Le *Virtual Participant* [7] est un exemple d'utilisation de l'enseignement à partir de cas. Il s'agit d'un assistant virtuel intervenant dans les conférences virtuelles, qui peut proposer des cas aux apprenants. La base de cas est constituée de problèmes posés par les personnes participant à la conférence et de solutions (proposées par ces mêmes intervenants). Cet environnement a été expérimenté dans un cours où les conférences de type forum étaient utilisées. D'une année à l'autre, les promotions rencontrent des problèmes identiques ; ainsi, le *Virtual Participant* permet de répondre plus efficacement à leurs attentes en proposant des cas similaires rencontrés par des étudiants des années antérieures.

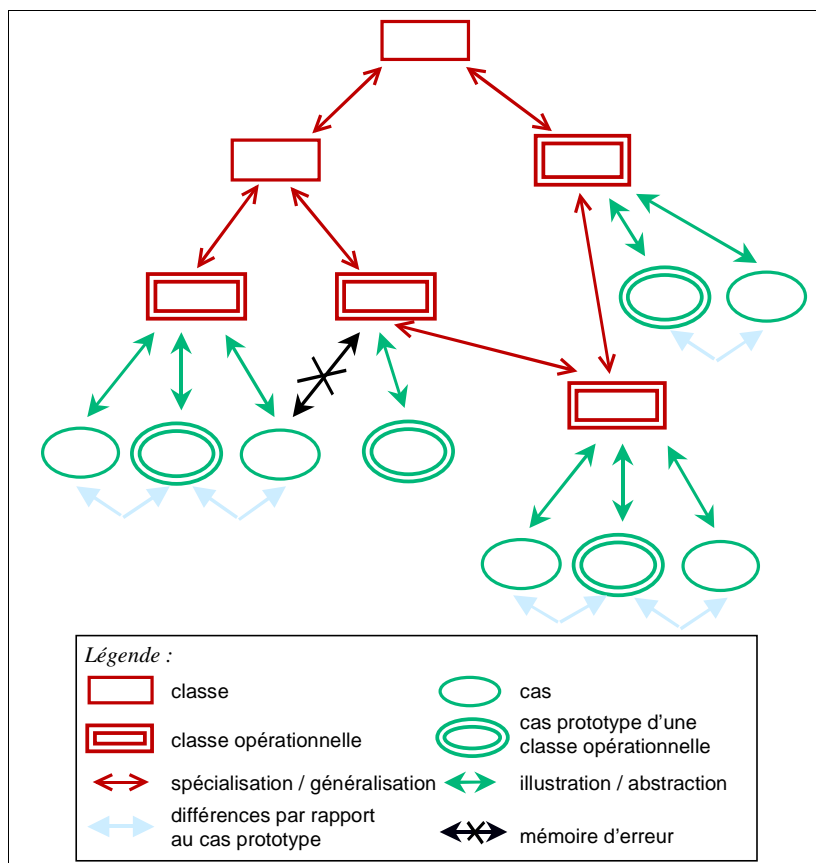


FIG. 1 - Modèle des connaissances de l'élève – expert.

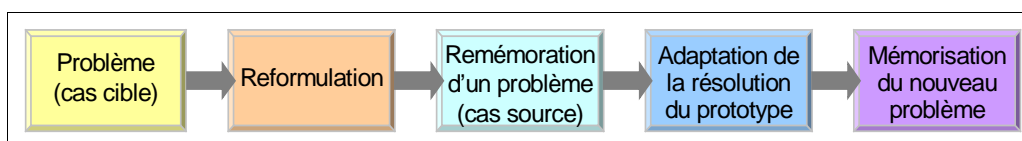


FIG. 2 - Les étapes de la résolution.

