LE PROJET APLUSIX

Jean-François NICAUD

1. INTRODUCTION

APLUSIX est un projet de recherche dont les objectifs sont d'étudier et de modéliser les processus d'apprentissage humains ainsi que de réaliser des EIAO (Environnements Interactifs d'Apprentissage avec Ordinateur). Le domaine d'APLUSIX est composé des champs de problèmes d'algèbre se résolvant par transformation d'expressions symboliques. Actuellement, les sous-domaines abordés sont la factorisation d'expressions polynomiales et la résolution d'équations.

Le projet APLUSIX a vu le jour au Laboratoire de Recherche en Informatique (LRI) à Orsay en 1983 dans l'équipe Intelligence Artificielle et Systèmes d'Inférence. Il a pris à partir de 1988 une dimension pluridisciplinaire, avec la participation de chercheurs en Psychologie Cognitive et en Didactique des Mathématiques. Les chercheurs travaillant sur ce projet appartiennent au LRI (Orsay), à l'équipe de Psychologie Cognitive de l'Université de Paris VIII, à l'INRP, à l'IMAG-LSD2 (Grenoble) et au LAFORIA (Paris VI). Outre l'auteur, les personnes ayant particulièrement travaillé sur les parties présentées ici sont : J.M. Gélis et M. Saïdi (conception des modèles et réalisation des prototypes), A. Nguyen-Xuan et F. Joly (modélisation de l'élève), C. Aubertin et P. Wach (conduite des expérimentations).

2. PROBLÉMATIQUE ET MODÈLE GÉNÉRAL

2.1. Problématique

Le cadre théorique se compose d'objets mathématiques, par exemple des fonctions ou des polynômes sur un ensemble de nombres, et de représentations de ces objets, les expressions algébriques.

Certains types de problèmes se définissent directement dans le cadre des expressions. C'est le cas, par exemple, des problèmes de réduc tion d'expressions (réduire une expression qui représente un objet mathématique, c'est trouver une représentation plus simple de cet objet, c'est donc bien un problème au niveau des expressions).

D'autres types de problèmes se définissent dans le cadre mathématique. Par exemple, dans l'ensemble des polynômes, on définit la divisibilité des polynômes et les polynômes premiers ; on montre la propriété de décomposition d'un polynôme en un produit de polynômes premiers, ce qui permet de définir le type de problème *factoriser polynôme* avec l'assurance qu'un problème de ce type a toujours une solution. La recherche des solutions d'un problème passe généralement par la manipulation d'expressions, elle consiste à transformer l'expression initiale jusqu'à l'obtention d'une forme résolue. Les transformations se font à l'aide d'opérateurs qui conservent l'équivalence des expressions, c'est-à-dire qui remplacent une expression par une expression représentant le même objet mathématique. Cependant, au niveau de l'enseignement, ces problèmes sont souvent abordés avant qu'il y ait connaissance de la théorie. Cela les place d'emblée dans le cadre des manipulations d'expressions et se fait éventuellement en attribuant une autre signification aux solutions. Par exemple, en factorisation de polynômes, on exigera qu'il y ait mise sous la forme d'un produit ou d'une puissance, sans exiger que les facteurs soient premiers, mais on exigera en plus une mise en facteur des constantes et une réduction des facteurs.

2.2. Modèle général

Nous nous plaçons au niveau des expressions en considérant que les objets mathématiques interviennent seulement pour donner du sens aux opérations effectuées.

Les objets du modèle sont les expressions qui sont construites à l'aide de symboles : des symboles pour représenter les constantes (exemple : 12), des symboles pour représenter des variables mathématiques (comme une indéterminée ou une inconnue x) et des symboles pour représenter des fonctions mathématiques (comme +, -, √, sin). La structure des expressions est récursive : une expression qui n'est ni un symbole de constante ni un symbole de variable est un symbole de fonction avec des arguments qui sont des expressions. La représentation la plus fidèle de cette structure est un arbre dont les feuilles sont les symboles de constantes et les symboles de variables de l'expression et dont les autres noeuds sont les symboles de fonctions. La représentation habituelle est d'une autre nature, incluant des symboles sous-entendus, des parenthèses, des priorités d'opérateurs.

