Conception d'une ingénierie didactique pour l'introduction d'éléments d'algorithmique et de programmation dans l'enseignement secondaire des mathématiques

Alain Birebent, Thanh Nguyen Chi

Les cahiers Leibniz

Conception d'une ingénierie didactique pour l'introduction d'éléments d'algorithmique et de programmation dans l'enseignement secondaire des mathématiques

Alain BIREBENT et NGUYEN Chi Thanh

Laboratoire Leibniz-IMAG, 46 av. Félix Viallet, 38000 GRENOBLE, France -
ISSN : 1298-020X

n° 125
Juin 2005

Site internet : http://www-leibniz.imag.fr

Conception d'une ingénierie didactique pour l'introduction d'éléments d'algorithmique et de programmation dans l'enseignement secondaire des mathématiques

Alain BIREBENT et NGUYEN Chi Thanh
Equipe DDM, Laboratoire Leibniz, Grenoble

Résumé : Il y a entre les mathématiques et l'informatique une solidarité fondamentale qui repose sur l'histoire et sur les pratiques actuelles de ces disciplines. Une preuve en est le recours constant aux algorithmes dans les résolutions de problèmes mathématiques fondamentaux et l'existence de l'algorithmique comme domaine constitutif de l'informatique aux côtés d'autres, la théorie des langages, la théorie des automates, etc.

Nous présentons[1] ici partiellement les premiers résultats d'une recherche qui étudie la question de l'introduction d'éléments d'algorithmique et de programmation dans l'enseignement mathématique secondaire. Cette recherche construit une ingénierie didactique autour d'un problème de calcul numérique effectif. Le domaine mathématique choisi est fortement associé dans l'enseignement mathématique secondaire à la présence d'algorithmes et à l'intervention de la calculatrice. Après avoir décrit le contexte institutionnel et justifié certaines des orientations de la recherche, nous exposons la conception de cette ingénierie didactique.

1. L'informatique dans l'enseignement secondaire des mathématiques : une nouvelle proposition noosphérienne

Nous nous référons ici au rapport (2001) de la commission de réflexion sur l'enseignement des mathématiques, dite commission Kahane, entièrement disponible sur le site de la SMF (Société de Mathématique de France) http://smf.emath.fr/Enseignement/CommissionKahane/

La commission propose "d'introduire une part d'informatique dans l'enseignement des sciences mathématiques et dans la formation des maîtres" en faisant évoluer progressivement les contenus "pour intégrer de nouveaux objets et notions d'algorithmique et programmation". *Rapport de la commission Kahane 2001.*

Le rapport désigne à l'enseignement "dans le cadre des sciences mathématiques" ce qu'il appelle les concepts de base d'algorithmique et de programmation :

- A propos de programmation : des structures de contrôle (boucles et branchements) et la récursivité
- A propos d'algorithmique : des structures de données et la complexité

Les arguments avancés par la commission pour les enseigner se déclinent ainsi :

- "à cause de leur caractère fondamental et universel : ils sont le domaine d'interaction entre la logique, algorithmique et informatique"
- "pour permettre une meilleure compréhension et appréhension des ordinateurs et calculatrices comme aide à l'enseignement des mathématiques"
- pour revisiter certains concepts mathématiques (*exemples* : récurrence, variable)

En affirmant la nécessité de répandre l'usage d'outils informatiques dans l'enseignement des mathématiques, cette proposition s'inscrit en continuité par rapport aux choix noosphériens

[1] Nous remercions ici Annie Bessot pour sa forte implication dans la rédaction de cet article.

passés et actuels. Mais, en rupture avec ces mêmes choix, elle vise à accentuer la construction de liens théoriques qui unissent l'informatique aux mathématiques.
La comparaison avec le rapport d'une commission précédente permet d'apprécier la nature de cette rupture :

> Nous ne pensons pas que l'informatique doive être enseignée comme discipline (théorique) en tant que telle à ces niveaux de formation. L'informatique enseignée à ce niveau contient des risques de formalisme encore plus graves que les mathématiques. L'argument des élèves mauvais en mathématiques qui se rattraperaient par l'informatique est peu fondé. Par contre l'introduction d'outils informatiques peut "débloquer" des élèves en difficulté et en motiver d'autres. *Rapport de la mission Dacunha-Castelle, 1989*

La commission Kahane fait d'ailleurs le procès de cette introduction d'outils informatiques sans ambition d'enseignement sur les objets de l'informatique :

