

OBJETS DE SAVOIR DE NATURE INFORMATIQUE DANS L'ENSEIGNEMENT SECONDAIRE

André Rouchier

La didactique de l'informatique est en voie de construction. Elle doit mettre en avant des objets de savoir qui s'avèrent facilitants pour les apprentissages des élèves. La récursivité est certainement l'un de ces objets, permettant la construction économique de figures géométriques. Son élaboration par les élèves est facilitée par l'utilisation de LOGO, mais il n'est pas simple d'apporter la preuve de la réussite de l'apprentissage. Cet exemple, entre autres, montre la difficulté pour la recherche en didactique à créer de nouveaux objets de savoir, puis à les transférer dans l'enseignement.

L'apport des approches actuelles de la didactique des mathématiques à la résolution et à l'étude de questions d'enseignement se situe autant du côté de l'organisation du champ comme domaine de recherche autonome que du côté de la mise en question des formes de mise en rapport des savoirs et des objets qui les constituent (réalisées dans tout enseignement) avec les choix didactiques qui le structurent. Il s'agit de séparer, afin de mieux l'étudier, le didactique de l'apprentissage lui-même. Il ne saurait être question de soutenir que le didactique ne s'accompagne d'aucun effet d'apprentissage. Par contre on se propose d'appréhender de façon plus rigoureuse les parts respectives et les jeux mutuels des savoirs et des pratiques sociales qui leur servent de références, de l'apprentissage et des mécanismes qu'il met en oeuvre, du didactique c'est-à-dire de l'entrée dans les problèmes et les situations qui leur sont associées (dévolution) et font vivre les savoirs, autant que de l'identification de ces savoirs eux-mêmes (institutionnalisation) au cours d'un enseignement.

l'informatique,
objet d'étude
didactique

On dispose, à l'heure actuelle, de plusieurs niveaux de saisie et de description de phénomènes de nature didactique associés à des savoirs déterminés, isolés dans la sphère savante ou dans des sphères professionnelles. Il se trouve que l'informatique présente, de façon massive, cette dualité savant-professionnel dans sa nature même : théorique dans les objets propres qu'elle constitue, pratique par une exigence d'effectivité (le passage sur machine) et par l'existence d'un secteur professionnel fortement développé.

Il n'existe pas actuellement de didactique générale constituée et reconnue. Toutefois, on ne saurait dans un domaine de recherches vivant et dynamique, se passer de faire circuler et

d'emprunter d'un domaine disciplinaire à l'autre, les concepts, les problématiques et les méthodes. Didactique des mathématiques et didactique de la physique représentent le meilleur exemple pour le moment. Le domaine de l'informatique est beaucoup moins avancé, en particulier du côté de la recherche. Nous allons chercher à examiner quelques raisons de ce retard. Nous allons ensuite développer quelques éléments d'un cadre problématique et présenter ce qui nous paraît ressortir sur le plan des objets de savoir mis en oeuvre, d'une recherche effectuée il y a quelques années chez des débutants en LOGO au niveau collège.

1. PLACEMENT DE L'INFORMATIQUE DANS UN CHAMP D'INTERROGATION DIDACTIQUE

Le placement de l'informatique dans un enseignement, à un niveau donné, s'accompagne de décisions curriculaires et de la génération, création et mise en oeuvre de tout un ensemble de décisions qui relèvent de l'action didactique.

s'il y a
enseignement de
l'informatique, il y
a didactique de
l'informatique

En fait, il y a une didactique de l'informatique, il s'agit de l'ensemble des actions et pratiques d'enseignement qui ont actuellement pour objet l'informatique. Elles sont largement développées, notamment dans des secteurs d'enseignements professionnels à haut niveau. Pour dire autrement, l'enseignement existe, il y a donc une didactique. Cet enseignement évolue, il s'accompagne d'un travail de réflexion, par mise en rapport d'intentions et d'observations variées et par incorporation de nouveaux objets et de nouvelles techniques. L'intervention et le rôle d'une noosphère peuvent s'observer à travers la mise au point de curriculums, à travers la proposition et la conduite de choix pédagogiques, à travers l'expression des opinions à leur sujet. Contraint de s'effectuer au plus près d'une discipline qui évolue très rapidement, aussi bien du fait d'une demande sociale de compétence et de formation constamment en extension, que du fait de l'évolution des techniques et des réaménagements conceptuels, l'enseignement bouscule très rapidement des traditions qui n'ont, pour la plupart d'entre elles, que quelques années d'existence. Pour le didacticien, dans la mesure où il dispose des instruments d'analyse et des moyens pour les mettre en oeuvre, ainsi que des moyens matériels et humains de son travail, il y a là un terrain d'observation d'une très grande richesse. Il reste malheureusement en grande partie inexploité aujourd'hui. Nous ne pouvons que le regretter car les exigences de l'enseignement ont, au cours de l'histoire, joué un rôle important dans le développement de la science elle-même. L'intégration de l'informatique à la culture scientifique et technique s'effectue en très grande partie par son placement dans le champ de la connaissance, par les modifications qu'elle apporte dans les relations de l'homme et de son environnement, par les pratiques sociales

un enseignement
interrogé par
l'environnement

qu'elle institue, notamment dans sa fonction de service, c'est-à-dire par ses aspects techniques, industriels et sociaux. Cette intégration s'effectue aussi à travers l'appareil de formation et d'enseignement dans sa fonction d'intégration et de formation des générations nouvelles à l'ensemble théorico-technique qui lui est associé. Ce travail incorpore ses résultats dans la discipline elle-même par les réaménagements notionnels et conceptuels qu'il provoque. Son étude nous paraît, au moins à ce titre, particulièrement importante.