Les opérateurs sur les expressions sont des règles de transformation. Une première forme de règles est celle des règles de réécriture, forme $g \rightarrow d$ dans laquelle g et d sont des expressions contenant des variables de réécriture, par exemple $A^2-B^2 \rightarrow (A-B)(A+B)$ qui contient les variables de réécriture A et B. Les variables de réécriture sont faites pour être remplacées par des expressions. Un mécanisme d'appariement d'expressions permet de déterminer quand une règle est applicable, ce mécanisme n'est pas uniquement syntaxique, il peut faire appel à des concepts sur les expressions, ce qui permet par exemple d'apparier A^2-B^2 avec $4x^2-9$ (syntaxiquement A^2 ne s'apparie pas avec $4x^2$ et B^2 ne s'apparie pas avec 9) en utilisant un concept de carré. Une deuxième forme de règles est composée de généralisations de règles de réécriture, par exemple, *mettre A en facteur dans B.* Les règles de transformation valides sont issues d'identités, égalités universellement quantifiées du niveau mathématique, leur mode d'utilisation, apporté par le niveau mathématique, est *le remplacement d'égal à égal* qui est applicable sur toute sous-expression.

Un type de problème est défini par la donnée d'une caractérisation de ses solutions, c'est-à-dire d'un mécanisme permettant de déterminer quand une expression est résolue pour ce type de problème.

Dans ce cadre, la résolution d'un problème est une activité de *recherche heuristique* : on transforme des objets avec des opérateurs jusqu'à l'obtention d'une solution. Selon le type de problèmes et le niveau de connaissance, on dispose, ou on ne dispose pas de méthode permettant d'aller directement vers une solution. L'élève doit se constituer des connaissances stratégiques pour choisir les opérations à effectuer. Il doit parfois effectuer des retours en arrière, lorsqu'une direction de recherche semble être une impasse, et se constituer des connaissances méta-stratégiques pour cela. Un retour en arrière n'est pas une erreur, c'est une opération normale d'un mécanisme de résolution ne disposant pas d'une méthode directe.

3. LE PROTOTYPE APLUSIX M0-V2

A partir du modèle général, nous avons conçu un modèle plus précis, appelé APLUSIX M0 (M0 signifiant *modèle d'ordre 0*), que nous avons implanté dans des prototypes qui ont été expérimentés.

3.1. Le modèle APLUSIX M0

Le modèle APLUSIX M0 comporte des connaissances stratégiques qui ne sont pas structurées. Le choix de la transformation à appliquer est effectué en appliquant des règles stratégiques aux transformations possibles, ce qui a pour effet de leur attribuer un *intérêt*. Lorsque la meilleure transformation possible sur l'expression courante a un intérêt faible ou moyen et qu'une transformation possible sur une autre expression *e* a un intérêt plus fort, il y a abandon de l'expression courante au profit de l'expression *e*. Nous considérons que ce modèle est pertinent à certains niveaux de l'apprentissage.

3.2. Description du prototype APLUSIX M0-V2

Le prototype APLUSIX M0-V2 est une mise en oeuvre du modèle précédent dans le domaine des factorisations d'expressions polynomiales d'une variable à coefficients entiers, au niveau des classes de seconde. Les opérateurs sont regroupés en quatre familles : les factorisations qui comportent $A^2\text{-}B^2 \rightarrow (A\text{-}B)(A\text{+}B)$, $A^2\text{+}2AB\text{+}B^2 \rightarrow (A\text{+}B)^2$, $A^2\text{-}2AB\text{+}B^2 \rightarrow (A\text{-}B)^2$ et *mettre A en facteur* ; un unique opérateur de développement regroupant tous les cas de suppression de parenthèses ; un unique opérateur de réduction regroupant toutes les règles de réduction ; un opérateur de développement-réduction permettant de combiner développement et réduction pour les expressions du premier degré.

Le mode d'interaction du prototype APLUSIX M0-V2 est l'apprentissage par l'action : l'élève résout des exercices pas à pas en appliquant, à chaque pas, un opérateur qu'il choisit dans un menu à une sous-expression qu'il sélectionne avec la souris. Lorsqu'il choisit une factorisation, il doit indiquer l'appariement en donnant au clavier les valeurs de A et B pour les trois premières règles, la valeur de A pour *mettre A en facteur*. Toutes ces informations constituent la requête de l'élève qui est analysée par le système. Lorsque la requête est invalide, une rétroaction immédiate est effectuée (voir figure 1) ; lorsqu'elle est valide, l'opérateur est appliqué par le système (l'élève ne fait pas les calculs). Le raisonnement de l'élève est représenté sous la forme d'un arbre dont les arcs indiquent les transformations réalisées (voir figure 2). L'élève peut effectuer un retour en arrière à tout moment.