> Les programmes de mathématiques des années 90 stipulent que l'élève devra maîtriser l'usage d'une calculatrice scientifique (seconde et première) et d'une calculatrice programmable (terminable). Si cette requête est affichée dans l'introduction du document, il n'en est plus fait mention dans l'évocation des contenus et des activités. Comment le législateur imaginait-il cet apprentissage ? Il est peut-être temps de faire entrer ce souhait dans les faits en s'en donnant les moyens. *Rapport de la commission Kahane*

2. L'informatique dans l'enseignement mathématique secondaire : constats et questions

De façon très synthétique, dans EMS (l'institution Enseignement Mathématique Secondaire français) actuelle, on constate qu'il y a :

- des algorithmes mais pas d'algorithmique enseignée
- des programmes de calcul pour des calculatrices programmables ou des tableurs mais pas de programmation enseignée
- des machines-ordinateurs mais l'architecture d'une telle machine n'est pas un objet d'enseignement

Un tel constat ne renvoie pas essentiellement à des faiblesses du législateur comme le suggère le rapport de la commission Kahane mais aux conditions écologiques et économiques du système didactique concerné. C'est ce que nous rappelle Chevallard (1986) :

> On réforme les programmes, on y introduit de nouveaux “objets d'enseignement”. Or, bien souvent, ceux-ci se révèlent trop “gros”. Et parce qu'il s'aperçoit que leur gestion dans la classe est lourde, invalidante, impossible, l'enseignant doit bien vite les apprêter, les “dégraisser”, les calibrer, voire se résoudre à les écarter. Le problème didactique ainsi posé est, d'une certaine manière, bien connu. Mais, pour l'énoncer correctement, il convient de voir qu'il ne surgit pas seulement de manière anecdotique (aussi ne parlerai-je pas, ici, de la trop fameuse “droite affine en quatrième”), mais bien de manière *systématique*. Il convient de ne pas y voir seulement l'effet de quelque décision irréfléchie des rédacteurs des programmes, mais bien de la conséquence *régulière* des lois spécifiques du fonctionnement didactique.

Nous nous poserons donc les questions suivantes :

Quelles sont les conditions qui peuvent permettre à des objets nouveaux comme les structures de données et les structures de contrôle, de vivre dans EMS ? Avec quels problèmes mathématiques ? Avec quelles organisations mathématiques et didactiques ?

Nous allons développer ces questions en fixant :

- un domaine des mathématiques : le Calcul Numérique
- un technologie informatique : la calculatrice

Justifions ces choix.

3. Calcul numérique, algorithmes et instruments de calcul

Calcul numérique et algorithmes

Le calcul numérique est un lieu privilégié de production d'algorithmes notamment d'algorithmes itératifs où il y a répétition d’un invariant de calcul (division de deux nombres entiers, approximation d'une racine d'une équation numérique, etc.).
La répétition s’appuie sur une discrétisation des ensembles de nombres et sur la détermination d’actions effectuables et reproductibles. Sa formulation, en vue d'obtenir un résultat numérique ou une preuve, produit des algorithmes de calcul.
Dans les praxéologies de calcul numérique élaborées par EMS (l'approximation décimale, la construction d'une table numérique d'une fonction numérique etc.)[2], où sont présents des algorithmes, la répétition peut être outillée par divers objets mathématiques comme les formules et les fonctions, les suites et la notion de récurrence.

Comment la présence d'éléments d’algorithmique et de programmation peut-elle s'appuyer sur ces praxéologies ? Et comment peut-elle les transformer ?

Calcul numérique et instruments de calculs

Une part plus ou moins importante de la répétition peut être déléguée à un ou des instruments de calculs qui permettent à un opérateur humain de réaliser un calcul par l’intermédiaire d’un algorithme (explicite ou cristallisé dans l’instrument) : par exemple table numérique, boulier, calculatrice non programmable, etc. Cette capacité a été exploitée dans chaque tentative d'introduire un instrument de calcul nouveau dans l'enseignement des mathématiques pour renforcer l'effectivité du calcul numérique. L'introduction de l'instrument calculatrice dans une institution d'enseignement des mathématiques comme EMS procède d'une genèse institutionnelle qui réorganise les savoirs et les techniques de calcul numérique. Inversement on peut s'attendre à ce que l'introduction d’éléments d’algorithmique et de programmation modifie l'instrumentation des calculs numériques et la prise en charge institutionnelle de leur effectivité.