Science des représentations, de leurs transformations et des calculs sur ces représentations, l'informatique possède un versant pratique constitué de la programmation et de l'ensemble des usages effectifs de l'ordinateur. Cette particularité lui confère une place singulière parmi les sciences. Elle va devoir trouver une contrepartie du côté des choix multiples qui sont effectués pour l'enseignement et qui vont contribuer à constituer son image. C'est ainsi qu'on va chercher à tenir compte de la contrainte représentée par la nature double de l'informatique, à partir des traditions, des objectifs et des formes qu'ils représentent du fait de leur insertion institutionnelle. On pourra observer, par exemple, dans le cas de l'enseignement français, une différenciation entre les choix opérés en lycée dans le cadre de l'option informatique et ceux qui sont effectués dans l'enseignement post-baccalauréat. Lieux d'enseignement distincts donc de transpositions didactiques assujetties aux éléments généraux des contrats qui y prévalent. Une analyse de ces lieux d'enseignement permettrait de mettre à jour la relation qui se réalise effectivement entre certains éléments de contrats didactiques prévalant et les objets de savoirs qui sont retenus ainsi que les mises en rapports au savoir qui sont proposées. Il s'agit en l'occurrence de l'analyse écologique des conditions et des contraintes qui règlent le choix, l'apparition ou la disparition d'objets de savoir. La problématique et les moyens d'étude proposés dans Rajoson (1988) paraissent particulièrement adaptés à cet enseignement jeune où les manifestations de certains phénomènes sont plus apparentes. Une observation rapide de l'enseignement post-baccalauréat montre, à cet égard, quelques effets intéressants. Par exemple, au niveau des premiers cycles universitaires, on a pu créer des "objets de savoir pour l'enseignement", c'est le cas de la machine-ruban décrite dans Lucas, Peyrin, Scholl (1983) qui, au moment où elle apparaît dans l'enseignement ne peut être mise en relation avec aucun objet de savoir déjà existant. Dans les classes préparatoires au contraire, on ne crée pas d'objet abstrait à caractère général, mais on s'intéresse directement aux algorithmes numériques classiques et à leur transcription dans un langage de programmation pour passage sur machine. Autrement dit, les objets de savoir promus au titre d'objets d'enseignement sont des objets de savoirs qui peuvent être reçus dans une certaine culture didactique, celle du lieu dans lequel ils doivent fonctionner, où ils vont être accueillis et trouver leur place. Ils sont créés et transformés dans ce but.

l'enseignement
peut créer des
objets nouveaux
de savoir

l'informatique est plus jeune que la mathématique, on ne peut donc autant en exiger

Nous ne décrivons ici que quelques éléments d'une situation dont tout conduit à penser qu'elle n'est pas définitive. En effet, l'enseignement de l'informatique, en dépit d'une progression spectaculaire, n'est pas encore "installé". Il n'a pas encore conquis son "territoire" ou du moins un "territoire" qui ne se déplace pas trop autant en extension qu'en compréhension. Ce "territoire", au sens où nous l'employons ici, c'est la mise en relation, pour l'enseignement, d'un niveau d'étude, de savoirs associés et d'un type de rapport (officiel) à ce savoir proposé aux élèves. Il est un fait que l'informatique connaît une communauté savante dynamique qui travaille dans un cadre où des découpages d'objets paraissent fonctionner. L'enseignement de l'informatique lui, ne s'est pas encore institué, identifié en institution symbolique avec la même force que l'ont fait les mathématiques. Cela tient naturellement au caractère récent de l'informatique. Cela tient aussi à une importance instrumentale qui la rattache aux domaines de la "techné" plus qu'à ceux de l'"épistémé". Son insertion, son admission même dans cette partie de la culture qui s'identifie fortement à une culture d'origine scolaire ne saurait s'effectuer selon les mêmes modalités et atteindre les mêmes formes que pour les mathématiques. Il n'y a pas le même imaginaire. Il n'y a pas la même ancienneté. Leur présence n'a pas la même visibilité.

A côté d'une observation de la didactique qui se fait (les découpages de savoirs, leurs mises au régime didactique), il est légitime de poser la question de la constitution d'une didactique expérimentale mobilisée dans une perspective de construction, de proposition et d'intervention. Elle se place dans une dynamique de repérage et d'élaboration de formations épistémologiques propres à assurer un statut à l'informatique comme discipline d'enseignement aux niveaux primaire et secondaire. Il ne s'agit nullement de soutenir la thèse selon laquelle il serait nécessaire de faire de l'informatique une matière de l'enseignement obligatoire. Il s'agit au contraire d'identifier les conditions qui permettraient à une décision en ce sens d'être opératoire.

Partant de la situation actuelle, marquée par l'absence de l'informatique comme discipline enseignée dans l'enseignement obligatoire, il n'est pas possible de développer un système d'interrogation qui s'appuie sur un savoir scolaire ou un savoir culturel ordinaire. Par conséquent, toute expérimentation dans les domaines qu'une interrogation classique renvoie du côté des "conceptions" du sujet, doit s'appuyer sur des séquences d'enseignement souvent brèves et espacées. Elles ne pourraient que difficilement être mises en relation avec d'autres contenus, au risque d'en perturber l'apprentissage. Les choix qui s'offrent à l'intervention expérimentale vont être très largement marqués par ces contraintes. Ils n'ont pas forcément une signification culturelle importante. On a pu utiliser, dans certaines recherches, des dispositifs programmables très spécifiques, très éloignés des machines en usage dans le monde professionnel.