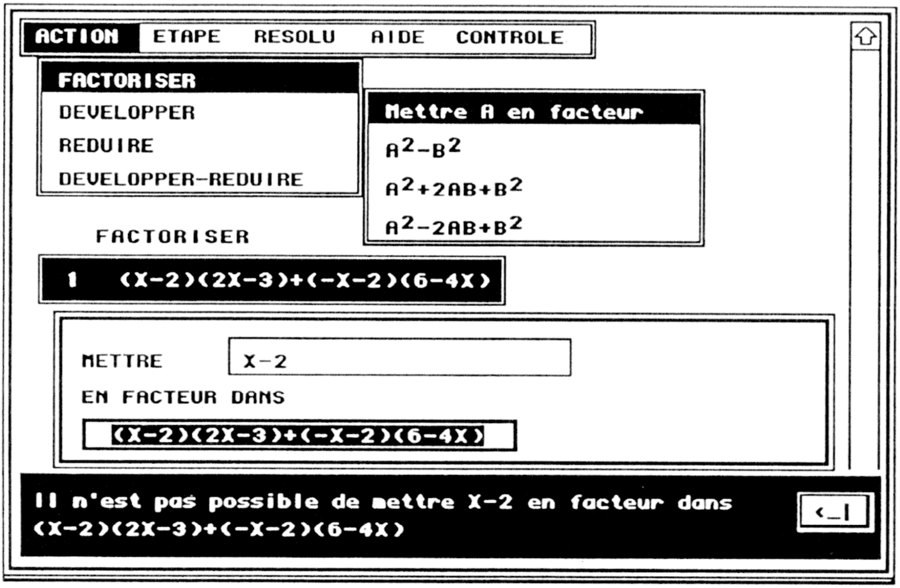

Figure 1

Un écran montrant les menus, une requête erronée et une rétroaction.

Une aide a été implantée dans ce prototype, elle est accessible à tout moment à l'aide d'un menu. Elle s'appuie sur les connaissances du résolveur du système. Comme nous l'avons dit, ce résolveur attribue, à l'aide de règles stratégiques, un intérêt aux transformations possibles. Au niveau considéré, nous avons choisi l'intérêt *t-bien* pour les réductions et les développement-réductions, l'intérêt *bien* pour les factorisations (sauf certaines, comme la mise en facteur de constantes ou la mise en facteur de x sur x^2+2x dans x^2+2x-3, qui sont écartées), l'intérêt *faible* pour les développements.

Remarques : La mise en facteur de constantes est écartée car le résolveur est capable de raisonner modulo cette opération. La mise en facteur de x sur x^2+2x dans x^2+2x-3 est écartée car elle est considérée comme étant sans intérêt. Les transformations qui sont écartées par le résolveur sont permises à l'élève, par exemple, pour l'expression (x-2)(x+4)-(4-2x)(x+1), le résolveur a la mise en facteur de x-2 sur l'expression entière comme transformation possible et n'a pas celle de 2 sur 4-2x ; cependant l'élève peut mettre 2 en facteur sur 4-2x.

L'aide comporte deux niveaux. Le premier niveau répond au choix *les transformations de l'étape* et consiste à présenter ce qui est raisonnablement possible à l'étape, c'est-à-dire toutes les transformations de l'étape qui n'ont été ni écartées par le résolveur, ni déjà appliquées par l'élève. Le deuxième niveau répond au choix *informations générales* et consiste à présenter, sur chaque expression de l'arbre, les transformations d'intérêt bien ou t-bien qui n'ont pas déjà été appliquées par l'élève. C'est donc une aide guidée par les connaissances. Lorsque les problèmes ont des difficultés stratégiques, elle ne place pas forcément l'élève sur une direction aboutissant à une solution ; elle lui montre les directions que le résolveur considère comme étant prometteuses au moment de la demande.

APLUSIX M0-V2, comme tous les prototypes du projet APLUSIX, est développé sur Macintosh avec le générateur de systèmes experts SIM réalisé par l'auteur.