Comment l'introduction d’éléments d’algorithmique et de programmation peut-elle s'appuyer sur l'intégration actuelle de la calculatrice dans l'enseignement secondaire des mathématiques ? Et comment peut-elle la transformer ?

[2] Travaux de la thèse d'Alain Birebent (2001)

4. Algorithme, machine, programmation et algorithmique

Une hypothèse de base de notre recherche est que l'introduction de concepts d'algorithmique et de programmation dans l'enseignement mathématique secondaire doit s'appuyer sur le recours à des machines fictives et/ou réelles, telles les calculatrices, ceci compte tenu de la très faible familiarisation des élèves avec les algorithmes mathématiques et de l'absence institutionnelle de tout langage de programmation. Cette hypothèse est renforcée par l'analyse de la genèse historique des machines-ordinateurs.

Du projet de la machine Analytique de Babbage à l'écriture d'un programme pour cette machine par Ada Lovelace

Les premières machines arithmétiques avant la machine Analytique (par exemple, celle fabriquée en 1623 par Schickard W., la fameuse pascaline de Pascal B. en 1643, ensuite celle conçue en 1673 par Leibniz G.W.) pouvaient effectuer des algorithmes élémentaires à savoir les quatre opérations arithmétiques sur les nombres de plusieurs chiffres introduits à la main par l'utilisateur. Le projet de la machine Analytique vient de la double ambition de Babbage :

- de faire varier la compétence de la machine pour qu'elle puisse exécuter n'importe quel type d'algorithme et non plus seulement quelques traitements particuliers. *Moreau 1987, p. 12*
- de résoudre n'importe quelle équation et d'exécuter les opérations les plus compliquées de l'analyse mathématique. *Babbage 1839, cité dans Moreau 1987, p. 14*

Pour ce faire Babbage doit imposer à la machine :

- de posséder un mécanisme permettant d'exécuter les opérations de base de l'arithmétique.
- de pouvoir exécuter non plus une seule opération par traitement comme toutes les machines existant alors, mais des séquences d'opérations variant d'un traitement à un autre. *Moreau 1987, p. 14*.

Et, en conséquence :

> il [lui] fallait imaginer des dispositifs qui permettraient cet enchaînement. […] et, avant chaque traitement, pouvoir indiquer à la machine quelle séquence d'opérations elle devait cette fois exécuter. *Moreau 1987, p. 14*

Babbage invente alors trois différentes cartes permettant de programmer la machine : des cartes d'opération arithmétique et de transfert ; des cartes constantes; des cartes variables. Grâce à ces cartes, sa machine est capable de choisir seule les opérations à effectuer, soit à partir d'une règle qui lui a été donnée, soit en fonction du résultat de ses calculs précédents. Une telle possibilité, connue sous le terme de « branchement » ou de « saut conditionnel » permet à la machine analytique de poursuivre seule sa tâche, après sélection de la décision appropriée.

> L'apparition de cette faculté de décision constitue l'une des caractéristiques principales de la machine Analytique. Elle la distingue catégoriquement de toutes les machines à calculer mécaniques précédentes comme de bien d'autres qui suivront encore et qui doivent toujours être guidées, pas à pas, par l'utilisateur. *Ligonnère 1987*

Cette faculté de décision s'appuie sur l'existence des résultats intermédiaires stockés dans une mémoire de la machine. Plusieurs programmes ont été écrits pour cette machine par sa collaboratrice Ada Lovelace, dans lesquels on peut attester de la première apparition des notions non mathématiques de boucles, et de variables. La présence de ces objets que nous qualifions de premiers objets informatiques est rendue possible par l'intention d'Ada d'écrire pour une machine possédant des mémoires. Voici, par exemple, celui qui concerne les nombres de Bernoulli :

Partant de la formule connue : $\frac{x}{e^x-1} = 1 - \frac{x}{2} + B_1\frac{x^2}{2} + B_3\frac{x^4}{2.3.4} + B_5\frac{x^6}{2.3.4.5.6} + \ldots$ Ada déduit une formule de récurrence : $-\frac{1}{2}.\frac{2n-1}{2n+1} + B_1\frac{2n}{2} + B_3\frac{2n.(2n-1).(2n-2)}{2.3.4} + \ldots + B_{2n-1} = 0$ soit, $A_0 + A_1B_1 + A_3B_3 + \ldots B_{2n-1} = 0$.