Cet éloignement, s'il a pu apparaître nécessaire a conduit à retarder l'étude de certains aspects des rapports entre l'élève et l'informatique. Ceux-ci ne se réduisent pas, en effet, à la seule dimension de la programmation. Un programme écrit doit être aussi rentré en machine, exécuté, modifié, mis en fichier, etc.

des dispositifs
d'étude
didactique sont
mis en place

Nous l'avons constaté, des dispositifs didactiques se sont mis en place, et des solutions ont été apportées à des questions d'enseignement. En fait, le champ d'action et d'interrogation d'une didactique de l'informatique peut être très vaste et couvrir une "distance de développement" qui va de la première moitié de l'enseignement obligatoire (fin de l'école élémentaire, début du collège) à des enseignements spécialisés de haut niveau. Au cours de cette période l'élève peut être confronté à des dispositifs programmables de nature très diverse. Certains peuvent être simplifiés et d'accès direct, d'autres animés par des langages de commande, d'autres enfin pour lesquels il est possible d'obtenir des effets uniquement à partir de descriptions générales d'objets et de processus. Au cours d'une aussi longue période, l'écologie des objets de savoirs informatiques va se révéler très complexe. En effet, ces objets de savoir vont être proposés en référence à des problèmes, et des dispositifs programmables dont les contraintes d'interface et d'interdépendance conceptuelle peuvent être très divers. C'est ainsi que mathématique et informatique constituent deux grands "ensembles instrumentaux" pour la modélisation. La modélisation, dont nous savons qu'elle est actuellement un "point obscur" dans l'enseignement, consiste souvent à mettre en forme (ce qui ne se réduit pas à la seule formalisation) des relations qui ne sont ni "données", ni "définies" a priori dans les termes du problème qui est soumis à traitement. L'identification de ces relations suppose la mise en oeuvre d'un ou plusieurs domaines conceptuels avant traitement, qu'il soit direct par l'intermédiaire d'un calcul numérique, algébrique, ou les deux, qu'il soit indirect par le recours à une programmation. Le travail de modélisation présente des difficultés spécifiques, parfois inattendues. On a pu constater par exemple, lors de l'introduction de l'informatique dans les entreprises et notamment les PME qui ont cherché à mettre en oeuvre un processus d'informatisation, que l'obligation de s'expliquer, de "se livrer" n'allait pas sans réticences de leur part. L'obligation de s'insérer dans des pratiques sociales d'un type nouveau, qui apparaissent consécutives à un savoir, ne va pas de soi. Modéliser avec l'ordinateur, modéliser pour l'ordinateur, sont des activités qui ne comportent pas que des difficultés conceptuelles. S'il existe de nombreuses "expériences" d'enseignement qui fournissent un matériau extraordinaire pour l'analyse didactique, la reprise des questions sous-jacentes dans un système d'interrogation constitué (problématique) nous paraît un problème très actuel pour la recherche. On pourra mesurer le chemin à parcourir pour une étude des problèmes liés à une modélisation appuyée sur l'informatique en se référant aux analyses menées depuis plusieurs années

l'informatique,
outil de
modélisation,
nécessite
l'explicitation de
relations

par des équipes de l'IREM de Marseille et dont certains éléments sont donnés par Chevallard (1989). En tout état de cause, si on pense pouvoir se limiter au seul domaine de l'informatique, ce qui n'est pas assuré dès lors qu'on s'intéresse aux seuls enseignements secondaires, on devra assurer et étudier des transpositions didactiques qui prennent en charge l'empirie propre à la discipline. Cette empirie est réalisée à partir d'éléments de pratique, convertis et transposés pour des besoins d'enseignement. C'est à ce niveau que seront retrouvés, sans qu'il y ait prise en charge théorique, ce qui est nécessaire à l'utilisation effective de l'ordinateur, moyen et terme de l'empirie.

2. LES OBJETS INFORMATIQUES DANS L'ÉCRITURE DE PROGRAMMES ÉLÉMENTAIRES

L'enseignement, la didactique expérimentale, la psychologie effectuent un découpage du savoir en objets qui seront soumis à des régimes d'action ou d'interrogation qui leur sont propres. Dans les trois démarches il y a une référence au savoir. Cette référence s'inscrit dans l'ensemble des dispositifs mis en oeuvre pour l'action (enseignement) ou pour l'expérimentation (didactique, psychologie) F. Conne (1989) a mis en évidence le double mouvement dont procèdent, chacune, psychologie et didactique. Dans le premier cas, il s'agit de *"montrer comment le développement des connaissances du sujet épistémique, ..., imprime sa nature jusque dans la construction (sociale) des savoirs scientifiques les plus élaborés."* Dans le second cas, on cherchera à *"montrer comment les formes et les normes préétablies des savoirs (considérés d'emblée comme produits d'une histoire et d'une culture) peuvent être adaptées, à leur tour, à ces mécanismes sans dénaturer la connaissance ni surtout désacculturer les savoirs"*.