3 Le cycle de RàPC dans AMBRE

Dans le but de faire acquérir à l'apprenant une méthode de résolution pour les problèmes d'un domaine, on commence par lui présenter de manière interactive la résolution de quelques problèmes types. On l'accompagne ensuite dans la résolution de nouveaux problèmes : la résolution du problème par l'apprenant est guidée par l'environnement en suivant chaque étape du cycle de RàPC (cf. FIG. 2). Élaboration, remémoration, adaptation et mémorisation sont effectuées par l'apprenant, mais guidées par le système.

3.1 Élaboration du cas cible

La première étape consiste à reformuler le problème à résoudre. L'apprenant est invité à construire une nouvelle formulation du problème qui lui est posé, étape par étape. Cette tâche consiste en fait à trouver les valeurs des attributs du modèle opérationnel du problème. Le système se fonde sur le graphe de classification pour guider cette reformulation. Il

diagnostique les réponses de l'élève et explique d'éventuelles erreurs afin d'obtenir un accord avec l'élève sur une reformulation correcte. Cette reformulation (le modèle opérationnel) est débarrassée de l'essentiel des traits de surface du problème initial, et devient une référence pour la suite de la résolution.

3.2 Remémoration d'un cas source

La deuxième étape de la résolution consiste pour l'apprenant à comparer le problème à résoudre à ceux déjà résolus. L'apprenant organise, à l'aide du logiciel, les problèmes résolus en groupes de problèmes, chaque groupe étant représenté par un problème prototype (cf. 3.4). Pendant cette étape de remémoration, l'apprenant doit choisir le problème prototype (cas source) qui lui semble le plus proche du problème à résoudre, cette proximité devant se baser sur les problèmes reformulés et non sur des traits de surface. Le système diagnostique la proposition de l'élève et propose éventuellement un meilleur choix en le justifiant, afin d'obtenir un accord avec l'élève sur un problème de référence.

3.3 Adaptation de la solution du cas source au cas cible

Pour achever la résolution, l'apprenant doit adapter la solution du cas source (le problème de référence) au cas cible (le problème à résoudre). Le système prépare cette adaptation en mettant en évidence différences et similarités entre le cas source et le cas cible. Lorsque les deux cas relèvent de la même classe opérationnelle, adapter la solution du problème prototype revient à appliquer la technique de résolution associée à la classe à laquelle il appartient, aux valeurs des attributs significatifs du problème à résoudre (ceux qui caractérisent le problème par rapport à la classe dont il est une instance). La technique de résolution, connue du système ne l'est pas explicitement de l'apprenant. C'est pourquoi l'apprenant doit s'appuyer sur la solution du cas source pour l'adapter au cas cible.

On peut envisager que, dans un premier temps, le système montre à l'apprenant comment effectuer cette étape d'adaptation sur un exemple, de manière à pouvoir présenter les outils de l'interface permettant d'effectuer cette adaptation. Lorsque l'apprenant propose une adaptation débouchant sur une proposition de solution, le système diagnostique cette proposition et éventuellement donne des explications sur les corrections à effectuer pour obtenir une solution correcte.

3.4 Mémorisation du cas cible

Une fois le problème résolu, l'apprenant doit l'ajouter à la base de cas, c'est-à-dire l'insérer dans l'organisation des problèmes qu'il a construite. Cette tâche consistera souvent à attacher le nouveau cas à un problème prototype, donc à l'insérer dans un groupe de problèmes existant. Un groupe de problèmes regroupe tous les problèmes qui relèvent de la même classe opérationnelle. Il faut bien entendu donner la possibilité à l'apprenant de modifier le choix du problème prototype qui représente un groupe. Il est par ailleurs peut-être souhaitable que les problèmes prototypes appartiennent tous à la même famille de problèmes (au sens des traits de surface), de manière à pouvoir mieux insister sur les différences de structure.

Il est possible que le nouveau cas n'appartienne pas à un groupe de problèmes existant de l'organisation, et l'apprenant peut alors créer un nouveau groupe dont le problème prototype est ce problème qu'il vient de résoudre. Il est aussi possible qu'il soit nécessaire de créer un (ou plusieurs) sous-groupes de problèmes au sein d'un groupe, le sous-groupe étant également représenté par un problème prototype. Il s'agit de représenter une classe opérationnelle fille d'une autre classe opérationnelle ; à la classe-fille est associée une méthode de résolution plus spécifique que celle associée à la classe-mère.