Cette formule lui permet de :

- transformer une formule contenant une infinité dénombrable de valeurs en un processus de calcul fini. Ce processus est basé sur la formule de récurrence double : A_{2i+1}en fonction de A_{2i-1} ($i \geq 1$) et B_{2i+1} en fonction de B_{2j+1} et de A_{2j+1} ($j = 1 \ldots i-1$)
- avoir la valeur finie n comme condition d'arrêt du processus de calcul.

Ada Lovelace propose alors un programme pour n = 4 et l'exécute "à la main". L'écriture du programme s'accompagne donc d'un retour réflexif sur les objets mathématiques de la formule afin de dégager la condition d'arrêt et l'invariant d'une boucle.

La force de cet exemple historique nous conduit à formuler un postulat et une hypothèse de recherche :

> ***Postulat*** *: l'écriture d'un message à une machine pour obtenir l'exécution d'un algorithme itératif est une condition nécessaire à l'émergence en acte des notions relevant des structures des données (variables, constantes, listes etc.) et des structures de contrôles (boucles, branchements et conditions, test d'arrêt etc.)*
>
> ***Hypothèse de recherche*** *: la formulation de la condition d'arrêt et de l'invariant de la répétition dans l'écriture du message à une machine s'élabore dans l'interaction avec un travail réflexif sur les objets mathématiques de la solution mathématique présente.*

Quelle machine dans un enseignement d'objets algorithmiques et de programmation ?

En ce qui concerne la présence de la machine dans l'enseignement algorithmique, l'analyse des ouvrages universitaires met en évidence deux stratégies d'enseignement :

- une machine de référence : réelle, ou virtuelle (c'est-à-dire non construite matériellement : par exemple la machine de Babagge), ou idéale (non construite mais représentante d'une classe de machines réelles : voir plus loin la machine MIX de Knuth)
- pas de machine de référence : on vise un langage de programmation représentatif à une classe de langages existants (cf. SPARKS de Horowitz), ce qui suppose la connaissance préalable d'un langage de programmation[3] à l'enseignement algorithmique.

L'exemple de MIX comme machine fictive idéale

Knuth (1968) introduit dans son ouvrage[4] une machine fictive MIX qu'il décrit ainsi :

- une machine idéale : "mythical computer" et puissante "an algorithm in MIX should work properly regardless of how big a byte is".
- un modèle de toutes les machines existantes : "MIX is very much like nearly every computer now in existence (except it is, perhaps, nicer)".

[3] L'absence de machines peut, à un niveau plus élémentaire, donner lieu à l'utilisation d'un "langage" par organigrammes.
[4] Nous regardons particulièrement cet ouvrage car nous le considérons comme un traité au sens de Neyret (1995)

- un langage simple et puissant : "the language of MIX has been designed to be powerful enough to allow brief programs to be written for most algorithms, yet simple enough so that its operations are easily learned".
- le langage a des caractéristiques communes avec les autres langages existants : "MIX has been specially designed to be so much like most existing machine languages that its characteristics are easy to assimilate".

Cette machine et son langage ont été conçus selon le principe de Von Neumann pour un traitement d'information automatique :

> - la machine fonctionne et est guidée par des programmes enregistrés dans la mémoire
> - les données au sens large (entrée, sortie, résultas, programmes etc.) sont stockées dans les cases de la mémoire dont l'accès se fait par des adresses de cases de mémoire. *Ho Si Dam et al. 1990*

Quelle machine choisir pour l'objectif d'introduction des objets d'algorithmique et de programmation dans EMS ?

Comme nous l'avons déjà avancé, nous rejetons l'absence de machine de référence qui suppose des savoirs déjà institués en algorithmique et programmation ; notre choix se porte alors sur :

- une machine réelle ou virtuelle de référence ;
- une machine :
  - qui intègre un artefact de calcul numérique pour réaliser effectivement les calculs numériques (machine arithmétique) et muni de mémoires
  - qui intègre un automate capable de :
    - prendre la décision d'exécuter les instructions sur l'artefact (machine de Babbage)
    - d'interagir, au cours de l'exécution des instructions, avec les programmes enregistrés (machine de Von Neumann)

Quel style de programmation choisir ?

Nous avons choisi le style impératif qui est plus proche de la machine par rapport à d'autres styles sans être un langage de la machine. Ce choix est conforme à notre hypothèse de recherche.