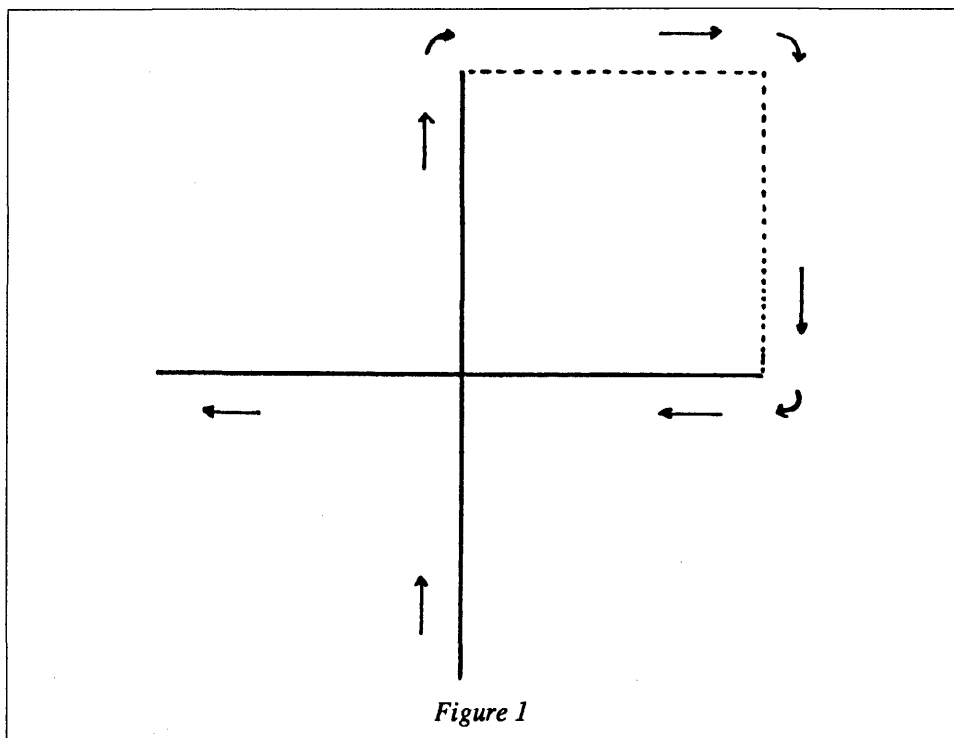
Le découpage, dans ce savoir déterminé qu'est l'informatique, d'objets pour lesquels on va développer des stratégies d'enseignement ou pour lesquels on va élaborer des dispositifs d'expérimentation, s'est appuyé pendant longtemps sur les notions qui structurent les rapports avec l'ordinateur et qui se retrouvent dans la syntaxe des programmes : instruction, suite d'instructions, exécution, paramètres et variables, premiers éléments de structuration de programmes, coordinations intra et inter-programmes. On doit aussi tenir compte de ce qui relève de la connaissance du domaine de savoir ou de pratique dont tel ou tel problème de programmation est issu. C'est en effet au niveau de l'apprenti programmeur un élément essentiel. Programmer a une fin et vise la production d'un programme qui doit être exécutable. Cette exécution fournit quelque chose qui est de l'ordre du résultat et qui témoigne d'un certain niveau de réussite. Par exemple, chez les débutants en LOGO, la programmation consiste à faire produire graphiquement des objets

la programmation est une fin, régie par une syntaxe stricte

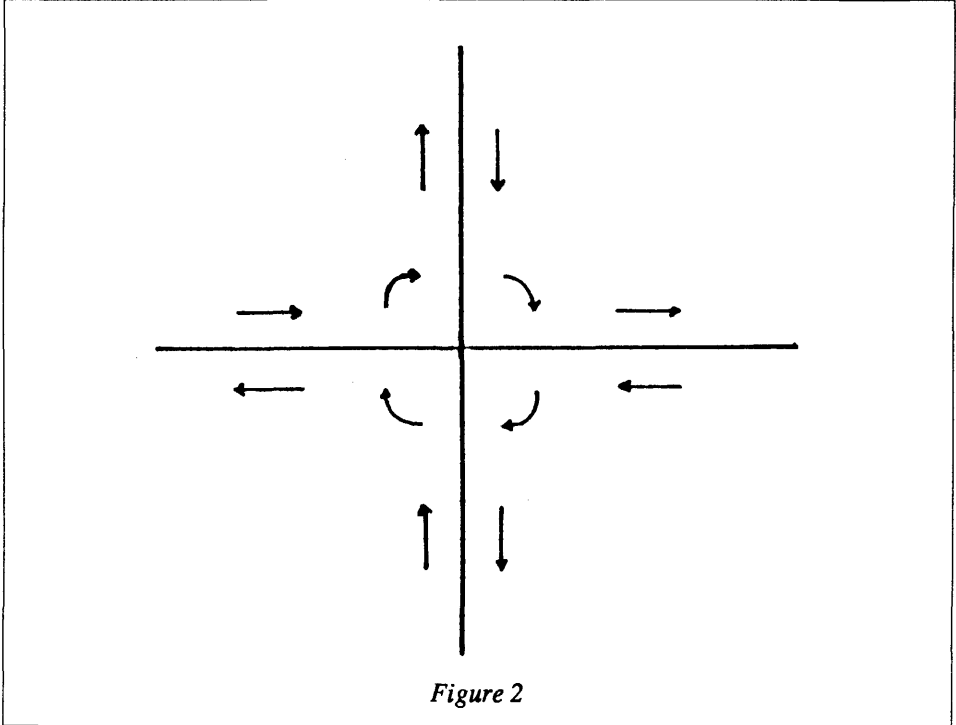
...dont
l'apprentissage
peut être facilité
par une
motivation

de nature géométrique dont certains peuvent aussi être réalisés par un tracé dans l'espace de la feuille de papier. On pourra s'interroger alors sur la manière dont un sujet jeune, débutant en programmation, développera une interaction opératoire avec un objet pour lequel il dispose en général d'un mode de production dans une instrumentation différente. Pour un objet dont on veut obtenir une réalisation graphique dans l'espace de l'écran, l'instrumentation familière est associée aux tracés sur papier, elle utilise un registre de description-production de nature géométrique dans laquelle interviennent des moyens physiques : papier, crayon, règle, compas, etc. Le développement de l'interaction opératoire se marque dans l'évolution des réponses. Au début, il peut y avoir simple transcription d'une méthode de production (tracé) qui serait réalisée à l'aide des instruments usuels de la géométrie dans les termes de l'instrumentation graphique (par exemple celle de la TORTUE), qui paraissent permettre d'obtenir le même résultat. Plus tard on aboutira à l'écriture d'un programme dans lequel seront mis en oeuvre les moyens de structuration propres au langage de programmation.