Il faut enfin donner à l'apprenant les outils qui lui permettront, s'il le désire, de caractériser les groupes de problèmes les uns par rapport aux autres, en construisant un début de hiérarchie. Il est souhaitable qu'il utilise

pour cela des attributs des modèles opérationnels, qu'il connaît grâce à l'étape de reformulation du problème.

4 Architecture de l'EIAH et bases de connaissances

Pour que l'environnement AMBRE puisse assurer les fonctionnalités décrites dans la section précédente, nous proposons ici l'architecture que nous envisageons actuellement pour le système. Nous décrivons les différents modules constituant cette architecture, les bases de connaissances et les processus de raisonnement utilisés par chacun de ces modules, ainsi que leurs interactions.

4.1 Le module expert

Le module expert contient une base de connaissances équivalente à celle de la figure 1 : une hiérarchie de classes de problèmes contenant des classes opérationnelles auxquelles sont attachés des cas représentant les problèmes déjà résolus au cours de la session.

Un résolveur issu de l'architecture SYRCLAD peut résoudre des problèmes en utilisant la hiérarchie de classes de problèmes. On peut lui adjoindre un résolveur de type RàPC qui pourra raisonner de manière indépendante sur les cas.

Ces deux résolveurs sont capables de résoudre le problème parallèlement à l'apprenant. Les autres modules (diagnostic, aide) peuvent ainsi utiliser les résolutions de ces résolveurs ainsi que la base de connaissances.

4.2 Le modèle de l'apprenant

Le modèle de l'apprenant contient une représentation de l'expérience de l'apprenant : les problèmes qu'il a résolus, avec une trace des éventuelles erreurs qu'il a commises pendant la résolution. Ces problèmes constituent une base de cas que l'apprenant doit organiser autour de problèmes prototypes avec éventuellement un début de hiérarchie. Bien que contenant les mêmes problèmes que la base de connaissances du module expert, cette base de cas est organisée de manière différente. En effet, l'apprenant n'étant pas expert, il organise les problèmes différemment du système.

Ces éléments doivent être interprétés afin de chercher à identifier les connaissances de l'apprenant.

4.3 Le module de diagnostic

Le rôle du module de diagnostic est d'analyser la validité des réponses de l'apprenant à chaque étape de la résolution (cf. section 3). Pour cela, il compare ces réponses à celles que propose le module expert. Si la réponse de l'élève n'est pas correcte, il fournit au module d'aide les données nécessaires à la construction d'une explication appropriée.

4.4 Le module pédagogique

Le module pédagogique a pour rôle de choisir le problème posé à l'apprenant, et de choisir le type d'aide et d'explications adapté si cela s'avère nécessaire pendant la résolution.

Pour choisir le problème à poser, le module pédagogique dispose de la hiérarchie des classes de problèmes et d'une base de problèmes cibles attachés aux classes opérationnelles. Il utilise les connaissances du modèle de l'apprenant sur les problèmes que celui-ci a déjà traités et sur les difficultés qu'il a rencontrées.

Choisir le type d'aide consiste principalement à déterminer la granularité des indications fournies, en laissant au module d'aide le soin de construire ces indications.

4.5 Le module d'aide et d'explications

Le module d'aide doit permettre de proposer des aides et des explications adaptées aux difficultés rencontrées par l'apprenant. Les explications peuvent s'appuyer sur le résolveur dans la mesure où le fonctionnement de celui-ci est fondé sur le raisonnement humain, et plus spécifiquement sur les connaissances des apprenants. Pour fournir de l'aide à l'apprenant à chaque étape de la résolution, le module d'aide utilise la base de connaissances et les résolveurs du module expert, le modèle de l'apprenant, et éventuellement les informations fournies par le module de diagnostic.