> Le style apparu historiquement le premier fut qualifié d'impératif parce qu'il ramène l'écriture d'un algorithme à une succession d'ordres impérieux intimés à la machine. *Ganascia 1998, p210*

5. Construction d'une ingénierie didactique

La possibilité d'introduction des objets d'algorithmique et de programmation dans le système d'enseignement actuel passe par des choix sur ces objets et les problèmes mathématiques pour les faire vivre. L'ingénierie didactique a pour ambition d'opérer de tels choix en conformité aux hypothèses de la recherche et de les mettre à l'épreuve d'une réalisation didactique dans le système d'enseignement lui-même.

Les analyses précédentes nous conduisent à concevoir une ingénierie didactique qui soit pour l'élève une genèse expérimentale de la machine de Von Neumann en lui faisant formuler des messages à des machines successivement dotées de caractéristiques différentes.

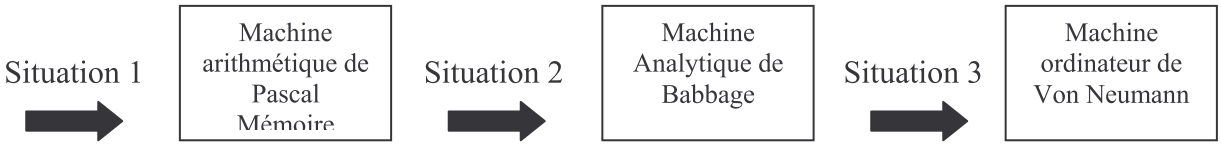


Figure 1. Les différentes machines dans les trois situations

L'artefact commun à ces trois machines est une calculatrice que nous avons nommée Alpro. C'est un artefact familier des élèves avec les commandes numériques habituelles d'une calculatrice notamment les mémoires.

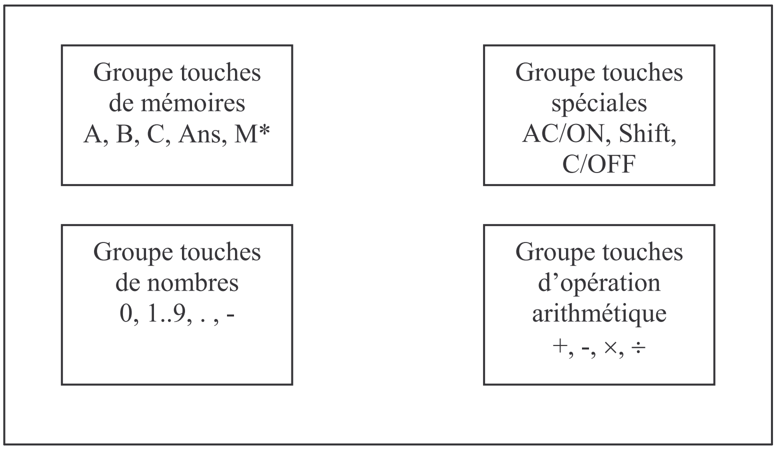
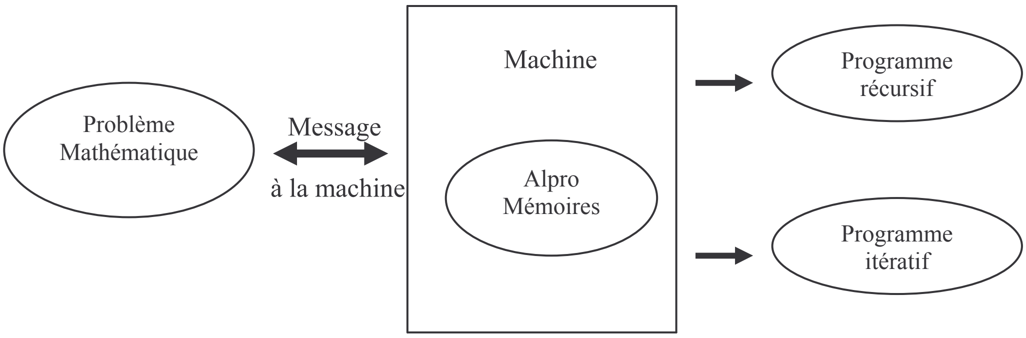


Figure 2. Le modèle de l'artefact Alpro

La gestion des mémoires de la calculatrice est pensée dans l'ingénierie comme un accès possible au concept de variable informatique.
Le problème choisi, celui de la tabulation d'une fonction numérique, cherche à déclencher la fabrication de répétitions à travers de l'itération ou de la récursivité. Mais le nombre limite de mémoires d'Alpro privilégie l'algorithme itératif au détriment de l'algorithme récursif.
L'enjeu de la situation 1 est la familiarisation avec Alpro et de ses différentes mémoires.
L'enjeu de la situation 2 est la formulation de l'invariant opératoire dans le message à la machine de Babbage, condition nécessaire, d'après notre postulat, à la conceptualisation des notions de boucle et variable informatique.
L'enjeu de la situation 3 est la formulation des conditions d'arrêt et d'initialisation des variables informatiques dans le message à la machine de Von Neumann.