Nous pouvons rappeler ici, à titre d'exemple, l'évolution des réponses au problème du tracé d'une croix en LOGO : on peut observer un passage du schème propre au système papier-crayon, c'est-à-dire :



(les flèches indiquent le sens de parcours de la TORTUE, alors que les pointillés marquent les parcours effectués sans laisser de trace et les changements d'état, par rotation par exemple) à un schème beaucoup plus spécifique des compétences du dispositif telles que le langage de programmation les réalise.



l'enseignement
vise la pratique
d'un système
opérateur

Les objets de savoir qui vont être mis en oeuvre dans la programmation vont devoir tenir compte de cette remarque. Ce que l'enseignement va viser c'est la pratique d'un système opératoire identifié comme objet d'une culture théorique et pratique appuyée sur l'usage de l'ordinateur. C'est dans ce système que vont se réaliser la représentation et le traitement des objets qui interviennent dans les problèmes. On fera intervenir des opérateurs généraux, par exemple la structuration d'ensembles d'actions. Celle-ci peut emprunter une forme organisée : répétition et boucle. Les formes opératoires propres à la programmation dans un langage déterminé vont provoquer la décomposition des objets pour en extraire variables et relations qui rendent compte du traitement à effectuer avant tout codage. Cela correspond à un élargissement des capacités opératoires, rendu effectif par l'utilisation du dispositif programmable donc de l'informatique qu'il réalise. Il va même être possible de produire des objets plus nombreux et plus complexes à partir de programmes structurés de manière adéquate. Il peut

répétition et
récursivité, deux
formes différentes
de boucles

même y avoir invention de formes nouvelles. L'opérateur de structuration le plus puissant est la répétition. Dans ce cas, le passage du simple au complexe c'est aller d'une répétition finie, dont les éléments sont fixés une fois pour toutes et qui associe le programme à un seul objet ou à un seul processus, à la répétition finie indéterminée qui réalise en un seul programme toute une famille de programmes effectifs qu'il est possible d'appeler, de faire exécuter, pour obtenir une réalisation particulière de l'objet ou du processus. L'augmentation de la capacité opératoire est visible. En informatique, l'instrument de cette structuration est la boucle de programmation. Elle peut être réalisée soit sous forme itérative soit sous forme récursive, formes qui du point de vue conceptuel sont profondément différentes.

Dans le premier cas, la composition de la succession des blocs d'opérations élémentaires est de type sommatif : il y a un cumul d'ensembles d'opérations élémentaires. Au terme de l'écriture le tout est différent, quantitativement et qualitativement, de ses constituants (blocs d'opérations élémentaires), selon le mode :

tant que < condition > faire < suite d'instructions >

Dans le second cas, il s'agit de processus auto-référents, c'est-à-dire que le processus intervient dans sa propre production, à travers ses réalisations antérieures, selon le mode :

faire < processus > au niveau n =

faire < processus > au niveau (n-1) plus quelque chose.

La structure itérative a été un des objets couramment isolés dans les recherches portant sur les débuts de l'informatique et de la programmation. Les éléments structurels (par lesquels passe la représentation) et les éléments fonctionnels (par lesquels passe le traitement) ne sont pas séparables dans la plupart des situations de résolution de problèmes. La variable de gestion de boucle est un invariant, au sens où il faut en assurer la gestion, donc la faire intervenir explicitement, quel que soit le langage de programmation choisi.

un nouveau
rapport au savoir
s'établit

La structure alors se fait "moyen" pour travailler sur des objets nouveaux pour lesquels les schèmes opératoires généraux font défaut. C'est une des voies de son institutionnalisation, par laquelle elle devient un objet du savoir courant. Les objets auxquels elle va être confrontée ne sont pas toujours complètement nouveaux et inconnus. Par contre les rapports qu'on est amené à établir avec eux, dans le cadre des problèmes à traiter, sont complètement nouveaux. Dans le langage LOGO, par exemple, on pourra s'intéresser aux mots. On constate alors que, plus encore que dans le tracé d'un carré, le savoir-faire usuel requiert l'utilisation d'une instrumentation externe (épeler un nom est une opération dont la familiarité rend le schème opératoire trop bien dissimulé). L'occasion d'appliquer un schème itératif, par exemple sous forme récursive, fonde une vraie redécouverte de l'objet "mot". La curiosité n'est plus du côté de la structure mais de celui des objets (mots, listes) qu'elle permet d'étudier. Ils sont constitués et institués en objets

récurifs ce qui contribue à "découvrir" et à mettre en forme certaines de leurs propriétés. Nous ne sommes plus tout à fait du côté des programmes élémentaires pour débutants. Une pratique de la programmation appuyée sur l'utilisation d'écritures récurives va être un moyen de faire entrer des objets dans un champ de représentation et de traitement où existe un mode d'engendrement qui les admette comme objets récurifs. Si les objets sur lesquels on opère à l'aide d'écritures récurives (comme réalisation d'itérations) sont des objets de savoir, ou sont des éléments constitutifs d'objets de savoirs reconnus et réalisés dans une transposition didactique déterminée, alors on peut mettre en évidence et proposer, de cette manière, de nouveaux "habitats" pour des objets de savoir déjà connus (Rajson, 1988).