5 Perspectives

Nous avons présenté dans cet article le projet d'EIAH AMBRE, qui s'appuie sur le cycle de résolution du RàPC pour faire acquérir à l'apprenant une méthode de résolution de problèmes fondée sur une classification des problèmes du domaine.

Une première maquette de cet EIAH est en cours de réalisation sur des problèmes de dénombrement en terminale scientifique. Cette maquette sera testée dans des classes de terminale en mai 2001. Ces expérimentations, conduites selon des techniques expérimentales de psychologie cognitive, ont pour objectif de tester l'efficacité pour l'apprentissage humain de la démarche que nous avons présentée dans cet article.

Références

- [1] V. Aleven et K.D. Ashley. Teaching Case-Based Argumentation through a Model and Examples - Empirical Evaluation of an Intelligent Learning Environment. *Artificial Intelligence in Education* (B. du Boulay and R. Mizoguchi Eds.), pages 87-94, IOS Press, 1997.
- [2] É. Delozanne. Explications en EIAO : études à partir d'ÉLISE, un logiciel pour s'entraîner à une méthode de calcul de primitives. Thèse de doctorat, Université du Maine, Le Mans, 1992.
- [3] J.E Gilbert. Case-Based Reasoning Applied to Instruction Method Selection for Intelligent Tutoring Systems. *Workshop 5 : Case-Based Reasoning in Intelligent Training Systems, ITS'2000*, Montreal, pages 11-15, 2000.
- [4] N. Guin-Duclosson. SYRCLAD : une architecture de résolveurs de problèmes permettant d'explicitier des connaissances de classification, reformulation et résolution. *Revue d'Intelligence Artificielle*, vol 13-2, pages 225-282, Paris : Hermès, 1999.
- [5] J.M. Héraud. Projet d'intégration de l'expérience pour l'enseignement à distance : présentation du prototype. *RàPC'2000*, Toulouse, pages 33-37, 2000.
- [6] T.M. Khan. Case-Based Evaluation for Student Modelling. *Workshop 5 : Case-Based Reasoning in Intelligent Training Systems, ITS'2000*, Montreal, pages 16-22, 2000.
- [7] S. Masterton. The Virtual Participant : Lessons to be Learned from a Case-Based Tutor's Assistant. *Computer Support for Collaborative Learning*, Toronto, 1997.
- [8] E. Melis, J. Lieber and A. Napoli. Reformulation in Case-Based Reasoning, *Fourth European Workshop on Case-Based Reasoning*, EWCBR 98, pages 172-183, 1998.
- [9] A. Mille. Associer expertise et expérience pour assister les tâches de l'utilisateur, *Habilitation à diriger des recherches*, Université Claude Bernard, Lyon1, 1998.
- [10] M. Rogalski. Enseigner des méthodes en mathématiques. *Commission Inter-Irem Université, Enseigner autrement les mathématiques en Deug A première année*, bulletin Inter-Irem, pages 65-79, 1990.
- [11] M. Rogalski. Les concepts de l'EIAO sont-ils indépendants du domaine ? L'exemple d'enseignement de méthodes en analyse. *Recherches en Didactiques des Mathématiques*, vol 14 n°1.2, pages 43-66, 1994.
- [12] R. Schank et D. Edelson. A Role for AI in Education : Using Technology to Reshape Education. *Journal of Artificial Intelligence in Education* vol 1.2, pages 3-20, 1990.
- [13] A. Shiri, E. Aimeur et C. Frasson. SARA : A Cased-Based Student Modelling System. *Fourth European Workshop on Case-Based Reasoning*, Lecture Notes in Artificial Intelligence, n° 1488, pages 425-436, Dublin, Ireland, 1998.
- [14] N. Tourigny et L. Capus. Towards Making Intelligent Training Systems Using Examples more Flexible and Reusable by Exploiting Case-Based Reasoning. *Workshop 5 : Case-Based Reasoning in Intelligent Training Systems, ITS'2000*, Montreal, pages 23-28, 2000.