Ainsi l'ingénierie est-elle bâtie sur les hypothèses de recherche élaborées par les analyses institutionnelle et épistémologique. Elle a été expérimentée dans une classe de première scientifique d'un lycée français et a donné lieu à des observations. Dans sa confrontation avec le système d'enseignement, l'ingénierie permet un retour sur les hypothèses. Les analyses des séances expérimentées sont en cours et feront l'objet d'un prochain article.

RÉFÉRENCES BIBLIOGRAPHIQUES

Birebent A. (2001), *Articulation entre la calculatrice et l'approximation décimale dans les calculs numériques de l'enseignement secondaire français.* Thèse de doctorat, Université Joseph Fourier, Grenoble.

Ho Si Dam et al. (1990), *Tin học đại cương*, Département d'Informatique et de Mathématique, Université nationale de Hanoi.

Chevallard Y. (1986),Les programmes et la transposition didactique. Illusions, contraintes et possibles. *Bulletin APMEP*, n° 352 pp. 32-50.

Ganascia J-G. (1998), *Dictionnaire de l'informatique et des sciences de l'information*, Editions Flammarion

Knuth (1968), *The art of computer programming fundamental algorithms*, Addison-Wesley Publishing Company.

Ligonnère R. (1987), *Préhistoire et histoire des ordinateurs*, Editions Robert Laffont, Paris.

Moreau R. (1987), *Ainsi naquit l'informatique,* Editions Dunod, Paris.

Neyret R. (1995), *Contraintes et déterminations des processus de formation des enseignants*, Thèse de doctorat, Université Joseph Fourier, Grenoble.

Nguyen C. T. (2002), *La notion d'algorithme dans l'enseignement des mathématiques au lycée. Comment l'émergence des notions de boucles et de variables s'articule à des connaissances en mathématiques ?* Mémoire de DEA EIAHD, Université Joseph Fourier, Grenoble.

Nguyen C. T., Bessot A. (2003), La prise en compte des notions de boucle et de variable informatique dans l'enseignement des mathématiques au lycée, *Petit x* n° 62, Irem de Grenoble.

Les Cahiers Leibniz

Le **Laboratoire Leibniz** est fortement pluridisciplinaire. Son activité scientifique couvre un large domaine qui comprend des thèmes fondamentaux aussi bien en informatique qu'en mathématiques, avec une ouverture sur l'apprentissage machine, la modélisation de systèmes complexes adaptatifs, et les applications aux environnements informatiques pour l'apprentissage humain.

Les **Cahiers Leibniz** ont pour vocation la diffusion de rapports de recherche, de supports de cours, de textes de séminaires ou de projets de publications réalisés par des membres du laboratoire. Ils peuvent accueillir aussi des textes de chercheurs n'appartenant pas au laboratoire Leibniz mais qui travaillent sur des thèmes proches et ne disposent pas de tels supports de publication. Ces chercheurs sont priés de contacter un des membres du comité éditorial ; le comité décidera de l'acceptation du texte proposé.

Le contenu des textes publiés dans les **Cahiers Leibniz** relève de la seule responsabilité de leurs auteurs.

The research at **Laboratoire Leibniz** is multidisciplinary. It covers a large domain of fundamental and applied subjects in informatics an mathematics, with openings to machine learning, the modelisation of adaptive complex systems, and applications to teaching software.

The **Cahiers Leibniz** aim at diffusing research reports, lectures, and texts of conferences or pre-prints of the members of the laboratory. Moreover, the **Cahiers** welcome manuscripts of researchers belonging to other laboratories, working on subjects close to ours, but not disposing of such a medium. These researchers should contact one of the members of the editorial board; the latter will decide whether to accept the proposal.

The responsibility of the contents of the **Cahiers** lies exclusively with the authors.

Réalisation : Jacky Coutin

ISSN : 1298-020X - © laboratoire Leibniz