3. PRATIQUES DE VALIDATION DANS L'ENSEIGNEMENT DE LOGO AU NIVEAU DU PREMIER CYCLE (COLLÈGES)

La "distance" entre le programme, c'est-à-dire l'ensemble des instructions qui y figurent explicitement, et un objet graphique réalisé à travers les instructions qui sont effectivement exécutées commence à devenir importante dès qu'on utilise une structure de programmation de type répétition. C'est une première étape du détachement du programme et de son objet, c'est-à-dire de la nécessité de prendre en compte le dispositif de traitement, à travers la réalisation que le langage utilisé actualise. Comment faut-il "parler" à l'ordinateur pour obtenir ce qu'on souhaite, dès lors que la construction des éléments du discours (la programmation) fait appel à des éléments de structuration différents de la simple juxtaposition des instructions. Dans le cas de la répétition définie on constate chez de jeunes enfants (niveau CM1-CM2 notamment) quelques difficultés à entrer dans un codage de la répétition où les instructions "actives" sont placées après l'instruction REPETE. Cela tient en grande partie au rapport que l'élève doit désormais établir entre l'usage courant, culturel, du mot REPETE qui n'a pas été, jusque là, mis en opposition à un usage savant, contrôlé. L'entrée dans une pratique cognitive, contrainte par un processus technique, représente un changement dans l'usage. Il devient nécessaire d'apprendre à dissocier deux contextes, celui où il est possible de répéter une action déjà accomplie, et celui où on structure et on annonce l'usage du bloc d'instructions à répéter. Ce changement d'usage ne va pas sans difficultés qui sont rendues manifestes par l'apparition de réponses erronées au moment de l'écriture d'un programme répétitif simple. C'est le cas, par exemple, de la suivante : "REPETE 3 (AV 100 TD 90)" pour la programmation d'un carré. Ce type d'erreur est beaucoup moins fréquent, comme on peut s'y attendre, lors de la lecture d'un programme comportant cette répétition. Ces difficultés peuvent perdurer pendant un intervalle de temps relativement important, et mises par l'enseignant sur le compte de la distraction.

l'utilisation de mots du langage courant est, ici aussi, une contrainte

la récursivité est plus complexe, par sa condensation d'écriture

Le passage à la répétition indéterminée représente un tout autre ordre de difficulté. La condensation d'écriture est encore plus manifeste. Il en va de même pour l'ensemble des opérations cognitives qu'il faut conduire pour passer d'une description d'objet effectuée dans un cadre opératoire usuel à un autre projet, celui de la réalisation par l'intermédiaire du système de représentation et de traitement, concrétisé par le couple langage-ordinateur. Deux types de structuration sont possibles : structuration itérative et structuration récursive. Pour la première nous disposons d'un ensemble de précurseurs possibles, par exemple la notion commune de répétition ou bien encore la commande "REPETE". Pour la seconde il n'y a pas de précurseur opératoire simple. Nous allons voir dans quelle mesure ces deux structures sont différentes du point de vue cognitif. Il y aura, du point de vue d'un enseignement, un "objet à connaître pour opérer avec", dans un ensemble de problèmes comportant des écritures et des analyses de programmes. Nous allons partir d'une figure simple : une suite de carrés emboîtés :

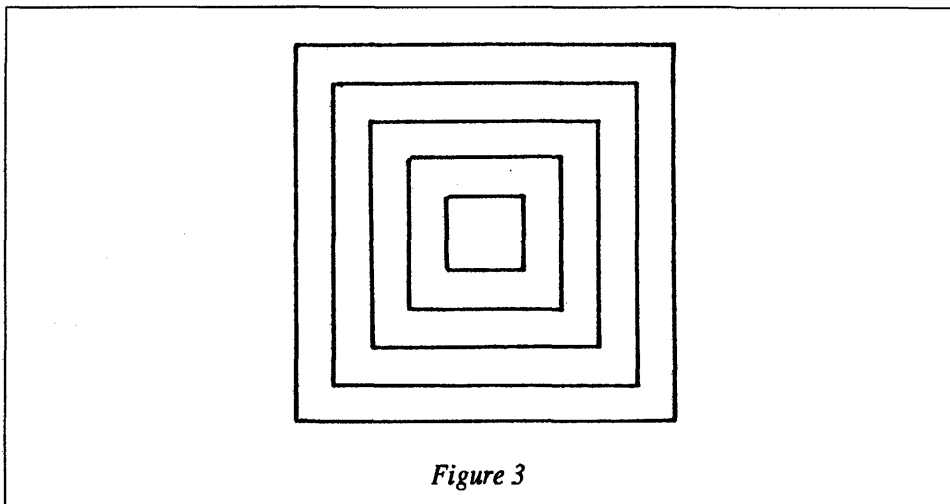


Figure 3

Elle correspond à la suite des ordres graphiques
 CARRE 10 CARRE 20 CARRE 30 CARRE 40 CARRE 50
 Cette dernière peut être associée au programme suivant :

```

    POUR FIGURE :N
      SI :N = 0 [STOP]
      CARRE :N*10
      FIGURE :N-1
    FIN
  (P)
  
```

dont elle représente un résultat d'exécution, correspondant au choix N=5.