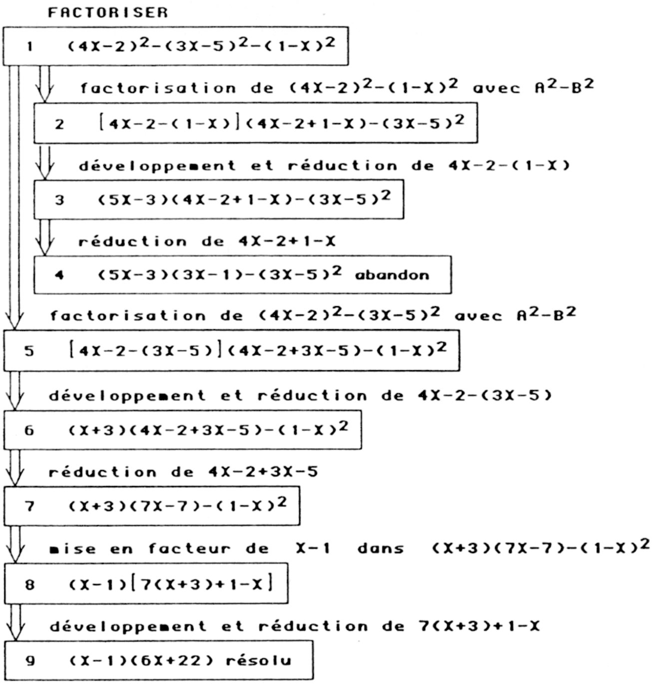

Figure 2

Une résolution telle qu'elle apparaît à l'écran.

3.3. Les expérimentations

Le prototype APLUSIX M0-V2, et son prédécesseur APLUSIX M0-V1, ont fait l'objet d'expérimentations en Avril 1990 et en Mai 1992. Dans les deux cas, des élèves de seconde ont utilisé le logiciel, pendant quatre séances d'une heure, pour résoudre des exercices de factorisation, chaque élève disposant d'une machine.

L'expérimentation d'Avril 1990 s'est déroulée avec APLUSIX M0-V1 et 24 élèves du lycée du Parc de Vilgénis à Massy en utilisant une suite figée d'exercices. Son but principal était de voir s'il y avait apprentissage lors de l'utilisation du prototype. Pour cela, les élèves subirent un pré-test et un post-test à un mois d'intervalle, les tests étant donnés sous une forme classique papier-crayon. Les exercices proposés dans ces deux tests étaient rigoureusement identiques et plus difficiles que les exercices habituellement posés en classe de seconde (ils ne furent bien sûr ni rendus ni corrigés). L'étude des pré et post-tests montre un incontestable progrès global des élèves, la réussite passant de 10% à 34%. On a constaté en particulier des progrès au niveau des appariements, par exemple, dans l'exercice *factoriser (3-2X)(3X+6)-(8+4X)(-2X-3)*, les appariements faux du type *mise en facteur de 2X-3*, sont passés de 12 à 1.

L'expérimentation de Mai 1992 s'est déroulée avec APLUSIX M0-V2 et 46 élèves du lycée Saint Exupéry de Créteil. Son but principal était d'enregistrer les interactions entre les élèves et le système afin de pouvoir les analyser ensuite. L'enchaînement des exercices était assuré par des automates dont les états étaient constitués d'exercices et les transitions de conditions portant sur l'historique de l'élève (nombre d'aides demandées, abandon éventuel, erreurs commises) pendant la résolution de l'exercice. La stratégie pédagogique adoptée consistait à permettre aux élèves faibles d'apprendre ou de renforcer les connaissances de base manquantes ou déficientes à l'aide d'exercices semblables ou plus simples, les autres accédant plus rapidement à des exercices ayant des difficultés stratégiques.

3.4. La modélisation de l'élève

En EIAO, on appelle *modélisation de l'élève* un processus permettant d'établir des informations sur les connaissances de l'élève. L'ensemble de ces informations est appelé *modèle de l'élève* et constitue le point de vue du système sur l'élève. Le mécanisme d'élaboration du modèle est appelé *diagnostic cognitif*.