La forme du programme (P) est très contraignante. Elle réalise le principe de récursivité et nous oblige, dès lors que nous

un problème à
plusieurs analyses
didactiques
possibles

voulons l'utiliser, à une analyse qui n'est plus strictement itérative de l'objet à produire, ou de la dynamique de sa production. On ne superpose plus des carrés dont les côtés sont en progression arithmétique, on compose le carré final avec une réalisation inachevée de l'objet. Pour mesurer la difficulté de cette approche nous pouvons partir du fait qu'il existe une stratégie de base dans l'élaboration d'une réponse : celle de la juxtaposition ordonnée des carrés. Cette stratégie peut être conçue par une très grande majorité d'élèves dans les classes du premier cycle de l'enseignement secondaire. Elle peut permettre de réussir dans la production de la figure 3 avec un répertoire assez faible de commandes géométriques et d'éléments de structuration. Pour déplacer les pratiques associées à cette stratégie et le "modèle d'engendrement" qu'elle réalise, vers des pratiques qui conduiraient au programme (P), plusieurs systèmes didactiques sont possibles. Le moins contraignant pour l'enseignant paraît celui dans lequel le problème est traité devant les élèves. Ces derniers sont, de fait, mis en situation d'imiter les conduites du maître. Dans un autre cas on pourra chercher à suppléer au manque de précurseur opératoire de la récursivité en fournissant des procédures toutes faites. Les élèves auront à les utiliser en faisant varier certains éléments. Les relations pertinentes entre ces éléments, leur place dans le programme et les effets observés réalisent un rapport au savoir visé, rapport qu'il faudra ensuite faire évoluer pour aborder des problèmes d'écriture de programmes.

difficultés d'une
approche de
construction par
les élèves

Nous avons cherché à construire une autre approche dans laquelle la structure récursive peut apparaître comme une réponse à un certain type de problèmes. La solution qui a été élaborée dans le cas des écritures récursives linéaires (un seul appel récursif dans le corps du programme) qui est décrite dans Rouchier (1988) n'est pas totalement satisfaisante. Faute d'un problème de bonne facture inductive qui permettrait l'écriture d'un programme qui ne soit pas trop complexe, il faut effectuer des décompositions du problème de base et jouer sur l'information donnée aux élèves pour que certaines interrogations, utiles du point de vue de certaines connaissances à mettre en œuvre, soient rencontrées. Si on accorde quelque crédit à la thèse selon laquelle il serait important de lier à chaque connaissance (nouvelle) une situation fondamentale composée d'un ou plusieurs problèmes, associée à un mode de mise en rapport avec ce(s) problème(s), ce qui est notre cas, il ne nous semble pas que nous ayons isolé, pour le moment les éléments de cette situation. Il y a là indice de difficulté qui peut concerner autant la qualité du travail conduit que la notion de situation fondamentale elle-même. La connaissance en cause n'est pas, en l'occurrence, le facteur le moins important. A cet égard, nous avons mis en évidence une différence structurale-fonctionnelle importante dans la relation entre un objet itérativo-récursif du type de celui qui est décrit plus haut et les parties qui le composent selon qu'on le traite comme la totalité de la suite des carrés (lue dans le sens des côtés croissants ou dans le sens des

côtés décroissants) ou bien comme un terme dans la double suite coordonnée des FIGURES et des CARRES. Cette analyse peut être résumée dans les deux schémas suivants :

CARRE 10 -> CARRE 20 -> CARRE 30 -> CARRE 40 -> CARRE 50

Schéma n° 1

CARRE 10 CARRE 20 CARRE 30 CARRE 40
 FIGURE1 FIGURE2 FIGURE3

Schéma n° 2

la validation
exige plus que la
reproduction

L'expérience de l'élève doit se constituer autour d'un objet qui est composé de la double suite, du programme et de la relation de nécessité qui les unit dans un problème de programmation, relatif à ce niveau d'intervention didactique. L'instrument d'articulation de ces éléments est une forme de validation. Le travail de validation ne saurait, du point de vue notionnel, en rester au niveau du constat visuel de l'adéquation d'un objet produit à l'écran avec ce qui est fourni dans la formulation de la consigne ("Ecrire un programme dont l'appel, par une valeur déterminée de la variable, produise une figure analogue à la figure donnée"). Il est indispensable d'élaborer des instruments de contrôle plus intellectuels, sinon plus rigoureux. Le moyen que nous avons retenu au cours des quatre séquences d'enseignement associées à la recherche sur "didactique et apprentissage de la récursivité au niveau du collège", est basé sur la demande d'un "équivalent-exécution" composé des ordres graphiques exécutés à l'appel de telle ou telle procédure. Un contrat va donc s'établir au niveau de la capacité et produire cette séquence à la demande, production qui pourra être assortie d'une justification. Cette justification va évoluer, partant d'une exécution réalisée effectivement, à l'usage d'un modèle appelé modèle relationnel, par lequel la suite ordonnée des appels à la procédure extérieure (il s'agit de l'exemple de la procédure CARRE) est associée au couple (appel récursif, appel extérieur) selon un rapport schématisé comme suit :

relations structurelles "récursives"	<=>	relations figurales "itératives"
(appel récursif, appel extérieur)	<->	suite de carrés tracés dans le sens croissant des côtés
(appel extérieur, appel récursif)	<->	suite des carrés tracés dans le sens décroissant des côtés

...en demandant
aux élèves de
produire deux
procédures

Cette correspondance ordonne autant la vérification qu'elle permet la production de programmes ou l'anticipation d'un résultat. Nous rencontrons alors deux types de problèmes :

- ceux qui sont posés par l'extension de son domaine d'utilisation, pour la récursivité linéaire, à des contenus de programmation non graphiques, au sens de LOGO. Il faut alors fonder en analogie opératoire, dans le nouveau contexte, une réalisation de la relation figurale itérative.
- ceux associés à l'établissement d'un autre rapport qui serait du type :

relations figurales récursives <=> relations structurelles récursives

et correspondrait à ce qu'on appelle traditionnellement l'intériorisation (par l'élève) de la structure récursive.