L'environnement APLUSIX a deux particularités. D'une part les opérateurs sont supposés connus de l'élève (ils sont en plus rappelés dans les menus), d'autre part l'élève est libéré des calculs. De ce fait, les connaissances opératoires importantes sont les connaissances en appariement, pour déterminer les transformations possibles, et les connaissances stratégiques, pour obtenir une bonne efficacité. Le diagnostic a porté sur ces deux composantes, plus précisément sur l'appariement ainsi que sur les heuristiques de choix d'une transformation à appliquer, laissant les heuristiques de retour en arrière pour des travaux ultérieurs. Ces deux types de connaissances opératoires ne sont pas indépendants. En effet, pour choisir entre plusieurs transformations, il faut d'abord être capable de les reconnaître. Par exemple, deux des transformations de $1 + 3(8x^2 - 8x + 2) - 4x^2$ sont les factorisations de $1 - 4x^2$ avec $A^2\text{-}B^2 \rightarrow (A+B)(A\text{-}B)$ et de $8x^2 - 8x + 2$ avec $A^2 - 2AB + B^2 \rightarrow (A\text{-}B)^2$. Supposons que l'élève ait correctement effectué la seconde, nous pouvons dire qu'il sait apparier $A^2- 2AB + B^2$ à $8x^2- 8x + 2$; nous pouvons parler de choix entre les deux transformations seulement si nous sommes certains qu'il a aussi reconnu que l'on peut apparier $A^2\text{-}B^2$ à $1 - 4x^2$. Ainsi pour diagnostiquer les connaissances heuristiques, ou tout au moins décider si l'élève a un ordre de préférence stable entre les transformations possibles, il est nécessaire de connaître ses connaissances en appariement. Mais comme les deux types de connaissance se construisent au fur et à mesure de l'apprentissage, en interaction l'un avec l'autre, il n'est pas pertinent de chercher à diagnostiquer les connaissances heuristiques seulement lorsqu'on est sûr que l'élève possède bien les connaissances sur les appariements.

Pour réaliser le mécanisme de diagnostic, nous avons utilisé 17 catégories d'appariement (définies auparavant et déjà utilisées dans le système) ainsi que 10 classes stratégiques de transformations (à partir d'une étude manuelle des interactions enregistrées). Nous avons écrit des règles permettant, à partir de chaque action de l'élève, de faire évoluer des scores sur les catégories d'appariement ainsi que des scores de préférence entre les classes stratégiques. Nous avons fait fonctionner ce mécanisme de modélisation sur les interactions enregistrées, ce qui nous a permis d'observer l'évolution des connaissances des élèves sur les différentes composantes indiquées ci-dessus. Ce mécanisme sera intégré ultérieurement au système pour permettre un choix des exercices à partir d'informations individuelles précises.

4. LE PROTOTYPE APLUSIX M1-V1

Le modèle APLUSIX M1 (*modèle d'ordre 1*) est une particularisation plus complexe du modèle général que nous avons implanté dans un prototype qui n'a pas encore été expérimenté.

4.1. Le modèle APLUSIX M1

Le modèle APLUSIX M1 comporte des connaissances stratégiques structurées : d'une part des sous-problèmes apparaissant comme des sous-buts à résoudre, d'autre part des *plans mémorisés* qui sont des successions d'actions qui ont été mémorisées du fait de leur intérêt pour le type de problème. Les problèmes sont séparés en deux classes, les problèmes simples (exemple : réduire), pour lesquels on a des règles dont on est sûr de la terminaison, et les problèmes complexes (exemples : factoriser, résoudre équation), pour lesquels on n'a pas cette assurance. Le modèle comporte quatre interpréteurs : un superviseur qui gère le lancement des autres interpréteurs, un interpréteur de problèmes simples qui applique les règles du problème concerné jusqu'à terminaison, un interpréteur de problèmes complexes qui évalue les plans possibles du problème avec des règles stratégiques (par un mécanisme analogue à celui du modèle 0 pour les transformations) et un interpréteur de plans qui gère le passage d'une action à l'autre à l'intérieur d'un plan.

4.2. Description du prototype APLUSIX M1-V1

Le prototype APLUSIX M1-V1 est une mise en oeuvre de ce modèle dans le domaine des factorisations d'expressions polynomiales (une variable, coefficients entiers) et des équations (une variable, expressions polynomiales ou avec des radicaux) au niveau des classes de première et terminale.

Le mode d'interaction du prototype APLUSIX M1-V1 est l'apprentissage par l'exemple : le système montre comment il résout des problèmes. Pour cela, il effectue une résolution pas à pas, donnant la possibilité à l'élève, à chaque pas, de demander des explications. Les explications sont engendrées par une base de connaissances qui effectue un raisonnement sur le raisonnement de résolution et apporte des informations complémentaires faisant apparaître les sous-problèmes, les plans et les règles stratégiques dans un discours en langage naturel. La figure 3 est un exemple d'explication produite par ce prototype.

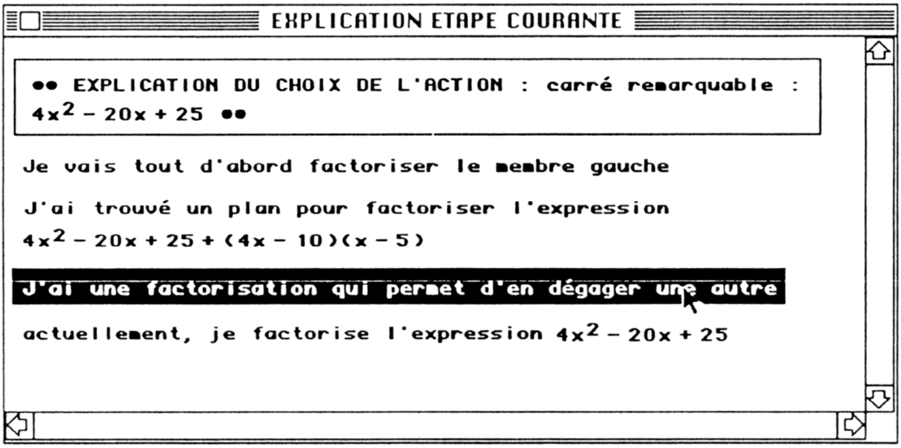

Figure 3

Explication de la factorisation d'une partie d'une somme. La première ligne indique un sous-problème, la deuxième et la troisième un plan, la quatrième l'étape courante du plan. L'élève peut cliquer sur une explication s'il veut une explication complémentaire de cette partie.

5. CONCLUSION

Nos travaux se situent dans une problématique de recherche fondamentale en EIAO. Ils comportent l'élaboration de cadres théoriques et de modèles à caractère cognitif (modèles produisant des comportements assez proches de ceux des humains) et computationnel (modèles pouvant être implantés dans des systèmes informatiques). Ils comportent aussi la réalisation d'environnements d'apprentissage avec des modes d'interaction qui cherchent à exploiter les possibilités des interactions homme-machine. Ils comportent enfin des expérimentations pour évaluer les réalisations ainsi que pour essayer de mieux comprendre la façon dont les élèves apprennent.

Cette problématique de recherche fondamentale a empêché pour l'instant la réalisation de produits commercialisés. Nos prototypes sont des réalisations soignées, car nous pensons qu'il n'est pas possible de conduire des expérimentations significatives si l'on n'a pas une bonne fidélité avec les objets usuels du domaine ainsi qu'une une bonne qualité générale, mais il reste un travail de développement, de test et de documentation important pour transformer les prototypes en produits. Ce travail sera peut-être réalisé un jour, cela permettrait une utilisation importante des logiciels et un recueil massif de données sur les élèves. L'étude de ces données permettrait certainement d'affiner les modèles sur le plan cognitif et d'améliorer les systèmes.

Jean-François NICAUD
L.R.I., CNRS URA 410
Bât 490, Université de Paris 11
91405 Orsay cedex, France

6. BIBLIOGRAPHIE

Le lecteur intéressé par des développements plus détaillés pourra lire les publications suivantes :

« Résolution et explication avec le modèle d'ordre 1 » : J.F. Nicaud, M. Saïdi : Explication en résolution d'exercices d'algèbre. *Revue d'Intelligence Artificielle*, Vol 4, n° 2, juin 1990.

« Description de prototypes avec le modèle d'ordre 0 et de la mise en oeuvre » : *Systèmes Experts et EAO* (M. Quéré ed.), OPHRYS, 1991.

« Résolution et explication avec le modèle d'ordre 0 » : M. Saïdi : *Planification et explication du raisonnement d'un résolveur complexe en algèbre : application aux factorisations de polynômes et aux résolutions d'équations*. Thèse de l'Université de Paris XI, Décembre 1992.

« Modélisation de l'élève » : A. Nguyen-Xuan, F. Joly, J.F. Nicaud, J.M. Gélis : Une méthode de diagnostic des connaissances en algèbre pour un module de modélisation de l'élève. *Environnements Interactifs d'Apprentissage avec Ordinateur* (M. Baron, R. Gras, J.F. Nicaud eds) Eyrolles, 1993.