On peut penser que des problèmes de programmation portant sur des objets pour lesquels une description basée sur des relations figurales itératives est inadéquate, vont permettre de passer à une description basée sur d'autres relations figurales, en l'occurrence des relations récursives. L'arbre linéaire est un de ces objets possibles, étudié par C. Dupuis et D. Guin (1988, 1989). L'ordonnement du questionnement se réalise comme celui que nous avons proposé dans deux textes (1988a, 1989b). On constate la persistance, qui paraît normale, de ce que les auteurs appellent "*appréhension séquentielle*" et qui relève d'une lecture des relations figurales selon le mode itératif, et son transport dans le domaine de la désignation des variables sous la forme du codage descriptif. C'est témoigner largement à la fois de l'intérêt, et des difficultés, que peut rencontrer la mise en place d'objets de savoir aussi nouveaux que la forme d'"élémentarisation" de la récursivité que représente la pratique des écritures récursives en situation de programmation.

CONCLUSION

Tracer les éléments d'un passage de l'expérimentation organisée à l'enseignement suppose une délimitation des problèmes qui peuvent être rencontrés au cours de cette opération. Ils sont du même ordre que ceux qui se manifesteraient dans la didactique d'autres disciplines scientifiques à cela près qu'il s'agit de savoirs dont le mode d'insertion dans un réel a une spécificité particulière. Le double repérage, interne, articulé sur l'écologie des savoirs en cause, externe articulé sur des éléments culturels donc sur un ordre de reconnaissance dans la noosphère est une nécessité. Dans cet article nous avons essayé de mettre en évidence, par une présentation de quelques travaux, le prix à payer pour la nouveauté, en tant qu'objet de savoir, de notions comme la récursivité. Il nous semble que dans l'état actuel de l'enseignement scientifique, elle ne pourrait prétendre à un "habitat" de taille suffisante. Elle ne pourrait donc pas survivre.

La tâche de la recherche reste, toutefois, toujours la même, explorer des réalisations didactiques ainsi que leurs conditions d'existence qui permettent le double mouvement d'insertion d'objets de savoir dans des curriculums existants ou de construction de curriculums.

André ROUCHIER
Université d'Orléans

BIBLIOGRAPHIE

CHEVALLARD Y. (1989) *Arithmétique, Algèbre, Modélisation*. Publications de l'IREM de Marseille.

CONNE F. (1989) *Un grain de sel à propos de la transposition didactique*. CIRADE. Université de Montréal.

DUPUIS C., GUIN D. (1988) "Découverte de la récursivité en LOGO dans une classe" in C. LABORDE. *Actes du premier colloque franco-allemand de Didactique des Mathématiques et de l'Informatique*. Grenoble. La Pensée Sauvage.

DUPUIS C., GUIN D. (1989) *Gestion des relations entre variables dans un environnement de programmation LOGO*. Vol. 20, n° 3, pp. 293-316.

LABORDE C., BALACHEFF N., MEJIAS B. (1985) "Genèse du concept d'itération : une approche expérimentale". *Enfance* n° 2.3., pp. 223-239.

LUCAS M., PEYRIN J.P., SCHOLL P.C. (1983) "Algorithmique et représentation des données". Tome 1. *Files Automates d'états finis*. Paris. Masson.

MENDELSON P. (1985) "L'analyse psychologique des activités de programmation chez l'enfant de CM1 et CM2". *Enfance* n° 2.3., pp. 213-221.

PEA D. ROY., KURLAND D. MIAU (1984) "On the cognitive effects of learning computer programming". *New Ideas in Psychology*. Vol.2, n°2, pp. 137-168.

RAJOSON L. (1988) *L'analyse écologique des conditions et des contraintes dans l'étude des phénomènes de transposition didactique : trois études de cas*. Thèse de 3e cycle. Université d'Aix-Marseille. Faculté de Luminy.

ROUCHIER A., SAMURCAY R. (1984) *Concepts informatiques et programmation. Une première analyse en classe de seconde des lycées*. Rapport de recherche, Projet E.A.O. 696 ABC, CNRS, EHESS et IREM d'Orléans.

ROUCHIER A. (1988a) *Savoirs et Savoir-faire en programmation au collège. La récursivité en 4e, 3e*. Rapport de Recherche MRT.MEN (non publié).

ROUCHIER A. (1988b) "Représentation et mise en scène d'objets informatiques pour l'enseignement", in LABORDE C. *Actes du premier colloque franco-allemand de Didactique des Mathématiques et de l'Informatique*. Grenoble. La Pensée Sauvage.

SAMURCAY R., ROUCHIER A. (1985) "De "faire" à "faire faire" : planification d'actions dans la situation de programmation". *Enfance* n° 2.3., pp. 241-254.

SAMURCAY R. (1988) "Modèles cognitifs dans l'acquisition de concepts informatique", in C. LABORDE. *Actes du premier colloque franco-allemand de Didactique des Mathématiques et de l'Informatique*. Grenoble. La Pensée Sauvage.

SAMURCAY R., ROUCHIER A., (à paraître) "Apprentissage de la programmation au collège : un élargissement du champ de fonctionnement du schéma récursif". *Recherches en Didactique des Mathématiques*.